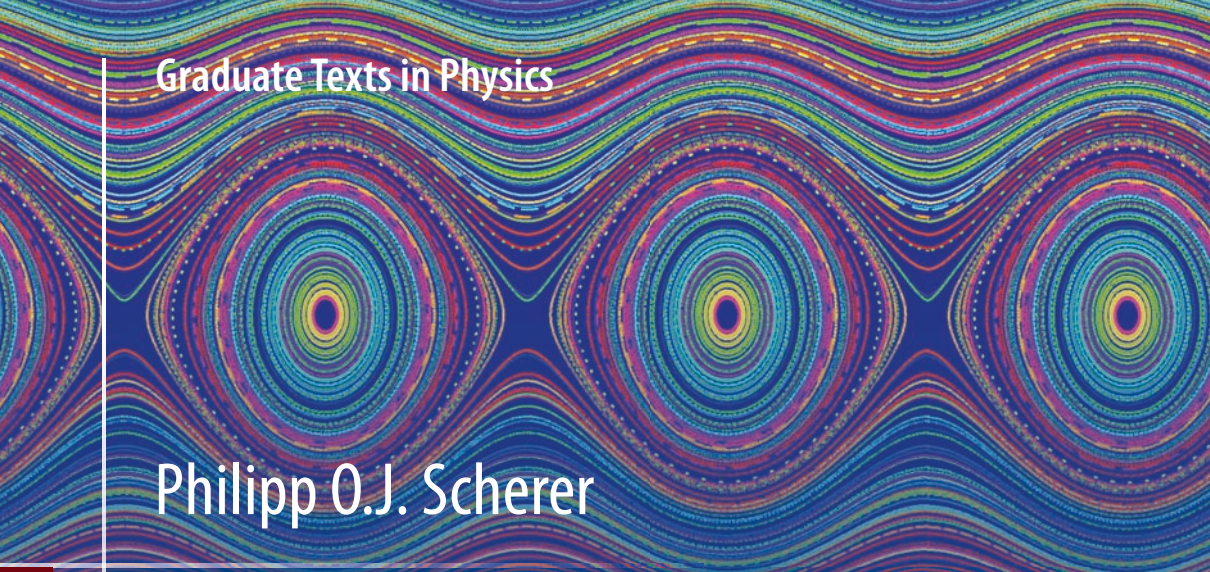
Graduate Texts in Physics



Philipp O.J. Scherer

Computational Physics

Simulation of Classical and Quantum Systems

Third Edition

Springer

Graduate Texts in Physics

Graduate Texts in Physics

Graduate Texts in Physics publishes core learning/teaching material for graduate- and advanced-level undergraduate courses on topics of current and emerging fields within physics, both pure and applied. These textbooks serve students at the MS- or PhD-level and their instructors as comprehensive sources of principles, definitions, derivations, experiments and applications (as relevant) for their mastery and teaching, respectively. International in scope and relevance, the textbooks correspond to course syllabi sufficiently to serve as required reading. Their didactic style, comprehensiveness and coverage of fundamental material also make them suitable as introductions or references for scientists entering, or requiring timely knowledge of, a research field.

More information about this series at http://www.springer.com/series/8431

Philipp O.J. Scherer

Computational Physics

Simulation of Classical and Quantum Systems

Third Edition

Springer

Philipp O.J. Scherer
Physikdepartment T38
Technische Universität München
Garching
Germany

ISSN 1868-4513 ISSN 1868-4521 (electronic)
Graduate Texts in Physics
ISBN 978-3-319-61087-0 ISBN 978-3-319-61088-7 (eBook)
DOI 10.1007/978-3-319-61088-7

Library of Congress Control Number: 2017944306

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To Christine

Preface to the Third Edition

While the first edition of this textbook was based on a one-year course in computational physics with a rather limited scope, its extent has been increased substantially in the third edition, offering the possibility to select from a broader range of computer experiments and to deepen the understanding of the important numerical methods. The computer experiments have always been a central part of my concepts for this book. Since Java applets, which are very convenient otherwise, have become more or less deprecated and their usage in a browser is no longer recommended for security issues, I decided to use standalone Java programs instead and to rewrite all of the old examples. These can also been edited and compiled with the "netbeans" environment and offer the same possibilities to generate a graphical user interface in short time.

The major changes in the third edition are as follows.

In the first part, a new chapter is devoted to the time-frequency analysis of experimental data. While the classical Fourier transform allows the calculation of the spectrum of a stationary signal, it is not so useful for nonstationary signals with significant variation of the momentaneous frequency distribution. Application of the Fourier transformation to short time windows, a method which is known as short-time Fourier transformation (STFT), allows analyzing the frequency content of a signal as a function of time. Good time resolution, of course, always comes together with a loss in frequency resolution (this is well known as "uncertainty principle"). The STFT method uses the same window for the whole spectrum, therefore the absolute time and frequency resolution is the same for low- and high-frequency components and the time resolution is limited by the period of the lowest frequencies of interest. Analysis of a signal with wavelets, on the other hand, uses shorter windows for the higher frequencies and keeps the relative frequency resolution constant while increasing the time resolution of the high-frequency components. The continuous wavelet transform can be very time consuming since it involves a convolution integral and is highly redundant. The discrete wavelet

transform uses a finite number of orthogonal basis function and can be performed much faster by calculating scalar products. It is closely related to multiresolution analysis which analyzes a signal in terms of a basic approximation and details of increasing resolution. Such methods are very popular in signal processing, especially of audio and image data but also in medical physics and seismology. The principles of the construction of orthogonal wavelet families are explained in detail, but without too many mathematical proofs. Several popular kinds of wavelets are discussed, like those by Haar, Meyer and Daubechies and their application is explored in a series of computer experiments.

In the second part, two new chapters have been added. First I included a discussion of the advection equation. Several methods to solve the one-dimensional problem are discussed from very simple straightforward differencing to quite sophisticated Galerkin-Taylor methods. The properties of these methods are demonstrated in computer experiments, as well by programs in the problems section as by numerous figures in the text. The extension to more dimensions by finite volume methods and dimensional splitting are discussed. A profound understanding of the advection equation and its numerical solution is also the basis for the more complex convection and Navier–Stokes equations.

Another chapter was added to the application of variational methods for quantum systems. The variational principle is very useful to calculate the groundstate energy. Two different types of computer experiments are performed. First we use the variational quantum Monte Carlo method (VQMC) for small atomic and molecular systems like the Helium atom and the Hydrogen molecule. We use trial functions which treat electron correlation explicitly by introducing a Jastrow factor which depends on the electron-electron distances. Such trial functions lead to nonseparable multidimensional integrals which can be efficiently calculated with the VQMC method. A second series of computer experiments studies exciton-phonon coupling in molecular aggregates which are of large interest for energy transfer in artificial and biological systems. The non-Born-Oppenheimer character of the wavefunction makes it necessary to optimize a large number of parameters. Different kinds of trial functions are applied to aggregates of up to 100 molecules to study the localization of the lowest state (so called "self-trapping").

Apart from these newly added chapters, further improvements have been made throughout the book. The chapter on random numbers now discusses in more detail the principles of modern random number generators, especially the xorshift, multiply with carry (MWC) and complementary multiply with carry (CMWC) methods. Nonstationary iterative Krylov-space methods for systems of linear equations are discussed systematically with a focus on the conjugate gradients (CG) and general minimum residual (GMRES) methods. The QR method for eigenvalue problems is now discussed in much more detail together with its connection to the power iteration method and the Krylov-space methods by Arnoldi and Lanczos.

Finally, I included a computer experiment simulating the transition between two states with wave packet dynamics, which is very helpful to understand the semi-classical approximation, especially the Landau–Zener model, which is the subject of another computer experiment.

Garching, Germany Philipp O.J. Scherer
March 2017

Preface to the Second Edition

This textbook introduces the main principles of computational physics, which include numerical methods and their application to the simulation of physical systems. The first edition was based on a one-year course in computational physics where I presented a selection of only the most important methods and applications. Approximately one-third of this edition is new. I tried to give a larger overview of the numerical methods, traditional ones as well as more recent developments. In many cases it is not possible to pin down the "best" algorithm, since this may depend on subtle features of a certain application, the general opinion changes from time to time with new methods appearing and computer architectures evolving, and each author is convinced that his method is the best one. Therefore I concentrated on a discussion of the prevalent methods and a comparison for selected examples. For a comprehensive description I would like to refer the reader to specialized textbooks like "Numerical Recipes" or elementary books in the field of the engineering sciences.

The major changes are as follows.

A new chapter is dedicated to the discretization of differential equations and the general treatment of boundary value problems. While finite differences are a natural way to discretize differential operators, finite volume methods are more flexible if material properties like the dielectric constant are discontinuous. Both can be seen as special cases of the finite element methods which are omnipresent in the engineering sciences. The method of weighted residuals is a very general way to find the "best" approximation to the solution within a limited space of trial functions. It is relevant for finite element and finite volume methods but also for spectral methods which use global trial functions like polynomials or Fourier series.

Traditionally, polynomials and splines are very often used for interpolation. I included a section on rational interpolation which is useful to interpolate functions with poles but can also be an alternative to spline interpolation due to the recent development of barycentric rational interpolants without poles.

The chapter on numerical integration now discusses Clenshaw-Curtis and Gaussian methods in much more detail, which are important for practical applications due to their high accuracy.

Besides the elementary root finding methods like bisection and Newton–Raphson, also the combined methods by Dekker and Brent and a recent extension by Chandrupatla are discussed in detail. These methods are recommended in most text books. Function minimization is now discussed also with derivative free methods, including Brent's golden section search method. Quasi-Newton methods for root finding and function minimizing are thoroughly explained.

Eigenvalue problems are ubiquitous in physics. The QL-method, which is very popular for not too large matrices is included as well as analytic expressions for several differentiation matrices.

The discussion of Singular value decomposition was extended and its application to low rank matrix approximation and linear fitting is discussed.

For the integration of equations of motion (i.e. of initial value problems) many methods are available, often specialized for certain applications. For completeness, I included the predictor-corrector methods by Nordsieck and Gear which have been often used for molecular dynamics and the backward differentiation methods for stiff problems.

A new chapter is devoted to molecular mechanics, since this is a very important branch of current computational physics. Typical force field terms are discussed as well as the calculation of gradients which are necessary for molecular dynamics simulations.

The simulation of waves now includes three additional two-variable methods which are often used in the literature and are based on generally applicable schemes (leapfrog, Lax–Wendroff, Crank–Nicolson).

The chapter on simple quantum systems was rewritten. Wave packet simulation has become very important in theoretical physics and theoretical chemistry. Several methods are compared for spatial discretization and time integration of the one-dimensional Schroedinger equation. The dissipative two-level system is used to discuss elementary operations on a Qubit.

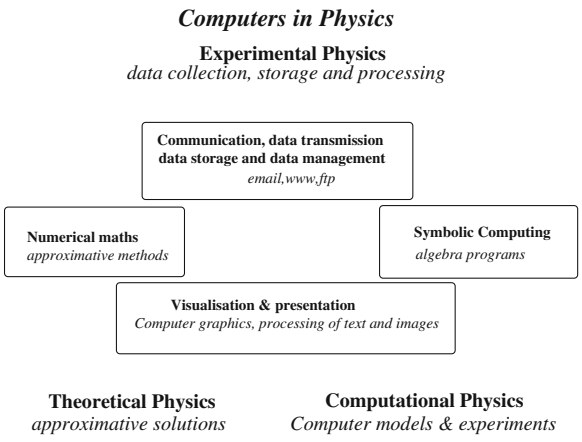
The book is accompanied by many computer experiments. For those readers who are unable to try them out, the essential results are shown by numerous figures.

This book is intended to give the reader a good overview over the fundamental numerical methods and their application to a wide range of physical phenomena. Each chapter now starts with a small abstract, sometimes followed by necessary physical background information. Many references, original work as well as specialized text books, are helpful for more deepened studies.

Garching, Germany Philipp O.J. Scherer
February 2013

Preface to the First Edition

Computers have become an integral part of modern physics. They help to acquire, store and process enormous amounts of experimental data. Algebra programs have become very powerful and give the physician the knowledge of many mathematicians at hand. Traditionally physics has been divided into experimental physics which observes phenomena occurring in the real world and theoretical physics which uses mathematical methods and simplified models to explain the experimental findings and to make predictions for future experiments. But there is also a new part of physics which has an ever growing importance. Computational physics combines the methods of the experimentalist and the theoretician. Computer simulation of physical systems helps to develop models and to investigate their properties.



This book is a compilation of the contents of a two-part course on computational physics which I have given at the TUM (Technische Universität München) for several years on a regular basis. It attempts to give the undergraduate physics students a profound background in numerical methods and in computer simulation

methods but is also very welcome by students of mathematics and computational science who want to learn about applications of numerical methods in physics. This book may also support lecturers of computational physics and bio-computing. It tries to bridge between simple examples which can be solved analytically and more complicated but instructive applications which provide insight into the underlying physics by doing computer experiments.

The first part gives an introduction into the essential methods of numerical mathematics which are needed for applications in physics. Basic algorithms are explained in detail together with limitations due to numerical inaccuracies. Mathematical explanations are supplemented by numerous numerical experiments.

The second part of the book shows the application of computer simulation methods for a variety of physical systems with a certain focus on molecular biophysics. The main object is the time evolution of a physical system. Starting from a simple rigid rotor or a mass point in a central field, important concepts of classical molecular dynamics are discussed. Further chapters deal with partial differential equations, especially the Poisson–Boltzmann equation, the diffusion equation, nonlinear dynamic systems and the simulation of waves on a 1-dimensional string. In the last chapters simple quantum systems are studied to understand e.g. exponential decay processes or electronic transitions during an atomic collision. A two-state quantum system is studied in large detail, including relaxation processes and excitation by an external field. Elementary operations on a quantum bit (Qubit) are simulated.

Basic equations are derived in detail and efficient implications are discussed together with numerical accuracy and stability of the algorithms. Analytical results are given for simple test cases which serve as a benchmark for the numerical methods. Many computer experiments are provided realized as Java applets which can be run in the web browser. For a deeper insight the source code can be studied and modified with the free "netbeans"[1] environment.

Garching, Germany
April 2010

Philipp O.J. Scherer

[1] www.netbeans.org.

Contents

Part I
Numerical Methods

Chapter 1
Error Analysis

Several sources of errors are important for numerical data processing:

Experimental uncertainty: *Input data from an experiment have a limited precision. Instead of the vector of exact values* $\mathbf{x}$ the calculation uses $\mathbf{x}+\Delta\mathbf{x}$, with an uncertainty $\Delta\mathbf{x}$. *This can lead to large uncertainties of the calculated results if an unstable algorithm is used or if the unavoidable error inherent to the problem is large.*

Rounding errors: *The arithmetic unit of a computer uses only a subset of the real numbers, the so called machine numbers* $A \subset \Re$. *The input data as well as the results of elementary operations have to be represented by machine numbers whereby rounding errors can be generated. This kind of numerical error can be avoided in principle by using arbitrary precision arithmetics*[1] *or symbolic algebra programs. But this is unpractical in many cases due to the increase in computing time and memory requirements.*

Truncation errors: *Results from more complex operations like square roots or trigonometric functions can have even larger errors since series expansions have to be truncated and iterations can accumulate the errors of the individual steps.*

1.1 Machine Numbers and Rounding Errors

Floating point numbers are internally stored as the product of sign, mantissa and a power of 2. According to the IEEE754 standard [1] single, double and quadruple precision numbers are stored as 32, 64 or 128 bits (Table 1.1).

The sign bit s is 0 for positive and 1 for negative numbers. The exponent b is biased by adding E which is half of its maximum possible value (Table 1.2).[2] The value of a number is given by

[1] *For instance the open source GNU MP bignum library.*

[2] In the following the usual hexadecimal notation is used which represents a group of 4 bits by one of the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_1

Table 1.1 Binary floating-point formats

Format	Sign	Exponent	Hidden bit	Fraction	Precision ε_M
Float	s	$b_0 \cdots b_7$	1	$a_0 \cdots a_{22}$	$2^{-24} = 5.96\text{E}^{-8}$
Double	s	$b_0 \cdots b_{10}$	1	$a_0 \cdots a_{51}$	$2^{-53} = 1.11\text{E}^{-16}$
Quadruple	s	$b_0 \cdots b_{14}$	1	$a_0 \cdots a_{111}$	$2^{-113} = 9.63\text{E}^{-35}$

Table 1.2 Exponent bias E

Decimal value	Binary value	Hexadecimal value	Data type
127_{10}	1111111_2	\$ 3F	Single
1023_{10}	1111111111_2	\$ 3FF	Double
16383_{10}	11111111111111_2	\$3FFF	Quadruple

Table 1.3 Special double precision numbers

Hexadecimal value	Symbolic value
\$ 000 0000000000000	+0
\$ 080 00000000000000	−0
\$ 7FF 0000000000000	+inf
\$ FFF 0000000000000	-inf
\$ 7FF 0000000000001 $\cdots$ \$ 7FF FFFFFFFFFFFF	NAN
\$ 001 0000000000000	Min_Normal
\$ 7FE FFFFFFFFFFFF	Max_Normal
\$ 000 0000000000001	Min_Subnormal
\$ 000 FFFFFFFFFFFF	Max_Subnormal

$$x = (-)^s \times a \times 2^{b-E}. \tag{1.1}$$

The mantissa a is normalized such that its first bit is 1 and its value is between 1 and 2

$$1.000_2 \cdots 0 \leq a \leq 1.111 \cdots 1_2 < 10.0_2 = 2_{10}. \tag{1.2}$$

Since the first bit of a normalized floating point number always is 1, it is not necessary to store it explicitly (hidden bit or J-bit). However, since not all numbers can be normalized, only the range of exponents from \$001 $\cdots$ \$7FE is used for normalized numbers. An exponent of \$000 signals that the number is not normalized (zero is an important example, there exist even two zero numbers with different sign) whereas the exponent \$7FF is reserved for infinite or undefined results (Table 1.3). The range of normalized double precision numbers is between

$$\text{Min\_Normal} = 2.2250738585072014 \times 10^{-308}$$

and

$$\text{Max\_Normal} = 1.7976931348623157E \times 10^{308}.$$

Example
Consider the following bit pattern which represents a double precision number:

$$\$4059000000000000.$$

The exponent is $100\,0000\,0101_2 - 011\,1111\,1111_2 = 110_2$ and the mantissa including the J-bit is $1\,1001\,0000\,0000\cdots_2$. Hence the decimal value is

$$1.5625 \times 2^6 = 100_{10}.$$

Input numbers which are not machine numbers have to be rounded to the nearest machine number. This is formally described by a mapping $\Re \to A$

$$x \to rd(x)$$

with the property[3]

$$|x - rd(x)| \leq |x - g| \text{ for all } g \in A. \tag{1.3}$$

For the special case that x is exactly in the middle between two successive machine numbers, a tie-breaking rule is necessary. The simplest rules are to round up always (*round-half-up*) or always down *(round-half-down)*. However, these are not symmetric and produce a bias in the average round-off error. The IEEE-754 standard [1] recommends the *round-to-nearest-even* method, i.e. the least significant bit of the rounded number should always be zero. Alternatives are *round-to-nearest-odd,* stochastic rounding and alternating rounding.

The cases of *exponent overflow* and *exponent underflow* need special attention:

Whenever the exponent b has the maximum possible value $b = b_{\max}$ and $a = 1.11\cdots11$ has to be rounded to $a' = 10.00\cdots0$, the rounded number is not a machine number and the result is $\pm\inf$.

Numbers in the range $2^{b_{\min}} > |x| \geq 2^{b_{\min}-t}$ can be represented with loss of accuracy by denormalized machine numbers. Their mantissa cannot be normalized since it is $a < 1$ and the exponent has the smallest possible value $b = b_{\min}$. Even smaller numbers with $|x| < 2^{-t+b_{\min}}$ have to be rounded to ± 0.

[3] Sometimes rounding is replaced by a simpler truncation operation which, however leads to significantly larger rounding errors.

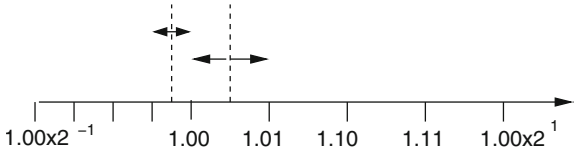


Fig. 1.1 (Round to nearest) Normalized machine numbers with $t = 3$ binary digits are shown. Rounding to the nearest machine number produces a round-off error which is bounded by half the spacing of the machine numbers

The maximum rounding error for normalized numbers with t binary digits

$$a' = s \times 2^{b-E} \times 1.a_1 a_2 \cdots a_{t-1} \tag{1.4}$$

is given by (Fig. 1.1)

$$|a - a'| \leq 2^{b-E} \times 2^{-t} \tag{1.5}$$

and the relative error is bounded by

$$\left| \frac{rd(x) - x}{x} \right| \leq \frac{2^{-t} \times 2^b}{|a| \times 2^b} \leq 2^{-t}. \tag{1.6}$$

The error bound determines the relative machine precision[4]

$$\varepsilon_M = 2^{-t} \tag{1.7}$$

and the rounding operation can be described by

$$rd(x) = x(1 + \varepsilon) \text{ with } |\varepsilon| \leq \varepsilon_M. \tag{1.8}$$

The round-off error takes its maximum value if the mantissa is close to 1. Consider a number

$$x = 1 + \varepsilon.$$

If $\varepsilon < \varepsilon_M$ then $\mathrm{rd}(x) = 1$ whereas for $\varepsilon > \varepsilon_M$ rounding gives $\mathrm{rd}(x) = 1 + 2^{1-t}$ (Fig. 1.2). Hence ε_M is given by the largest number ε for which $\mathrm{rd}(1.0 + \varepsilon) = 1.0$ and is therefore also called *unit roundoff*.

[4] Also known as machine epsilon.

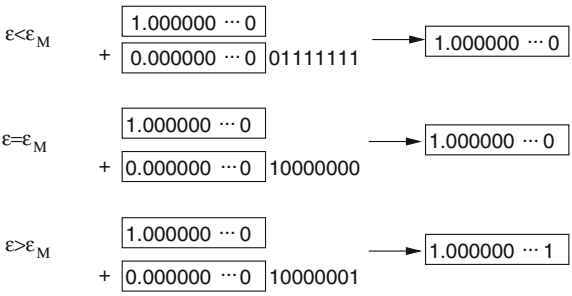


Fig. 1.2 (Unit round off)

1.2 Numerical Errors of Elementary Floating Point Operations

Even for two machine numbers $x, y \in A$ the results of addition, subtraction, multiplication or division are not necessarily machine numbers. We have to expect some additional round-off errors from all these elementary operations [2]. We assume that the results of elementary operations are approximated by machine numbers as precisely as possible. The IEEE754 standard [1] requires that the exact operations $x + y, x - y, x \times y, x \div y$ are approximated by floating point operations $A \to A$ with the property:

$$\begin{aligned} fl_+(x, y) &= rd(x + y) \\ fl_-(x, y) &= rd(x - y) \\ fl_*(x, y) &= rd(x \times y) \\ fl_\div(x, y) &= rd(x \div y). \end{aligned} \tag{1.9}$$

1.2.1 Numerical Extinction

For an addition or subtraction one summand has to be denormalized to line up the exponents (for simplicity we consider only the case $x > 0, y > 0$)

$$x + y = a_x 2^{b_x - E} + a_y 2^{b_y - E} = (a_x + a_y 2^{b_y - b_x}) 2^{b_x - E}. \tag{1.10}$$

If the two numbers differ much in their magnitude, numerical extinction can happen. Consider the following case:

$$y < 2^{b_x - E} \times 2^{-t} \tag{1.11}$$

$$a_y 2^{b_y - b_x} < 2^{-t}.$$

The mantissa of the exact sum is

$$a_x + a_y 2^{b_y - b_x} = 1.\alpha_2 \cdots \alpha_{t-1} 01 \beta_2 \cdots \beta_{t-1}. \tag{1.12}$$

Rounding to the nearest machine number gives

$$rd(x + y) = 2^{b_x} \times (1.\alpha_2 \cdots \alpha_{t-1}) = x \tag{1.13}$$

since

$$\begin{aligned} &|0.01\beta_2 \cdots \beta_{t-1} - 0| \leq |0.011 \cdots 1| = 0.1 - 0.00 \cdots 01 \\ &|0.01\beta_2 \cdots \beta_{t-1} - 1| \geq |0.01 - 1| = 0.11. \end{aligned} \tag{1.14}$$

Consider now the case

$$y < x \times 2^{-t-1} = a_x \times 2^{b_x - E - t - 1} < 2^{b_x - E - t}. \tag{1.15}$$

For normalized numbers the mantissa is in the interval

$$1 \leq |a_x| < 2 \tag{1.16}$$

hence we have

$$rd(x + y) = x \text{ if } \frac{y}{x} < 2^{-t-1} = \frac{\varepsilon_M}{2}. \tag{1.17}$$

Especially for $x = 1$ we have

$$rd(1 + y) = 1 \text{ if } y < 2^{-t} = 0.00 \cdots 0_{t-1} 1_t 000 \cdots \tag{1.18}$$

2^{-t} could be rounded to 0 or to 2^{1-t} since the distance is the same $|2^{-t} - 0| = |2^{-t} - 2^{1-t}| = 2^{-t}$.

The smallest machine number with $fl_+(1, \varepsilon) > 1$ is either $\varepsilon = 0.00 \cdots 1_t 0 \cdots = 2^{-t}$ or $\varepsilon = 0.00 \cdots 1_t 0 \cdots 01_{2t-1} = 2^{-t}(1 + 2^{1-t})$. Hence the machine precision ε_M can be determined by looking for the smallest (positive) machine number ε for which $fl_+(1, \varepsilon) > 1$.

1.2.2 Addition

Consider the sum of two floating point numbers

$$y = x_1 + x_2. \tag{1.19}$$

First the input data have to be approximated by machine numbers:

$$x_1 \to rd(x_1) = x_1(1+\varepsilon_1)$$
$$x_2 \to rd(x_2) = x_2(1+\varepsilon_2) \tag{1.20}$$

The addition of the two summands may produce another error α since the result has to be rounded. The numerical result is

$$\tilde{y} = fl_+(rd(x_1), rd(x_2)) = (x_1(1+\varepsilon_1) + x_2(1+\varepsilon_2))(1+\alpha). \tag{1.21}$$

Neglecting higher orders of the error terms we have in first order

$$\tilde{y} = x_1 + x_2 + x_1\varepsilon_1 + x_2\varepsilon_2 + (x_1 + x_2)\alpha \tag{1.22}$$

and the relative error of the numerical sum is

$$\frac{\tilde{y} - y}{y} = \frac{x_1}{x_1 + x_2}\varepsilon_1 + \frac{x_2}{x_1 + x_2}\varepsilon_2 + \alpha. \tag{1.23}$$

If $x_1 \approx -x_2$ then numerical extinction can produce large relative errors and uncertainties of the input data can be strongly enhanced.

1.2.3 Multiplication

Consider the multiplication of two floating point numbers

$$y = x_1 \times x_2. \tag{1.24}$$

The numerical result is

$$\tilde{y} = fl_*(rd(x_1), rd(x_2)) = x_1(1+\varepsilon_1)x_2(1+\varepsilon_2)(1+\mu) \approx x_1x_2(1+\varepsilon_1+\varepsilon_2+\mu) \tag{1.25}$$

with the relative error

$$\frac{\tilde{y} - y}{y} = 1 + \varepsilon_1 + \varepsilon_2 + \mu. \tag{1.26}$$

The relative errors of the input data and of the multiplication just add up to the total relative error. There is no enhancement. Similarly for a division

$$y = \frac{x_1}{x_2} \tag{1.27}$$

the relative error is

$$\frac{\tilde{y} - y}{y} = 1 + \varepsilon_1 - \varepsilon_2 + \mu. \tag{1.28}$$

1.3 Error Propagation

Consider an algorithm consisting of a sequence of elementary operations. From the set of input data which is denoted by the vector

$$\mathbf{x} = (x_1 \cdots x_n) \tag{1.29}$$

a set of output data is calculated

$$\mathbf{y} = (y_1 \cdots y_m). \tag{1.30}$$

Formally this can be denoted by a vector function

$$\mathbf{y} = \varphi(\mathbf{x}) \tag{1.31}$$

which can be written as a product of r simpler functions representing the elementary operations

$$\varphi = \varphi^{(r)} \times \varphi^{(r-1)} \cdots \varphi^{(1)}. \tag{1.32}$$

Starting with $\mathbf{x}$ intermediate results $\mathbf{x}_i = (x_{i1}, \cdots x_{in_i})$ are calculated until the output data $\mathbf{y}$ result from the last step:

$$\begin{aligned} \mathbf{x}_1 &= \varphi^{(1)}(\mathbf{x}) \\ \mathbf{x}_2 &= \varphi^{(2)}(\mathbf{x}_1) \\ &\vdots \\ \mathbf{x}_{r-1} &= \varphi^{(r-1)}(\mathbf{x}_{r-2}) \\ \mathbf{y} &= \varphi^{(r)}(\mathbf{x}_{r-1}). \end{aligned} \tag{1.33}$$

In the following we analyze the influence of numerical errors onto the final results. We treat all errors as small quantities and neglect higher orders. Due to round-off errors and possible experimental uncertainties the input data are not exactly given by $\mathbf{x}$ but by

$$\mathbf{x} + \Delta\mathbf{x}. \tag{1.34}$$

The first step of the algorithm produces the result

$$\tilde{\mathbf{x}_1} = rd(\varphi^{(1)}(\mathbf{x} + \Delta\mathbf{x})). \tag{1.35}$$

Taylor series expansion gives in first order

$$\tilde{\mathbf{x}_1} = \left(\varphi^{(1)}(\mathbf{x}) + D\varphi^{(1)}\Delta\mathbf{x}\right)(1 + E_1) + \cdots \tag{1.36}$$

with the partial derivatives

$$D\varphi^{(1)} = \left(\frac{\partial x_{1i}}{\partial x_j}\right) = \begin{pmatrix} \frac{\partial x_{11}}{\partial x_1} & \cdots & \frac{\partial x_{11}}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_{1n_1}}{\partial x_1} & \cdots & \frac{\partial x_{1n_1}}{\partial x_n} \end{pmatrix} \tag{1.37}$$

and the round-off errors of the first step

$$E_1 = \begin{pmatrix} \varepsilon_1^{(1)} & & \\ & \ddots & \\ & & \varepsilon_{n_1}^{(1)} \end{pmatrix}. \tag{1.38}$$

The error of the first intermediate result is

$$\Delta\mathbf{x}_1 = \tilde{\mathbf{x}_1} - \mathbf{x}_1 = D\varphi^{(1)}\Delta\mathbf{x} + \varphi^{(1)}(\mathbf{x})E_1. \tag{1.39}$$

The second intermediate result is

$$\begin{aligned} \tilde{\mathbf{x}_2} &= \left(\varphi^{(2)}(\tilde{\mathbf{x}_1})\right)(1 + E_2) = \varphi^{(2)}(\mathbf{x}_1 + \Delta\mathbf{x}_1)(1 + E_2) \\ &= \mathbf{x}_2(1 + E_2) + D\varphi^{(2)}D\varphi^{(1)}\Delta\mathbf{x} + D\varphi^{(2)}\mathbf{x}_1E_1 \end{aligned} \tag{1.40}$$

with the error

$$\Delta\mathbf{x}_2 = \mathbf{x}_2E_2 + D\varphi^{(2)}D\varphi^{(1)}\Delta\mathbf{x} + D\varphi^{(2)}\mathbf{x}_1E_1. \tag{1.41}$$

Finally the error of the result is

$$\Delta\mathbf{y} = \mathbf{y}E_r + D\varphi^{(r)}\cdots D\varphi^{(1)}\Delta\mathbf{x} + D\varphi^{(r)}\cdots D\varphi^{(2)}\mathbf{x}_1E_1 + \cdots D\varphi^{(r)}\mathbf{x}_{r-1}E_{r-1}. \tag{1.42}$$

The product of the matrices $D\varphi^{(r)}\cdots D\varphi^{(1)}$ is the matrix which contains the derivatives of the output data with respect to the input data (chain rule)

$$D\varphi = D\varphi^{(r)} \cdots D\varphi^{(1)} = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_n} \end{pmatrix}. \tag{1.43}$$

The first two contributions to the total error do not depend on the way in which the algorithm is divided into elementary steps in contrary to the remaining summands. Hence the inevitable error which is inherent to the problem can be estimated as [2]

$$\Delta^{(\text{in})} y_i = \varepsilon_M |y_i| + \sum_{j=1}^{n} \left| \frac{\partial y_i}{\partial x_j} \right| |\Delta x_j| \tag{1.44}$$

or in case the error of the input data is dominated by the round-off errors $|\Delta x_j| \leq \varepsilon_M |x_j|$

$$\Delta^{(\text{in})} y_i = \varepsilon_M |y_i| + \varepsilon_M \sum_{j=1}^{n} \left| \frac{\partial y_i}{\partial x_j} \right| |x_j|. \tag{1.45}$$

Additional errors which are smaller than this inevitable error can be regarded as harmless. If all errors are harmless, the algorithm can be considered well behaved.

1.4 Stability of Iterative Algorithms

Often iterative algorithms are used which generate successive values starting from an initial value $\mathbf{x}_0$ according to an iteration method

$$\mathbf{x}_{j+1} = f(\mathbf{x}_j), \tag{1.46}$$

for instance to solve a large system of equations or to approximate a time evolution $\mathbf{x}_j \approx \mathbf{x}(j\Delta t)$. Consider first a linear iteration equation which can be written in matrix form as

$$\mathbf{x}_{j+1} = A\mathbf{x}_j. \tag{1.47}$$

If the matrix A is the same for all steps we have simply

$$\mathbf{x}_j = A^j \mathbf{x}_0. \tag{1.48}$$

Consider the unavoidable error originating from errors $\Delta\mathbf{x}$ of the start values:

$$\mathbf{x}_j = A^j(\mathbf{x}_0 + \Delta\mathbf{x}) = A^j\mathbf{x}_0 + A^j\Delta\mathbf{x}. \tag{1.49}$$

The initial errors $\Delta\mathbf{x}$ can be enhanced exponentially if A has at least one eigenvalue[5] λ with $|\lambda| > 1$. On the other hand the algorithm is conditionally stable if for all eigenvalues $|\lambda| \leq 1$ holds. For a more general nonlinear iteration

$$\mathbf{x}_{j+1} = \varphi(\mathbf{x}_j) \tag{1.50}$$

the error propagates according to

$$\begin{aligned}
&\mathbf{x}_1 = \varphi(\mathbf{x}_0) + D\varphi\Delta\mathbf{x} \\
&\mathbf{x}_2 = \varphi(\mathbf{x}_1) = \varphi(\varphi(\mathbf{x}_0)) + (D\varphi)^2\Delta x \\
&\vdots \\
&\mathbf{x}_j = \varphi(\varphi\cdots\varphi(\mathbf{x}_0)) + (D\varphi)^j\Delta x.
\end{aligned} \tag{1.51}$$

The algorithm is conditionally stable if all eigenvalues of the derivative matrix $D\varphi$ have absolute values $|\lambda| \leq 1$.

1.5 Example: Rotation

Consider a simple rotation in the complex plane. The equation of motion

$$\dot{z} = i\omega z \tag{1.52}$$

obviously has the exact solution

$$z(t) = z_0 \mathrm{e}^{i\omega t}. \tag{1.53}$$

As a simple algorithm for numerical integration we use a time grid

$$t_j = j\Delta t \quad j = 0, 1, 2\cdots \tag{1.54}$$

$$z_j = z(t_j) \tag{1.55}$$

and iterate the function values

$$z_{j+1} = z_j + \dot{z}(t_j) = (1 + i\omega\Delta t)z_j. \tag{1.56}$$

Since

$$|1 + i\omega\Delta t| = \sqrt{1 + \omega^2\Delta t^2} > 1 \tag{1.57}$$

[5]The eigenvalues of A are solutions of the eigenvalue equation $A\mathbf{x} = \lambda\mathbf{x}$ (Chap. 10).

uncertainties in the initial condition will grow exponentially and the algorithm is not stable. A stable method is obtained by taking the derivative in the middle of the time interval (p. 296)

$$\dot{z}\left(t+\frac{\Delta t}{2}\right)=i\omega z\left(t+\frac{\Delta t}{2}\right)$$

and making the approximation (p. 297)

$$z\left(t+\frac{\Delta t}{2}\right)\approx\frac{z(t)+z(t+\Delta t)}{2}.$$

This gives the implicit equation

$$z_{j+1}=z_j+i\omega\Delta t\frac{z_{j+1}+z_j}{2} \tag{1.58}$$

which can be solved by

$$z_{j+1}=\frac{1+\frac{i\omega\Delta t}{2}}{1-\frac{i\omega\Delta t}{2}}z_j. \tag{1.59}$$

Now we have

$$\left|\frac{1+\frac{i\omega\Delta t}{2}}{1-\frac{i\omega\Delta t}{2}}\right|=\frac{\sqrt{1+\frac{\omega^2\Delta t^2}{4}}}{\sqrt{1+\frac{\omega^2\Delta t^2}{4}}}=1 \tag{1.60}$$

and the calculated orbit is stable.

1.6 Truncation Error

The algorithm in the last example is stable but of course not perfect. Each step produces an error due to the finite time step. The exact solution

$$z(t+\Delta t)=z(t)\mathrm{e}^{i\omega\Delta t}=z(t)\left(1+i\omega\Delta t-\frac{\omega^2\Delta t^2}{2}+\frac{-i\omega^3\Delta t^3}{6}\cdots\right) \tag{1.61}$$

is approximated by

$$z(t+\Delta t)\approx z(t)\frac{1+\frac{i\omega\Delta t}{2}}{1-\frac{i\omega\Delta t}{2}}$$

$$= z(t)\left(1 + \frac{i\omega\Delta t}{2}\right)\left(1 + \frac{i\omega\Delta t}{2} - \frac{\omega^2\Delta t^2}{4} - \frac{i\omega^3\Delta t^3}{8} + \cdots\right) \quad (1.62)$$

$$= z(t)\left(1 + i\omega\Delta t - \frac{\omega^2\Delta t^2}{2} + \frac{-i\omega^3\Delta t^3}{4}\cdots\right) \quad (1.63)$$

which deviates from the exact solution by a term of the order $O(\Delta t^3)$, hence the *local error* order of this algorithm is $O(\Delta t^3)$ which is indicated by writing

$$z(t + \Delta t) = z(t)\frac{1 + \frac{i\omega\Delta t}{2}}{1 - \frac{i\omega\Delta t}{2}} + O(\Delta t^3). \quad (1.64)$$

Integration up to a total time $T = N\Delta t$ accumulates a *global error* of the order $N\Delta t^3 = T\Delta t^2$.

Problems

Problem 1.1 Machine Precision

In this computer experiment we determine the machine precision ε_M. Starting with a value of 1.0, x is divided repeatedly by 2 until numerical addition of 1 and $x = 2^{-M}$ gives 1. Compare single and double precision calculations.

Problem 1.2 Maximum and Minimum Integers

Integers are used as counters or to encode elements of a finite set like characters or colors. There are different integer formats available which store signed or unsigned integers of different length (Table 1.4). There is no infinite integer and addition of 1 to the maximum integer gives the minimum integer.

In this computer experiment we determine the smallest and largest integer numbers. Beginning with $I = 1$ we add repeatedly 1 until the condition $I + 1 > I$ becomes invalid or subtract repeatedly 1 until $I - 1 < I$ becomes invalid. For the 64 bit long integer format this takes too long. Here we multiply alternatively I by 2 until $I - 1 < I$ becomes invalid. For the character format the corresponding ordinal number is shown which is obtained by casting the character to an integer.

Table 1.4 Maximum and minimum integers

Java format	Bit length	Minimum	Maximum
Byte	8	−128	127
Short	16	−32768	32767
Integer	32	−2147483647	2147483648
Long	64	−9223372036854775808	9223372036854775807
Char	16	0	65535

Problem 1.3 Truncation Error

This computer experiment approximates the cosine function by a truncated Taylor series

$$\cos(x) \approx \text{mycos}(x, n_{\max}) = \sum_{n=0}^{n_{\max}} (-)^n \frac{x^{2n}}{(2n)!} = 1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \cdots \quad (1.65)$$

in the interval $-\pi/2 < x < \pi/2$. The function $\text{mycos}(x, n_{\max})$ is numerically compared to the intrinsic cosine function.

Chapter 2
Interpolation

Experiments usually produce a discrete set of data points $(\mathbf{x}_i, f_i)$ *which represent the value of a function* $f(\mathbf{x})$ *for a finite set of arguments* $\{\mathbf{x}_0 \ldots \mathbf{x}_n\}$. *If additional data points are needed, for instance to draw a continuous curve, interpolation is necessary. Interpolation also can be helpful to represent a complicated function by a simpler one or to develop more sophisticated numerical methods for the calculation of numerical derivatives and integrals. In the following we concentrate on the most important interpolating functions which are polynomials, splines and rational functions. Trigonometric interpolation is discussed in Chap.* 7. *An interpolating function reproduces the given function values at the interpolation points exactly* (*Fig.* 2.1). *The more general procedure of curve fitting, where this requirement is relaxed, is discussed in Chap.* 11.

The interpolating polynomial can be explicitly constructed with the Lagrange method. Newton's method is numerically efficient if the polynomial has to be evaluated at many interpolating points and Neville's method has advantages if the polynomial is not needed explicitly and has to be evaluated only at one interpolation point.

Polynomials are not well suited for interpolation over a larger range. Spline functions can be superior which are piecewise defined polynomials. Especially cubic splines are often used to draw smooth curves. Curves with poles can be represented by rational interpolating functions whereas a special class of rational interpolants without poles provides a rather new alternative to spline interpolation.

2.1 Interpolating Functions

Consider the following problem: Given are $n+1$ sample points (x_i, f_i) , $i = 0 \cdots n$ and a function of x which depends on $n+1$ parameters a_i:

$$\Phi(x; a_0 \cdots a_n). \tag{2.1}$$

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_2

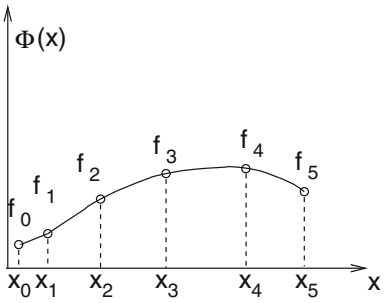


Fig. 2.1 (Interpolating function) The interpolating function $\Phi(x)$ reproduces a given data set $\Phi(x_i) = f_i$ and provides an estimate of the function $f(x)$ between the data points

The parameters are to be determined such that the interpolating function has the proper values at all sample points (Fig. 2.1)

$$\Phi(x_i; a_0 \cdots a_n) = f_i \quad i = 0 \cdots n. \tag{2.2}$$

An interpolation problem is called linear if the interpolating function is a linear combination of functions

$$\Phi(x; a_0 \cdots a_n) = a_0\Phi_0(x) + a_1\Phi_1(x) + \cdots a_n\Phi_n(x). \tag{2.3}$$

Important examples are

- polynomials

$$a_0 + a_1 x + \cdots a_n x^n \tag{2.4}$$

- trigonometric functions

$$a_0 + a_1 \mathrm{e}^{ix} + a_2 \mathrm{e}^{2ix} + \cdots a_n \mathrm{e}^{nix} \tag{2.5}$$

- spline functions which are piecewise polynomials, for instance the cubic spline

$$\mathrm{s}(x) = \alpha_i + \beta_i(x - x_i) + \gamma_i(x - x_i)^2 + \delta_i(x - x_i)^3 \quad x_i \le x \le x_{i+1}. \tag{2.6}$$

Important examples for nonlinear interpolating functions are

- rational functions

$$\frac{p_0 + p_1 x + \cdots p_M x^M}{q_0 + q_1 x + \cdots q_N x^N} \tag{2.7}$$

- exponential functions

$$a_0 e^{\lambda_0 x} + a_1 e^{\lambda_1 x} + \cdots . \tag{2.8}$$

where amplitudes a_i and exponents λ_i have to be optimized.

2.2 Polynomial Interpolation

For n + 1 sample points $(x_i, f_i), \quad i = 0 \cdots n, \quad x_i \neq x_j$ there exists exactly one interpolating polynomial of degree n with

$$p(x_i) = f_i, \quad i = 0 \cdots n. \tag{2.9}$$

2.2.1 Lagrange Polynomials

Lagrange polynomials [3] are defined as

$$\mathrm{L}_i(x) = \frac{(x - x_0) \cdots (x - x_{i-1})(x - x_{i+1}) \cdots (x - x_n)}{(x_i - x_0) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}. \tag{2.10}$$

They are of degree n and have the property

$$\mathrm{L}_i(x_k) = \delta_{i,k}. \tag{2.11}$$

The interpolating polynomial is given in terms of Lagrange polynomials by

$$p(x) = \sum_{i=0}^{n} f_i \mathrm{L}_i(x) = \sum_{i=0}^{n} f_i \prod_{k=0, k \neq i}^{n} \frac{x - x_k}{x_i - x_k}. \tag{2.12}$$

2.2.2 Barycentric Lagrange Interpolation

With the polynomial

$$\omega(x) = \prod_{i=0}^{n} (x - x_i) \tag{2.13}$$

the Lagrange polynomial can be written as

$$\mathrm{L}_i(x) = \frac{\omega(x)}{x - x_i} \frac{1}{\prod_{k=0, k\neq i}^{n} (x_i - x_k)} \tag{2.14}$$

which, introducing the Barycentric weights [4]

$$u_i = \frac{1}{\prod_{k=0, k\neq i}^{n} (x_i - x_k)} \tag{2.15}$$

becomes the first form of the barycentric interpolation formula

$$\mathrm{L}_i(x) = \omega(x) \frac{u_i}{x - x_i}. \tag{2.16}$$

The interpolating polynomial can now be evaluated according to

$$p(x) = \sum_{i=0}^{n} f_i \mathrm{L}_i(x) = \omega(x) \sum_{i=0}^{n} f_i \frac{u_i}{x - x_i}. \tag{2.17}$$

Having computed the weights u_i, evaluation of the polynomial only requires $O(n)$ operations whereas calculation of all the Lagrange polynomials requires $O(n^2)$ operations. Calculation of $\omega(x)$ can be avoided considering that

$$p_1(x) = \sum_{i=0}^{n} L_i(x) = \omega(x) \sum_{i=0}^{n} \frac{u_i}{x - x_i} \tag{2.18}$$

is a polynomial of degree n with

$$p_1(x_i) = 1 \quad i = 0 \ldots n. \tag{2.19}$$

But this is only possible if

$$p_1(x) = 1. \tag{2.20}$$

Therefore

$$p(x) = \frac{p(x)}{p_1(x)} = \frac{\sum_{i=0}^{n} f_i \frac{u_i}{x - x_i}}{\sum_{i=0}^{n} \frac{u_i}{x - x_i}} \tag{2.21}$$

which is known as the second form of the barycentric interpolation formula.

2.2.3 Newton's Divided Differences

Newton's method of divided differences [5] is an alternative for efficient numerical calculations [6]. Rewrite

$$f(x) = f(x_0) + \frac{f(x) - f(x_0)}{x - x_0}(x - x_0). \tag{2.22}$$

With the first order divided difference

$$f[x, x_0] = \frac{f(x) - f(x_0)}{x - x_0} \tag{2.23}$$

this becomes

$$f[x, x_0] = f[x_1, x_0] + \frac{f[x, x_0] - f[x_1, x_0]}{x - x_1}(x - x_1) \tag{2.24}$$

and with the second order divided difference

$$\begin{aligned} f[x, x_0, x_1] &= \frac{f[x, x_0] - f[x_1, x_0]}{x - x_1} = \frac{f(x) - f(x_0)}{(x - x_0)(x - x_1)} - \frac{f(x_1) - f(x_0)}{(x_1 - x_0)(x - x_1)} \\ &= \frac{f(x)}{(x - x_0)(x - x_1)} + \frac{f(x_1)}{(x_1 - x_0)(x_1 - x)} + \frac{f(x_0)}{(x_0 - x_1)(x_0 - x)} \end{aligned} \tag{2.25}$$

we have

$$f(x) = f(x_0) + (x - x_0)\, f[x_1, x_0] + (x - x_0)(x - x_1)\, f[x, x_0, x_1]. \tag{2.26}$$

Higher order divided differences are defined recursively by

$$f[x_1 x_2 \cdots x_{r-1} x_r] = \frac{f[x_1 x_2 \cdots x_{r-1}] - f[x_2 \cdots x_{r-1} x_r]}{x_1 - x_r}. \tag{2.27}$$

They are invariant against permutation of the arguments which can be seen from the explicit formula

$$f[x_1 x_2 \cdots x_r] = \sum_{k=1}^{r} \frac{f(x_k)}{\prod_{i \neq k}(x_k - x_i)}. \tag{2.28}$$

Finally we have

$$f(x) = p(x) + q(x) \tag{2.29}$$

with a polynomial of degree n

$$p(x) = f(x_0) + f[x_1, x_0](x - x_0) + f[x_2 x_1 x_0](x - x_0)(x - x_1) + \cdots$$

$$\cdots + f[x_n x_{n-1} \cdots x_0](x - x_0)(x - x_1) \cdots (x - x_{n-1}) \tag{2.30}$$

and the function

$$q(x) = f[x x_n \cdots x_0](x - x_0) \cdots (x - x_n). \tag{2.31}$$

Obviously $q(x_i) = 0 \quad , i = 0 \cdots n$, hence $p(x)$ is the interpolating polynomial.

Algorithm

The divided differences are arranged in the following way:

$$\begin{array}{llll} f_0 & & & \\ f_1 & f[x_0 x_1] & & \\ \vdots & \vdots & \ddots & \\ f_{n-1} & f[x_{n-2} x_{n-1}] & f[x_{n-3} x_{n-2} x_{n-1}] \ldots & f[x_0 \ldots x_{n-1}] \\ f_n & f[x_{n-1} x_n] & f[x_{n-2} x_{n-1} x_n] \cdots & f[x_1 \cdots x_{n-1} x_n] \quad f[x_0 x_1 \cdots x_{n-1} x_n] \end{array} \tag{2.32}$$

Since only the diagonal elements are needed, a one-dimensional data array $t[0] \cdots t[n]$ is sufficient for the calculation of the polynomial coefficients:

```
for i:=0 to n do begin
  t[i]:=f[i];
  for k:=i-1 downto 0 do
     t[k]:=(t[k+1]-t[k])/(x[i]-x[k]);
   a[i]:=t[0];
end;
```

The value of the polynomial is then evaluated by

```
p:=a[n];
for i:=n-1 downto 0 do
  p:=p*(x-x[i])+a[i];
```

2.2.4 Neville Method

The Neville method [7] is advantageous if the polynomial is not needed explicitly and has to be evaluated only at one point. Consider the interpolating polynomial for the points $x_0 \cdots x_k$, which will be denoted as $P_{0,1,\cdots k}(x)$. Obviously

$$P_{0,1,\cdots k}(x) = \frac{(x - x_0)P_{1\cdots k}(x) - (x - x_k)P_{0\cdots k-1}(x)}{x_k - x_0} \tag{2.33}$$

since for $x = x_1 \cdots x_{k-1}$ the right hand side is

$$\frac{(x - x_0)f(x) - (x - x_k)f(x)}{x_k - x_0} = f(x). \tag{2.34}$$

For $x = x_0$ we have

$$\frac{-(x_0 - x_k)f(x)}{x_k - x_0} = f(x) \tag{2.35}$$

and finally for $x = x_k$

$$\frac{(x_k - x_0)f(x)}{x_k - x_0} = f(x). \tag{2.36}$$

Algorithm:

We use the following scheme to calculate $P_{0,1\cdots n}(x)$ recursively:

$$\begin{array}{lllll} P_0 & & & & \\ P_1 & P_{01} & & & \\ P_2 & P_{12} & P_{012} & & \\ \vdots & \vdots & \vdots & \ddots & \\ P_n & P_{n-1,n} & P_{n-2,n-1,n} & \cdots & P_{01\cdots n} \end{array} \tag{2.37}$$

The first column contains the function values $P_i(x) = f_i$. The value $P_{01\cdots n}$ can be calculated using a 1-dimensional data array $p[0] \cdots p[n]$:

```
for i:=0 to n do begin
p[i]:=f[i];
for k:=i-1 downto 0 do
p[k]:=(p[k+1]*(x-x[k])-p[k]*(x-x[i]) )/(x[k]-x[i]);
end;
f:=p[0];
```

2.2.5 *Error of Polynomial Interpolation*

The error of polynomial interpolation [8] can be estimated with the help of the following theorem:

If $f(x)$ is $n + 1$ times differentiable then for each $\overline{x}$ there exists ξ within the smallest interval containing $\overline{x}$ as well as all the x_i with

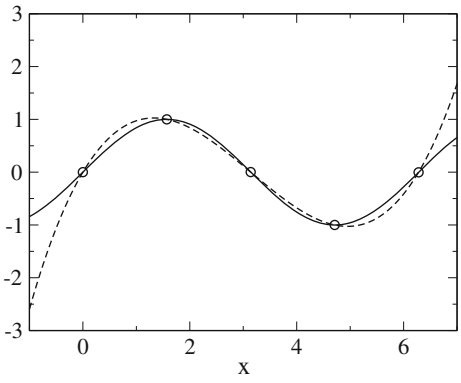


Fig. 2.2 (Interpolating polynomial) The interpolated function (*solid curve*) and the interpolating polynomial (*broken curve*) for the example (2.40) are compared

$$q(\overline{x}) = \prod_{i=0}^{n} (\overline{x} - x_i) \frac{f^{(n+1)}(\xi)}{(n+1)!}. \tag{2.38}$$

From a discussion of the function

$$\omega(x) = \prod_{i=0}^{n} (x - x_i) \tag{2.39}$$

it can be seen that the error increases rapidly outside the region of the sample points (extrapolation is dangerous!). As an example consider the sample points (Fig. 2.2)

$$f(x) = \sin(x) \quad x_i = 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}, 2\pi. \tag{2.40}$$

The maximum interpolation error is estimated by($|f^{(n+1)}| \leq 1$)

$$|f(x) - p(x)| \leq |\omega(x)| \frac{1}{120} \leq \frac{35}{120} \approx 0.3 \tag{2.41}$$

whereas the error increases rapidly outside the interval $0 < x < 2\pi$ (Fig. 2.3).

2.3 Spline Interpolation

Polynomials are not well suited for interpolation over a larger range. Often spline functions are superior which are piecewise defined polynomials [9, 10]. The simplest case is a linear spline which just connects the sampling points by straight lines:

$$p_i(x) = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x - x_i) \tag{2.42}$$

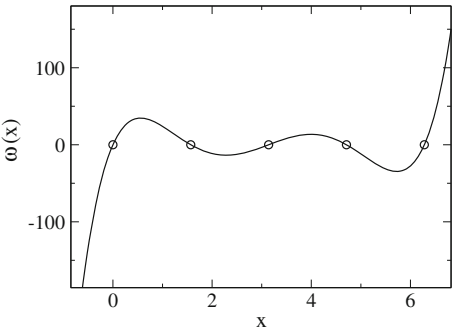


Fig. 2.3 (Interpolation error) The polynomial $\omega(x)$ is shown for the example (2.40). Its roots x_i are given by the x values of the sample points (*circles*). Inside the interval $x_0 \cdots x_4$ the absolute value of ω is bounded by $|\omega(x)| \leq 35$ whereas outside the interval it increases very rapidly

$$s(x) = p_i(x) \text{ where } x_i \leq x < x_{i+1}. \tag{2.43}$$

The most important case is the cubic spline which is given in the interval $x_i \leq x < x_{i+1}$ by

$$p_i(x) = \alpha_i + \beta_i(x - x_i) + \gamma_i(x - x_i)^2 + \delta_i(x - x_i)^3. \tag{2.44}$$

We want to have a smooth interpolation and assume that the interpolating function and their first two derivatives are continuous. Hence we have for the inner boundaries:

$$i = 0, \cdots n - 1$$

$$p_i(x_{i+1}) = p_{i+1}(x_{i+1}) \tag{2.45}$$

$$p_i'(x_{i+1}) = p_{i+1}'(x_{i+1}) \tag{2.46}$$

$$p_i''(x_{i+1}) = p_{i+1}''(x_{i+1}). \tag{2.47}$$

We have to specify boundary conditions at x_0 and x_n. The most common choice are natural boundary conditions $s''(x_0) = s''(x_n) = 0$, but also periodic boundary conditions $s''(x_0) = s''(x_n)$, $s'(x_0) = s'(x_n)$, $\quad s(x_0) = s(x_n)$ or given derivative values $s'(x_0)$ and $s'(x_n)$ are often used. The second derivative is a linear function [2]

$$p_i''(x) = 2\gamma_i + 6\delta_i(x - x_i) \tag{2.48}$$

which can be written using $h_{i+1} = x_{i+1} - x_i$ and $M_i = s''(x_i)$ as

$$p_i''(x) = M_{i+1}\frac{(x - x_i)}{h_{i+1}} + M_i\frac{(x_{i+1} - x)}{h_{i+1}} \quad i = 0 \cdots n - 1 \tag{2.49}$$

since

$$p_i''(x_i) = M_i \frac{x_{i+1} - x_i}{h_{i+1}} = s''(x_i) \tag{2.50}$$

$$p_i''(x_{i+1}) = M_{i+1} \frac{(x_{i+1} - x_i)}{h_{i+1}} = s''(x_{i+1}). \tag{2.51}$$

Integration gives with the two constants A_i and B_i

$$p_i'(x) = M_{i+1} \frac{(x - x_i)^2}{2h_{i+1}} - M_i \frac{(x_{i+1} - x)^2}{2h_{i+1}} + A_i \tag{2.52}$$

$$p_i(x) = M_{i+1} \frac{(x - x_i)^3}{6h_{i+1}} + M_i \frac{(x_{i+1} - x)^3}{6h_{i+1}} + A_i(x - x_i) + B_i. \tag{2.53}$$

From $s(x_i) = y_i$ and $s(x_{i+1}) = y_{i+1}$ we have

$$M_i \frac{h_{i+1}^2}{6} + B_i = y_i \tag{2.54}$$

$$M_{i+1} \frac{h_{i+1}^2}{6} + A_i h_{i+1} + B_i = y_{i+1} \tag{2.55}$$

and hence

$$B_i = y_i - M_i \frac{h_{i+1}^2}{6} \tag{2.56}$$

$$A_i = \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{h_{i+1}}{6} (M_{i+1} - M_i). \tag{2.57}$$

Now the polynomial is

$$\begin{aligned} p_i(x) &= \frac{M_{i+1}}{6h_{i+1}}(x - x_i)^3 - \frac{M_i}{6h_{i+1}}(x - x_i - h_{i+1})^3 + A_i(x - x_i) + B_i \\ &= (x - x_i)^3 \left(\frac{M_{i+1}}{6h_{i+1}} - \frac{M_i}{6h_{i+1}} \right) + \frac{M_i}{6h_{i+1}} 3h_{i+1}(x - x_i)^2 \\ &\quad + (x - x_i) \left(A_i - \frac{M_i}{6h_{i+1}} 3h_{i+1}^2 \right) + B_i + \frac{M_i}{6h_{i+1}} h_{i+1}^3. \end{aligned} \tag{2.58}$$

Comparison with

$$p_i(x) = \alpha_i + \beta_i(x - x_i) + \gamma_i(x - x_i)^2 + \delta_i(x - x_i)^3 \tag{2.59}$$

gives

$$\alpha_i = B_i + \frac{M_i}{6} h_{i+1}^2 = y_i \tag{2.60}$$

$$\beta_i = A_i - \frac{h_{i+1}M_i}{2} = \frac{y_{i+1} - y_i}{h_{i+1}} - h_{i+1}\frac{M_{i+1} + 2M_i}{6} \tag{2.61}$$

$$\gamma_i = \frac{M_i}{2} \tag{2.62}$$

$$\delta_i = \frac{M_{i+1} - M_i}{6h_{i+1}}. \tag{2.63}$$

Finally we calculate M_i from the continuity of $s'(x)$. Substituting for A_i in $p_i'(x)$ we have

$$p_i'(x) = M_{i+1}\frac{(x - x_i)^2}{2h_{i+1}} - M_i\frac{(x_{i+1} - x)^2}{2h_{i+1}} + \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{h_{i+1}}{6}(M_{i+1} - M_i) \tag{2.64}$$

and from $p_{i-1}'(x_i) = p_i'(x_i)$ it follows

$$\begin{aligned} &M_i\frac{h_i}{2} + \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{6}(M_i - M_{i-1}) \\ &= -M_i\frac{h_{i+1}}{2} + \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{h_{i+1}}{6}(M_{i+1} - M_i) \end{aligned} \tag{2.65}$$

$$M_i\frac{h_i}{3} + M_{i-1}\frac{h_i}{6} + M_i\frac{h_{i+1}}{3} + M_{i+1}\frac{h_{i+1}}{6} = \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i} \tag{2.66}$$

which is a system of linear equations for the M_i. Using the abbreviations

$$\lambda_i = \frac{h_{i+1}}{h_i + h_{i+1}} \tag{2.67}$$

$$\mu_i = 1 - \lambda_i = \frac{h_i}{h_i + h_{i+1}} \tag{2.68}$$

$$d_i = \frac{6}{h_i + h_{i+1}}\left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i}\right) \tag{2.69}$$

we have

$$\mu_i M_{i-1} + 2M_i + \lambda_i M_{i+1} = d_i \quad i = 1 \cdots n - 1. \tag{2.70}$$

We define for natural boundary conditions

$$\lambda_0 = 0 \quad \mu_n = 0 \quad d_0 = 0 \quad d_n = 0 \tag{2.71}$$

and in case of given derivative values

$$\lambda_0 = 1 \quad \mu_n = 1 \quad d_0 = \frac{6}{h_1}\left(\frac{y_1 - y_0}{h_1} - y_0'\right) \quad d_n = \frac{6}{h_n}\left(y_n' - \frac{y_n - y_{n-1}}{h_n}\right). \tag{2.72}$$

The system of equations has the form

$$\begin{bmatrix} 2 & \lambda_0 & & & & \\ \mu_1 & 2 & \lambda_1 & & & \\ & \mu_2 & 2 & \lambda_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & & \mu_n & 2 \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ \vdots \\ M_{n-1} \\ M_n \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix}. \tag{2.73}$$

For periodic boundary conditions we define

$$\lambda_n = \frac{h_1}{h_1 + h_n} \quad \mu_n = 1 - \lambda_n \quad d_n = \frac{6}{h_1 + h_n}\left(\frac{y_1 - y_n}{h_1} - \frac{y_n - y_{n-1}}{h_n}\right) \tag{2.74}$$

and the system of equations is (with $M_n = M_0$)

$$\begin{bmatrix} 2 & \lambda_1 & & & & \mu_1 \\ \mu_2 & 2 & \lambda_2 & & & \\ & \mu_3 & 2 & \lambda_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mu_{n-1} & 2 & \lambda_{n-1} \\ \lambda_n & & & & \mu_n & 2 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ \vdots \\ M_{n-1} \\ M_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix}. \tag{2.75}$$

All this tridiagonal systems can be easily solved with a special Gaussian elimination method (Sects. 5.3 and 5.4)

2.4 Rational Interpolation

The use of rational approximants allows to interpolate functions with poles, where polynomial interpolation can give poor results [2]. Rational approximants without poles [11] are also well suited for the case of equidistant x_i, where higher order polynomials tend to become unstable. The main disadvantages are additional poles which are difficult to control and the appearance of unattainable points. Recent developments using the barycentric form of the interpolating function [11–13] helped to overcome these difficulties.

2.4.1 Padé Approximant

The Padé approximant [14] of order $[M/N]$ to a function $f(x)$ is the rational function

$$R_{M/N}(x) = \frac{P_M(x)}{Q_N(x)} = \frac{p_0 + p_1 x + \ldots p_M x^M}{q_0 + q_1 x + \ldots q_N x^N} \tag{2.76}$$

which reproduces the McLaurin series (the Taylor series at $x = 0$) of

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots \tag{2.77}$$

up to order $M + N$, i.e.

$$\begin{aligned} & f(0) = R(0) \\ & \frac{\mathrm{d}}{\mathrm{d}x} f(0) = \frac{\mathrm{d}}{\mathrm{d}x} R(0) \\ & \vdots \\ & \frac{\mathrm{d}^{(M+N)}}{\mathrm{d}x^{(M+N)}} f(0) = \frac{\mathrm{d}^{(M+N)}}{\mathrm{d}x^{(M+N)}} R(0). \end{aligned} \tag{2.78}$$

Multiplication gives

$$p_0 + p_1 x + \cdots + p_M x^M = (q_0 + q_1 x + \cdots + q_N x^N)(a_0 + a_1 x + \ldots) \tag{2.79}$$

and collecting powers of x we find the system of equations

$$\begin{aligned} p_0 &= q_0 a_0 \\ p_1 &= q_0 a_1 + q_1 a_0 \\ p_2 &= q_0 a_2 + a_1 q_1 + a_0 q_2 \\ &\vdots \\ p_M &= q_0 a_M + a_{M-1} q_1 + \cdots + a_0 q_M \\ 0 &= q_0 a_{M+1} + q_1 a_M + \cdots + q_N a_{M-N+1} \\ &\vdots \\ 0 &= q_0 a_{M+N} + q_1 a_{M+N-1} + \cdots + q_N a_M \end{aligned} \tag{2.80}$$

where

$$a_n = 0 \quad \text{for } n < 0 \tag{2.81}$$

$$q_j = 0 \quad \text{for } j > N. \tag{2.82}$$

Example: Calculate the [3, 3] approximant to $\tan(x)$.
The Laurent series of the tangent is

$$\tan(x) = x + \frac{1}{3}x^3 + \frac{2}{15}x^5 + \dots. \tag{2.83}$$

We set $q_0 = 1$. Comparison of the coefficients of the polynomial

$$p_0 + p_1 x + p_2 x^2 + p_3 x^3 = (1 + q_1 x + q_2 x^2 + q_3 x^3)\left(x + \frac{1}{3}x^3 + \frac{2}{15}x^5\right) \tag{2.84}$$

gives the equations

$$\begin{aligned} x^0 &: p_0 = 0 \\ x^{1:} &\quad p_1 = 1 \\ x^2 &: p_2 = q_1 \\ x^3 &: p_3 = q_2 + \tfrac{1}{3} \\ x^4 &: 0 \;\; = q_3 + \tfrac{1}{3}q_1 \\ x^5 &: 0 \;\; = \tfrac{2}{15} + \tfrac{1}{3}q_2 \\ x^6 &: 0 \;\; = \tfrac{2}{15}q_1 + \tfrac{1}{3}q_3. \end{aligned} \tag{2.85}$$

We easily find

$$p_2 = q_1 = q_3 = 0 \quad q_2 = -\frac{2}{5} \quad p_3 = -\frac{1}{15} \tag{2.86}$$

and the approximant of order [3, 3] is

$$R_{3,3} = \frac{x - \frac{1}{15}x^3}{1 - \frac{2}{5}x^2}. \tag{2.87}$$

This expression reproduces the tangent quite well (Fig. 2.4). Its pole at $\sqrt{10}/2 \approx 1.581$ is close to the pole of the tangent function at $\pi/2 \approx 1.571$.

2.4.2 *Barycentric Rational Interpolation*

If the weights of the barycentric form of the interpolating polynomial (2.21) are taken as general parameters $u_i \neq 0$ it becomes a rational function

$$R(x) = \frac{\sum_{i=0}^{n} f_i \frac{u_i}{x - x_i}}{\sum_{i=0}^{n} \frac{u_i}{x - x_i}} \tag{2.88}$$

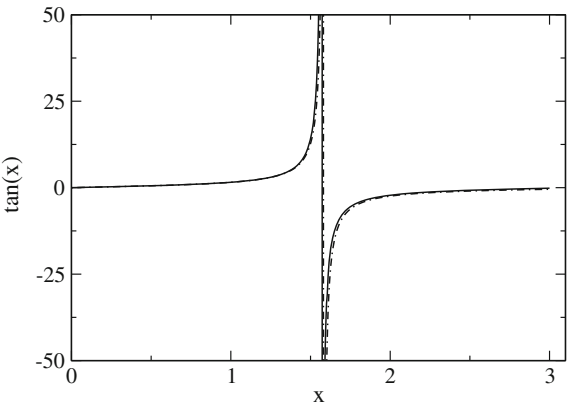


Fig. 2.4 (Padé approximation to $\tan(x)$) The Padé approximant (2.87, *dash dotted curve*) reproduces the tangent (*full curve*) quite well

which obviously interpolates the data points since

$$\lim_{x \to x_i} R(x) = f_i. \tag{2.89}$$

With the polynomials[1]

$$P(x) = \sum_{i=0}^{n} u_i f_i \prod_{j=0; j \neq i}^{n} (x - x_j) = \sum_{i=0}^{n} u_i f_i \frac{\omega(x)}{x - x_i}$$

$$Q(x) = \sum_{i=0}^{n} u_i \prod_{j=0; j \neq i}^{n} (x - x_j) = \sum_{i=0}^{n} u_i \frac{\omega(x)}{x - x_i}$$

a rational interpolating function is given by[2]

$$R(x) = \frac{P(x)}{Q(x)}.$$

Obviously there are infinitely different rational interpolating functions which differ by the weights $\mathbf{u} = (u_0, u_1 \ldots u_n)$ (an example is shown in Fig. 2.5). To fix the parameters u_i, additional conditions have to be imposed.

2.4.2.1 Rational Interpolation of Order [M, N]

One possibility is to assume that $P(x)$ and $Q(x)$ are of order $\leq M$ and $\leq N$, respectively with $M + N = n$. This gives n additional equations for the $2(n + 1)$

[1] $\omega(x) = \prod_{i=0}^{n} (x - x_i)$ as in (2.39).

[2] It can be shown that any rational interpolant can be written in this form.

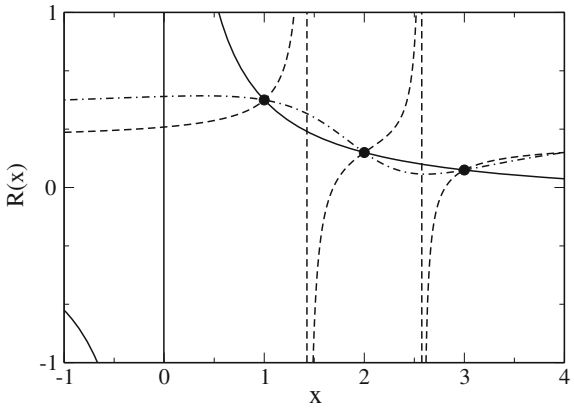


Fig. 2.5 (Rational interpolation) The data points $(1, \frac{1}{2})$, $(2, \frac{1}{5})$, $(3, \frac{1}{10})$ are interpolated by several rational functions. The [1, 1] approximant (2.95) corresponding to $\mathbf{u} = (5, -20, 15)$ is shown by the *solid curve*, the *dashed curve* shows the function $R(x) = \frac{8x^2-36x+38}{10(3x^2-12x+11)}$ which is obtained for $\mathbf{u} = (1, 1, 1)$ and the *dash dotted curve* shows the function $R(x) = \frac{4x^2-20x+26}{10(5-4x+x^2)}$ which follows for $\mathbf{u} = (1, -1, 1)$ and has no real poles

polynomial coefficients. The number of unknown equals $n + 1$ and the rational interpolant is uniquely determined up to a common factor in numerator and denominator.

Example Consider the data points $f(1) = \frac{1}{2}$, $f(2) = \frac{1}{5}$, $f(3) = \frac{1}{10}$.
The polynomials are

$$\begin{aligned} P(x) &= \frac{1}{2}u_0(x-2)(x-3) + \frac{1}{5}u_1(x-1)(x-3) + \frac{1}{10}u_2(x-1)(x-2) \\ &= 3u_0 + \frac{3}{5}u_1 + \frac{1}{5}u_2 + \left[-\frac{5}{2}u_0 - \frac{4}{5}u_1 - \frac{3}{10}u_2\right]x + \left[\frac{1}{2}u_0 + \frac{1}{5}u_1 + \frac{1}{10}u_2\right]x^2 \end{aligned} \tag{2.90}$$

$$\begin{aligned} Q(x) &= u_0(x-2)(x-3) + u_1(x-1)(x-3) + u_2(x-1)(x-2) \\ &= 6u_0 + 3u_1 + 2u_2 + [-5u_0 - 4u_1 - 3u_2]\,x + [u_0 + u_1 + u_2]\,x^2. \end{aligned} \tag{2.91}$$

To obtain a [1, 1] approximant we have to solve the equations

$$\frac{1}{2}u_0 + \frac{1}{5}u_1 + \frac{1}{10}u_2 = 0 \tag{2.92}$$

$$u_0 + u_1 + u_2 = 0 \tag{2.93}$$

which gives

$$u_2 = 3u_0 \quad u_1 = -4u_0 \tag{2.94}$$

and thus

$$R(x) = \frac{\frac{6}{5}u_0 - \frac{1}{5}u_0 x}{2u_0 x} = \frac{6-x}{10x}. \tag{2.95}$$

General methods to obtain the coefficients u_i for a given data set are described in [12, 13]. They also allow to determine unattainable points corresponding to $u_i = 0$ and to locate the poles. Without loss of generality it can be assumed [13] that $M \geq N$.[3]

Let $P(x)$ be the unique polynomial which interpolates the product $f(x)Q(x)$

$$P(x_i) = f(x_i)Q(x_i) \quad i = 0 \ldots M. \tag{2.96}$$

Then from (2.31) we have

$$f(x)Q(x) - P(x) = (fQ)[x_0 \cdots x_M, x](x - x_0) \cdots (x - x_M). \tag{2.97}$$

Setting

$$x = x_i \quad i = M+1, \ldots n \tag{2.98}$$

we have

$$f(x_i)Q(x_i) - P(x_i) = (fQ)[x_0 \ldots x_M, x_i](x_i - x_0) \ldots (x - x_M) \tag{2.99}$$

which is zero if $P(x_i)/Q(x_i) = f_i$ for $i = 0, \ldots n$. But then

$$(fQ)[x_0 \ldots x_M, x_i] = 0 \quad i = M+1, \ldots n. \tag{2.100}$$

The polynomial $Q(x)$ can be written in Newtonian form (2.30)

$$Q(x) = \sum_{i=0}^{N} \nu_i \prod_{j=0}^{i-1} (x - x_j) = \nu_0 + \nu_1(x - x_0) + \cdots + \nu_N(x - x_0) \ldots (x - x_{N-1}). \tag{2.101}$$

With the abbreviation

$$g_j(x) = x - x_j \quad j = 0 \ldots N \tag{2.102}$$

we find

[3] The opposite case can be treated by considering the reciprocal function values $1/f(x_i)$.

$$(fg_j)[x_0 \dots x_M, x_i] = \sum_{k=0\dots M,i} \frac{f(x_k)g(x_k)}{\prod_{r\neq k}(x_k - x_r)} = \sum_{k=0\dots M,i,k\neq j} \frac{f(x_k)}{\prod_{r\neq k, r\neq j}(x_k - x_r)}$$
$$= f[x_0 \dots x_{j-1}, x_{j+1} \dots x_M, x_i] \tag{2.103}$$

which we apply repeatedly to (2.100) to get the system of $n - M = N$ equations for $N + 1$ unknowns

$$\sum_{j=0}^{N} \nu_j f[x_j, x_{j+1} \dots x_M, x_i] = 0 \quad i = M + 1 \dots n \tag{2.104}$$

from which the coefficients ν_j can be found by Gaussian elimination up to a scaling factor. The Newtonian form of $Q(x)$ can then be converted to the barycentric form as described in [6].

2.4.2.2 Rational Interpolation without Poles

Polynomial interpolation of larger data sets can be ill behaved, especially for the case of equidistant $x-$values. Rational interpolation without poles can be a much better choice here (Fig. 2.6).

Berrut [15] suggested to choose the following weights

$$u_k = (-1)^k.$$

With this choice $Q(x)$ has no real roots. Floater and Horman [11] used the different choice

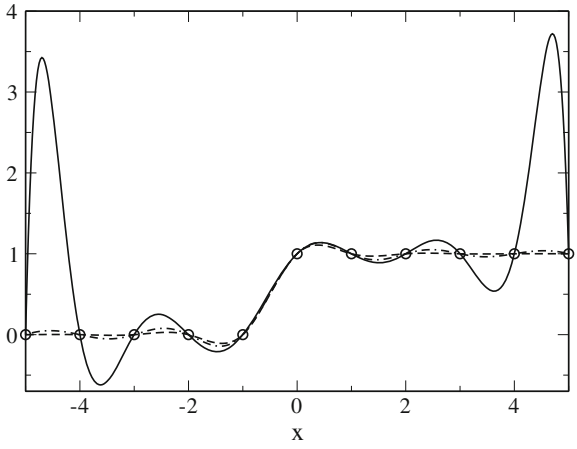


Fig. 2.6 (Interpolation of a step function) A step function with uniform x-values (*circles*) is interpolated by a polynomial (*full curve*), a cubic spline (*dashed curve*) and with the rational Floater–Horman $d = 1$ function (2.105, *dash-dotted curve*). The rational function behaves similar to the spline function but provides in addition an analytical function with continuous derivatives

Table 2.1 Floater-Horman weights for uniform data

$\lvert u_k \rvert$	d
1, 1, 1 . . . , 1, 1, 1	0
1, 2, 2, 2, . . . , 2, 2, 2, 1	1
1, 3, 4, 4, 4, . . . , 4, 4, 4, 3, 1	2
1, 4, 7, 8, 8, 8, . . . , 8, 8, 8, 7, 4, 1	3
1, 5, 11, 15, 16, 16, 16, . . . , 16, 16, 16, 15, 11, 5, 1	4

$$u_k = (-1)^{k-1}\left(\frac{1}{x_{k+1}-x_k}+\frac{1}{x_k-x_{k-1}}\right) \quad k = 1\ldots n-1$$
$$u_0 = -\frac{1}{x_1-x_0} \quad u_n = (-1)^{n-1}\frac{1}{x_n-x_{n-1}} \tag{2.105}$$

which becomes very similar for equidistant x-values.

Floater and Horman generalized this expression and found a class of rational interpolants without poles given by the weights

$$u_k = (-1)^{k-d}\sum_{i=\max(k-d,0)}^{\min(k,n-d)}\prod_{j=i,j\neq k}^{i+d}\frac{1}{|x_k-x_j|} \tag{2.106}$$

where $0 \le d \le n$ and the approximation order increases with d. In the uniform case this simplifies to (Table 2.1)

$$u_k = (-1)^{k-d}\sum_{i=\min(k-d,0)}^{\max(k,n-d)}\binom{d}{k-i}. \tag{2.107}$$

2.5 Multivariate Interpolation

The simplest 2-dimensional interpolation method is bilinear interpolation.[4] It uses linear interpolation for both coordinates within the rectangle $x_i \le x \le x_{i+1} \quad y_i \le y_i \le y_{i+1}$:

$$p(x_i+h_x, y_i+h_y) = p(x_i+h_x, y_i) + h_y\frac{p(x_i+h_x, y_{i+1}) - p(x_i+h_x, y_i)}{y_{i+1}-y_i}$$
$$= f(x_i, y_i) + h_x\frac{f(x_{i+1}, y_i) - f(x_i, y_i)}{x_{i+1}-x_i} \tag{2.108}$$

[4]Bilinear means linear interpolation in two dimensions. Accordingly linear interpolation in three dimensions is called trilinear.

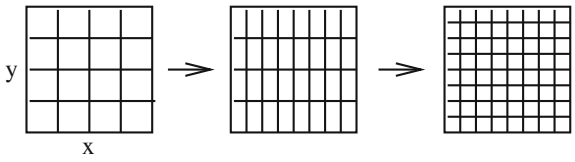


Fig. 2.7 Bispline interpolation

$$+h_y \frac{f(x_i, y_{i+1}) + h_x \frac{f(x_{i+1}, y_{i+1}) - f(x_i, y_{i+1})}{x_{i+1} - x_i} - f(x_i, y_i) - h_x \frac{f(x_{i+1}, y_i) - f(x_i, y_i)}{x_{i+1} - x_i}}{y_{i+1} - y_i}$$

which can be written as a two dimensional polynomial

$$p(x_i + h_x, y_i + h_y) = a_{00} + a_{10}h_x + a_{01}h_y + a_{11}h_x h_y \tag{2.109}$$

with

$$\begin{aligned} a_{00} &= f(x_i, y_i) \\ a_{10} &= \frac{f(x_{i+1}, y_i) - f(x_i, y_i)}{x_{i+1} - x_i} \\ a_{01} &= \frac{f(x_i, y_{i+1}) - f(x_i, y_i)}{y_{i+1} - y_i} \\ a_{11} &\frac{f(x_{i+1}, y_{i+1}) - f(x_i, y_{i+1}) - f(x_{i+1}, y_i) + f(x_i, y_i)}{(x_{i+1} - x_i)(y_{i+1} - y_i)}. \end{aligned} \tag{2.110}$$

Application of higher order polynomials is straightforward. For image processing purposes bicubic interpolation is often used.

If high quality is needed more sophisticated interpolation methods can be applied. Consider for instance two-dimensional spline interpolation on a rectangular mesh of data to create a new data set with finer resolution[5]

$$f_{i,j} = f(ih_x, jh_y) \text{ with } 0 \le i < N_x \quad 0 \le j < N_y. \tag{2.111}$$

First perform spline interpolation in x-direction for each data row j to calculate new data sets

$$f_{i',j} = s(x_{i'}, f_{ij}, 0 \le i < N_x) \quad 0 \le j \le N_y \quad 0 \le i' < N'_x \tag{2.112}$$

and then interpolate in y direction to obtain the final high resolution data (Fig. 2.7)

$$f_{i',j'} = s(y_{j'}, f_{i'j}, 0 \le j < N_y) \quad 0 \le i' < N'_x \quad 0 \le j' < N'_y. \tag{2.113}$$

[5] A typical task of image processing.

Problems

Problem 2.1 Polynomial Interpolation

This computer experiment interpolates a given set of n data points by

- a polynomial

$$p(x) = \sum_{i=0}^{n} f_i \prod_{k=0, k \neq i}^{n} \frac{x - x_k}{x_i - x_k}, \tag{2.114}$$

- a linear spline which connects successive points by straight lines

$$s_i(x) = a_i + b_i(x - x_i) \text{ for } x_i \leq x \leq x_{i+1} \tag{2.115}$$

- a cubic spline with natural boundary conditions

$$s(x) = p_i(x) = \alpha_i + \beta_i(x - x_i) + \gamma_i(x - x_i)^2 + \delta_i(x - x_i)^3 \quad x_i \leq x \leq x_{i+1} \tag{2.116}$$

$$s''(x_n) = s''(x_0) = 0 \tag{2.117}$$

- a rational function without poles

$$R(x) = \frac{\sum_{i=0}^{n} f_i \frac{u_i}{x - x_i}}{\sum_{i=0}^{n} \frac{u_i}{x - x_i}} \tag{2.118}$$

with weights according to Berrut

$$u_k = (-1)^k \tag{2.119}$$

or Floater–Hormann

$$u_k = (-1)^{k-1} \left(\frac{1}{x_{k+1} - x_k} + \frac{1}{x_k - x_{k-1}} \right) \quad k = 1 \ldots n - 1 \tag{2.120}$$

$$u_0 = -\frac{1}{x_1 - x_0} \quad u_n = (-1)^{n-1} \frac{1}{x_n - x_{n-1}}. \tag{2.121}$$

Table 2.2 Zener diode voltage/current data

Voltage	−1.5	−1.0	−0.5	0.0
Current	−3.375	−1.0	−0.125	0.0

Table 2.3 Additional voltage/current data

Voltage	1.0	2.0	3.0	4.0	4.1	4.2	4.5
Current	0.0	0.0	0.0	0.0	1.0	3.0	10.0

Table 2.4 Pulse and step function data

x	-3	-2	-1	0	1	2	3
y_{pulse}	0	0	0	1	0	0	0
y_{step}	0	0	0	1	1	1	1

Table 2.5 Data set for two-dimensional interpolation

x	0	1	2	0	1	2	0	1	2
y	0	0	0	1	1	1	2	2	2
f	1	0	-1	0	0	0	-1	0	1

- Interpolate the data (Table 2.2) in the range

$$-1.5 < x < 0.$$

- Now add some more sample points (Table 2.3) for $-1.5 < x < 4.5$
- Interpolate the function $f(x) = \sin(x)$ at the points $x = 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}, 2\pi$. Take more sample points and check if the quality of the fit is improved.
- Investigate the oscillatory behavior for a discontinuous pulse or step function as given by the data (Table 2.4)

Problem 2.3 Two-dimensional Interpolation

This computer experiment uses bilinear interpolation or bicubic spline interpolation to interpolate the data (Table 2.5)
on a finer grid $\Delta x = \Delta y = 0.1$.

Chapter 3
Numerical Differentiation

For more complex problems analytical derivatives are not always available and have to be approximated by numerical methods. Numerical differentiation is also very important for the discretization of differential equations (Sect. 12.2). The simplest approximation uses a forward difference quotient (Fig. 3.1) and is not very accurate. A symmetric difference quotient improves the quality. Even higher precision is obtained with the extrapolation method. Approximations to higher order derivatives can be obtained systematically with the help of polynomial interpolation.

3.1 One-Sided Difference Quotient

The simplest approximation of a derivative is the ordinary difference quotient which can be taken forward

$$\frac{\mathrm{d}f}{\mathrm{d}x}(x) \approx \frac{\Delta f}{\Delta x} = \frac{f(x+h) - f(x)}{h} \tag{3.1}$$

or backward

$$\frac{\mathrm{d}f}{\mathrm{d}x}(x) \approx \frac{\Delta f}{\Delta x} = \frac{f(x) - f(x-h)}{h}. \tag{3.2}$$

Its truncation error can be estimated from the Taylor series expansion

$$\begin{aligned} \frac{f(x+h) - f(x)}{h} &= \frac{f(x) + hf'(x) + \frac{h^2}{2} f''(x) + \cdots - f(x)}{h} \\ &= f'(x) + \frac{h}{2} f''(x) + \cdots . \end{aligned} \tag{3.3}$$

The error order is $O(h)$. The step width should not be too small to avoid rounding errors. Error analysis gives

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_3

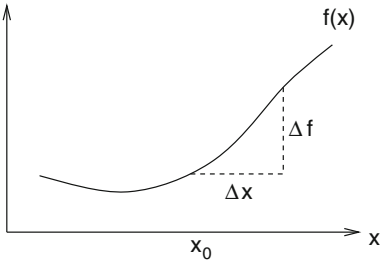


Fig. 3.1 (Numerical differentiation) Numerical differentiation approximates the differential quotient by a difference quotient $\frac{\mathrm{d}f}{\mathrm{d}x} \approx \frac{\Delta f}{\Delta x}$. However, approximation by a simple forward difference $\frac{\mathrm{d}f}{\mathrm{d}x}(x_0) \approx \frac{f(x_0+h)-f(x_0)}{h}$, is not very accurate

$$\begin{aligned}\widetilde{\Delta f} &= fl_-\left(f(x+h)(1+\varepsilon_1),\, f(x)(1+\varepsilon_2)\right)\\ &= (\Delta f + f(x+h)\varepsilon_1 - f(x)\varepsilon_2)(1+\varepsilon_3)\\ &= \Delta f + \Delta f\varepsilon_3 + f(x+h)\varepsilon_1 - f(x)\varepsilon_2 + \cdots\end{aligned} \tag{3.4}$$

$$\begin{aligned}fl_\div(\widetilde{\Delta f}, h(1+\varepsilon_4)) &= \frac{\Delta f + \Delta f\varepsilon_3 + f(x+h)\varepsilon_1 - f(x)\varepsilon_2}{h(1+\varepsilon_4)}(1+\varepsilon_5)\\ &= \frac{\Delta f}{h}(1+\varepsilon_5-\varepsilon_4+\varepsilon_3) + \frac{f(x+h)}{h}\varepsilon_1 - \frac{f(x)}{h}\varepsilon_2.\end{aligned} \tag{3.5}$$

The errors are uncorrelated and the relative error of the result can be estimated by

$$\frac{\left|\frac{\widetilde{\Delta f}}{\Delta x} - \frac{\Delta f}{\Delta x}\right|}{\frac{\Delta f}{\Delta x}} \leq 3\varepsilon_M + \left|\frac{f(x)}{\frac{\Delta f}{\Delta x}}\right| 2\frac{\varepsilon_M}{h}. \tag{3.6}$$

Numerical extinction produces large relative errors for small step width h. The optimal value of h gives comparable errors from rounding and truncation. It can be found from

$$\frac{h}{2}|f''(x)| = |f(x)|\frac{2\varepsilon_M}{h}. \tag{3.7}$$

Assuming that the magnitude of the function and the derivative are comparable, we have the rule of thumb

$$h_o = \sqrt{\varepsilon_M} \approx 10^{-8}$$

(double precision). The corresponding relative error is of the same order.

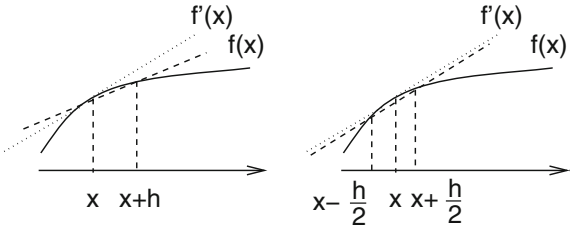


Fig. 3.2 (Difference quotient) The central difference quotient (***Right side***) approximates the derivative (*dotted*) much more accurately than the one-sided difference quotient (***Left side***)

3.2 Central Difference Quotient

Accuracy is much higher if a symmetric central difference quotient is used (Fig. 3.2):

$$\begin{aligned}\frac{\Delta f}{\Delta x} &= \frac{f(x+\frac{h}{2}) - f(x-\frac{h}{2})}{h} \\ &= \frac{f(x) + \frac{h}{2}f'(x) + \frac{h^2}{8}f''(x) + \cdots - \quad f(x) - \frac{h}{2}f'(x) + \frac{h^2}{8}f''(x) + \cdots}{h} \\ &= f'(x) + \frac{h^2}{24}f'''(x) + \cdots . \end{aligned} \tag{3.8}$$

The error order is $O(h^2)$. The optimal step width is estimated from

$$\frac{h^2}{24}|f'''(x)| = |f(x)|\frac{2\varepsilon_M}{h} \tag{3.9}$$

again with the assumption that function and derivatives are of similar magnitude as

$$h_0 =^3 \quad \overline{48\varepsilon_M} \approx 10^{-5}. \tag{3.10}$$

The relative error has to be expected in the order of $\frac{h_0^2}{24} \approx 10^{-11}$.

3.3 Extrapolation Methods

The Taylor series of the symmetric difference quotient contains only even powers of h:

$$D(h) = \frac{f(x+h) - f(x-h)}{2h} = f'(x) + \frac{h^2}{3!}f'''(x) + \frac{h^4}{5!}f^{(5)}(x) + \cdots . \tag{3.11}$$

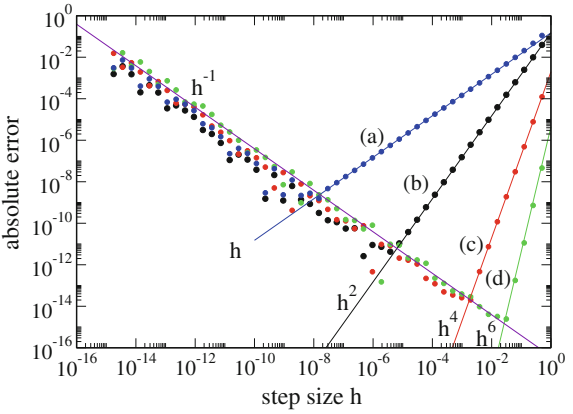


Fig. 3.3 (Numerical differentiation) The derivative $\frac{\mathrm{d}}{\mathrm{d}x}\sin(x)$ is calculated numerically using algorithms with increasing error order (3.1, 3.8, 3.14, 3.18). For very small step sizes the error increases as h^{-1} due to rounding errors (Problem 3.1)

The Extrapolation method [16] uses a series of step widths, e.g.

$$h_{i+1} = \frac{h_i}{2} \tag{3.12}$$

and calculates an estimate of $D(0)$ by polynomial interpolation (Fig. 3.3). Consider $D_0 = D(h_0)$ and $D_1 = D(\frac{h_0}{2})$. The polynomial of degree 1 (with respect to h^2) $p(h) = a + bh^2$ can be found by the Lagrange method

$$p(h) = D_0 \frac{h^2 - \frac{h_0^2}{4}}{h_0^2 - \frac{h_0^2}{4}} + D_1 \frac{h^2 - h_0^2}{\frac{h_0^2}{4} - h_0^2}. \tag{3.13}$$

Extrapolation for h = 0 gives

$$p(0) = -\frac{1}{3} D_0 + \frac{4}{3} D_1. \tag{3.14}$$

Taylor series expansion shows

$$p(0) = -\frac{1}{3}\left(f'(x) + \frac{h_0^2}{3!} f'''(x) + \frac{h_0^4}{5!} f^{(5)}(x) + \cdots\right) +$$
$$+ \frac{4}{3}\left(f'(x) + \frac{h_0^2}{4 \cdot 3!} f'''(x) + \frac{h_0^4}{16 \cdot 5!} f^{(5)}(x) + \cdots\right) \tag{3.15}$$
$$= f'(x) - \frac{1}{4}\frac{h_0^4}{5!} f^{(5)}(x) + \cdots \tag{3.16}$$

that the error order is $O(h_0^4)$. For 3 step widths $h_0 = 2h_1 = 4h_2$ we obtain the polynomial of second order (in h^2)

$$p(h) = D_0 \frac{(h^2 - \frac{h_0^2}{4})(h^2 - \frac{h_0^2}{16})}{(h_0^2 - \frac{h_0^2}{4})(h_0^2 - \frac{h_0^2}{16})} + D_1 \frac{(h^2 - h_0^2)(h^2 - \frac{h_0^2}{16})}{(\frac{h_0^2}{4} - h_0^2)(\frac{h_0^2}{4} - \frac{h_0^2}{16})} + D_2 \frac{(h^2 - h_0^2)(h^2 - \frac{h_0^2}{4})}{(\frac{h_0^2}{16} - h_0^2)(\frac{h_0^2}{16} - \frac{h_0^2}{4})} \tag{3.17}$$

and the improved expression

$$p(0) = D_0 \frac{\frac{1}{64}}{\frac{3}{4} \cdot \frac{15}{16}} + D_1 \frac{\frac{1}{16}}{\frac{-3}{4} \cdot \frac{3}{16}} + D_2 \frac{\frac{1}{4}}{\frac{-15}{16} \cdot \frac{-3}{16}} =$$

$$= \frac{1}{45} D_0 - \frac{4}{9} D_1 + \frac{64}{45} D_2 = f'(x) + O(h_0^6). \tag{3.18}$$

Often used is the following series of step widths:

$$h_i^2 = \frac{h_0^2}{2^i}. \tag{3.19}$$

The Neville method

$$P_{i\cdots k}(h^2) = \frac{(h^2 - \frac{h_0^2}{2^i}) P_{i+1\cdots k}(h^2) - (h^2 - \frac{h_0^2}{2^k}) P_{i\cdots k-1}(h^2)}{\frac{h_0^2}{2^k} - \frac{h_0^2}{2^i}} \tag{3.20}$$

gives for h=0

$$P_{i\cdots k} = \frac{P_{i\cdots k-1} - 2^{k-i} P_{i+1\cdots k}}{1 - 2^{k-i}} \tag{3.21}$$

which can be written as

$$P_{i\cdots k} = P_{i+1\cdots k} + \frac{P_{i\cdots k-1} - P_{i+1\cdots k}}{1 - 2^{k-i}} \tag{3.22}$$

and can be calculated according to the following scheme:

$$\begin{array}{llll} P_0 = D(h^2) & P_{01} & P_{012} & P_{0123} \\ P_1 = D(\frac{h^2}{2}) & P_{12} & P_{123} & \\ P_2 = D(\frac{h^2}{4} & P_{23} & & \\ \vdots & \vdots & \vdots & \ddots \end{array} \tag{3.23}$$

Here the values of the polynomials are arranged in matrix form

$$P_{i\cdots k} = T_{i,k-i} = T_{i,j} \tag{3.24}$$

with the recursion formula

$$T_{i,j} = T_{i+1,j-1} + \frac{T_{i,j-1} - T_{i+1,j}}{1 - 2^j}. \tag{3.25}$$

3.4 Higher Derivatives

Difference quotients for higher derivatives can be obtained systematically using polynomial interpolation. Consider equidistant points

$$x_n = x_0 + nh = \cdots x_0 - 2h, x_0 - h, x_0, x_0 + h, x_0 + 2h, \cdots . \tag{3.26}$$

From the second order polynomial

$$\begin{aligned} p(x) &= y_{-1}\frac{(x - x_0)(x - x_1)}{(x_{-1} - x_0)(x_{-1} - x_1)} + y_0\frac{(x - x_{-1})(x - x_1)}{(x_0 - x_{-1})(x_0 - x_1)} \\ &+ y_1\frac{(x - x_{-1})(x - x_0)}{(x_1 - x_{-1})(x_1 - x_0)} = \\ &= y_{-1}\frac{(x - x_0)(x - x_1)}{2h^2} + y_0\frac{(x - x_{-1})(x - x_1)}{-h^2} \\ &+ y_1\frac{(x - x_{-1})(x - x_0)}{2h^2} \end{aligned} \tag{3.27}$$

we calculate the derivatives

$$p'(x) = y_{-1}\frac{2x - x_0 - x_1}{2h^2} + y_0\frac{2x - x_{-1} - x_1}{-h^2} + y_1\frac{2x - x_{-1} - x_0}{2h^2} \tag{3.28}$$

$$p''(x) = \frac{y_{-1}}{h^2} - 2\frac{y_0}{h^2} + \frac{y_1}{h^2} \tag{3.29}$$

which are evaluated at x_0:

$$f'(x_0) \approx p'(x_0) = -\frac{1}{2h}y_{-1} + \frac{1}{2h}y_1 = \frac{f(x_0 + h) - f(x_0 - h)}{2h} \tag{3.30}$$

$$f''(x_0) \approx p''(x_0) = \frac{f(x_0 - h) - 2f(x_0) + f(x_0 + h)}{h^2}. \tag{3.31}$$

Higher order polynomials can be evaluated with an algebra program. For five sample points

$$x_0 - 2h, x_0 - h, x_0, x_0 + h, x_0 + 2h$$

we find

$$f'(x_0) \approx \frac{f(x_0-2h)-8f(x_0-h)+8f(x_0+h)-f(x_0+2h)}{12h} \tag{3.32}$$

$$f''(x_0) \approx \frac{-f(x_0-2h)+16f(x_0-h)-30f(x_0)+16f(x_0+h)-f(x_0+2h)}{12h^2} \tag{3.33}$$

$$f'''(x_0) \approx \frac{-f(x_0-2h)+2f(x_0-h)-2f(x_0+h)+f(x_0+2h)}{2h^3} \tag{3.34}$$

$$f^{(4)}(x_0) \approx \frac{f(x_0-2h)-4f(x_0-h)+6f(x_0+h)-4f(x_0+h)+f(x_0+2h)}{h^4}.$$

3.5 Partial Derivatives of Multivariate Functions

Consider polynomials of more than one variable. In two dimensions we use the Lagrange polynomials

$$L_{i,j}(x,y) = \prod_{k \neq i} \frac{(x-x_k)}{(x_i-x_k)} \prod_{j \neq l} \frac{(y-y_l)}{(y_j-y_l)}. \tag{3.35}$$

The interpolating polynomial is

$$p(x,y) = \sum_{i,j} f_{i,j} L_{i,j}(x,y). \tag{3.36}$$

For the nine sample points

$$\begin{matrix} (x_{-1},y_1) & (x_0,y_1) & (x_1,y_1) \\ (x_{-1},y_0) & (x_0,y_0) & (x_1,y_0) \\ (x_{-1},y_{-1}) & (x_0,y_{-1}) & (x_1,y_{-1}) \end{matrix} \tag{3.37}$$

we obtain the polynomial

$$p(x,y) = f_{-1,-1} \frac{(x-x_0)(x-x_1)(y-y_0)(y-y_1)}{(x_{-1}-x_0)(x_{-1}-x_1)(y_{-1}-y_0)(y_{-1})-y_1)} + \cdots \tag{3.38}$$

which gives an approximation to the gradient

$$\mathrm{grad} f(x_0 y_0) \approx \mathrm{grad} p(x_0 y_0) = \begin{matrix} \frac{f(x_0+h,y_0)-f(x_0-h,y_0)}{2h} \\ \frac{f(x_0,y_0+h)-f(x_0,y_0-h)}{2h} \end{matrix}, \tag{3.39}$$

the Laplace operator

$$\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right) f(x_0, y_0) \approx \left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right) p(x_0, y_0)$$
$$= \frac{1}{h^2}\left(f(x_0, y_0+h)+f(x_0, y_0-h)+f(x_0, y_0+h)+f(x_0, y_0-h)-4f(x_0, y_0)\right) \tag{3.40}$$

and the mixed second derivative

$$\frac{\partial^2}{\partial x \partial y} f(x_0, y_0) \approx \frac{\partial^2}{\partial x \partial y} p(x_0, y_0)$$
$$= \frac{1}{4h^2}\left(f(x_0+h, y_0+h)+f(x_0-h, y_0-h)-f(x_0-h, y_0+h)-f(x_0+h, y_0-h)\right). \tag{3.41}$$

Problems

Problem 3.1 Numerical Differentiation

In this computer experiment we calculate the derivative of $f(x) = \sin(x)$ numerically with

- the single sided difference quotient

$$\frac{\mathrm{d}f}{\mathrm{d}x} \approx \frac{f(x+h)-f(x)}{h}, \tag{3.42}$$

- the symmetrical difference quotient

$$\frac{\mathrm{d}f}{\mathrm{d}x} \approx D_h f(x) = \frac{f(x+h)-f(x-h)}{2h}, \tag{3.43}$$

- higher order approximations which can be derived using the extrapolation method

$$-\frac{1}{3}D_h f(x)+\frac{4}{3}D_{h/2} f(x) \tag{3.44}$$
$$\frac{1}{45}D_h f(x)-\frac{4}{9}D_{h/2} f(x)+\frac{64}{45}D_{h/4} f(x). \tag{3.45}$$

The error of the numerical approximation is shown on a log-log plot as a function of the step width h.

Chapter 4
Numerical Integration

Physical simulations often involve the calculation of definite integrals over complicated functions, for instance the Coulomb interaction between two electrons. Integration is also the elementary step in solving equations of motion.

An integral over a finite interval $[a, b]$ *can always be transformed into an integral over* $[0, 1]$ or $[-1, 1]$

$$\int_a^b f(x)dx = \int_0^1 f(a + (b-a)t)\,(b-a)dt$$
$$= \int_{-1}^1 f\left(\frac{a+b}{2} + \frac{b-a}{2}t\right)\frac{b-a}{2}dt. \tag{4.1}$$

An Integral over an infinite interval may have to be transformed into an integral over a finite interval by substitution of the integration variable, for example

$$\int_0^\infty f(x)dx = \int_0^1 f\left(\frac{t}{1-t}\right)\frac{dt}{(1-t)^2} \tag{4.2}$$
$$\int_{-\infty}^\infty f(x)dx = \int_{-1}^1 f\left(\frac{t}{1-t^2}\right)\frac{t^2+1}{(t^2-1)^2}dt. \tag{4.3}$$

In general a definite integral can be approximated numerically as the weighted average over a finite number of function values

$$\int_a^b f(x)dx \approx \sum_{x_i} w_i f(x_i). \tag{4.4}$$

Specific sets of quadrature points x_i *and quadrature weights* w_i *are known as "integral rules". Newton–Cotes rules like the trapezoidal rule, the midpoint rule or Simpson's rule, use equidistant points* x_i *and are easy to apply. Accuracy can be improved by dividing the integration range into sub-intervals and applying*

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_4

composite Newton–Cotes rules. Extrapolation methods reduce the error almost to machine precision but need many function evaluations. Equidistant sample points are convenient but not the best choice. Clenshaw–Curtis expressions use non uniform sample points and a rapidly converging Chebyshev expansion. Gaussian integration fully optimizes the sample points with the help of orthogonal polynomials.

4.1 Equidistant Sample Points

For equidistant points

$$x_i = a + ih \quad i = 0 \ldots N \quad h = \frac{b-a}{N} \tag{4.5}$$

the interpolating polynomial of order N with $p(x_i) = f(x_i)$ is given by the Lagrange method

$$p(x) = \sum_{i=0}^{N} f_i \prod_{k=0, k\neq i}^{N} \frac{x - x_k}{x_i - x_k}. \tag{4.6}$$

Integration of the polynomial gives

$$\int_a^b p(x)dx = \sum_{i=0}^{N} f_i \int_a^b \prod_{k=0, k\neq i}^{N} \frac{x - x_k}{x_i - x_k} dx. \tag{4.7}$$

After substituting

$$\begin{aligned} x &= a + hs \\ x - x_k &= h(s - k) \\ x_i - x_k &= (i - k)h \end{aligned} \tag{4.8}$$

we have

$$\int_a^b \prod_{k=0, k\neq i}^{N} \frac{x - x_k}{x_i - x_k} dx = \int_0^N \prod_{k=0, k\neq i}^{N} \frac{s - k}{i - k} h ds = h\alpha_i \tag{4.9}$$

and hence

$$\int_a^b p(x)dx = (b-a) \sum_{i=0}^{N} f_i \alpha_i. \tag{4.10}$$

The weight factors are given by

$$w_i = (b-a)\alpha_i = Nh\alpha_i. \tag{4.11}$$

4.1.1 Closed Newton–Cotes Formulae

For $N = 1$ the polynomial is

$$p(x) = f_0 \frac{x - x_1}{x_0 - x_1} + f_1 \frac{x - x_0}{x_1 - x_0} \tag{4.12}$$

and the integral is

$$\begin{aligned} \int_a^b p(x)dx &= f_0 \int_0^1 \frac{s-1}{0-1} hds + f_1 \int_0^1 \frac{s-0}{1-0} hds \\ &= -f_0 h \left(\frac{(1-1)^2}{2} - \frac{(0-1)^2}{2} \right) + f_1 h \left(\frac{1^2}{2} - \frac{0^2}{2} \right) \\ &= h \frac{f_0 + f_1}{2} \end{aligned} \tag{4.13}$$

which is known as the trapezoidal rule (Fig. 4.1). $N = 2$ gives Simpson's rule

$$2h \, \frac{f_0 + 4f_1 + f_2}{6}. \tag{4.14}$$

Larger N give further integration rules

$$\begin{aligned} 3h \, \frac{f_0 + 3f_1 + 3f_2 + f_3}{8} & \qquad 3/8 - \text{rule} \\ 4h \, \frac{7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4}{90} & \qquad \text{Milne} - \text{rule} \\ 5h \, \frac{19f_0 + 75f_1 + 50f_2 + 50f_3 + 75f_4 + 19f_5}{288} & \\ 6h \, \frac{41f_0 + 216f_1 + 27f_2 + 272f_3 + 27f_4 + 216f_5 + 41f_6}{840} & \qquad \text{Weddle} - \text{rule}. \end{aligned} \tag{4.15}$$

For even larger N negative weight factors appear and the formulas are not numerically stable.

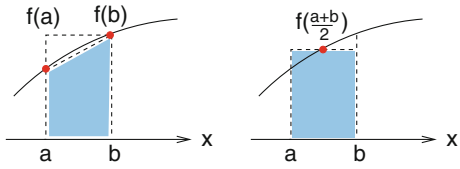


Fig. 4.1 (Trapezoidal rule and midpoint rule) The trapezoidal rule (***Left***) approximates the integral by the average of the function values at the boundaries. The midpoint rule (***Right***) evaluates the function in the center of the interval and has the same error order

4.1.2 Open Newton–Cotes Formulae

Alternatively, the integral can be computed from only interior points

$$x_i = a + ih \quad i = 1, 2, \ldots N \quad h = \frac{b-a}{N+1}. \tag{4.16}$$

The simplest case is the midpoint rule (Fig. 4.1)

$$\int_a^b f(x)dx \approx 2hf_1 = (b-a)f\left(\frac{a+b}{2}\right). \tag{4.17}$$

The next two are

$$\frac{3h}{2}\left(f_1 + f_2\right) \tag{4.18}$$

$$\frac{4h}{3}\left(2f_1 - f_2 + 2f_3\right). \tag{4.19}$$

4.1.3 Composite Newton–Cotes Rules

Newton–Cotes formulas are only accurate, if the step width is small. Usually the integration range is divided into small sub-intervals

$$[x_i, x_{i+1}] \quad x_i = a + ih \quad i = 0 \ldots N \tag{4.20}$$

for which a simple quadrature formula can be used. Application of the trapezoidal rule for each interval

$$I_i = \frac{h}{2}\Big(f(x_i) + f(x_{i+1})\Big) \tag{4.21}$$

gives the composite trapezoidal rule

$$T = h\left(\frac{f(a)}{2} + f(a+h) + \cdots f(b-h) + \frac{f(b)}{2}\right) \tag{4.22}$$

with error order $O(h^2)$. Repeated application of Simpson's rule for $[a, a+2h]$, $[a+2h, a+4h]\ldots$ gives the composite Simpson's rule

$$S = \frac{h}{3}\Big(f(a) + 4f(a+h) + 2f(a+2h) + 4f(a+3h) + \\ \cdots + 2f(b-2h) + 4f(b-h) + f(b)\Big) \tag{4.23}$$

with error order $O(h^4)$.[1]

Repeated application of the midpoint rule gives the composite midpoint rule

$$M = 2h\Big(f(a+h) + f(a+3h) + \cdots f(b-h)\Big) \tag{4.24}$$

with error order $O(h^2)$.

4.1.4 Extrapolation Method (Romberg Integration)

For the trapezoidal rule the Euler–McLaurin expansion exists which for a $2m$ times differentiable function has the form

$$\int_{x_0}^{x_N} f(x)dx - T = \alpha_2 h^2 + \alpha_4 h^4 + \cdots \alpha_{2m-2} h^{2m-2} + O(h^{2m}). \tag{4.25}$$

Therefore extrapolation methods are applicable. From the composite trapezoidal rule for h and $h/2$ an approximation of error order $O(h^4)$ results:

[1] The number of sample points must be even.

$$\int_{x_0}^{x_N} f(x)dx - T(h) = \alpha_2 h^2 + \alpha_4 h^4 + \cdots \tag{4.26}$$

$$\int_{x_0}^{x_N} f(x)dx - T(h/2) = \alpha_2 \frac{h^2}{4} + \alpha_4 \frac{h^4}{16} + \cdots \tag{4.27}$$

$$\int_{x_0}^{x_N} f(x)dx - \frac{4T(h/2) - T(h)}{3} = -\alpha_4 \frac{h^4}{4} + \cdots \tag{4.28}$$

More generally, for the series of step widths

$$h_k = \frac{h_0}{2^k} \tag{4.29}$$

the Neville method gives the recursion for the interpolating polynomial

$$P_{i\cdots k}(h^2) = \frac{(h^2 - \frac{h_0^2}{2^{2i}})P_{i+1\cdots k}(h^2) - (h^2 - \frac{h_0^2}{2^{2k}})P_{i\cdots k-1}(h^2)}{\frac{h_0^2}{2^{2k}} - \frac{h_0^2}{2^{2i}}} \tag{4.30}$$

which for $h = 0$ becomes the higher order approximation to the integral (Fig. 4.2)

$$\begin{aligned} P_{i\cdots k} &= \frac{2^{-2k}P_{i\cdots k-1} - 2^{-2i}P_{i+1\cdots k}}{2^{-2k} - 2^{-2i}} = \frac{P_{i\cdots k-1} - 2^{2k-2i}P_{i+1\cdots k}}{1 - 2^{2k-2i}} \\ &= P_{i+1\cdots k} + \frac{P_{i\cdots k-1} - P_{i+1\cdots k}}{1 - 2^{2k-2i}}. \end{aligned} \tag{4.31}$$

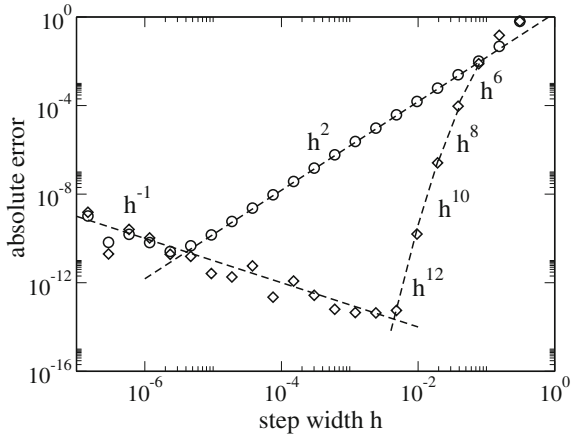


Fig. 4.2 (Romberg integration) The integral $\int_0^{\pi^2} \sin(x^2)dx$ is calculated numerically. *Circles* show the absolute error of the composite trapezoidal rule (4.22) for the step size sequence $h_{i+1} = h_i/2$. *Diamonds* show the absolute error of the extrapolated value (4.31). The error order of the trapezoidal rule is $O(h^2)$ whereas the error order of the Romberg method increases by factors of h^2. For very small step sizes the rounding errors dominate which increase as h^{-1}

The polynomial values can again be arranged in matrix form

$$\begin{matrix} P_0 & P_{01} & P_{012} & \cdots \\ P_1 & P_{12} & & \\ P_2 & & & \\ \vdots & & & \end{matrix} \tag{4.32}$$

with

$$T_{i,j} = P_{i\cdots i+j} \tag{4.33}$$

and the recursion formula

$$T_{i,0} = P_i = T_s\left(\frac{h_0}{2^i}\right) \tag{4.34}$$

$$T_{i,j} = T_{i+1,j-1} + \frac{T_{i,j-1} - T_{i+1,j-1}}{1 - 2^{2j}}. \tag{4.35}$$

4.2 Optimized Sample Points

The Newton–Cotes method integrates polynomials of order up to $N - 1$ exactly, using N equidistant sample points. Unfortunately the polynomial approximation converges slowly, at least for not so well behaved integrands. The accuracy of the integration can be improved by optimizing the sample point positions. Gaussian quadrature determines the N positions and N weights such, that a polynomial of order $2N - 1$ is integrated exactly. The Clenshaw–Curtis and the related Fejer methods use the roots or the extrema of the Chebyshev polynomials as nodes and determine the weights to integrate polynomials of order N. However, since the approximation by Chebyshev polynomials usually converges very fast, the accuracy is in many cases comparable to the Gaussian method [17, 18]. In the following we restrict the integration interval to $[-1, 1]$. The general case $[a, b]$ is then given by a simple change of variables.

4.2.1 Clenshaw–Curtis Expressions

Clenshaw and Curtis [19] make the variable substitution

$$x = \cos\theta \quad dx = -\sin\theta\, d\theta \tag{4.36}$$

for the integral

$$\int_{-1}^{1} f(x)dx = \int_{0}^{\pi} f(\cos t)\ \sin t\, dt \tag{4.37}$$

and approximate the function by the trigonometric polynomial (7.19 with $N = 2M, T = 2\pi$)

$$f(\cos t) = \frac{1}{2M}c_0 + \frac{1}{M}\sum_{j=1}^{M-1} c_j \cos(j\, t) + \frac{1}{2M}c_M \cos(Mt) \tag{4.38}$$

which interpolates (Sect. 7.2.1) $f(\cos t)$ at the sample points

$$t_n = n\Delta t = n\frac{\pi}{M} \quad \text{with} n = 0, 1, \ldots M \tag{4.39}$$

$$x_n = \cos t_n = \cos\left(n\frac{\pi}{M}\right) \tag{4.40}$$

and where the Fourier coefficients are given by (7.17)

$$c_j = f_0 + 2\sum_{n=1}^{M-1} f(\cos(t_n))\cos(\frac{\pi}{M}jn) + f_M \cos(j\pi). \tag{4.41}$$

The function $\cos(j\ t)$ is related to the Chebyshev polynomials of the first kind which for $-1 \leq x \leq 1$ are given by the trigonometric definition

$$T_j(x) = \cos(j\ \arccos(x)) \tag{4.42}$$

and can be calculated recursively

$$T_0(x) = 1 \tag{4.43}$$

$$T_1(x) = x \tag{4.44}$$

$$T_{j+1}(x) = 2x\, T_j(x) - T_{j-1}(x). \tag{4.45}$$

Substituting $x = \cos t$ we find

$$T_j(\cos t) = \cos(j\ t). \tag{4.46}$$

Hence the Fourier series (4.38) corresponds to a Chebyshev approximation

$$f(x) = \sum_{j=0}^{M} a_j T_j(x) = \frac{c_0}{2M} T_0(x) + \sum_{j=1}^{M-1} \frac{c_j}{M} T_j(x) + \frac{c_M}{2M} T_M(x) \tag{4.47}$$

and can be used to approximate the integral

$$\int_{-1}^{1} f(x)dx \approx \int_0^{\pi} \left\{ \frac{1}{2M} c_0 + \frac{1}{M} \sum_{j=1}^{M-1} c_j \cos(j\,t) + \frac{1}{2M} c_M \cos(Mt) \right\} \sin\theta \, d\theta \tag{4.48}$$

$$= \frac{1}{M} c_0 + \frac{1}{M} \sum_{j=1}^{M-1} c_j \frac{\cos(j\pi) + 1}{1 - j^2} + \frac{1}{2M} c_M \frac{\cos(M\pi) + 1}{1 - M^2} \tag{4.49}$$

where, in fact, only the even j contribute.

Example Clenshaw Curtis quadrature for $M = 5$

The function has to be evaluated at the sample points $x_k = \cos(\frac{\pi}{5}k) = (1, 0.80902, 0.30902, -0.30902, -0.80902, -1)$. The Fourier coefficients are given by

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 2 & 2 & 2 & 1 \\ 1 & 1.618 & 0.618 & -0.618 & -1.618 & -1 \\ 1 & 0.618 & -1.618 & -1.618 & 0.618 & 1 \\ 1 & -0.618 & -1.618 & 1.618 & 0.618 & -1 \\ 1 & -1.618 & 0.618 & 0.618 & -1.618 & 1 \\ 1 & -2 & 2 & -2 & 2 & -1 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} \tag{4.50}$$

and the integral is approximately

$$\int_{-1}^{1} f(x)dx \approx \left(\tfrac{1}{5} \; 0 \; -\tfrac{2}{15} \; 0 \; -\tfrac{2}{75} \; 0 \right) \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix}$$

$$= 0.0400\, f_0 + 0.3607\, f_1 + 0.5993\, f_2 + 0.5993\, f_3 + 0.3607\, f_4 + 0.0400\, f_5. \tag{4.51}$$

Clenshaw Curtis weights of very high order can be calculated efficiently [20, 21] using the FFT algorithm (fast Fourier transformation, Sect. 7.3.2).

4.2.2 Gaussian Integration

Now we will optimize the positions of the N quadrature points x_i to obtain the maximum possible accuracy. We approximate the integral by a sum

$$\int_a^b f(x)dx \approx \sum_{i=1}^{N} f(x_i)w_i \tag{4.52}$$

and determine the $2N$ parameters x_i and w_i such that a polynomial of order $2N-1$ is integrated exactly. This can be achieved with the help of a set of polynomials which are orthogonal with respect to the scalar product

$$< fg >= \int_a^b f(x)g(x)w(x)dx \tag{4.53}$$

where the weight function $w(x)$ and the interval $[a, b]$ determine a particular set of orthogonal polynomials.

4.2.2.1 Gauss–Legendre Integration

Again we restrict the integration interval to $[-1, 1]$ in the following. For integrals with one or two infinite boundaries see Sect. 4.2.2. The simplest choice for the weight function is

$$w(x) = 1. \tag{4.54}$$

An orthogonal system of polynomials on the interval $[-1, 1]$ can be found using the Gram–Schmidt method:

$$P_0 = 1 \tag{4.55}$$

$$P_1 = x - \frac{P_0}{< P_0 P_0 >} \int_{-1}^{1} x P_0(x)dx = x \tag{4.56}$$

$$\begin{aligned} P_2 &= x^2 - \frac{P_1}{< P_1 P_1 >} \int_{-1}^{1} x^2 P_1(x)dx - \frac{P_0}{< P_0 P_0 >} \int_{-1}^{1} x^2 P_0(x)dx \\ &= x^2 - \frac{1}{3} \end{aligned} \tag{4.57}$$

$$P_n = x^n - \frac{P_{n-1}}{< P_{n-1} P_{n-1} >} \int_{-1}^{1} x^n P_{n-1}(x) dx$$
$$- \frac{P_{n-2}}{< P_{n-2} P_{n-2} >} \int_{-1}^{1} x^n P_{n-2}(x) dx - \cdots . \tag{4.58}$$

The P_n are known as Legendre-polynomials. Consider now a polynomial $p(x)$ of order $2N-1$. It can be interpolated at the N quadrature points x_i using the Lagrange method by a polynomial $\widetilde{p}(x)$ of order $N-1$:

$$\widetilde{p}(x) = \sum_{j=1}^{N} L_j(x) p(x_j). \tag{4.59}$$

Then $p(x)$ can be written as

$$p(x) = \widetilde{p}(x) + (x - x_1)(x - x_2) \ldots (x - x_N) q(x). \tag{4.60}$$

Obviously $q(x)$ is a polynomial of order $(2N-1) - N = N - 1$. Now choose the positions x_i as the roots of the Legendre polynomial of order N

$$(x - x_1)(x - x_2) \ldots (x - x_N) = P_N(x). \tag{4.61}$$

Then we have

$$\int_{-1}^{1} (x - x_1)(x - x_2) \ldots (x - x_N) q(x) dx = 0 \tag{4.62}$$

since P_N is orthogonal to the polynomial of lower order. But now

$$\int_{-1}^{1} p(x) dx = \int_{-1}^{1} \widetilde{p}(x) dx = \int_{-1}^{1} \sum_{j=1}^{N} p(x_j) L_j(x) dx = \sum_{j=1}^{N} w_j p(x_j) \tag{4.63}$$

with the weight factors

$$w_j = \int_{-1}^{1} L_j(x) dx. \tag{4.64}$$

Example (Gauss–Legendre integration with two quadrature points) The 2nd order Legendre polynomial

$$P_2(x) = x^2 - \frac{1}{3} \tag{4.65}$$

has two roots

$$x_{1,2} = \pm\sqrt{\frac{1}{3}}. \tag{4.66}$$

The Lagrange polynomials are

$$L_1 = \frac{x - \sqrt{\frac{1}{3}}}{-\sqrt{\frac{1}{3}} - \sqrt{\frac{1}{3}}} \quad L_2 = \frac{x + \sqrt{\frac{1}{3}}}{\sqrt{\frac{1}{3}} + \sqrt{\frac{1}{3}}} \tag{4.67}$$

and the weights

$$w_1 = \int_{-1}^{1} L_1 dx = -\frac{\sqrt{3}}{2}\left(\frac{x^2}{2} - \sqrt{\frac{1}{3}}x\right)_{-1}^{1} = 1 \tag{4.68}$$

$$w_2 = \int_{-1}^{1} L_2 dx = \frac{\sqrt{3}}{2}\left(\frac{x^2}{2} + \sqrt{\frac{1}{3}}x\right)_{-1}^{1} = 1. \tag{4.69}$$

This gives the integral rule

$$\int_{-1}^{1} f(x)dx \approx f\left(-\sqrt{\frac{1}{3}}\right) + f\left(\sqrt{\frac{1}{3}}\right). \tag{4.70}$$

For a general integration interval we substitute

$$x = \frac{a+b}{2} + \frac{b-a}{2}u \tag{4.71}$$

and find the approximation

$$\int_a^b f(x)dx = \int_{-1}^{1} f\left(\frac{a+b}{2} + \frac{b-a}{2}u\right)\frac{b-a}{2}du$$

$$\approx \frac{b-a}{2}\left(f\left(\frac{a+b}{2} - \frac{b-a}{2}\sqrt{\frac{1}{3}}\right) + f\left(\frac{a+b}{2} + \frac{b-a}{2}\sqrt{\frac{1}{3}}\right)\right). \tag{4.72}$$

The next higher order Gaussian rule is given by

$$n = 3 : w_1 = w_3 = 5/9,\ w_2 = 8/9,\ x_3 = -x_1 = 0.77459\ldots,\ x_2 = 0. \tag{4.73}$$

4.2.2.2 Other Types of Gaussian Integration

Further integral rules can be obtained by using other sets of orthogonal polynomials, for instance

Chebyshev Polynomials

$$w(x) = \frac{1}{\sqrt{1-x^2}} \tag{4.74}$$

$$\int_{-1}^{1} f(x)dx = \int_{-1}^{1} f(x)\sqrt{1-x^2}w(x)dx \tag{4.75}$$

$$T_{n+1}(x) = 2x\,T_n(x) - T_{n-1}(x) \tag{4.76}$$

$$x_k = \cos\left(\frac{2k-1}{2N}\pi\right) \quad w_k = \frac{\pi}{N}. \tag{4.77}$$

Hermite Polynomials

$$w(x) = e^{-x^2} \tag{4.78}$$

$$\int_{-\infty}^{\infty} f(x)dx = \int_{-\infty}^{\infty} f(x)e^{x^2}w(x)dx \tag{4.79}$$

$$H_0(x) = 1, \quad H_1(x) = 2x, \quad H_{n+1}(x) = 2x\,H_n(x) - 2n\,H_{n-1}(x).$$

Laguerre Polynomials

$$w(x) = e^{-x} \tag{4.80}$$

$$\int_{0}^{\infty} f(x)dx = \int_{0}^{\infty} f(x)e^{x}w(x)dx \tag{4.81}$$

$$L_0(x) = 1, \quad L_1(x) = 1 - x, \quad L_{n+1}(x) = \frac{1}{n+1}\left((2n+1-x)L_n(x) - nL_{n-1}(x)\right). \tag{4.82}$$

4.2.2.3 Connection with an Eigenvalue Problem

The determination of quadrature points and weights can be formulated as an eigenvalue problem [22, 23]. Any set of orthogonal polynomials $P_n(x)$ with

$$\int_a^b P_m(x)P_n(x)w(x)dx = \delta_{m,n} \tag{4.83}$$

satisfies a three term recurrence relation

$$P_{n+1}(x) = (a_{n+1}x + b_{n+1})\,P_n(x) - c_{n+1}P_{n-1}(x) \tag{4.84}$$

with $a_n > 0, c_n > 0$ which can be written in matrix form [24]

$$x\begin{pmatrix} P_0(x) \\ P_1(x) \\ \\ \\ P_{N-1}(x) \end{pmatrix} = \begin{pmatrix} -\frac{b_1}{a_1} & \frac{1}{a_1} & & & & \\ \frac{c_2}{a_2} & -\frac{b_2}{a_2} & \frac{1}{a_2} & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \frac{c_{N-1}}{a_{N-1}} & -\frac{b_{N-1}}{a_{N-1}} & \frac{1}{a_{N-1}} \\ & & & & \frac{c_N}{a_N} & -\frac{b_N}{a_N} \end{pmatrix} \begin{pmatrix} P_0(x) \\ P_1(x) \\ \\ \\ P_{N-1}(x) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ \frac{1}{a_N}P_N(x) \end{pmatrix} \tag{4.85}$$

or shorter

$$x\mathbf{P}(x) = T\mathbf{P}(x) + \frac{1}{a_N}P_N(x)\mathbf{e}_{N-1} \tag{4.86}$$

with a tridiagonal matrix T. Obviously $P_N(x) = 0$ if and only if

$$x_j\mathbf{P}(x_j) = T\mathbf{P}(x_j), \tag{4.87}$$

hence the roots of $P_N(x)$ are given by the eigenvalues of T. The matrix T is symmetric if the polynomials are orthonormal, otherwise it can be transformed into a symmetric tridiagonal matrix by an orthogonal transformation [24]. Finally the quadrature weight corresponding to the eigenvalue x_j can be calculated from the first component of the corresponding eigenvector $\mathbf{u}_j$ [22] as

$$w_j = u_{j,1}^2 \times \int_a^b w(x)dx. \tag{4.88}$$

Problems

Problem 4.1 Romberg Integration

Use the trapezoidal rule

$$T(h) = h\left(\frac{1}{2}f(a) + f(a+h) + \cdots f(b-h) + \frac{1}{2}f(b)\right) = \int_a^b f(x)dx + \cdots \tag{4.89}$$

with the step sequence

$$h_i = \frac{h_0}{2^i} \tag{4.90}$$

and calculate the elements of the triangular matrix

$$T(i,0) = T(h_i) \tag{4.91}$$

$$T(i,k) = T(i+1,k-1) + \frac{T(i,k-1) - T(i+1,k-1)}{1 - \frac{h_i^2}{h_{i+k}^2}} \tag{4.92}$$

to obtain the approximations

$$T_{01} = P_{01}, T_{02} = P_{012}, T_{03} = P_{0123}, \ldots \tag{4.93}$$

- calculate

$$\int_0^{\pi^2} \sin(x^2)dx = 0.6773089370468890331\ldots \tag{4.94}$$

  and compare the absolute error of the trapezoidal sums $T(h_i) = T_{i,0}$ and the extrapolated values $T_{0,i}$.
- calculate

$$\int_\varepsilon^1 \frac{dx}{\sqrt{x}} \tag{4.95}$$

  for $\varepsilon = 10^{-3}$. Compare with the composite midpoint rule

$$T(h) = h\left(f\left(a + \frac{h}{2}\right) + f\left(a + \frac{3h}{2}\right) + \cdots + f\left(b - \frac{3h}{2}\right) + f\left(b - \frac{h}{2}\right)\right) \tag{4.96}$$

Chapter 5
Systems of Inhomogeneous Linear Equations

Many problems in physics and especially computational physics involve systems of linear equations

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &= b_1 \\ \vdots \quad \vdots \quad \vdots & \\ a_{n1}x_1 + \cdots + a_{nn}x_n &= b_n \end{aligned} \tag{5.1}$$

or shortly in matrix form

$$A\mathbf{x} = \mathbf{b} \tag{5.2}$$

which arise e.g. from linearization of a general nonlinear problem like (Sect. 22.2)

$$0 = \begin{pmatrix} F_1(x_1 \ldots x_n) \\ \vdots \\ F_n(x_1 \ldots x_n) \end{pmatrix} = \begin{pmatrix} F_1(x_1^{(0)} \ldots x_n^{(0)}) \\ \vdots \\ F_n(x_1^{(0)} \ldots x_n^{(0)}) \end{pmatrix} + \begin{pmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial x_1} & \cdots & \frac{\partial F_n}{\partial x_n} \end{pmatrix} \begin{pmatrix} x_1 - x_1^{(0)} \\ \vdots \\ x_n - x_n^{(0)} \end{pmatrix} + \ldots \tag{5.3}$$

or from discretization of differential equations like

$$\begin{aligned} 0 = \frac{\partial f}{\partial x} - g(x) &\to \begin{pmatrix} \vdots \\ \frac{f((j+1)\Delta x) - f(j\Delta x)}{\Delta x} - g(j\Delta x) \\ \vdots \end{pmatrix} \\ &= \begin{pmatrix} \ddots & & & & \\ & -\frac{1}{\Delta x} & \frac{1}{\Delta x} & & \\ & & -\frac{1}{\Delta x} & \frac{1}{\Delta x} & \\ & & & \ddots & \end{pmatrix} \begin{pmatrix} \vdots \\ f_j \\ f_{j+1} \\ \vdots \end{pmatrix} - \begin{pmatrix} \vdots \\ g_j \\ g_{j+1} \\ \vdots \end{pmatrix}. \end{aligned} \tag{5.4}$$

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_5

If the matrix A is non singular and has full rank, (5.2) *can be formally solved by matrix inversion*

$$\mathbf{x} = A^{-1}\mathbf{b}. \tag{5.5}$$

If the matrix is singular or the number of equations smaller than the number of variables, a manifold of solutions exists which can be found efficiently by singular value decomposition (Sect. 11.2*). The general solution is given by a particular solution and the nullspace of A*

$$\mathbf{x} = \mathbf{x}_p + \mathbf{z} \; with \; A\mathbf{x}_p = \mathbf{b} \; and \; A\mathbf{z} = 0. \tag{5.6}$$

If the number of equations is larger than the number of variables there exists no unique solution. The "best possible solution" can be determined by minimizing the residual

$$|A\mathbf{x} - \mathbf{b}| = \min \tag{5.7}$$

which leads to a least squares problem (Sect. 11.1.1*).*

In the following we discuss several methods to solve non singular systems. If the dimension is not too large, direct methods like Gaussian elimination or QR decomposition are sufficient. Systems with a tridiagonal matrix are important for cubic spline interpolation and numerical second derivatives. They can be solved very efficiently with a specialized Gaussian elimination method. Practical applications often involve very large dimensions and require iterative methods. Stationary methods apply a simple iteration scheme repeatedly. The slow convergence of the methods by Jacobi and Gauss-Seidel can be improved with relaxation or overrelaxation. Non-stationary methods construct a sequence of improved approximations within a series of increasing subspaces of $\mathbb{R}^N$*. Modern Krylov-space methods minimize the residual* $\mathbf{r} = A\mathbf{x} - \mathbf{b}$ *within the sequence of Krylov-spaces* $K_n(A, \mathbf{r}^{(0)}) = \text{span}(\mathbf{r}^{(0)}, A\mathbf{r}^{(0)}, \ldots A^{n-1}\mathbf{r}^{(0)})$*. We discuss the conjugate gradients method (CG* [25]*) for symmetric positive definite matrices and the method of general minimal residuals (GMRES* [26]*) for non symmetric matrices. Other popular methods are the methods of bi-conjugate gradients (BiCG* [27] *BiCGSTAB* [28]*), conjugate residuals (CR* [29]*) , minimal residual (MINRES* [30]*), quasi-minimal residual (QMR* [31]*), the symmetric LQ-method (SYMMLQ* [32]*) and Lanzcos type product methods (LTPM* [33–35]*).*

5.1 Gaussian Elimination Method

A series of linear combinations of the equations transforms the matrix A into an upper triangular matrix. Start with subtracting a_{i1}/a_{11} times the first row from rows $2 \cdots n$

$$\begin{pmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_n^T \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{a}_1^T \\ \mathbf{a}^{\mathbf{T}}{}_2 - l_{21}\mathbf{a}_1^T \\ \vdots \\ \mathbf{a}_n^T - l_{n1}\mathbf{a}_1^T \end{pmatrix} \tag{5.8}$$

which can be written as a multiplication

$$A^{(1)} = L_1 A \tag{5.9}$$

with the Frobenius matrix

$$L_1 = \begin{pmatrix} 1 & & & \\ -l_{21} & 1 & & \\ -l_{31} & & 1 & \\ \vdots & & & \ddots \\ -l_{n1} & & & 1 \end{pmatrix} \qquad l_{i1} = \frac{a_{i1}}{a_{11}}. \tag{5.10}$$

The result has the form

$$A^{(1)} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n-1} & a_{1n} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n-1}^{(1)} & a_{2n}^{(1)} \\ 0 & a_{32}^{(1)} & \cdots & \cdots & a_{3n}^{(1)} \\ 0 & \vdots & & & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & \cdots & a_{nn}^{(1)} \end{pmatrix}. \tag{5.11}$$

Now subtract $\frac{a_{i2}}{a_{22}}$ times the second row from rows $3 \cdots n$. This can be formulated as

$$A^{(2)} = L_2 A^{(1)} = L_2 L_1 A \tag{5.12}$$

with

$$L_2 = \begin{pmatrix} 1 & & & & \\ 0 & 1 & & & \\ 0 & -l_{32} & 1 & & \\ \vdots & \vdots & & \ddots & \\ 0 & -l_{n2} & & & 1 \end{pmatrix} \qquad l_{i2} = \frac{a_{i2}^{(1)}}{a_{22}^{(1)}}. \tag{5.13}$$

The result is

$$A^{(2)} = \begin{pmatrix} a_{11}^{(2)} & a_{12}^{(2)} & a_{13}^{(2)} & \cdots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} \end{pmatrix}. \tag{5.14}$$

Continue until an upper triangular matrix results after n−1 steps:

$$A^{(n-1)} = L_{n-1} A^{(n-2)} \tag{5.15}$$

$$L_{n-1} = \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & -l_{n,n-1} & 1 \end{pmatrix} \qquad l_{n,n-1} = \frac{a_{n,n-1}^{(n-2)}}{a_{n-1,n-1}^{(n-2)}} \tag{5.16}$$

$$A^{(n-1)} = L_{n-1} L_{n-2} \cdots L_2 L_1 A = U \tag{5.17}$$

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ & u_{22} & u_{23} & \cdots & u_{2n} \\ & & u_{33} & \cdots & u_{3n} \\ & & & \ddots & \vdots \\ & & & & u_{nn} \end{pmatrix}. \tag{5.18}$$

The transformed system of equations

$$U\mathbf{x} = \mathbf{y} \qquad \mathbf{y} = L_{n-1} L_{n-1} \cdots L_2 L_1 \mathbf{b} \tag{5.19}$$

can be solved easily by backward substitution:

$$x_n = \frac{1}{u_{nn}} y_n \tag{5.20}$$

$$x_{n-1} = \frac{y_{n-1} - x_n u_{n-1,n}}{u_{n-1,n-1}} \tag{5.21}$$

$$\vdots \tag{5.22}$$

Alternatively the matrices L_i can be inverted:

$$L_1^{-1} = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ l_{31} & & 1 & \\ \vdots & & & \ddots \\ l_{n1} & & & 1 \end{pmatrix} \cdots L_{n-1}^{-1} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \\ & & l_{n,n-1} & 1 \end{pmatrix}. \tag{5.23}$$

This gives

$$A = L_1^{-1} L_2^{-1} \cdots L_{n-1}^{-1} U. \tag{5.24}$$

The product of the inverted matrices is a lower triangular matrix:

$$L_1^{-1} L_2^{-1} = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ l_{31} & l_{32} & 1 & \\ \vdots & \vdots & & \ddots \\ l_{n1} & l_{n2} & & 1 \end{pmatrix}$$

$$\vdots$$

$$L = L_1^{-1} L_2^{-1} \cdots L_{n-1}^{-1} = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & \vdots & \ddots & \\ l_{n-1,1} & l_{n-1,2} & \cdots & 1 \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} \; 1 \end{pmatrix}. \tag{5.25}$$

Hence the matrix A becomes decomposed into a product of a lower and an upper triangular matrix

$$A = LU \tag{5.26}$$

which can be used to solve the system of (5.2).

$$A\mathbf{x} = LU\mathbf{x} = \mathbf{b} \tag{5.27}$$

in two steps:

$$L\mathbf{y} = \mathbf{b} \tag{5.28}$$

which can be solved from the top

$$y_1 = b_1 \tag{5.29}$$

$$y_2 = b_2 - l_{21} y_1 \tag{5.30}$$

$$\vdots \tag{5.31}$$

and

$$U\mathbf{x} = \mathbf{y} \tag{5.32}$$

which can be solved from the bottom

$$x_n = \frac{1}{u_{nn}} y_n \tag{5.33}$$

$$x_{n-1} = \frac{y_{n-1} - x_n u_{n-1,n}}{u_{n-1,n-1}}. \tag{5.34}$$

$$\vdots \tag{5.35}$$

5.1.1 Pivoting

To improve numerical stability and to avoid division by zero pivoting is used. Most common is partial pivoting. In every step the order of the equations is changed in order to maximize the pivoting element $a_{k,k}$ in the denominator. This gives LU decomposition of the matrix PA where P is a permutation matrix. P is not needed explicitly. Instead an index vector is used which stores the new order of the equations

$$P \begin{pmatrix} 1 \\ \vdots \\ N \end{pmatrix} = \begin{pmatrix} i_1 \\ \vdots \\ i_N \end{pmatrix}. \tag{5.36}$$

Total pivoting exchanges rows and columns of A. This can be time consuming for larger matrices.

If the elements of the matrix are of different orders of magnitude it can be necessary to balance the matrix, for instance by normalizing all rows of A. This can be also achieved by selecting the maximum of

$$\frac{a_{ik}}{\sum_j |a_{ij}|} \tag{5.37}$$

as the pivoting element.

5.1.2 Direct LU Decomposition

LU decomposition can be also performed in a different order [36]. For symmetric positive definite matrices there exists the simpler and more efficient Cholesky method decomposes the matrix into the product LL^T of a lower triangular matrix and its transpose [37].

5.2 QR Decomposition

The Gaussian elimination method can become numerically unstable [38]. An alternative method to solve a system of linear equations uses the decomposition [39]

$$A = QR \tag{5.38}$$

with a unitary matrix $Q^\dagger Q = 1$ (an orthogonal matrix $Q^T Q = 1$ if A is real) and an upper right triangular matrix R. The system of linear equations (5.2) is simplified by multiplication with $Q^\dagger = Q^{-1}$

$$QR\mathbf{x} = A\mathbf{x} = \mathbf{b} \tag{5.39}$$

$$R\mathbf{x} = Q^\dagger \mathbf{b}. \tag{5.40}$$

Such a system with upper triangular matrix is easily solved (see 5.32).

5.2.1 QR Decomposition by Orthogonalization

Gram-Schmidt orthogonalization [2, 39] provides a simple way to perform a QR decomposition. It is used for symbolic calculations and also for least square fitting (11.1.2) but can become numerically unstable.

From the decomposition $A = QR$ we have

$$a_{ik} = \sum_{j=1}^{k} q_{ij} r_{jk} \tag{5.41}$$

$$\mathbf{a}_k = \sum_{j=1}^{k} r_{jk} \mathbf{q}_j \tag{5.42}$$

which gives the k-th column vector $\mathbf{a}_k$ of A as a linear combination of the orthonormal vectors $\mathbf{q}_1 \cdots \mathbf{q}_k$. Similarly $\mathbf{q}_k$ is a linear combination of the first k columns of A. With the help of the Gram-Schmidt method r_{jk} and $\mathbf{q}_j$ are calculated as follows:

$$r_{11} := |a_1| \tag{5.43}$$

$$\mathbf{q}_1 := \frac{\mathbf{a}_1}{r_{11}} \tag{5.44}$$

For $k = 2, \cdots n$:

$$r_{ik} := \mathbf{q}_i \mathbf{a}_k \quad i = 1 \cdots k-1 \tag{5.45}$$

$$\mathbf{b}_k := \mathbf{a}_k - r_{1k}\mathbf{q}_1 - \cdots r_{k-1,k}\mathbf{q}_{k-1} \tag{5.46}$$

$$r_{kk} := |\mathbf{b}_k| \tag{5.47}$$

$$\mathbf{q}_k := \frac{\mathbf{b}_k}{r_{kk}}. \tag{5.48}$$

Obviously now

$$\mathbf{a}_k = r_{kk}\mathbf{q}_k + r_{k-1,k}\mathbf{q}_{k-1} + \cdots r_{1k}\mathbf{q}_1 \tag{5.49}$$

since per definition

$$\mathbf{q}_i \mathbf{a}_k = r_{ik} \quad i = 1 \cdots k \tag{5.50}$$

and

$$r_{kk}^2 = |\mathbf{b}_k|^2 = |\mathbf{a}_k|^2 + r_{1k}^2 + \cdots r_{k-1,k}^2 - 2r_{1k}^2 - \cdots - 2r_{k-1,k}^2. \tag{5.51}$$

Hence

$$\mathbf{q}_k \mathbf{a}_k = \frac{1}{r_{kk}}(\mathbf{a}_k - r_{1k}\mathbf{q}_1 \cdots r_{k-1,k}\mathbf{q}_{k-1})\mathbf{a}_k = \frac{1}{r_{kk}}(|a_k|^2 - r_{1k}^2 \cdots - r_{k-1,k}^2) = r_{kk}. \tag{5.52}$$

Orthogonality gives

$$\mathbf{q}_i \mathbf{a}_k = 0 \quad i = k+1 \cdots n. \tag{5.53}$$

In matrix notation we have finally

$$A = (\mathbf{a}_1 \cdots \mathbf{a}_n) = (\mathbf{q}_1 \cdots \mathbf{q}_n) \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{pmatrix}. \tag{5.54}$$

If the columns of A are almost linearly dependent, numerical stability can be improved by an additional orthogonalization step

$$\mathbf{b}_k \to \mathbf{b}_k - (\mathbf{q}_1 \mathbf{b}_k)\mathbf{q}_1 - \cdots (\mathbf{q}_{k-1}\mathbf{b}_k)\mathbf{q}_{k-1} \tag{5.55}$$

after (5.46) which can be iterated several times to improve the results [2, 40].

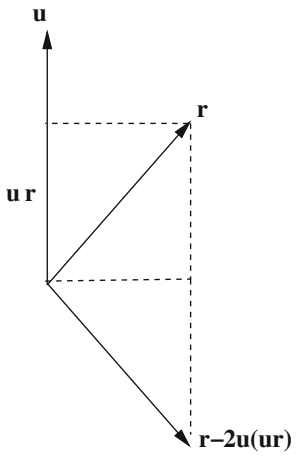


Fig. 5.1 (Householder transformation) Geometrically the Householder transformation (5.56) is a mirror operation with respect to a plane with normal vector **u**

5.2.2 *QR Decomposition by Householder Reflections*

Numerically stable algorithms use a series of transformations with unitary matrices, mostly Householder reflections (Fig. 5.1) [2][1] which have the form

$$P = P^T = 1 - 2\mathbf{u}\mathbf{u}^T \tag{5.56}$$

with a unit vector

$$|\mathbf{u}| = 1. \tag{5.57}$$

Obviously P is an orthogonal matrix since

$$P^T P = (1 - 2\mathbf{u}\mathbf{u}^T)(1 - 2\mathbf{u}\mathbf{u}^T) = 1 - 4\mathbf{u}\mathbf{u}^T + 4\mathbf{u}\mathbf{u}^T\mathbf{u}\mathbf{u}^T = 1. \tag{5.58}$$

In the first step we try to find a vector **u** such that the first column vector of A

$$\mathbf{a}_1 = \begin{pmatrix} a_{11} \\ \vdots \\ a_{n1} \end{pmatrix} \tag{5.59}$$

[1] Alternatively Givens rotations [39] can be employed which need slightly more floating point operations.

is transformed into a vector along the 1-axis

$$P\mathbf{a}_1 = \left(1 - 2\mathbf{u}\mathbf{u}^T\right)\mathbf{a}_1 = k\mathbf{e}_1 = \begin{pmatrix} k \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{5.60}$$

Multiplication with the transpose vector gives

$$k^2 = (P\mathbf{a}_1)^T\, P\mathbf{a}_1 = \mathbf{a}_1^T P^T P\mathbf{a}_1 = |\mathbf{a}_1|^2 \tag{5.61}$$

and

$$k = \pm|\mathbf{a}_1| \tag{5.62}$$

can have both signs. From (5.60) we have

$$\mathbf{a}_1 - 2\mathbf{u}(\mathbf{u}\mathbf{a}_1) = k\mathbf{e}_1. \tag{5.63}$$

Multiplication with $\mathbf{a}_1^T$ gives

$$2(\mathbf{u}\mathbf{a}_1)^2 = |\mathbf{a}_1|^2 - k(\mathbf{a}_1\mathbf{e}_1) \tag{5.64}$$

and since

$$|\mathbf{a}_1 - k\mathbf{e}_1|^2 = |\mathbf{a}_1|^2 + k^2 - 2k(\mathbf{a}_1\mathbf{e}_1) = 2|\mathbf{a}_1|^2 - 2k(\mathbf{a}_1\mathbf{e}_1) \tag{5.65}$$

we have

$$2(\mathbf{u}\mathbf{a}_1)^2 = \frac{1}{2}|\mathbf{a}_1 - k\mathbf{e}_1|^2 \tag{5.66}$$

and from (5.63) we find

$$\mathbf{u} = \frac{\mathbf{a}_1 - k\mathbf{e}_1}{2\mathbf{u}\mathbf{a}_1} = \frac{\mathbf{a}_1 - k\mathbf{e}_1}{|\mathbf{a}_1 - k\mathbf{e}_1|}. \tag{5.67}$$

To avoid numerical extinction the sign of k is chosen such that

$$\sigma = \text{sign}(k) = -\text{sign}(a_{11}). \tag{5.68}$$

Then,

$$\mathbf{u} = \frac{1}{\sqrt{2(a_{11}^2 + \cdots + a_{n1}^2) + 2|a_{11}|\sqrt{a_{11}^2 + \cdots + a_{n1}^2}}} \begin{pmatrix} \text{sign}(a_{11})\left(|a_{11}| + \sqrt{a_{11}^2 + a_{21}^2 + \ldots a_{n1}^2}\right) \\ a_{21} \\ \vdots \\ a_{n1} \end{pmatrix} \tag{5.69}$$

$$2\mathbf{u}\mathbf{u}^T\mathbf{a}_1 = \begin{pmatrix} \text{sign}(a_{11})\left(|a_{11}| + \sqrt{a_{11}^2 + a_{21}^2 + \ldots a_{n1}^2}\right) \\ a_{21} \\ \vdots \\ a_{n1} \end{pmatrix} \times \frac{1}{(a_{11}^2 + \cdots + a_{n1}^2) + |a_{11}|\sqrt{a_{11}^2 + \cdots + a_{n1}^2}} \left(a_{11}^2 + |a_{11}|\sqrt{a_{11}^2 + \ldots a_{n1}^2} + a_{21}^2 + \cdots + a_{n1}^2\right) \tag{5.70}$$

and the Householder transformation of the first column vector of A gives

$$(1 - 2\mathbf{u}\mathbf{u}^T)\mathbf{a}_1 = \begin{pmatrix} -\text{sign}(a_{11})\sqrt{a_{11}^2 \ldots a_{n1}^2} \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{5.71}$$

Thus after the first step a matrix results of the form

$$A^{(1)} = P_1 A = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \ldots & a_{1n}^{(1)} \\ 0 & a_{22}^{(1)} & & a_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n2}^{(1)} & \ldots & a_{nn}^{(1)} \end{pmatrix}.$$

In the following (n-2) steps further Householder reflections are applied in the subspace $k \leq i, j \leq n$ to eliminate the elements

$$a_{k+1,k} \ldots a_{n,k}$$

of the $k - th$ row vector below the diagonal of the matrix:

$$A^{(k-1)} = P_{k-1} \dots P_1 A = \begin{pmatrix} a_{11}^{(1)} & \dots & a_{1,k-1}^{(1)} & a_{1,k}^{(1)} & \dots & a_{1,n}^{(1)} \\ 0 & \ddots & \vdots & \vdots & & \vdots \\ \vdots & \ddots & a_{k-1,k-1}^{(k-1)} & a_{k-1,k}^{(k-1)} & & a_{k-1,n}^{(k-1)} \\ \vdots & \vdots & 0 & a_{k,k}^{(k-1)} & & a_{k,n}^{(k-1)} \\ \vdots & \vdots & \vdots & a_{k+1,k}^{(k-1)} & & a_{k+1,n}^{(k-1)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & a_{n,k}^{(k-1)} & \dots & a_{n,n}^{(k-1)} \end{pmatrix} \tag{5.72}$$

$$P_k = \begin{pmatrix} 1_1 & & & \\ & \ddots & & \\ & & 1_{k-1} & \\ & & & 1 - 2\mathbf{u}\mathbf{u}^T \end{pmatrix}.$$

Finally an upper triangular matrix results

$$A^{(n-1)} = (P_{n-1} \dots P_1)A = R = \begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \dots & a_{1,n-1}^{(1)} & a_{1,n}^{(1)} \\ 0 & a_{22}^{(2)} & \dots & a_{2,n-1}^{(2)} & a_{2,n}^{(2)} \\ \vdots & 0 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & a_{n-1,n-1}^{(n-1)} & a_{n-1,n}^{(n-1)} \\ 0 & 0 & \dots & 0 & a_{n,n}^{(n-1)} \end{pmatrix}. \tag{5.73}$$

If the orthogonal matrix Q is needed explicitly additional numerical operations are necessary to form the product

$$Q = (P_{n-1} \dots P_1)^T. \tag{5.74}$$

5.3 Linear Equations with Tridiagonal Matrix

Linear equations with the form

$$b_1 x_1 + c_1 x_2 = r_1 \tag{5.75}$$

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = r_i \quad i = 2 \cdots (n-1) \tag{5.76}$$

$$a_n x_{n-1} + b_n x_n = r_n \tag{5.77}$$

can be solved very efficiently with a specialized Gaussian elimination method.[2] They are important for cubic spline interpolation or second derivatives. We begin by eliminating a_2. To that end we multiply the first line with a_2/b_1 and subtract it from the first line. The result is the equation

$$\beta_2 x_2 + c_2 x_3 = \rho_2 \tag{5.78}$$

with the abbreviations

$$\beta_2 = b_2 - \frac{c_1 a_2}{b_1} \quad \rho_2 = r_2 - \frac{r_1 a_2}{b_1}. \tag{5.79}$$

We iterate this procedure

$$\beta_i x_i + c_i x_{i+1} = \rho_i \tag{5.80}$$

$$\beta_i = b_i - \frac{c_{i-1} a_i}{\beta_{i-1}} \quad \rho_i = r_i - \frac{\rho_{i-1} a_i}{\beta_{i-1}} \tag{5.81}$$

until we reach the n-th equation, which becomes simply

$$\beta_n x_n = \rho_n \tag{5.82}$$

$$\beta_n = b_n - \frac{c_{n-1} a_n}{\beta_{n-1}} \quad \rho_n = r_n - \frac{\rho_{n-1} a_n}{\beta_{n-1}}. \tag{5.83}$$

Now we immediately have

$$x_n = \frac{\rho_n}{\beta_n} \tag{5.84}$$

and backward substitution gives

$$x_{i-1} = \frac{\rho_{i-1} - c_{i-1} x_i}{\beta_{i-1}} \tag{5.85}$$

and finally

$$x_1 = \frac{r_1 - c_1 x_2}{\beta_2}. \tag{5.86}$$

This algorithm can be formulated as LU decomposition: Multiplication of the matrices

[2] This algorithm is only well behaved if the matrix is diagonal dominant $|b_i| > |a_i| + |c_i|$.

$$L = \begin{pmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & l_3 & 1 & & \\ & & \ddots & \ddots & \\ & & & l_n & 1 \end{pmatrix} \qquad U = \begin{pmatrix} \beta_1 & c_1 & & & \\ & \beta_2 & c_2 & & \\ & & \beta_3 & c_3 & \\ & & & \ddots & \\ & & & & \beta_n \end{pmatrix} \tag{5.87}$$

gives

$$LU = \begin{pmatrix} \beta_1 & c_1 & & & & & \\ & \ddots & \ddots & & & & \\ & & \ddots & \ddots & & & \\ & & l_i\beta_{i-1} & (l_i c_{i-1} + \beta_i) & c_i & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & & l_n\beta_{n-1} & (l_n c_{n-1} + \beta_n) \end{pmatrix} \tag{5.88}$$

which coincides with the matrix

$$A = \begin{pmatrix} b_1 & c_1 & & & & & \\ a_2 & \ddots & \ddots & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & a_i & b_i & c_i & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & & a_n & b_n \end{pmatrix} \tag{5.89}$$

if we choose

$$l_i = \frac{a_i}{\beta_{i-1}} \tag{5.90}$$

since then from (5.81)

$$b_i = \beta_i + l_i c_{i-1} \tag{5.91}$$

and

$$l_i \beta_{i-1} = a_i. \tag{5.92}$$

5.4 Cyclic Tridiagonal Systems

Periodic boundary conditions lead to a small perturbation of the tridiagonal matrix

$$A = \begin{pmatrix} b_1 & c_1 & & & & a_1 \\ a_2 & \ddots & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & a_i & b_i & c_i & \\ & & & \ddots & \ddots & \ddots \\ & & & & a_{n-1} & b_{n-1} & c_{n-1} \\ c_n & & & & & a_n & b_n \end{pmatrix}. \tag{5.93}$$

The system of equations

$$A\mathbf{x} = \mathbf{r} \tag{5.94}$$

can be reduced to a tridiagonal system [41] with the help of the Sherman–Morrison formula [42], which states that if A_0 is an invertible matrix and $\mathbf{u}$, $\mathbf{v}$ are vectors and

$$1 + \mathbf{v}^T A_0^{-1}\mathbf{u} \neq 0 \tag{5.95}$$

then the inverse of the matrix[3]

$$A = A_0 + \mathbf{u}\mathbf{v}^T \tag{5.96}$$

is given by

$$A^{-1} = A_0^{-1} - \frac{A_0^{-1}\mathbf{u}\mathbf{v}^T A_0^{-1}}{1 + \mathbf{v}^T A_0^{-1}\mathbf{u}}. \tag{5.97}$$

We choose

$$\mathbf{u}\mathbf{v}^T = \begin{pmatrix} \alpha \\ 0 \\ \vdots \\ 0 \\ c_n \end{pmatrix} \begin{pmatrix} 1\ 0 \cdots 0\ \frac{a_1}{\alpha} \end{pmatrix} = \begin{pmatrix} \alpha & a_1 \\ & \\ & \\ c_n & \frac{a_1 c_n}{\alpha} \end{pmatrix}. \tag{5.98}$$

[3] Here $\mathbf{u}\mathbf{v}^T$ is the outer or matrix product of the two vectors.

Then

$$A_0 = A - \mathbf{u}\mathbf{v}^T = \begin{pmatrix} (b_1-\alpha) & c_1 & & & & 0 \\ a_2 & \ddots & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & a_i & b_i & c_i & \\ & & & \ddots & \ddots & \ddots \\ & & & & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & & & & a_n & (b_n - \frac{a_1 c_n}{\alpha}) \end{pmatrix} \tag{5.99}$$

is tridiagonal. The free parameter α has to be chosen such that the diagonal elements do not become too small. We solve the system (5.94) by solving the two tridiagonal systems

$$\begin{aligned} A_0\mathbf{x}_0 &= \mathbf{r} \\ A_0\mathbf{q} &= \mathbf{u} \end{aligned} \tag{5.100}$$

and compute $\mathbf{x}$ from

$$\mathbf{x} = A^{-1}\mathbf{r} = A_0^{-1}\mathbf{r} - \frac{(A_0^{-1}\mathbf{u})\mathbf{v}^T(A_0^{-1}\mathbf{r})}{1+\mathbf{v}^T(A_0^{-1}\mathbf{u})} = \mathbf{x}_0 - \mathbf{q}\frac{\mathbf{v}^T\mathbf{x}_0}{1+\mathbf{v}^T\mathbf{q}}. \tag{5.101}$$

5.5 Linear Stationary Iteration

Discretized differential equations often lead to systems of equations with large sparse matrices, which have to be solved by iterative methods which, starting from an initial guess $\mathbf{x}_0$ (often simply $\mathbf{x}_0 = 0$ or $\mathbf{x}_0 = \mathbf{b}$) construct a sequence of improved vectors by the iteration

$$\mathbf{x}^{(n+1)} = \Phi(\mathbf{x}^{(n)}, \mathbf{b}) \tag{5.102}$$

which under certain conditions converges to the fixed point

$$\mathbf{x}_{FP} = \Phi(\mathbf{x}_{FP}, \mathbf{b}) \tag{5.103}$$

which solves the system of equations

$$A\mathbf{x} = \mathbf{b}. \tag{5.104}$$

A linear iterative method is called stationary, if it has the form

$$\mathbf{x}^{(n+1)} = B\mathbf{x}^{(n)} + C\mathbf{b} \tag{5.105}$$

where the matrices B (the so called iteration matrix) and C are constant and do not depend on the iteration count n. A fixed point of (5.105) solves (5.104) and hence the method is consistent, if

$$\mathbf{x} = B\mathbf{x} + C\mathbf{b} = B\mathbf{x} + CA\mathbf{x} \tag{5.106}$$

and hence

$$B = I - CA \tag{5.107}$$

which brings the iteration to the general form

$$\mathbf{x}^{(n+1)} = (I - CA)\mathbf{x}^{(n)} + C\mathbf{b} = \mathbf{x}^{(n)} - C(A\mathbf{x}^{(n)} - \mathbf{b}) \tag{5.108}$$

$$\mathbf{r}^{(n+1)} = (1 - AC)\mathbf{r}^{(n)}. \tag{5.109}$$

5.5.1 Richardson-Iteration

The simplest of these methods uses $C = \omega I$ with a damping parameter ω. It iterates according to

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \omega(A\mathbf{x}^{(n)} - \mathbf{b}) \tag{5.110}$$

$$\mathbf{r}^{(n+1)} = (1 - \omega A)\mathbf{r}^{(n)}. \tag{5.111}$$

The Richardson iteration is not of much practical use. It serves as the prototype of a linear stationary method. To improve convergence (5.104) usually has to be preconditioned by multiplication with a suitable matrix P

$$PA\mathbf{x} = P\mathbf{b} \tag{5.112}$$

for which the Richardson iteration

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \omega P(A\mathbf{x}^{(n)} - \mathbf{b}) \tag{5.113}$$

is of the general form (5.108).

5.5.2 Matrix Splitting Methods

Let us divide the matrix A into two (non singular) parts [2]

$$A = A_1 + A_2 \tag{5.114}$$

where A_1 can be easily inverted and rewrite (5.104) as

$$A_1\mathbf{x} = \mathbf{b} - A_2\mathbf{x} \tag{5.115}$$

which defines the iteration

$$\Phi(\mathbf{x}) = -A_1^{-1}A_2\mathbf{x} + A_1^{-1}\mathbf{b} \tag{5.116}$$

$$= -A_1^{-1}(A - A_1)\mathbf{x} + A_1^{-1}\mathbf{b} = \mathbf{x} - A_1^{-1}(A\mathbf{x} - \mathbf{b}). \tag{5.117}$$

5.5.3 Jacobi Method

Jacobi divides the matrix A into its diagonal and two triangular matrices [43]:

$$A = L + U + D. \tag{5.118}$$

For A_1 the diagonal part is chosen

$$A_1 = D \tag{5.119}$$

giving

$$\mathbf{x}^{(n+1)} = -D^{-1}(A - D)\mathbf{x}^{(n)} + D^{-1}\mathbf{b} \tag{5.120}$$

which reads explicitly

$$x_i^{(n+1)} = -\frac{1}{a_{ii}}\sum_{j\neq i} a_{ij}x_j^{(n)} + \frac{1}{a_{ii}}b_i. \tag{5.121}$$

This method is stable but converges rather slowly. Reduction of the error by a factor of 10^{-p} needs about $\frac{pN}{2}$ iterations. N grid points have to be evaluated in each iteration and the method scales with $O(N^2)$ [44].

5.5.4 Gauss-Seidel Method

With

$$A_1 = D + L \tag{5.122}$$

the iteration becomes

$$(D + L)\mathbf{x}^{(n+1)} = -U\mathbf{x}^{(n)} + \mathbf{b} \tag{5.123}$$

which has the form of a system of equations with triangular matrix [45]. It reads explicitly

$$\sum_{j \le i} a_{ij} x_j^{(n+1)} = -\sum_{j > i} a_{ij} x_j^{(n)} + b_i. \tag{5.124}$$

Forward substitution starting from x_1 gives

$$\begin{aligned}
i = 1: \quad & x_1^{(n+1)} = \frac{1}{a_{11}} \left(-\sum_{j \ge 2} a_{ij} x_j^{(n)} + b_1 \right) \\
i = 2: \quad & x_2^{(n+1)} = \frac{1}{a_{22}} \left(-a_{21} x_1^{(n+1)} - \sum_{j \ge 3} a_{ij} x_j^{(n)} + b_2 \right) \\
i = 3: \quad & x_3^{(n+1)} = \frac{1}{a_{33}} \left(-a_{31} x_1^{(n+1)} - a_{32} x_2^{(n+1)} - \sum_{j \ge 4} a_{ij} x_j^{(n)} + b_3 \right) \\
& \vdots \\
& x_i^{(n+1)} = \frac{1}{a_{ii}} \left(-\sum_{j < i} a_{ij} x_j^{(n+1)} - \sum_{j > i} a_{ij} x_j^{(n)} + b_i \right).
\end{aligned} \tag{5.125}$$

This looks very similar to the Jacobi method. But here all changes are made immediately. Convergence is slightly better (roughly a factor of 2) and the numerical effort is reduced [44].

5.5.5 Damping and Successive Over-relaxation

Convergence can be improved [44] by combining old and new values. Starting from the iteration

$$A_1\mathbf{x}^{(n+1)} = (A_1 - A)\mathbf{x}^{(n)} + \mathbf{b} \tag{5.126}$$

we form a linear combination with

$$D\mathbf{x}^{(n+1)} = D\mathbf{x}^{(n)} \tag{5.127}$$

giving the new iteration equation

$$((1-\omega)D + \omega A_1)\mathbf{x}^{(n+1)} = ((1-\omega)D + \omega A_1 - \omega A)\mathbf{x}^{(n)} + \omega\mathbf{b}. \tag{5.128}$$

In case of the Jacobi method with $D = A_1$ we have

$$D\mathbf{x}^{(n+1)} = (D - \omega A)\mathbf{x}^{(n)} + \omega\mathbf{b} \tag{5.129}$$

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \omega D^{-1}(A\mathbf{x} - \mathbf{b}) \tag{5.130}$$

which can be also obtained directly from (5.113).

Explicitly,

$$x_i^{(n+1)} = (1-\omega)x_i^{(n)} + \frac{\omega}{a_{ii}}\left(-\sum_{j\neq i} a_{ij}x_j^{(n)} + b_i\right). \tag{5.131}$$

The changes are damped ($0 < \omega < 1$) or exaggerated ($1 < \omega < 2$).

In case of the Gauss-Seidel method with $A_1 = D + L$ the new iteration[4] (5.128) is

$$(D + \omega L)\mathbf{x}^{(n+1)} = (D + \omega L - \omega A)\mathbf{x}^{(n)} + \omega\mathbf{b} = (1-\omega)D\mathbf{x}^{(n)} - \omega U\mathbf{x}^{(n)} + \omega\mathbf{b} \tag{5.132}$$

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \left(\frac{1}{\omega}D + L\right)^{-1}(A\mathbf{x}^{(n)} - \mathbf{b}) \tag{5.133}$$

or explicitly

$$x_i^{(n+1)} = (1-\omega)x_i^{(n)} + \frac{\omega}{a_{ii}}\left(-\sum_{j<i} a_{ij}x_j^{(n+1)} - \sum_{j>i} a_{ij}x_j^{(n)} + \mathbf{b}\right). \tag{5.134}$$

[4] This is also known as the method of successive over-relaxation (SOR) and differs from the damped Gauss-Seidel method which follows from (5.113).

It can be shown that the successive over-relaxation method converges only for $0 < \omega < 2$. For optimal choice of ω about $\frac{1}{3}p\sqrt{N}$ iterations are needed to reduce the error by a factor of 10^{-p}. The order of the method is $O(N^{\frac{3}{2}})$ which is comparable to the most efficient matrix inversion methods [44].

5.6 Non Stationary Iterative Methods

Non stationary methods use a more general iteration

$$\mathbf{x}^{(n+1)} = \Phi_n(\mathbf{x}_n) \tag{5.135}$$

where the iteration function differs from step to step. The method can be formulated as a direction search

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \lambda_n \mathbf{s}_n \tag{5.136}$$

with direction vectors $\mathbf{s}_n$ and step lengths λ_n. The residual

$$\mathbf{r}^{(n+1)} = A\mathbf{x}^{(n+1)} - \mathbf{b} = \mathbf{r}^{(n)} + \lambda_n A\mathbf{s}_n \tag{5.137}$$

is used as a measure of the remaining error since the exact solution $\mathbf{x}_{FP}$ together with the error vector

$$\mathbf{d}^{(n)} = \mathbf{x}^{(n)} - \mathbf{x}_{FP} \tag{5.138}$$

are unknown. Both are, however, related by

$$A\mathbf{d}^{(n)} = A\mathbf{x}^{(n)} - A\mathbf{x}_{FP} = A\mathbf{x}^{(n)} - \mathbf{b} = \mathbf{r}^{(n)}. \tag{5.139}$$

5.6.1 Krylov Space Methods

Solution of the linear system

$$A\mathbf{x} = \mathbf{b} \tag{5.140}$$

can be formulated as a search for the minimum

$$\min_{\mathbf{x}\in\mathbb{R}^{\mathbb{N}}} ||A\mathbf{x} - \mathbf{b}||. \tag{5.141}$$

General iterative methods look for the minimum residual in a subspace of $\mathbb{R}^{\mathbb{N}}$ which is increased at each step. The Richardson iteration, e.g. iterates

$$\begin{aligned}
&\mathbf{x}^{(n+1)} = (1-\omega A)\mathbf{x}^{(n)} + \omega\mathbf{b} = \mathbf{x}^{(n)} - \omega\mathbf{r}^{(n)} \\
&\mathbf{r}^{(n+1)} = A(\mathbf{x}^{(n)} - \omega\mathbf{r}^{(n)}) - \mathbf{b} = (1-\omega A)\mathbf{r}^{(n)} \\
&\mathbf{r}_0 = A\mathbf{x}_0 - \mathbf{b} \\
&\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \omega\mathbf{r}^{(0)} \\
&\mathbf{r}^{(1)} = \mathbf{r}^{(0)} - \omega A\mathbf{r}^{(0)} \\
&\mathbf{x}^{(2)} = \mathbf{x}^{(1)} - \omega\mathbf{r}^{(1)} = \mathbf{x}^{(0)} - 2\omega\mathbf{r}^{(0)} + \omega^2 A\mathbf{r}^{(0)} \\
&\mathbf{r}^{(2)} = (1-\omega A)\mathbf{r}^{(1)} = \mathbf{r}^{(0)} - 2\omega A\mathbf{r}^{(0)} + \omega^2 A^2\mathbf{r}^{(0)} \\
&\vdots
\end{aligned} \tag{5.142}$$

Obviously,

$$\mathbf{x}^{(n)} - \mathbf{x}_0 \in K_n(A, \mathbf{r}^{(0)}) \quad \mathbf{r}^{(n)} \in K_{n+1}(A, \mathbf{r}^{(0)})$$

with the definition of the n-th Krylov subspace[5]

$$K_n(A, \mathbf{r}^{(0)}) = \text{span}\{\mathbf{r}^{(0)}, A\mathbf{r}^{(0)}, A^2\mathbf{r}^{(0)}, \ldots A^{n-1}\mathbf{r}^{(0)}\}. \tag{5.143}$$

5.6.2 *Minimization Principle for Symmetric Positive Definite Systems*

If the matrix A is symmetric and positive definite, we consider the quadratic form defined by

$$h(\mathbf{x}) = h_0 - \mathbf{x}^T\mathbf{b} + \frac{1}{2}\mathbf{x}^T A\mathbf{x}. \tag{5.144}$$

At a local minimum the gradient

$$\nabla h(\mathbf{x}) = A\mathbf{x} - \mathbf{b} = \mathbf{r} \tag{5.145}$$

is zero and therefore the minimum of h is also a solution of the linear system of equations

$$A\mathbf{x} = \mathbf{b}. \tag{5.146}$$

[5]For the most common choice $\mathbf{x}_0 = 0$ we have $\mathbf{r}^{(0)} = -\mathbf{b}$ and $\mathbf{x}_0 + K_n(A, \mathbf{r}^{(0)}) = K_n(A, \mathbf{r}^{(0)}) = K_n(A, \mathbf{b})$.

5.6.3 Gradient Method

The simple Richardson iteration (Sect. 5.5.1) uses

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - \omega \mathbf{r}^{(n)} \tag{5.147}$$

with a constant value of ω.

$$\mathbf{r}^{(n+1)} = \mathbf{r}^{(n)} - \omega A \mathbf{r}^{(n)}. \tag{5.148}$$

Let us now optimize the step width along the direction of the gradient vector. From

$$\begin{aligned} 0 &= \frac{\mathrm{d}}{\mathrm{d}\omega} |\mathbf{r}^{(n+1)}|^2 = \mathbf{r}^{(n)T}(1-\omega A)(1-\omega A)\mathbf{r}^{(n)} \\ &= \mathbf{r}^{(n)T}(-2A + 2\omega A^2)\mathbf{r}^{(n)} = -2\mathbf{r}^{(n)T} A \mathbf{r}^{(n)} + 2\omega |A\mathbf{r}^{(n)}|^2 \\ \mathbf{r} &= 2\mathbf{r}^{(n)T}(-1+\omega A) A \mathbf{r}^{(n)} \end{aligned} \tag{5.149}$$

we find the optimum value

$$\omega^{(n)} = \frac{\mathbf{r}^{(n)T} A \mathbf{r}^{(n)}}{|A\mathbf{r}^{(n)}|^2}. \tag{5.150}$$

The residuals[6]

$$\mathbf{r}_0 = -\mathbf{b} \tag{5.151}$$

$$\mathbf{r}^{(1)} = -\mathbf{b} + \omega^{(1)} A \mathbf{b} \tag{5.152}$$

$$\mathbf{r}^{(2)} = -\mathbf{b} + (\omega^{(1)} + \omega^{(2)}) A \mathbf{b} - \omega^{(2)} \omega^{(1)} A^2 \mathbf{b} \tag{5.153}$$

$$\vdots$$

etc. obviously are in the Krylov subspace

$$\mathbf{r}^{(n)} \in K_{n+1}(A, \mathbf{b}) \tag{5.154}$$

and so are the approximate solutions

$$\mathbf{x}^{(1)} = \omega^{(1)} \mathbf{b} \tag{5.155}$$

$$\mathbf{x}^{(2)} = (\omega^{(1)} + \omega^{(2)}) \mathbf{b} - \omega^{(2)} \omega^{(1)} A \mathbf{b} \tag{5.156}$$

$$\vdots$$

$$\mathbf{x}^{(n)} \in K_n(A, \mathbf{b}). \tag{5.157}$$

[6] We assume $\mathbf{x}_0 = 0$.

5.6.4 Conjugate Gradients Method

The gradient method gives an approximate solution

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(0)} - \omega^{(0)}\mathbf{r}^{(0)} - \omega^{(1)}\mathbf{r}^{(1)} \cdots - \omega^{(n)}\mathbf{r}^{(n)} \tag{5.158}$$

but the previously chosen $\omega^{(0)} \ldots \omega^{(n-1)}$are not optimal since the gradient vectors $\mathbf{r}^{(0)} \ldots \mathbf{r}^{(n)}$are not orthogonal. We want to optimize the solution within the space spanned by the gradients for which a new basis $\mathbf{s}^{(0)} \ldots \mathbf{s}^{(n)}$is introduced which will be determined later

$$K_{n+1} = \text{span}(\mathbf{r}^{(0)} \ldots \mathbf{r}^{(n)}) = \text{span}(\mathbf{s}^{(0)} \ldots \mathbf{s}^{(n)}). \tag{5.159}$$

Using $\mathbf{s}^{(n)}$as search direction the iteration becomes

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \lambda^{(n)}\mathbf{s}^{(n)} \tag{5.160}$$

$$\mathbf{r}^{(n+1)} = \mathbf{r}^{(n)} + \lambda^{(n)}A\mathbf{s}^{(n)}. \tag{5.161}$$

After n+1 steps

$$\mathbf{r}^{(n+1)} = \mathbf{r}^{(0)} + \sum \lambda^{(j)}A\mathbf{s}^{(\mathbf{j})} = A(\mathbf{d}^{(0)} + \sum \lambda^{(j)}\mathbf{s}^{(j)}). \tag{5.162}$$

Multiplication with $\mathbf{s}^{(m)}$ gives

$$\mathbf{s}^{(m)T}\mathbf{r}^{(n+1)} = \mathbf{s}^{(m)T}A\mathbf{d}^{(0)} + \sum \lambda^{(j)}\mathbf{s}^{(m)T}A\mathbf{s}^{(j)} \tag{5.163}$$

which, after introduction of the $A-$scalar product which is defined for a symmetric and positive definite matrix A as

$$(\mathbf{x}, \mathbf{y})_A = \mathbf{x}^T A\mathbf{y} \tag{5.164}$$

becomes

$$\mathbf{s}^{(m)T}\mathbf{r}^{(n+1)} = (\mathbf{s}^{(m)}, \mathbf{d}^{(0)})_A + \sum_{j=0}^{n} \lambda^{(j)}(\mathbf{s}^{(m)}, \mathbf{s}^{(j)})_A$$

which simplifies considerably if we assume A-orthogonality of the search directions

$$(\mathbf{s}^{(m)}, \mathbf{s}^{(j)}) = 0 \quad \text{for } m \neq j \tag{5.165}$$

because then

$$\mathbf{s}^{(m)T}\mathbf{r}^{(n+1)} = (\mathbf{s}^{(m)}, \mathbf{d}^{(0)})_A + \lambda^{(m)}(\mathbf{s}^{(m)}, \mathbf{s}^{(m)})_A = \mathbf{s}^{(m)T}\mathbf{r}^{(0)} + \lambda^{(m)}\mathbf{s}^{(m)T}A\mathbf{s}^{(m)}. \tag{5.166}$$

If we choose

$$\lambda^{(m)} = -\frac{\mathbf{s}^{(m)T}\mathbf{r}^{(0)}}{\mathbf{s}^{(m)T}A\mathbf{s}^{(m)}} \tag{5.167}$$

the projection of the new residual $\mathbf{r}^{(n+1)}$onto K_{n+1} vanishes, i.e. this is the optimal choice of the parameters $\lambda^{(0)} \ldots \lambda^{(n)}$.

Obviously the first search vector must have the direction of $\mathbf{r}^{(0)}$ to span the same one-dimensional space. Therefore we begin the iteration with

$$\mathbf{s}^{(0)} = \mathbf{r}^{(0)} \tag{5.168}$$

$$\lambda^{(0)} = -\frac{|\mathbf{r}^{(0)}|^2}{\mathbf{r}^{(0)T}A\mathbf{r}^{(0)}} \tag{5.169}$$

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \lambda^{(0)}\mathbf{s}^{(\mathbf{0})}. \tag{5.170}$$

For the next step we apply Gram-Schmidt orthogonalization

$$\mathbf{s}^{(1)} = \mathbf{r}^{(1)} - \mathbf{s}^{(0)}\frac{(\mathbf{r}^{(1)}, \mathbf{s}^{(0)})_A}{(\mathbf{s}^{(0)}, \mathbf{s}^{(0)})_A}. \tag{5.171}$$

For all further steps we have to orthogonalize $\mathbf{s}^{(n+1)}$with respect to all of $\mathbf{s}^{(n)} \ldots \mathbf{s}^{(0)}$. But, fortunately, the residual $\mathbf{r}^{(n+1)}$is already $A-$orthogonal to $\mathbf{s}^{(n-1)} \ldots \mathbf{s}^{(0)}$. This can be seen from (5.161)

$$\mathbf{r}^{(j+1)} - \mathbf{r}^{(j)} = \lambda^{(j)}A\mathbf{s}^{(j)} \tag{5.172}$$

$$(\mathbf{r}^{(n+1)}, \mathbf{s}^{(j)})_A = \mathbf{r}^{(n+1)T}A\mathbf{s}^{(j)} = \frac{1}{\lambda^{(j)}}\mathbf{r}^{(n+1)T}(\mathbf{r}^{(j+1)} - \mathbf{r}^{(j)}). \tag{5.173}$$

We already found, that $\mathbf{r}^{(n+1)}$ is orthogonal to K_{n+1} , hence to all $\mathbf{r}^{(n)}, \ldots \mathbf{r}^{(0)}$. Therefore we conclude

$$(\mathbf{r}^{(n+1)}, \mathbf{s}^{(j)})_A = 0 \quad \text{for } j+1 \leq n. \tag{5.174}$$

Finally we end up with the following procedure

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \lambda^{(n)}\mathbf{s}^{(n)} \quad \text{with } \lambda^{(n)} = -\frac{\mathbf{s}^{(n)T}\mathbf{r}^{(0)}}{\mathbf{s}^{(n)T}A\mathbf{s}^{(n)}} \tag{5.175}$$

$$\mathbf{r}^{(n+1)} = \mathbf{r}^{(n)} + \lambda^{(n)}A\mathbf{s}^{(n)} \tag{5.176}$$

$$\mathbf{s}^{(n+1)} = \mathbf{r}^{(n+1)} - \beta^{(n)}\mathbf{s}^{(n)} \quad \text{with } \beta^{(n)} = \frac{\mathbf{r}^{(n+1)T} A\mathbf{s}^{(n)}}{\mathbf{s}^{(n)T} A\mathbf{s}^{(n)}}. \tag{5.177}$$

This method [25] solves a linear system without storing the matrix A itself. Only the product $A\mathbf{s}$ is needed. In principle the solution is reached after $N = \dim(A)$ steps, but due to rounding errors more steps can be necessary and the method has to be considered as an iterative one.

The expressions for λ and β can be brought to numerically more efficient form. From (5.162) we find

$$\mathbf{s}^{(n)T}\mathbf{r}^{(0)} = \mathbf{s}^{(n)T}\left(\mathbf{r}^{(n)} - \sum_{j=0}^{n-1} \lambda^{(j)} A\mathbf{s}^{(\mathbf{j})}\right). \tag{5.178}$$

But due to A-orthogonality of the search directions

$$\mathbf{s}^{(n)T}\mathbf{r}^{(0)} = \mathbf{s}^{(n)T}\mathbf{r}^{(n)} = \mathbf{r}^{(n)T}\mathbf{r}^{(n)} \tag{5.179}$$

which simplifies

$$\lambda^{(n)} = -\frac{\mathbf{r}^{(n)T}\mathbf{r}^{(n)}}{\mathbf{s}^{(n)T} A\mathbf{s}^{(n)}}. \tag{5.180}$$

Furthermore, from (5.176) and orthogonality of the residual vectors

$$\mathbf{r}^{(n+1)T}\mathbf{r}^{(n+1)} = \lambda^{(n)}\mathbf{r}^{(n+1)T} A\mathbf{s}^{(n)} \tag{5.181}$$

from which

$$\begin{aligned}\beta^{(n)} &= \frac{\mathbf{r}^{(n+1)T} A\mathbf{s}^{(n)}}{\mathbf{s}^{(n)T} A\mathbf{s}^{(n)}} = \frac{-\frac{1}{\lambda^{(n)}}\mathbf{r}^{(n+1)T}\mathbf{r}^{(n+1)}}{\mathbf{s}^{(n)T} A\mathbf{s}^{(n)}} \\ &= -\frac{\mathbf{r}^{(n+1)T}\mathbf{r}^{(n+1)}}{\mathbf{r}^{(n)T}\mathbf{r}^{(n)}}.\end{aligned} \tag{5.182}$$

The conjugate gradients method is not useful for non symmetric systems. It can be applied to the normal equations (11.32)

$$A^T A\mathbf{x} = A^T\mathbf{b} \tag{5.183}$$

which, for a full-rank non singular matrix have the same solution. The condition number (Sect. 5.7), however, is

$$\text{cond}(A^T A) = (\text{cond}A)^2 \tag{5.184}$$

and the problem may be ill conditioned.

5.6.5 Non Symmetric Systems

The general minimum residual method (GMRES) searches directly for the minimum of $||A\mathbf{x}-\mathbf{b}||$ in the Krylov spaces of increasing order $K_n(A, \mathbf{r}^{(0)})$. To avoid problems with linear dependency, first an orthogonal basis

$$K_n(A, \mathbf{r}^{(0)}) = \text{span}(\mathbf{q}_1, \mathbf{q}_2, \ldots \mathbf{q}_n) \tag{5.185}$$

is constructed with Arnoldi's method, a variant of Gram-Schmidt orthogonalization. Starting from the normalized vector

$$\mathbf{q}_1 = \frac{\mathbf{r}^{(0)}}{|\mathbf{r}^{(0)}|} \tag{5.186}$$

the dimension is iteratively increased by orthogonalizing the vector $A\mathbf{q}_n$ with respect to $K_n(A, \mathbf{r}^{(0)})$[7]

$$\tilde{\mathbf{q}}_{n+1} = A\mathbf{q}_n - \sum_{j=1}^{n} (\mathbf{q}_j, \mathbf{q}_n)_A \mathbf{q}_j = A\mathbf{q}_n - \sum_{j=1}^{n} (\mathbf{q}_j^T A\mathbf{q}_n)\mathbf{q}_j = A\mathbf{q}_n - \sum_{j=1}^{n} h_{jn}\mathbf{q}_j \tag{5.187}$$

and normalizing this vector

$$h_{n+1,n} = |\tilde{\mathbf{q}}_{n+1}| \quad \mathbf{q}_{n+1} = \frac{\tilde{\mathbf{q}}_{n+1}}{h_{n+1,n}}. \tag{5.188}$$

Then

$$A\mathbf{q}_n = h_{n+1,n}\mathbf{q}_{n+1} + \sum_{j=1}^{n} h_{jn}\mathbf{q}_j \tag{5.189}$$

which explicitly reads

$$A\mathbf{q}_1 = h_{21}\mathbf{q}_2 + h_{11}\mathbf{q}_1 = (\mathbf{q}_1, \mathbf{q}_2)\begin{pmatrix} h_{11} \\ h_{21} \end{pmatrix} \tag{5.190}$$

$$A\mathbf{q}_2 = h_{32}\mathbf{q}_3 + h_{12}\mathbf{q}_1 + h_{22}\mathbf{q}_2 = (\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)\begin{pmatrix} h_{12} \\ h_{22} \\ h_{32} \end{pmatrix} \tag{5.191}$$

$$\vdots$$

[7] $\mathbf{q}_2$ is a linear combination of $\mathbf{q}_1$ and $A\mathbf{q}_1$, $\mathbf{q}_3$ one of $\mathbf{q}_1$, $\mathbf{q}_2$ and $A\mathbf{q}_2$ hence also of $\mathbf{q}_1$, $A\mathbf{q}_1$, $A^2\mathbf{q}_1$ etc. which proves the validity of (5.185).

$$A\mathbf{q}_n = h_{n+1,n}\mathbf{q}_{n+1} + h_{1n}\mathbf{q}_1 + \ldots h_{nn}\mathbf{q}_n. \tag{5.192}$$

The new basis vectors are orthogonal[8] since

$$\mathbf{q}_1^T\tilde{\mathbf{q}}_2 = \mathbf{q}_1^T\left[A\mathbf{q}_1 - (\mathbf{q}_1^T A\mathbf{q}_1)\mathbf{q}_1\right] = (\mathbf{q}_1^T A\mathbf{q}_1)(1 - |\mathbf{q}_1|^2) = 0$$

and induction shows for $k = 1 \ldots n$

$$\mathbf{q}_k^T\tilde{\mathbf{q}}_{n+1} = \mathbf{q}_k^T A\mathbf{q}_n - \sum_{j=1}^{n}(\mathbf{q}_j^T A\mathbf{q}_n)(\mathbf{q}_k^T\mathbf{q}_j) = \mathbf{q}_k^T A\mathbf{q}_n - \sum_{j=1}^{n}(\mathbf{q}_j^T A\mathbf{q}_n)\delta_{k,j} = 0.$$

We collect the new basis vectors $\mathbf{q}_1 \ldots \mathbf{q}_n$ into a matrix

$$U_n = (\mathbf{q}_1, \ldots \mathbf{q}_n) \tag{5.193}$$

and obtain from (5.190) to (5.192)

$$AU_n = U_{n+1}H \tag{5.194}$$

with the $(n+1) \times n$ upper Hessenberg matrix

$$H = \begin{pmatrix} h_{11} & h_{12} & \ldots & h_{1n} \\ h_{21} & h_{22} & & h_{2n} \\ & h_{32} & \ddots & \vdots \\ & & \ddots & h_{nn} \\ & & & h_{n+1,n} \end{pmatrix}. \tag{5.195}$$

Since

$$\mathbf{x}^{(n)} - \mathbf{x}_0 \in K_n(A, \mathbf{r}^{(0)}) \tag{5.196}$$

it can be written as a linear combination of $\mathbf{q}_1 \ldots \mathbf{q}_n$

$$\mathbf{x}^{(n)} - \mathbf{x}_0 = (\mathbf{q}_1 \ldots \mathbf{q}_n)\mathbf{v}. \tag{5.197}$$

The residual becomes

$$\begin{aligned}\mathbf{r}^{(n)} &= A(\mathbf{q}_1 \ldots \mathbf{q}_n)\mathbf{v} + A\mathbf{x}_0 - \mathbf{b} = A(\mathbf{q}_1 \ldots \mathbf{q}_n)\mathbf{v} + \mathbf{r}^{(0)} \\ &= U_{n+1}H\mathbf{v} + |\mathbf{r}^{(0)}|\mathbf{q}_1\end{aligned}$$

[8] If $\mathbf{q}_{n+1} = 0$ the algorithm has to stop and the Krylov space has the full dimension of the matrix.

$$= U_{n+1} \left[H\mathbf{v} + |\mathbf{r}^{(0)}| \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \right] \tag{5.198}$$

hence

$$|\mathbf{r}^{(n)}|^2 = \left[H\mathbf{v} + |\mathbf{r}^{(0)}| \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \right]^T U_{n+1}^T U_{n+1} \left[H\mathbf{v} + |\mathbf{r}^{(0)}| \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \right]. \tag{5.199}$$

But since

$$U_{n+1}^T U_{n+1} = \begin{pmatrix} \mathbf{q_1}^T \\ \vdots \\ \mathbf{q}_n^T \end{pmatrix} (\mathbf{q}_1 \dots \mathbf{q}_n) = \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix} \tag{5.200}$$

is a $n \times n$ unit matrix, we have to minimize

$$|\mathbf{r}^{(n)}| = \left| H\mathbf{v} + |\mathbf{r}^{(0)}| \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \right| \tag{5.201}$$

which is a least square problem, since there are $n + 1$ equations for the n unknown components of $\mathbf{v}$. It can be solved efficiently with the help of QR decomposition (11.36)

$$H = Q \begin{pmatrix} R \\ 0 \end{pmatrix} \quad R = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ & \ddots & \vdots \\ & & r_{nn} \end{pmatrix} \tag{5.202}$$

with a $(n + 1) \times (n + 1)$ orthogonal matrix Q. The norm of the residual vector becomes

$$|\mathbf{r}^{(n)}| = \left| Q \begin{pmatrix} R \\ 0 \end{pmatrix} \mathbf{v} + |\mathbf{r}^{(0)}| \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \right| = \left| \begin{pmatrix} R\mathbf{v} \\ 0 \end{pmatrix} + |\mathbf{r}^{(0)}| Q^T \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \right|. \tag{5.203}$$

Substituting

$$|\mathbf{r}^{(0)}|Q^T \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{y} \\ y_{n+1} \end{pmatrix} \tag{5.204}$$

we have

$$|\mathbf{r}^{(n)}| = \sqrt{y_{n+1}^2 + |R\mathbf{v} + \mathbf{y}|^2} \tag{5.205}$$

which is obviously minimized by solving the triangular system

$$R\mathbf{v} + \mathbf{y} = 0. \tag{5.206}$$

The GMRES method usually has to be preconditioned (cf.5.112) to improve convergence. Often it is restarted after a small number (e.g.20) of iterations which avoids the necessity to store a large orthogonal basis.

5.7 Matrix Inversion

LU and QR decomposition can be also used to calculate the inverse of a non singular matrix

$$AA^{-1} = 1. \tag{5.207}$$

The decomposition is performed once and then the column vectors of A^{-1} are calculated similar to (5.27)

$$L\left(UA^{-1}\right) = 1 \tag{5.208}$$

or (5.40)

$$RA^{-1} = Q^{\dagger}. \tag{5.209}$$

Consider now a small variation of the right hand side of (5.2)

$$\mathbf{b} + \Delta\mathbf{b}. \tag{5.210}$$

Instead of

$$A^{-1}\mathbf{b} = \mathbf{x} \tag{5.211}$$

the resulting vector is

$$A^{-1}(\mathbf{b} + \Delta\mathbf{b}) = \mathbf{x} + \Delta\mathbf{x} \tag{5.212}$$

and the deviation can be measured by[9]

$$||\Delta\mathbf{x}|| = ||A^{-1}||\,||\Delta\mathbf{b}|| \tag{5.213}$$

and since

$$||A||\,||\mathbf{x}|| = ||\mathbf{b}|| \tag{5.214}$$

the relative error becomes

$$\frac{||\Delta\mathbf{x}||}{||\mathbf{x}||} = ||A||\,||A^{-1}||\,\frac{||\Delta\mathbf{b}||}{||\mathbf{b}||}. \tag{5.215}$$

The relative error of $\mathbf{b}$ is multiplied by the condition number for inversion

$$\text{cond}(A) = ||A||\,||A^{-1}||. \tag{5.216}$$

Problem

Problem 5.1 (Comparison of different direct Solvers, Fig. 5.2) In this computer experiment we solve the system of equations

$$A\mathbf{x} = \mathbf{b} \tag{5.217}$$

with several methods:

- Gaussian elimination without pivoting (Sect. 5.1),
- Gaussian elimination with partial pivoting (Sect. 5.1.1),
- QR decomposition with Householder reflections (Sect. 5.2.2),
- QR decomposition with Gram-Schmidt orthogonalization (Sect. 5.2.1),
- QR decomposition with Gram-Schmidt orthogonalization with extra orthogonalization step (5.55).

The right hand side is chosen as

[9] The vector norm used here is not necessarily the Euclidean norm.

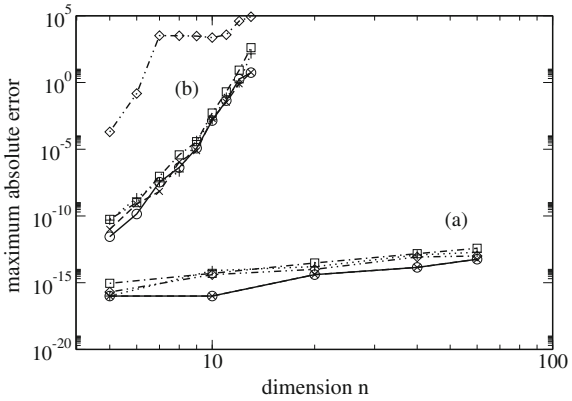


Fig. 5.2 (Comparison of different direct solvers) Gaussian elimination without (*circles*) and with (x) pivoting, QR decomposition with Householder reflections (*squares*), with Gram-Schmidt orthogonalization (*diamonds*) and including extra orthogonalization (+) are compared. The maximum difference $\max_{i=1\ldots n}(|x_i - x_i^{exact}|)$ increases only slightly with the dimension n for the well behaved matrix (5.224,a) but quite dramatically for the ill conditioned Hilbert matrix (5.226,b)

$$\mathbf{b} = A \begin{pmatrix} 1 \\ 2 \\ \vdots \\ n \end{pmatrix} \tag{5.218}$$

hence the exact solution is

$$\mathbf{x} = \begin{pmatrix} 1 \\ 2 \\ \vdots \\ n \end{pmatrix}. \tag{5.219}$$

Several test matrices can be chosen:

- Gaussian elimination is theoretically unstable but is stable in many practical cases. The instability can be demonstrated with the example [38]

$$A = \begin{pmatrix} 1 & & & & 1 \\ -1 & 1 & & & 1 \\ -1 & -1 & 1 & & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & -1 & 1 \end{pmatrix}. \tag{5.220}$$

  No pivoting takes place in the LU decomposition of this matrix and the entries in the last column double in each step:

$$A^{(1)} = \begin{pmatrix} 1 & & & 1 \\ & 1 & & 2 \\ -1 & 1 & & 2 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & 2 \end{pmatrix} \quad A^{(2)} = \begin{pmatrix} 1 & & & 1 \\ & 1 & & 2 \\ & & 1 & 4 \\ & & \vdots & \ddots & \vdots \\ & & -1 & -1 & 4 \end{pmatrix} \ldots A^{(n-1)} = \begin{pmatrix} 1 & & & 1 \\ & 1 & & 2 \\ & & \ddots & & 4 \\ & & & \ddots & \vdots \\ & & & & 2^{n-1} \end{pmatrix}. \tag{5.221}$$

Since the machine precision is $\epsilon_M = 2^{-53}$ for double precision calculations we have to expect numerical inaccuracy for dimension $n > 53$.

- Especially well conditioned are matrices [46] which are symmetric

$$A_{ij} = A_{ji} \tag{5.222}$$

and also diagonal dominant

$$\sum_{j \neq i} |A_{ij}| < |A_{ii}|. \tag{5.223}$$

We use the matrix

$$A = \begin{pmatrix} n & 1 & \ldots & 1 & 1 \\ 1 & n & \ldots & 1 & 1 \\ \vdots & & \ddots & & \vdots \\ 1 & 1 & 1 & n & 1 \\ 1 & 1 & 1 & 1 & n \end{pmatrix} \tag{5.224}$$

which can be inverted explicitly by

$$A^{-1} = \begin{pmatrix} a & b & \ldots & b & b \\ b & a & & b & b \\ \vdots & & \ddots & & \\ b & b & b & a & b \\ b & b & b & b & a \end{pmatrix} \quad a = \frac{1}{n - \frac{1}{2}}, b = -\frac{1}{2n^2 - 3n + 1} \tag{5.225}$$

and has a condition number[10] which is proportional to the dimension n (Fig. 5.3).

- The Hilbert matrix [47, 48]

[10] Using the Frobenius norm $||A|| = \sqrt{\sum_{ij} A_{ij}^2}$.

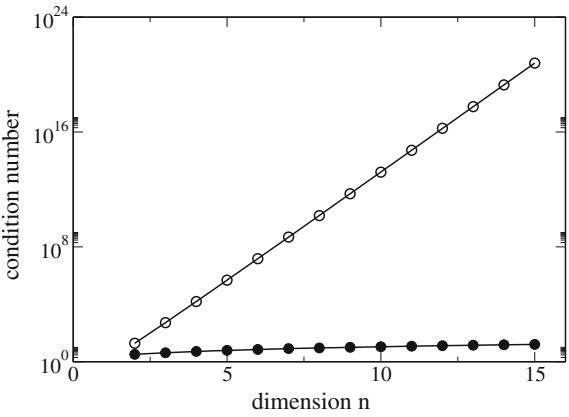


Fig. 5.3 (Condition numbers) The condition number $cond(A)$ increases only linearly with the dimension n for the well behaved matrix (5.224, *full circles*) but exponentially for the ill conditioned Hilbert matrix (5.226, *open circles*)

$$A = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & & \frac{1}{n+2} \\ \vdots & & & \ddots & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n-1} \end{pmatrix} \tag{5.226}$$

is especially ill conditioned [49] even for moderate dimension. It is positive definite and therefore the inverse matrix exists and even can be written down explicitly [50]. Its column vectors are very close to linearly dependent and the condition number grows exponentially with its dimension (Fig. 5.3). Numerical errors are large for all methods compared (Fig. 5.2).

- random matrices

$$A_{ij} = \xi \in [-1, 1]. \tag{5.227}$$

Chapter 6
Roots and Extremal Points

In computational physics very often roots of a function, i.e. solutions of an equation like

$$f(x_1 \cdots x_N) = 0 \tag{6.1}$$

have to be determined. A related problem is the search for local extrema (Fig. 6.1)

$$\max f(x_1 \cdots x_N) \quad \min f(x_1 \cdots x_N) \tag{6.2}$$

which for a smooth function are solutions of the equations

$$\frac{\partial f(x_1 \cdots x_N)}{\partial x_i} = 0, \quad i = 1 \ldots N. \tag{6.3}$$

In one dimension bisection is a very robust but rather inefficient root finding method. If a good starting point close to the root is available and the function smooth enough, the Newton–Raphson method converges much faster. Special strategies are necessary to find roots of not so well behaved functions or higher order roots. The combination of bisection and interpolation like in Dekker's and Brent's methods provides generally applicable algorithms. In multidimensions calculation of the Jacobian matrix is not always possible and Quasi-Newton methods are a good choice. Whereas local extrema can be found as the roots of the gradient, at least in principle, direct optimization can be more efficient. In one dimension the ternary search method or Brent's more efficient golden section search method can be used. In multidimensions the class of direction set search methods is very popular which includes the methods of steepest descent and conjugate gradients, the Newton–Raphson method and, if calculation of the full Hessian matrix is too expensive, the Quasi-Newton methods.

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_6

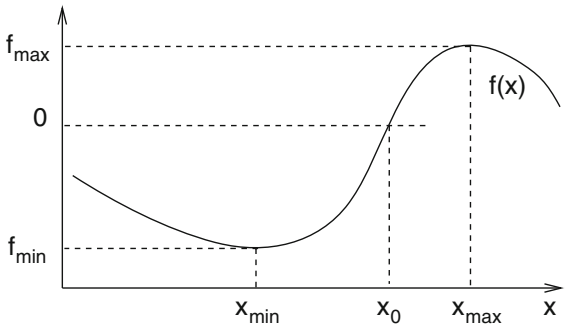


Fig. 6.1 Roots and local extrema of a function

6.1 Root Finding

If there is exactly one root in the interval $a_0 < x < b_0$ then one of the following methods can be used to locate the position with sufficient accuracy. If there are multiple roots, these methods will find one of them and special care has to be taken to locate the other roots.

6.1.1 Bisection

The simplest method [51] to solve

$$f(x) = 0 \tag{6.4}$$

uses the following algorithm (Fig. 6.2):

(1) Determine an interval $[a_0, b_0]$, which contains a sign change of $f(x)$. If no such interval can be found then $f(x)$ does not have any zero crossings
(2) Divide the interval into $[a_0, a_0 + \frac{b_0 - a_0}{2}]$ $[a_0 + \frac{b_0 - a_0}{2}, b_0]$ and choose that interval $[a_1, b_1]$, where $f(x)$ changes its sign.
(3) repeat until the width $b_n - a_n < \varepsilon$ is small enough.[1]

The bisection method needs two starting points which bracket a sign change of the function. It converges but only slowly since each step reduces the uncertainty by a factor of 2.

[1] Usually a combination like $\varepsilon = 2\varepsilon_M + |b_n|\varepsilon_r$ of an absolute and a relative tolerance is taken.

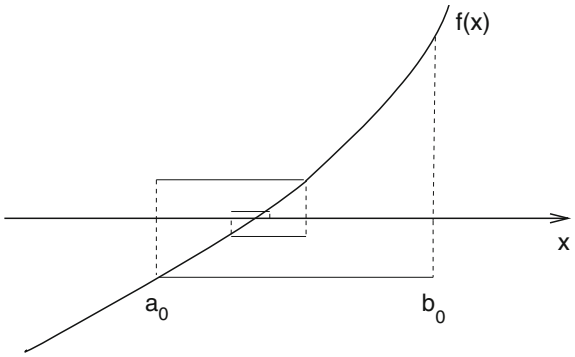


Fig. 6.2 Root finding by bisection

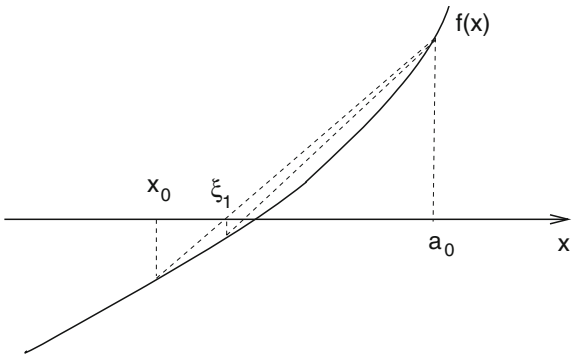


Fig. 6.3 Regula falsi method

6.1.2 Regula Falsi (False Position) Method

The regula falsi [52] method (Fig. 6.3) is similar to the bisection method [51]. However, polynomial interpolation is used to divide the interval $[x_r, a_r]$ with $f(x_r) f(a_r) < 0$. The root of the linear polynomial

$$p(x) = f(x_r) + (x - x_r) \frac{f(a_r) - f(x_r)}{a_r - x_r} \tag{6.5}$$

is given by

$$\xi_r = x_r - f(x_r) \frac{a_r - x_r}{f(a_r) - f(x_r)} = \frac{a_r f(x_r) - x_r f(a_r)}{f(x_r) - f(a_r)} \tag{6.6}$$

which is inside the interval $[x_r, a_r]$. Choose the sub-interval which contains the sign change:

$$f(x_r)f(\xi_r) < 0 \rightarrow [x_{r+1}, a_{r+1}] = [x_r, \xi_r]$$
$$f(x_r)f(\xi_r) > 0 \rightarrow [x_{r+1}, a_{r+1}] = [\xi_r, a_r]. \tag{6.7}$$

Then ξ_r provides a series of approximations with increasing precision to the root of $f(x) = 0$.

6.1.3 Newton–Raphson Method

Consider a function which is differentiable at least two times around the root ξ. Taylor series expansion around a point x_0 in the vicinity

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{1}{2}(x - x_0)^2 f''(x_0) + \cdots \tag{6.8}$$

gives for $x = \xi$

$$0 = f(x_0) + (\xi - x_0)f'(x_0) + \frac{1}{2}(\xi - x_0)^2 f''(x_0) + \cdots . \tag{6.9}$$

Truncation of the series and solving for ξ gives the first order Newton–Raphson [51, 53] method (Fig. 6.4)

$$x_{r+1} = x_r - \frac{f(x_r)}{f'(x_r)} \tag{6.10}$$

and the second order Newton–Raphson method (Fig. 6.4)

$$x_{r+1} = x_r - \frac{f'(x_r) \pm \sqrt{f'(x_r)^2 - 2f(x_r)f''(x_r)}}{f''(x_r)}. \tag{6.11}$$

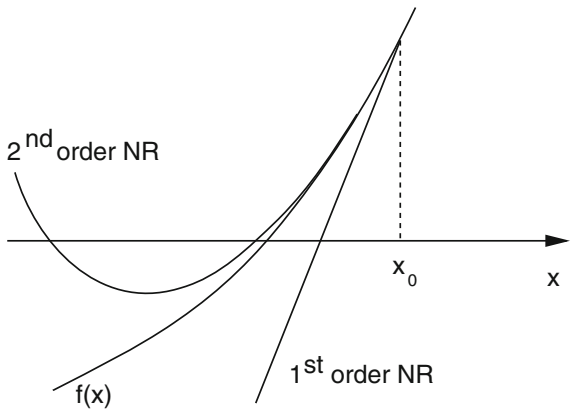


Fig. 6.4 Newton–Raphson method

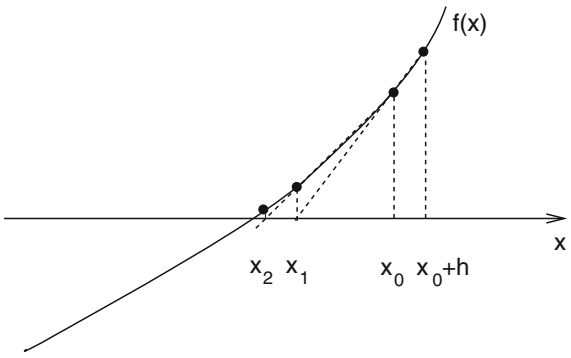


Fig. 6.5 Secant method

The Newton–Raphson method converges fast if the starting point is close enough to the root. Analytic derivatives are needed. It may fail if two or more roots are close by.

6.1.4 Secant Method

Replacing the derivative in the first order Newton Raphson method by a finite difference quotient gives the secant method [51] (Fig. 6.5) which has been known for thousands of years before [54]

$$x_{r+1} = x_r - f(x_r)\frac{x_r - x_{r-1}}{f(x_r) - f(x_{r-1})}. \tag{6.12}$$

Round-off errors can become important as $|f(x_r) - f(x_{r-1})|$ gets small. At the beginning choose a starting point x_0 and determine

$$x_1 = x_0 - f(x_0)\frac{2h}{f(x_0 + h) - f(x_0 - h)} \tag{6.13}$$

using a symmetrical difference quotient.

6.1.5 Interpolation

The secant method is also obtained by linear interpolation

$$p(x) = \frac{x - x_r}{x_{r-1} - x_r} f_{r-1} + \frac{x - x_{r-1}}{x_r - x_{r-1}} f_r. \tag{6.14}$$

The root of the polynomial $p(x_{r+1}) = 0$ determines the next iterate x_{r+1}

$$x_{r+1} = \frac{1}{f_{r-1} - f_r}(x_r f_{r-1} - x_{r-1} f_r) = x_r - f_r \frac{x_r - x_{r-1}}{f_r - f_{r-1}}. \tag{6.15}$$

Quadratic interpolation of three function values is known as Muller's method [55]. Newton's form of the interpolating polynomial is

$$p(x) = f_r + (x - x_r) f[x_r, x_{r-1}] + (x - x_r)(x - x_{r-1}) f[x_r, x_{r-1}, x_{r-2}] \tag{6.16}$$

which can be rewritten

$$\begin{aligned} p(x) &= f_r + (x - x_r) f[x_r, x_{r-1}] + (x - x_r)^2 f[x_r, x_{r-1}, x_{r-2}] \\ &+ (x_r - x_{r-1})(x - x_r) f[x_r, x_{r-1}, x_{r-2}] \\ &= f_r + (x - x_r)^2 f[x_r, x_{r-1}, x_{r-2}] + (x - x_r)\left(f[x_r, x_{r-1}] + f[x_r, x_{r-2}] - f[x_{r-1}, x_{r-2}]\right) \\ &= f_r + A(x - x_r) + B(x - x_r)^2 \end{aligned} \tag{6.17}$$

and has the roots

$$x_{r+1} = x_r - \frac{A}{2B} \pm \sqrt{\frac{A^2}{4B^2} - \frac{f_r}{B}}. \tag{6.18}$$

To avoid numerical cancellation, this is rewritten

$$\begin{aligned} x_{r+1} &= x_r + \frac{1}{2B}\left(-A \pm \sqrt{A^2 - 4Bf_r}\right) \\ &= x_r + \frac{-2f_r}{A^2 - (A^2 - 4Bf_r)}\left(A \mp \sqrt{A^2 - 4Bf_r}\right) \\ &= x_r + \frac{-2f_r}{A \pm \sqrt{A^2 - 4Bf_r}}. \end{aligned} \tag{6.19}$$

The sign in the denominator is chosen such that x_{r+1} is the root closer to x_r. The roots of the polynomial can become complex valued and therefore this method is useful to find complex roots.

6.1.6 *Inverse Interpolation*

Complex values of x_r can be avoided by interpolation of the inverse function instead

$$x = f^{-1}(y). \tag{6.20}$$

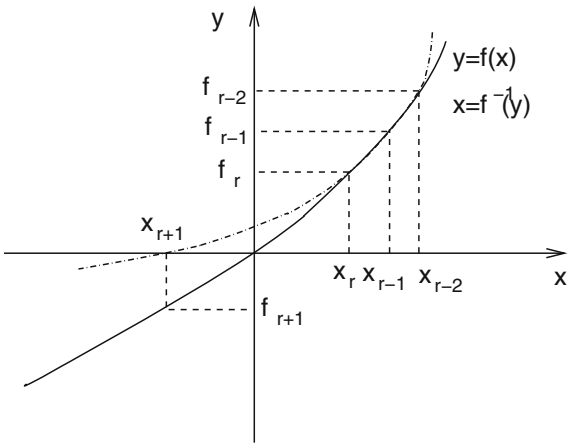


Fig. 6.6 Root finding by interpolation of the inverse function

Using the two points x_r, x_{r-1} the Lagrange method gives

$$p(y) = x_{r-1}\frac{y - f_r}{f_{r-1} - f_r} + x_r\frac{y - f_{r-1}}{f_r - f_{r-1}} \tag{6.21}$$

and the next approximation of the root corresponds again to the secant method (6.12)

$$x_{r+1} = p(0) = \frac{x_{r-1}f_r - x_r f_{r-1}}{f_r - f_{r-1}} = x_r + \frac{(x_{r-1} - x_r)}{f_r - f_{r-1}}f_r. \tag{6.22}$$

Inverse quadratic interpolation needs three starting points x_r, x_{r-1}, x_{r-2} together with the function values f_r, f_{r-1}, f_{r-2} (Fig. 6.6). The inverse function $x = f^{-1}(y)$ is interpolated with the Lagrange method

$$\begin{aligned} p(y) = {} & \frac{(y - f_{r-1})(y - f_r)}{(f_{r-2} - f_{r-1})(f_{r-2} - f_r)}x_{r-2} + \frac{(y - f_{r-2})(y - f_r)}{(f_{r-1} - f_{r-2})(f_{r-1} - f_r)}x_{r-1} \\ & + \frac{(y - f_{r-1})(y - f_{r-2})}{(f_r - f_{r-1})(f_r - f_{r-2})}x_r. \end{aligned} \tag{6.23}$$

For $y = 0$ we find the next iterate

$$\begin{aligned} x_{r+1} = p(0) = {} & \frac{f_{r-1}f_r}{(f_{r-2} - f_{r-1})(f_{r-2} - f_r)}x_{r-2} + \frac{f_{r-2}f_r}{(f_{r-1} - f_{r-2})(f_{r-1} - f_r)}x_{r-1} \\ & + \frac{f_{r-1}f_{r-2}}{(f_r - f_{r-1})(f_r - f_{r-2})}x_r. \end{aligned} \tag{6.24}$$

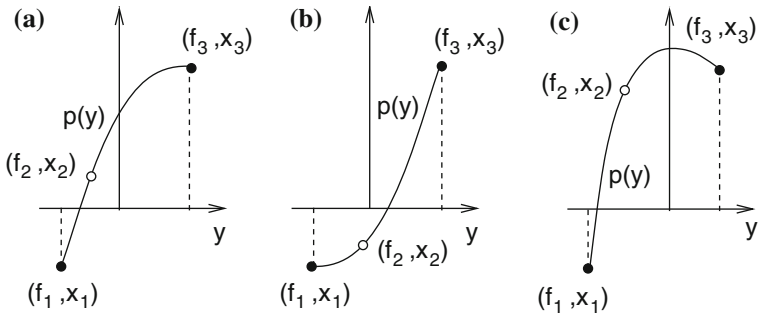


Fig. 6.7 (Validity of inverse quadratic interpolation) Inverse quadratic interpolation is only applicable if the interpolating polynomial $p(y)$ is monotonous in the range of the interpolated function values $f_1 \ldots f_3$. (**a**) and (**b**) show the limiting cases where the polynomial has a horizontal tangent at f_1 or f_3. (**c**) shows the case where the extremum of the parabola is inside the interpolation range and interpolation is not feasible

Inverse quadratic interpolation is only a good approximation if the interpolating parabola is single valued and hence if it is a monotonous function in the range of f_r, f_{r-1}, f_{r-2}. For the following discussion we assume that the three values of x are renamed such that $x_1 < x_2 < x_3$.

Consider the limiting case (a) in Fig. 6.7 where the polynomial has a horizontal tangent at $y = f_3$ and can be written as

$$p(y) = x_3 + (x_1 - x_3)\frac{(y - f_3)^2}{(f_1 - f_3)^2}. \tag{6.25}$$

Its value at $y = 0$ is

$$p(0) = x_3 + (x_1 - x_3)\frac{f_3^2}{(f_1 - f_3)^2} = x_1 + (x_3 - x_1)\left(1 - \frac{f_3^2}{(f_1 - f_3)^2}\right). \tag{6.26}$$

If f_1 and f_3 have different sign and $|f_1| < |f_3|$ (Sect. 6.1.7.2) we find

$$1 - \frac{f_3^2}{(f_1 - f_3)^2} < \frac{3}{4}. \tag{6.27}$$

Brent [56] used this as a criterion for the applicability of the inverse quadratic interpolation. However, this does not include all possible cases where interpolation is applicable. Chandrupatla [57] gave a more general discussion. The limiting condition is that the polynomial $p(y)$ has a horizontal tangent at one of the boundaries $x_{1,3}$. The derivative values are

$$\frac{dp}{dy}(y=f_1)=\frac{x_2(f_1-f_3)}{(f_2-f_1)(f_2-f_3)}+\frac{x_3(f_1-f_2)}{(f_3-f_1)(f_3-f_2)}+\frac{x_1}{f_1-f_2}+\frac{x_1}{f_1-f_3} \tag{6.28}$$

$$=\frac{(f_2-f_1)}{(f_3-f_1)(f_3-f_2)}\left[\frac{x_2(f_3-f_1)^2}{(f_2-f_1)^2}-x_3-\frac{x_1(f_3-f_1)^2-x_1(f_2-f_1)^2}{(f_2-f_1)^2}\right]$$

$$=\frac{(f_2-f_1)(x_2-x_1)}{(f_3-f_1)(f_3-f_2)}\left[\Phi^{-2}-\xi^{-1}\right]$$

$$\frac{dp}{dy}(y=f_3)=\frac{x_2(f_3-f_1)}{(f_2-f_1)(f_2-f_3)}+\frac{x_1(f_3-f_2)}{(f_1-f_2)(f_1-f_3)}+\frac{x_3}{f_3-f_2}+\frac{x_3}{f_3-f_1} \tag{6.29}$$

$$=\frac{(f_3-f_2)}{(f_2-f_1)(f_3-f_1)}\left[-\frac{x_2(f_3-f_1)^2}{(f_3-f_2)^2}+x_3\frac{(f_3-f_1)^2}{(f_3-f_2)^2}-x_3\frac{(f_3-f_2)^2}{(f_3-f_2)^2}+x_1\right]$$

$$=\frac{(f_3-f_2)(x_3-x_2)}{(f_2-f_1)(f_3-f_1)}\left[\left(\frac{1}{\Phi-1}\right)^2-\frac{1}{1-\xi}\right]$$

with [57]

$$\xi=\frac{x_2-x_1}{x_3-x_1}\quad \Phi=\frac{f_2-f_1}{f_3-f_1} \tag{6.30}$$

$$\xi-1=\frac{x_2-x_3}{x_3-x_1}\quad \Phi-1=\frac{f_2-f_3}{f_3-f_1}. \tag{6.31}$$

Since for a parabola either $f_1 < f_2 < f_3$ or $f_1 > f_2 > f_3$ the conditions for applicability of inverse interpolation finally become

$$\Phi^2 < \xi \tag{6.32}$$

$$1-\xi > (1-\Phi)^2 \tag{6.33}$$

which can be combined into

$$1-\sqrt{1-\xi} < |\Phi| < \sqrt{\xi}. \tag{6.34}$$

This method is usually used in combination with other methods (Sect. 6.1.7.2).

6.1.7 Combined Methods

Bisection converges slowly. The interpolation methods converge faster but are less reliable. The combination of both gives methods which are reliable and converge faster than pure bisection.

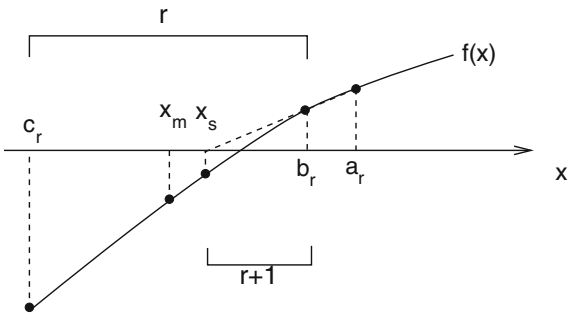


Fig. 6.8 Dekker's method

6.1.7.1 Dekker's Method

Dekker's method [58, 59] combines bisection and secant method. The root is bracketed by intervals $[c_r, b_r]$ with decreasing width where b_r is the best approximation to the root found and c_r is an earlier guess for which $f(c_r)$ and $f(b_r)$ have different sign. First an attempt is made to use linear interpolation between the points $(b_r, f(b_r))$ and $(a_r, f(a_r))$ where a_r is usually the preceding approximation $a_r = b_{r-1}$ and is set to the second interval boundary $a_r = c_{r-1}$ if the last iteration did not lower the function value (Fig. 6.8).

Starting from an initial interval $[x_0, x_1]$ with $\text{sign}(f(x_0)) \neq \text{sign}(f(x_1))$ the method proceeds as follows [59]:

initialization

$f_1 = f(x_1) \quad f_0 = f(x_0)$
if $|f_1| < |f_0|$ then {
$b = x_1 \quad c = a = x_0$
$f_b = f_1 \quad f_c = f_a = f_0\}$
$else\{$
$b = x_0 \quad c = a = x_1$
$f_b = f_0 \quad f_c = f_a = f_1\}$

iteration

$x_s = b - f_b \frac{b-a}{f_b - f_a}$
$x_m = \frac{c+b}{2}$.

If x_s is very close to the last b then increase the distance to avoid too small steps else choose x_s if it is between b and x_m, otherwise choose x_m (thus choosing the smaller interval)

$$x_r = \begin{cases} b + \delta \text{sign}(c-b) & \text{if abs}(x_s - b) < \delta \\ x_s \text{ if } b + \delta < x_s < x_m \text{ or } b - \delta > x_s > x_m & \\ x_m \text{ else} & \end{cases} .$$

Determine x_k as the latest of the previous iterates $x_0 \ldots x_{r-1}$ for which $\text{sign}(f(x_k)) \neq \text{sign}(f(x_r))$.

If the new function value is lower update the approximation to the root

$f_r = f(x_r)$
if $|f_r| < |f_k|$ then {
$a = b \quad b = x_r \quad c = x_k$
$f_a = f_b \quad f_b = f_r \quad f_c = f_k\}$

otherwise keep the old approximation and update the second interval boundary

if $|f_r| \geq |f_k|$ then {
$b = x_k \quad a = c = x_r$
$f_b = f_k \quad f_a = f_c = f_r\}$
repeat until $|c - b| < \varepsilon$ or $f_r = 0$.

6.1.7.2 Brent's Method

In certain cases Dekker's method converges very slowly making many small steps of the order ϵ. Brent [56, 59, 60] introduced some modifications to reduce such problems and tried to speed up convergence by the use of inverse quadratic interpolation (Sect. 6.1.6). To avoid numerical problems the iterate (6.24) is written with the help of a quotient

$$\begin{aligned} x_{r+1} &= \frac{f_b f_c}{(f_a - f_b)(f_a - f_c)} a + \frac{f_a f_c}{(f_b - f_a)(f_b - f_c)} b \\ &+ \frac{f_b f_a}{(f_c - f_b)(f_c - f_a)} c \\ &= b + \frac{p}{q} \end{aligned} \tag{6.35}$$

with

$$\begin{aligned} p &= \frac{f_b}{f_a}\left((c-b)\frac{f_a}{f_c}\left(\frac{f_a}{f_c} - \frac{f_b}{f_c}\right) - (b-a)\left(\frac{f_b}{f_c} - 1\right)\right) \\ &= (c-b)\frac{f_b(f_a - f_b)}{f_c^2} - (b-a)\frac{f_b(f_b - f_c)}{f_a f_c} \\ &= \frac{a f_b f_c(f_b - f_c) + b\left[f_a f_b(f_b - f_a) + f_b f_c(f_c - f_b)\right] + c f_a f_b(f_a - f_b)}{f_a f_c^2} \end{aligned} \tag{6.36}$$

$$q = -\left(\frac{f_a}{f_c} - 1\right)\left(\frac{f_b}{f_c} - 1\right)\left(\frac{f_b}{f_a} - 1\right) = -\frac{(f_a - f_c)(f_b - f_c)(f_b - f_a)}{f_a f_c^2}. \tag{6.37}$$

If only two points are available, linear interpolation is used. The iterate (6.22) then is written as

$$x_{r+1} = b + \frac{(a-b)}{f_b - f_a} f_b = b + \frac{p}{q} \tag{6.38}$$

with

$$p = (a-b)\frac{f_b}{f_a} \quad q = \left(\frac{f_b}{f_a} - 1\right). \tag{6.39}$$

The division is only performed if interpolation is appropriate and division by zero cannot happen. Brent's method is fast and robust at the same time. It is often recommended by text books. The algorithm is summarized in the following [61].
Start with an initial interval $[x_0, x_1]$ with $f(x_0)f(x_1) \leq 0$

initialization

$a = x_0 \quad b = x_1 \quad c = a$
$f_a = f(a) \quad f_b = f(b) \quad f_c = f_a$
$e = d = b - a$

iteration

If c is a better approximation than b exchange values

if $|f_c| < |f_b|$ then{
$a = b \quad b = c \quad c = a$
$f_a = f_b \quad f_b = f_c \quad f_c = f_a$}

calculate midpoint relative to b

$x_m = 0.5(c - b)$

stop if accuracy is sufficient

if $|x_m| < \varepsilon$ or $f_b = 0$ then exit

use bisection if the previous step width e was too small or the last step did not improve

if $|e| < \varepsilon$ or $|f_a| \leq |f_b|$ then{
$e = d = x_m$}

otherwise try interpolation

else {
if $a = c$ then {
$p = 2x_m \dfrac{f_b}{f_a} \quad q = \dfrac{f_b - f_a}{f_a}$}
else {
$p = 2x_m \dfrac{f_b(f_a - f_b)}{f_c^2} - (b - a)\dfrac{f_b(f_b - f_c)}{f_a f_c}$
$q = \left(\dfrac{f_a}{f_c} - 1\right)\left(\dfrac{f_b}{f_c} - 1\right)\left(\dfrac{f_b}{f_a} - 1\right)$}

make p a positive quantity

if $p > 0$ then $\{q = -q\}$ else $\{p = -p\}$

update previous step width
$s = e \quad e = d$
use interpolation if applicable, otherwise use bisection

if $2p < 3x_m q - |\varepsilon q|$ and $p < |0.5\, sq|$ then{
$d = \frac{p}{q}$}
else$\{e = d = x_m\}$
$a = b \quad f_a = f_b$

if $|d| > \varepsilon$ then {
$b = b + d$}
else {b = b + εsign (x_m)}

calculate new function value
$f_b = f(b)$
be sure to bracket the root
if sign(f_b) = sign(f_c) then {
$c = a \quad f_c = f_a$
$e = d = b - a$}

6.1.7.3 Chandrupatla's Method

In 1997 Chandrupatla [57] published a method which tries to use inverse quadratic interpolation whenever possible according to (6.34). He calculates the relative position of the new iterate as (Fig. 6.9).

$$t = \frac{x - c}{b - c}$$

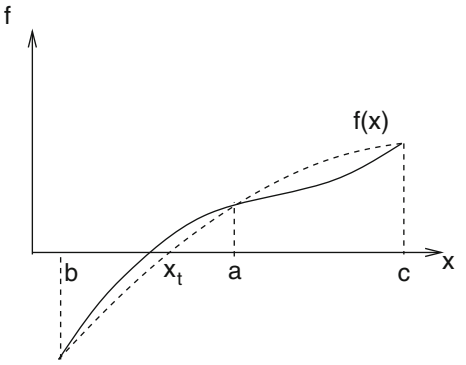


Fig. 6.9 Chandrupatla's method

$$= \frac{1}{b-c}\left[\frac{f_c f_b}{(f_a-f_b)(f_a-f_c)}a + \frac{f_a f_c}{(f_b-f_a)(f_b-f_c)}b + \frac{f_b f_a}{(f_c-f_b)(f_c-f_a)}c - c\right]$$
$$= \frac{a-c}{b-c}\frac{f_c}{f_c-f_a}\frac{f_b}{f_b-f_a} + \frac{f_a f_c}{(f_b-f_a)(f_b-f_c)}. \tag{6.40}$$

The algorithm proceeds as follows:
Start with an initial interval $[x_0, x_1]$ with $f(x_0)f(x_1) \leq 0$.

initialization

$b = x_0 \quad a = c = x_1$
$f_b = f(b) \quad f_a = f_c = f(c)$
$t = 0.5$
iteration

$x_t = a + t\,(b-a)$
$f_t = f(x_t)$
if sign(f_t) = sign(f_a)\{
$c = a \quad f_c = f_a$
$a = x_t \quad f_a = F_t$\}
else\{
$c = b \quad b = a \quad a = x_t$
$f_c = f_b \quad f_b = f_a \quad f_a = f_t$\}
$x_m = a \quad f_m = f_a$
if abs (f_b) < abs(f_a)\{
$x_m = b \quad f_m = f_b$\}
$tol = 2\epsilon_M|x_m| + \epsilon_a$
$t_l = \frac{tol}{|b-c|}$

if $t_l > 0.5$ or $f_m = 0$ exit
$\xi = \frac{a-b}{c-b} \quad \Phi = \frac{f_a - f_b}{f_c - f_b}$
if $1 - \sqrt{1-\xi} < \Phi < \sqrt{\xi}\{$
$t = \frac{f_a}{f_b - f_a}\frac{f_c}{f_b - f_c} + \frac{c-a}{b-a}\frac{f_a}{f_c - f_a}\frac{f_b}{f_c - f_b}\}$
else $\{t = 0.5\}$
if $t < t_l\{t = t_l\}$
if $t > (1 - t_l)\{t = 1 - t_l\}$

Chandrupatla's method is more efficient than Dekker's and Brent's, especially for higher order roots (Figs. 6.10, 6.11 and 6.12).

6.1.8 Multidimensional Root Finding

The Newton–Raphson method can be easily generalized for functions of more than one variable. We search for the solution of a system of n nonlinear equations in n variables x_i

$$\mathbf{f}(\mathbf{x}) = \begin{pmatrix} f_1(x_1 \cdots x_n) \\ \vdots \\ f_n(x_1 \cdots x_n) \end{pmatrix} = 0. \tag{6.41}$$

The first order Newton–Raphson method results from linearization of

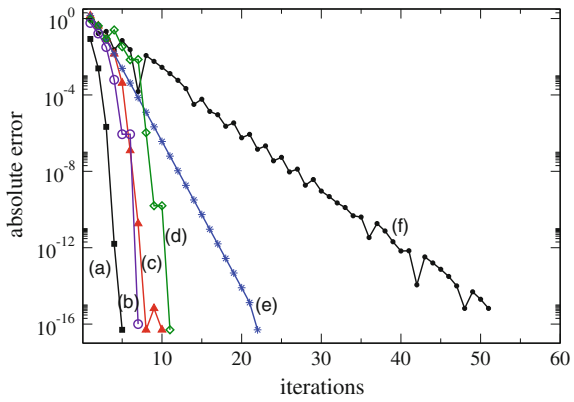


Fig. 6.10 (Comparison of different solvers) The root of the equation $f(x) = x^2 - 2$ is determined with different methods: Newton–Raphson (**a**) (*black squares*), Chandrupatla (**b**) (*indigo circles*), Brent (**c**) (*red triangles up*), Dekker (**d**) (*green diamonds*), regula falsi (**e**) (*blue stars*), pure bisection (**f**) (*black dots*). Starting values are $x_1 = -1$, $x_2 = 2$. The absolute error is shown as function of the number of iterations. For $x_1 = -1$, the Newton–Raphson method converges against $-\sqrt{2}$

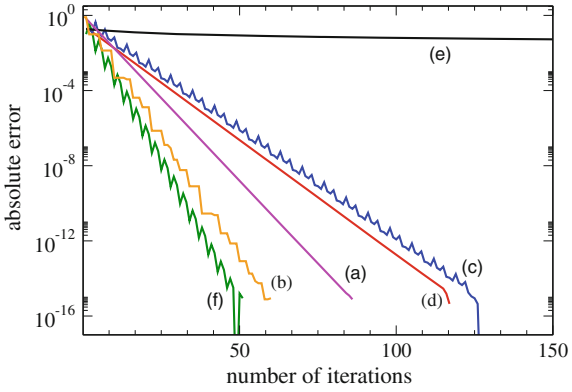


Fig. 6.11 (Comparison of different solvers for a third order root) The root of the equation $f(x) = (x-1)^3$ is determined with different methods: Newton–Raphson (**a**) (*magenta*), Chandrupatla (**b**) (*orange*), Brent (**c**) (*blue*), Dekker (**d**) (*red*), regula falsi (**e**) (*black*), pure bisection (**f**) (*green*). Starting values are $x_1 = 0, x_2 = 1.8$. The absolute error is shown as function of the number of iterations

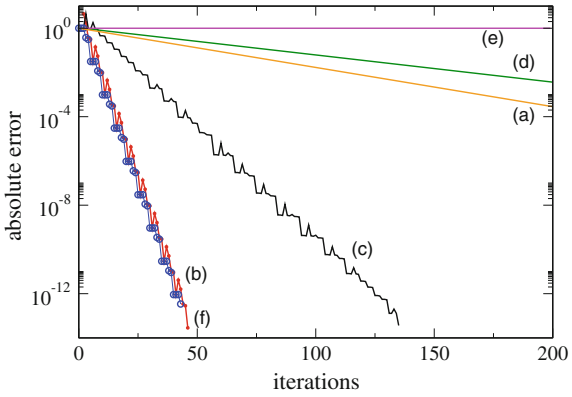


Fig. 6.12 (Comparison of different solvers for a high order root) The root of the equation $f(x) = x^{25}$ is determined with different methods: Newton–Raphson (**a**) (*orange*), Chandrupatla (**b**) (*blue circles*), Brent (**c**) (*black*), Dekker (**d**) (*green*), regula falsi (**e**) (*magenta*), pure bisection (**f**) (*red dots*). Starting values are $x_1 = -1, x_2 = 2$. The absolute error is shown as function of the number of iterations

$$0 = \mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}^0) + J(\mathbf{x}^0)(\mathbf{x} - \mathbf{x}^0) + \cdots \tag{6.42}$$

with the Jacobian matrix

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}. \tag{6.43}$$

If the Jacobian matrix is not singular the equation

$$0 = \mathbf{f}(\mathbf{x}^0) + J(\mathbf{x}^0)(\mathbf{x} - \mathbf{x}^0) \tag{6.44}$$

can be solved by

$$\mathbf{x} = \mathbf{x}^0 - \left(J(\mathbf{x}^0)\right)^{-1} \mathbf{f}(\mathbf{x}^0). \tag{6.45}$$

This can be repeated iteratively

$$\mathbf{x}^{(r+1)} = \mathbf{x}^{(r)} - (J(\mathbf{x}^{(r)}))^{-1}\mathbf{f}(\mathbf{x}^{(r)}). \tag{6.46}$$

6.1.9 Quasi-Newton Methods

Calculation of the Jacobian matrix can be very time consuming. Quasi-Newton methods use instead an approximation to the Jacobian which is updated during each iteration. Defining the differences

$$\mathbf{d}^{(r)} = \mathbf{x}^{(r+1)} - \mathbf{x}^{(r)} \tag{6.47}$$

$$\mathbf{y}^{(r)} = \mathbf{f}(\mathbf{x}^{(r+1)}) - \mathbf{f}(\mathbf{x}^{(r)}) \tag{6.48}$$

we obtain from the truncated Taylor series

$$\mathbf{f}(\mathbf{x}^{(r+1)}) = \mathbf{f}(\mathbf{x}^{(r)}) + J(\mathbf{x}^{(r)})(\mathbf{x}^{(r+1)} - \mathbf{x}^{(r)}) \tag{6.49}$$

the so called Quasi-Newton or secant condition

$$\mathbf{y}^{(r)} = J(\mathbf{x}^{(r)})\mathbf{d}^{(r)}. \tag{6.50}$$

We attempt to construct a family of successive approximation matrices J_r so that, if J were a constant, the procedure would become consistent with the quasi-Newton condition. Then for the new update J_{r+1} we have

$$J_{r+1}\mathbf{d}^{(r)} = \mathbf{y}^{(r)}. \tag{6.51}$$

Since $\mathbf{d}^{(r)}, \mathbf{y}^{(r)}$ are already known, these are only n equations for the n^2 elements of J_{r+1}. To specify J_{r+1} uniquely, additional conditions are required. For instance, it is reasonable to assume, that

$$J_{r+1}\mathbf{u} = J_r\mathbf{u} \quad \text{for all } \mathbf{u} \perp \mathbf{d}^{(r)}. \tag{6.52}$$

Then J_{r+1} differs from J_r only by a rank one updating matrix

$$J_{r+1} = J_r + \mathbf{u}\,\mathbf{d}^{(r)T}. \tag{6.53}$$

From the secant condition we obtain

$$J_{r+1}\mathbf{d}^{(r)} = J_r\mathbf{d}^{(r)} + \mathbf{u}(\mathbf{d}^{(r)}\mathbf{d}^{(r)T}) = \mathbf{y}^{(r)} \tag{6.54}$$

hence

$$\mathbf{u} = \frac{1}{|d^{(r)}|^2}\left(\mathbf{y}^{(r)} - J_r\mathbf{d}^{(r)}\right). \tag{6.55}$$

This gives Broyden's update formula [62]

$$J_{r+1} = J_r + \frac{1}{|d^{(r)}|^2}\left(\mathbf{y}^{(r)} - J_r\mathbf{d}^{(r)}\right)\mathbf{d}^{(r)T}. \tag{6.56}$$

To update the inverse Jacobian matrix, the Sherman–Morrison formula [42]

$$(A + \mathbf{u}\mathbf{v}^T)^{-1} = A^{-1} - \frac{A^{-1}\mathbf{u}\mathbf{v}^T A^{-1}}{1 + \mathbf{v}^T A^{-1}\mathbf{u}} \tag{6.57}$$

can be applied to have

$$\begin{aligned} J_{r+1}^{-1} &= J_r^{-1} - \frac{J_r^{-1}\frac{1}{|d^{(r)}|^2}\left(\mathbf{y}^{(r)} - J_r\mathbf{d}^{(r)}\right)\mathbf{d}^{(r)T}J_r^{-1}}{1 + \frac{1}{|d^{(r)}|^2}\mathbf{d}^{(r)T}J_r^{-1}\left(\mathbf{y}^{(r)} - J_r\mathbf{d}^{(r)}\right)} \\ &= J_r^{-1} - \frac{\left(J_r^{-1}\mathbf{y}^{(r)} - \mathbf{d}^{(r)}\right)\mathbf{d}^{(r)T}J_r^{-1}}{\mathbf{d}^{(r)T}J_r^{-1}\mathbf{y}^{(r)}}. \end{aligned} \tag{6.58}$$

6.2 Function Minimization

Minimization or maximization of a function[2] is a fundamental task in numerical mathematics and closely related to root finding. If the function $f(x)$ is continuously differentiable then at the extremal points the derivative is zero

$$\frac{\mathrm{d}f}{\mathrm{d}x} = 0. \tag{6.59}$$

Hence, in principle root finding methods can be applied to locate local extrema of a function. However, in some cases the derivative cannot be easily calculated or the

[2] In the following we consider only a minimum since a maximum could be found as the minimum of $-f(x)$.

function even is not differentiable. Then derivative free methods similar to bisection for root finding have to be used.

6.2.1 The Ternary Search Method

Ternary search is a simple method to determine the minimum of a unimodal function $f(x)$. Initially we have to find an interval $[a_0, b_0]$ which is certain to contain the minimum. Then the interval is divided into three equal parts $[a_0, c_0]$, $[c_0, d_0]$, $[d_0, b_0]$ and either the first or the last of the three intervals is excluded (Fig. 6.13). The procedure is repeated with the remaining interval $[a_1, b_1] = [a_0, d_0]$ or $[a_1, b_1] = [c_0, b_0]$.

Each step needs two function evaluations and reduces the interval width by a factor of $2/3$ until the maximum possible precision is obtained. It can be determined by considering a differentiable function which can be expanded around the minimum x_0 as

$$f(x) = f(x_0) + \frac{(x - x_0)^2}{2} f''(x_0) + \cdots . \tag{6.60}$$

Numerically calculated function values $f(x)$ and $f(x_0)$ only differ, if

$$\frac{(x - x_0)^2}{2} f''(x_0) > \varepsilon_M f(x_0) \tag{6.61}$$

which limits the possible numerical accuracy to

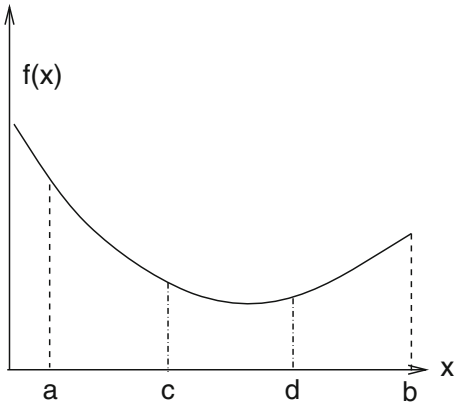


Fig. 6.13 Ternary search method

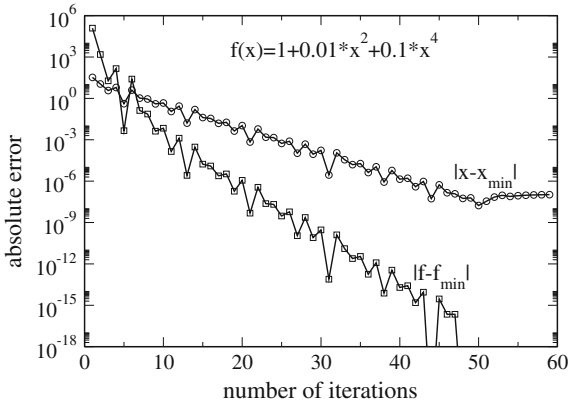


Fig. 6.14 (Ternary search method) The minimum of the function $f(x) = 1 + 0.01\,x^2 + 0.1\,x^4$ is determined with the ternary search method. Each iteration needs two function evaluations. After 50 iterations the function minimum $f_{\min} = 1$ is reached to machine precision $\varepsilon_M \approx 10^{-16}$. The position of the minimum $x_{\min}$ cannot be determined with higher precision than $\sqrt{\varepsilon_M} \approx 10^{-8}$ (6.63)

$$\varepsilon(x_0) = \min|x - x_0| = \sqrt{\frac{2f(x_0)}{f''(x_0)}\varepsilon_M} \tag{6.62}$$

and for reasonably well behaved functions (Fig. 6.14) we have the rule of thumb [63]

$$\varepsilon(x_0) \approx \sqrt{\varepsilon_M}. \tag{6.63}$$

However, it may be impossible to reach even this precision, if the quadratic term of the Taylor series vanishes (Fig. 6.15).

The algorithm can be formulated as follows:

iteration

if$(b-a) < \delta$ then exit
$c = a + \frac{1}{3}(b-a) \quad d = a + \frac{2}{3}(b-a)$
$f_c = f(c) \quad f_d = f(d)$
if $f_c < f_d$ then $b = d$ else $a = c$

6.2.2 *The Golden Section Search Method (Brent's Method)*

To bracket a local minimum of a unimodal function $f(x)$ three points a, b, c are necessary (Fig. 6.16) with

$$f(a) > f(b) \quad f(c) > f(b). \tag{6.64}$$

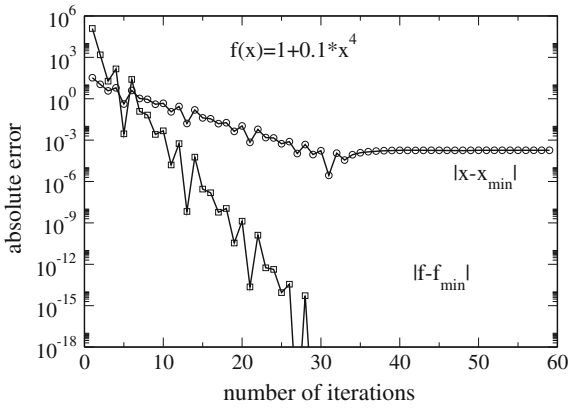


Fig. 6.15 (Ternary search method for a higher order minimum) The minimum of the function $f(x) = 1 + 0.1\,x^4$ is determined with the ternary search method. Each iteration needs two function evaluations. After 30 iterations the function minimum $f_{\min} = 1$ is reached to machine precision $\varepsilon_M \approx 10^{-16}$. The position of the minimum $x_{\min}$ cannot be determined with higher precision than $\sqrt[4]{\varepsilon_M} \approx 10^{-4}$

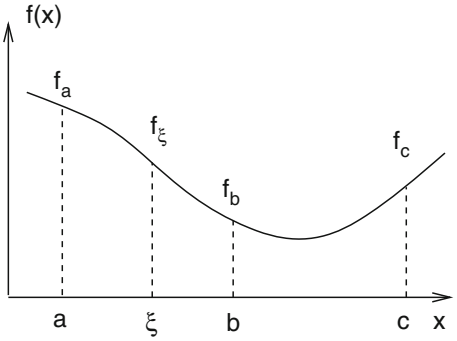


Fig. 6.16 (Golden section search method) A local minimum of the function $f(x)$ is bracketed by three points a, b, c. To reduce the uncertainty of the minimum position a new point ξ is chosen in the interval $a < \xi < c$ and either a or c is dropped according to the relation of the function values. For the example shown a has to be replaced by ξ

The position of the minimum can be determined iteratively by choosing a new value ξ in the interval $a < \xi < c$ and dropping either a or c, depending on the ratio of the function values. A reasonable choice for ξ can be found as follows (Fig. 6.17) [63, 64]. Let us denote the relative positions of the middle point and the trial point as

$$\frac{b-a}{c-a} = \beta \quad \frac{c-b}{c-a} = 1-\beta \quad \frac{b-a}{c-b} = \frac{\beta}{1-\beta} \quad \frac{\xi-b}{c-a} = t.$$
$$\frac{\xi-a}{c-a} = \frac{\xi-b+b-a}{c-a} = t+\beta. \tag{6.65}$$

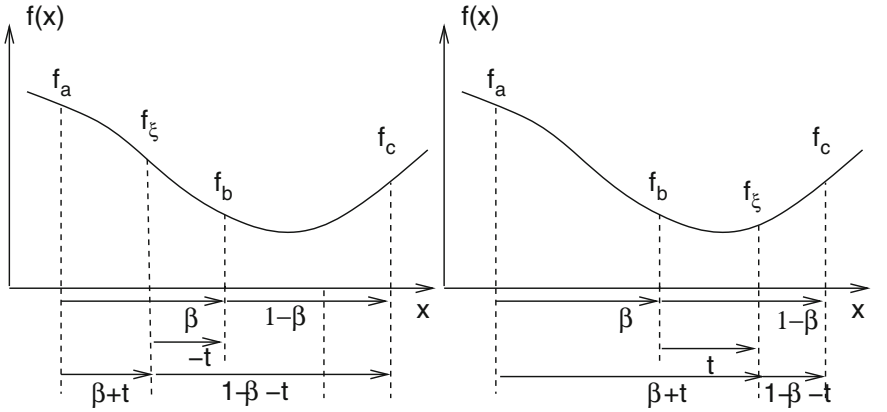


Fig. 6.17 Golden section search method

The relative width of the new interval will be

$$\frac{c-\xi}{c-a} = (1-\beta-t) \quad \text{or} \quad \frac{b-a}{c-a} = \beta \quad \text{if } a < \xi < b \tag{6.66}$$

$$\frac{\xi-a}{c-a} = (t+\beta) \quad \text{or} \quad \frac{c-b}{c-a} = (1-\beta) \quad \text{if } b < \xi < c. \tag{6.67}$$

The golden search method requires that

$$t = 1-2\beta = \frac{c+a-2b}{c-a} = \frac{(c-b)-(b-a)}{c-a}. \tag{6.68}$$

Otherwise it would be possible that the larger interval width is selected many times slowing down the convergence. The value of t is positive if $c-b > b-a$ and negative if $c-b < b-a$, hence the trial point always is in the larger of the two intervals. In addition the golden search method requires that the ratio of the spacing remains constant. Therefore we set

$$\frac{\beta}{1-\beta} = -\frac{t+\beta}{t} = -\frac{1-\beta}{t} \quad \text{if } a < \xi < b \tag{6.69}$$

$$\frac{\beta}{1-\beta} = \frac{t}{1-\beta-t} = \frac{t}{\beta} \quad \text{if } b < \xi < c. \tag{6.70}$$

Eliminating t we obtain for $a < \xi < b$ the equation

$$\frac{(\beta-1)}{\beta}(\beta^2+\beta-1) = 0. \tag{6.71}$$

Besides the trivial solution $\beta = 1$ there is only one positive solution

$$\beta = \frac{\sqrt{5}-1}{2} \approx 0.618. \tag{6.72}$$

For $b < \xi < c$ we end up with

$$\frac{\beta}{\beta - 1}(\beta^2 - 3\beta + 1) = 0 \tag{6.73}$$

which has the nontrivial positive solution

$$\beta = \frac{3-\sqrt{5}}{2} \approx 0.382. \tag{6.74}$$

Hence the lengths of the two intervals $[a, b], [b, c]$ have to be in the golden ratio $\varphi = \frac{1+\sqrt{5}}{2} \approx 1.618$ which gives the method its name. Using the golden ratio the width of the interval bracketing the minimum reduces by a factor of 0.618 (Figs. 6.18 and 6.19).

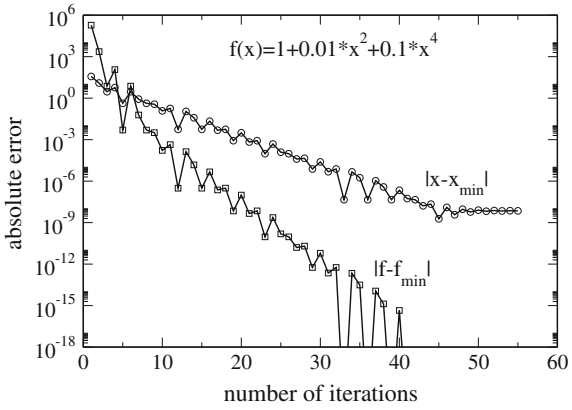


Fig. 6.18 (Golden section search method) The minimum of the function $f(x) = 1 + 0.01\,x^2 + 0.1\,x^4$ is determined with the golden section search method. Each iteration needs only one function evaluation. After 40 iterations the function minimum $f_{\min} = 1$ is reached to machine precision $\varepsilon_M \approx 10^{-16}$. The position of the minimum $x_{\min}$ cannot be determined to higher precision than $\sqrt{\varepsilon_M} \approx 10^{-8}$ (6.63)

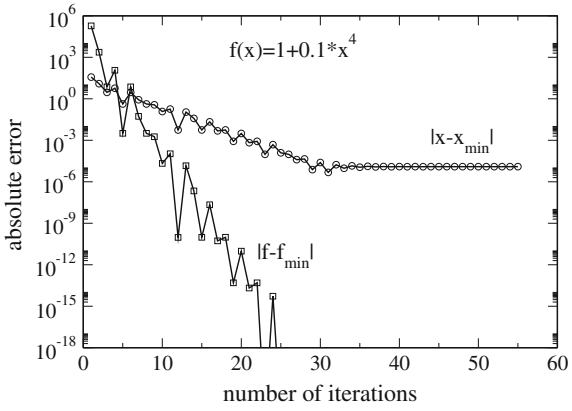


Fig. 6.19 (Golden section search for a higher order minimum) The minimum of the function $f(x) = 1 + 0.1\,x^4$ is determined with the golden section search method. Each iteration needs only one function evaluation. After 28 iterations the function minimum $f_{\min} = 1$ is reached to machine precision $\varepsilon_M \approx 10^{-16}$. The position of the minimum $x_{\min}$ cannot be determined to higher precision than $\sqrt[4]{\varepsilon_M} \approx 10^{-4}$

The algorithm can be formulated as follows:

if $c - a < \delta$ then exit
if $(b - a) \geq (c - b)$ then {
$x = 0.618\,b + 0.382\,a$
$f_x = f(x)$
if $f_x < f_b$ then $\{c = b \quad b = x \quad f_c = f_b \quad f_b = f_x\}$
else $a = x \quad f_a = f_x\}$
if $(b - a) < (c - b)$ then {
$x = 0.618\,b + 0.382\,c$
$f_x = f(x)$
if $f_x < f_b$ then $\{a = b \quad b = x \quad f_a = f_b \quad f_b = f_x\}$
else $c = x \quad f_c = f_x\}$

To start the method we need three initial points which can be found by Brent's exponential search method (Fig. 6.20). Begin with three points

$$a_0, b_0 = a_0 + h, c_0 + 1.618\,h \tag{6.75}$$

where h_0 is a suitable initial step width which depends on the problem. If the minimum is not already bracketed then if necessary exchange a_0 and b_0 to have

$$f(a_0) > f(b_0) > f(c_0). \tag{6.76}$$

Then replace the three points by

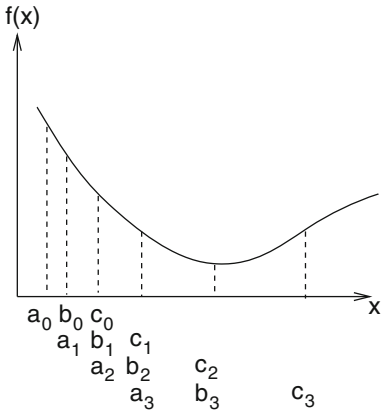


Fig. 6.20 Brent's exponential search

$$a_1 = b_0 \quad b_1 = c_0 \quad c_1 = c_0 + 1.618\,(c_0 - b_0) \tag{6.77}$$

and repeat this step until

$$f(b_n) < f(c_n) \tag{6.78}$$

or n exceeds a given maximum number. In this case no minimum can be found and we should check if the initial step width was too large.

Brent's method can be improved by making use of derivatives and by combining the golden section search with parabolic interpolation [63].

6.2.3 Minimization in Multidimensions

We search for local minima (or maxima) of a function

$$h(\mathbf{x})$$

which is at least two times differentiable. In the following we denote the gradient vector by

$$\mathbf{g}^T(\mathbf{x}) = \left(\frac{\partial h}{\partial x_1}, \cdots \frac{\partial h}{\partial x_n}\right) \tag{6.79}$$

and the matrix of second derivatives (Hessian) by

$$H = \left(\frac{\partial^2}{\partial x_i \partial x_j} h\right). \tag{6.80}$$

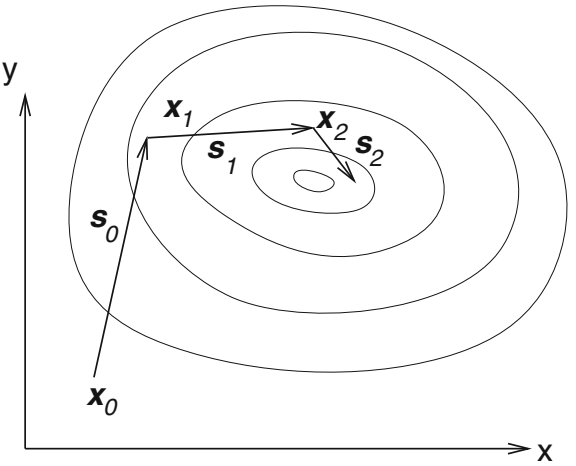


Fig. 6.21 (Direction set minimization) Starting from an initial guess $\mathbf{x}_0$ a local minimum is approached by making steps along a set of direction vectors $\mathbf{s}_r$

The very popular class of direction set methods proceeds as follows (Fig. 6.21). Starting from an initial guess $\mathbf{x}_0$ a set of direction vectors $\mathbf{s}_r$ and step lengths λ_r is determined such that the series of vectors

$$\mathbf{x}_{r+1} = \mathbf{x}_r + \lambda_r \mathbf{s}_r \tag{6.81}$$

approaches the minimum of $h(\mathbf{x})$. The method stops if the norm of the gradient becomes sufficiently small or if no lower function value can be found.

6.2.4 Steepest Descent Method

The simplest choice, which is known as the method of gradient descent or steepest descent[3] is to go in the direction of the negative gradient

$$\mathbf{s}_r = -\mathbf{g}_r \tag{6.82}$$

and to determine the step length by minimizing h along this direction

$$h(\lambda) = h(\mathbf{x}_r - \lambda \mathbf{g}_r) = \min. \tag{6.83}$$

Obviously two consecutive steps are orthogonal to each other since

$$0 = \frac{\partial}{\partial \lambda} h(\mathbf{x}_{r+1} - \lambda \mathbf{g}_r)_{|\lambda=0} = -\mathbf{g}_{r+1}^T \mathbf{g}_r. \tag{6.84}$$

[3] Which should not be confused with the method of steepest descent for the approximate calculation of integrals.

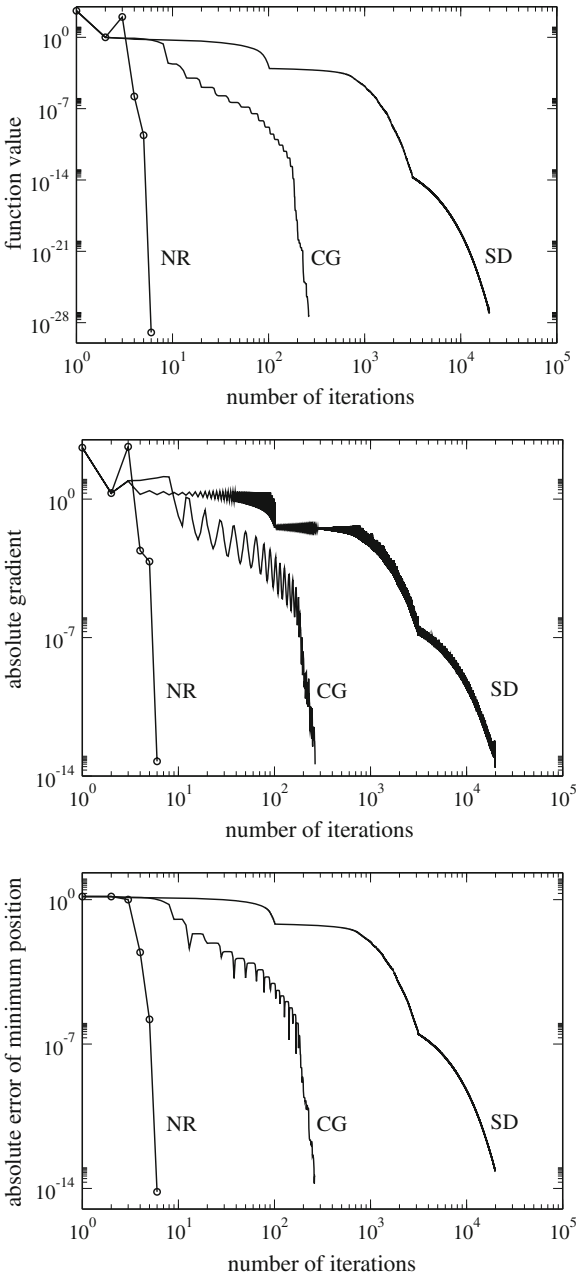


Fig. 6.22 (Function minimization) The minimum of the Rosenbrock function $h(x, y) = 100(y - x^2)^2 + (1 - x)^2$ is determined with different methods. Conjugate (CG) gradients converge much faster than steepest descent (SD). Starting at $(x, y) = (0, 2)$, Newton–Raphson (NR) reaches the minimum at $x = y = 1$ within only 5 iterations to machine precision

This can lead to a zig-zagging behavior and a very slow convergence of this method (Figs. 6.22 and 6.23).

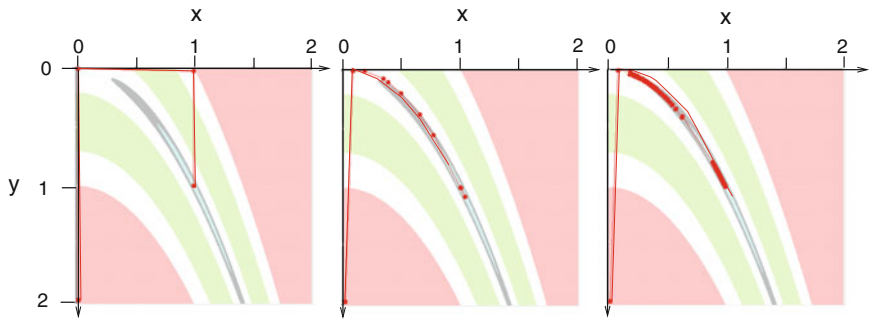


Fig. 6.23 (Minimization of the Rosenbrock function) ***Left*** Newton–Raphson finds the minimum after 5 steps within machine precision. ***Middle*** conjugate gradients reduce the gradient to 4×10^{-14} after 265 steps. ***Right*** The steepest descent method needs 20000 steps to reduce the gradient to 5×10^{-14}. *Red lines* show the minimization pathway. *Colored areas* indicate the function value (*light blue* < 0.1, *grey* $0.1 \ldots 0.5$, *green* $5 \ldots 50$, *pink* > 100). Screen shots taken from problem 6.2

6.2.5 Conjugate Gradient Method

This method is similar to the steepest descent method but the search direction is iterated according to

$$\mathbf{s}_0 = -\mathbf{g}_0 \tag{6.85}$$

$$\mathbf{x}_{r+1} = \mathbf{x}_r + \lambda_r \mathbf{s}_r \tag{6.86}$$

$$\mathbf{s}_{r+1} = -\mathbf{g}_{r+1} + \beta_{r+1} \mathbf{s}_r \tag{6.87}$$

where λ_r is chosen to minimize $h(\mathbf{x}_{r+1})$ and the simplest choice for β is made by Fletcher and Rieves [65]

$$\beta_{r+1} = \frac{g_{r+1}^2}{g_r^2}. \tag{6.88}$$

This method was devised to minimize a quadratic function and to solve the related system of linear equations, but it is also very efficient for more complicated functions (Sect. 5.6.4).

6.2.6 Newton–Raphson Method

The first order Newton–Raphson method uses the iteration

$$\mathbf{x}_{r+1} = \mathbf{x}_r - H(\mathbf{x}_r)^{-1} \mathbf{g}(\mathbf{x}_r). \tag{6.89}$$

The search direction is

$$\mathbf{s} = H^{-1}\mathbf{g} \tag{6.90}$$

and the step length is $\lambda = 1$. This method converges fast if the starting point is close to the minimum. However, calculation of the Hessian can be very time consuming (Fig. 6.22).

6.2.7 *Quasi-Newton Methods*

Calculation of the full Hessian matrix as needed for the Newton–Raphson method can be very time consuming. Quasi-Newton methods use instead an approximation to the Hessian which is updated during each iteration. From the Taylor series

$$h(\mathbf{x}) = h_0 + \mathbf{b}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T H\mathbf{x} + \cdots \tag{6.91}$$

we obtain the gradient

$$\mathbf{g}(\mathbf{x}_r) = \mathbf{b} + H\mathbf{x}_r + \cdots = \mathbf{g}(\mathbf{x}_{r-1}) + H(\mathbf{x}_r - \mathbf{x}_{r-1}) + \cdots . \tag{6.92}$$

Defining the differences

$$\mathbf{d}_r = \mathbf{x}_{r+1} - \mathbf{x}_r \tag{6.93}$$

$$\mathbf{y}_r = \mathbf{g}_{r+1} - \mathbf{g}_r \tag{6.94}$$

and neglecting higher order terms we obtain the quasi-Newton or secant condition

$$H\mathbf{d}_r = \mathbf{y}_r. \tag{6.95}$$

We want to approximate the true Hessian by a series of matrices H_r which are updated during each iteration to sum up all the information gathered so far. Since the Hessian is symmetric and positive definite, this also has to be demanded for the H_r.[4] This cannot be achieved by a rank one update matrix. Popular methods use a symmetric rank two update of the form

$$H_{r+1} = H_r + \alpha\mathbf{u}\mathbf{u}^T + \beta\mathbf{v}\mathbf{v}^T. \tag{6.96}$$

The Quasi-Newton condition then gives

$$H_{r+1}\mathbf{d}_r = H_r\mathbf{d}_r + \alpha(\mathbf{u}^T\mathbf{d}_r)\mathbf{u} + \beta(\mathbf{v}^T\mathbf{d}_r)\mathbf{v} = \mathbf{y}_r \tag{6.97}$$

[4]This is a major difference to the Quasi Newton methods for root finding (6.1.9).

hence $H_r\mathbf{d}_r - \mathbf{y}_r$ must be a linear combination of $\mathbf{u}$ and $\mathbf{v}$. Making the simple choice

$$\mathbf{u} = \mathbf{y}_r \quad \mathbf{v} = H_r\mathbf{d}_r \tag{6.98}$$

and assuming that these two vectors are linearly independent, we find

$$\beta = -\frac{1}{(\mathbf{v}^T\mathbf{d}_r)} \quad = -\frac{1}{(\mathbf{d}_r^T H_r\mathbf{d}_r)} \tag{6.99}$$

$$\alpha = \frac{1}{(\mathbf{u}^T\mathbf{d}_r)} = \frac{1}{(\mathbf{y}_r^T\mathbf{d}_r)} \tag{6.100}$$

which together defines the very popular BFGS (Broyden, Fletcher, Goldfarb, Shanno) method [66–69]

$$H_{r+1} = H_r + \frac{\mathbf{y_r}\mathbf{y}_r^T}{\mathbf{y}_r^T\mathbf{d}_r} - \frac{(H_r\mathbf{d}_r)(H_r\mathbf{d}_r)^T}{\mathbf{d}_r^T H_r\mathbf{d}_r}. \tag{6.101}$$

Alternatively the DFP method by Davidon, Fletcher and Powell, directly updates the inverse Hessian matrix $B = H^{-1}$ according to

$$B_{r+1} = B_r + \frac{\mathbf{d}_r\mathbf{d}_r^T}{\mathbf{y}_r^T\mathbf{d}_r} - \frac{(B_r\mathbf{y}_r)(B_r\mathbf{y}_r)^T}{\mathbf{y}_r^T B_r\mathbf{y}_r}. \tag{6.102}$$

Both of these methods can be inverted with the help of the Sherman–Morrison formula to give

$$B_{r+1} = B_r + \frac{(\mathbf{d}_r - B_r\mathbf{y}_r)\mathbf{d}_r^T + \mathbf{d}_r(\mathbf{d}_r - B\mathbf{y}_r)^T}{\mathbf{y}_r^T\mathbf{d}_r} - \frac{(\mathbf{d}_r - B_r\mathbf{y}_r)^T\mathbf{y}}{(\mathbf{y}_r^T\mathbf{d})^2}\mathbf{d}\mathbf{d}^T \tag{6.103}$$

$$H_{r+1} = H_r + \frac{(\mathbf{y}_r - H_r\mathbf{d}_r)\mathbf{y}_r^T + \mathbf{y}_r(\mathbf{y}_r - H_r\mathbf{d}_r)^T}{\mathbf{y}_r^T\mathbf{d}_r} - \frac{(\mathbf{y}_r - H_r\mathbf{d}_r)\mathbf{d}_r}{(\mathbf{y}_r^T\mathbf{d}_r)^2}\mathbf{y}_r\mathbf{y}_r^T. \tag{6.104}$$

Problems

Problem 6.1 Root Finding Methods

This computer experiment searches roots of several test functions:

$f(x) = x^n - 2 \quad n = 1, 2, 3, 4$ (Fig. 6.10)
$f(x) = 5\sin(5x)$
$f(x) = (\cos(2x))^2 - x^2$
$f(x) = 5\left(\sqrt{|x+2|} - 1\right)$
$f(x) = \mathrm{e}^{-x}\ln x$

$f(x) = (x-1)^3$ (Fig. 6.11)
$f(x) = x^{25}$ (Fig. 6.12)

You can vary the initial interval or starting value and compare the behavior of different methods:

- bisection
- regula falsi
- Dekker's method
- Brent's method
- Chandrupatla's method
- Newton–Raphson method

Problem 6.2 Stationary Points

This computer experiment searches a local minimum of the Rosenbrock function[5]

$$h(x, y) = 100(y - x^2)^2 + (1 - x)^2. \tag{6.105}$$

- The method of steepest descent minimizes $h(x, y)$ along the search direction

$$s_x^{(n)} = -g_x^{(n)} = -400x(x_n^2 - y_n) - 2(x_n - 1) \tag{6.106}$$
$$s_y^{(n)} = -g_y^{(n)} = -200(y_n - x_n^2). \tag{6.107}$$

- Conjugate gradients make use of the search direction

$$s_x^{(n)} = -g_x^{(n)} + \beta_n s_x^{(n-1)} \tag{6.108}$$
$$s_y^{(n)} = -g_y^{(n)} + \beta_n s_y^{(n-1)}. \tag{6.109}$$

- The Newton–Raphson method needs the inverse Hessian

$$H^{-1} = \frac{1}{\det(H)} \begin{pmatrix} h_{yy} & -h_{xy} \\ -h_{xy} & h_{xx} \end{pmatrix} \tag{6.110}$$

$$\det(H) = h_{xx}h_{yy} - h_{xy}^2 \tag{6.111}$$
$$h_{xx} = 1200x^2 - 400y + 2 \quad h_{yy} = 200 \quad h_{xy} = -400x \tag{6.112}$$

and iterates according to

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} - H^{-1} \begin{pmatrix} g_x^n \\ q_y^n \end{pmatrix}. \tag{6.113}$$

You can choose an initial point (x_0, y_0). The iteration stops if the gradient norm falls below 10^{-14} or if the line search fails to find a lower function value.

[5] A well known test function for minimization algorithms.

Chapter 7
Fourier Transformation

Fourier transformation is a very important tool for signal analysis but also helpful to simplify the solution of differential equations or the calculation of convolution integrals. In this chapter we discuss the discrete Fourier transformation as a numerical approximation to the continuous Fourier integral. It can be realized efficiently by Goertzel's algorithm or the family of fast Fourier transformation methods. Computer experiments demonstrate trigonometric interpolation and nonlinear filtering as applications.

7.1 Fourier Integral and Fourier Series

We use the symmetric definition of the Fourier transformation:

$$\widetilde{f}(\omega) = \mathcal{F}[f](\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) \mathrm{e}^{-\mathrm{i}\omega t} dt. \tag{7.1}$$

The inverse Fourier transformation

$$f(t) = \mathcal{F}^{-1}[f](t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(\omega) \mathrm{e}^{\mathrm{i}\omega t} d\omega \tag{7.2}$$

decomposes $f(t)$ into a superposition of oscillations. The Fourier transform of a convolution integral

$$g(t) = f(t) \otimes h(t) = \int_{-\infty}^{\infty} f(t')h(t-t')dt' \tag{7.3}$$

becomes a product of Fourier transforms:

$$\widetilde{g}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dt' f(t') \mathrm{e}^{-\mathrm{i}\omega t'} \int_{-\infty}^{\infty} h(t-t') \mathrm{e}^{-\mathrm{i}\omega(t-t')} d(t-t')$$

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_7

$$= \sqrt{2\pi}\,\widetilde{f}(\omega)\widetilde{h}(\omega). \tag{7.4}$$

A periodic function with $f(t+T) = f(t)$[1] is transformed into a Fourier series

$$f(t) = \sum_{n=-\infty}^{\infty} \mathrm{e}^{\mathrm{i}\omega_n t}\,\hat{f}(\omega_n) \;\text{ with}\, \omega_n = n\frac{2\pi}{T}, \quad \hat{f}(\omega_n) = \frac{1}{T}\int_0^T f(t)\mathrm{e}^{-\mathrm{i}\omega_n t}dt. \tag{7.5}$$

For a periodic function which in addition is real valued $f(t) = f(t)^*$ and even $f(t) = f(-t)$, the Fourier series becomes a cosine series

$$f(t) = \hat{f}(\omega_0) + 2\sum_{n=1}^{\infty} \hat{f}(\omega_n)\cos\omega_n t \tag{7.6}$$

with real valued coefficients

$$\hat{f}(\omega_n) = \frac{1}{T}\int_0^T f(t)\cos\omega_n t\,dt. \tag{7.7}$$

7.2 Discrete Fourier Transformation

We divide the time interval $0 \le t < T$ by introducing a grid of N equidistant points

$$t_n = n\Delta t = n\frac{T}{N} \quad \text{with} \quad n = 0, 1, \ldots N-1. \tag{7.8}$$

The function values (samples)

$$f_n = f(t_n) \tag{7.9}$$

are arranged as components of a vector

$$\mathbf{f} = \begin{pmatrix} f_0 \\ \vdots \\ f_{N-1} \end{pmatrix}.$$

With respect to the orthonormal basis

$$\mathbf{e}_n = \begin{pmatrix} \delta_{0,n} \\ \vdots \\ \delta_{N-1,n} \end{pmatrix}, \quad n = 0, 1, \ldots N-1 \tag{7.10}$$

[1] This could also be the periodic continuation of a function which is only defined for $0 < t < T$.

f is expressed as a linear combination

$$\mathbf{f} = \sum_{n=0}^{N-1} f_n \mathbf{e}_n. \tag{7.11}$$

The discrete Fourier transformation is the transformation to an orthogonal base in frequency space

$$\mathbf{e}_{\omega_j} = \sum_{n=0}^{N-1} \mathrm{e}^{\mathrm{i}\omega_j t_n} \mathbf{e}_n = \begin{pmatrix} 1 \\ \mathrm{e}^{\mathrm{i}\frac{2\pi}{N}j} \\ \vdots \\ \mathrm{e}^{\mathrm{i}\frac{2\pi}{N}j(N-1)} \end{pmatrix} \tag{7.12}$$

with

$$\omega_j = \frac{2\pi}{T} j. \tag{7.13}$$

These vectors are orthogonal

$$\mathbf{e}_{\omega_j} \mathbf{e}^*_{\omega_{j'}} = \sum_{n=0}^{N-1} \mathrm{e}^{\mathrm{i}(j-j')\frac{2\pi}{N}n} = \begin{array}{ll} \frac{1-\mathrm{e}^{\mathrm{i}(j-j')2\pi}}{1-\mathrm{e}^{\mathrm{i}(j-j')2\pi/N}} = 0 & for \quad j - j' \neq 0 \\ N \quad for & j - j' = 0 \end{array} \tag{7.14}$$

$$\mathbf{e}_{\omega_j} \mathbf{e}^*_{\omega_{j'}} = N \delta_{j,j'}. \tag{7.15}$$

Alternatively a real valued basis can be defined:

$$\begin{aligned} &\cos \quad \frac{2\pi}{N} jn \qquad\qquad j = 0, 1, \ldots j_{\max} \\ &\sin \quad \frac{2\pi}{N} jn \qquad\qquad j = 1, 2 \ldots j_{\max} \\ &\qquad\qquad j_{\max} = \tfrac{N}{2} (\text{even N}) \quad j_{\max} = \tfrac{N-1}{2} (\text{odd N}). \end{aligned} \tag{7.16}$$

The components of **f** in frequency space are given by the scalar product

$$\widetilde{f}_{\omega_j} = \mathbf{f}\mathbf{e}_{\omega_j} = \sum_{n=0}^{N-1} f_n \mathrm{e}^{-\mathrm{i}\omega_j t_n} = \sum_{n=0}^{N-1} f_n \mathrm{e}^{-\mathrm{i}j\frac{2\pi}{T}n\frac{T}{N}} = \sum_{n=0}^{N-1} f_n \mathrm{e}^{-\mathrm{i}\frac{2\pi}{N}jn}. \tag{7.17}$$

From

$$\sum_{j=0}^{N-1} f_{\omega_j} \mathrm{e}^{\mathrm{i}\omega_j t_n} = \sum_{n'} \sum_{\omega_j} f_{n'} \mathrm{e}^{-\mathrm{i}\omega_j t_{n'}} \mathrm{e}^{\mathrm{i}\omega_j t_n} = N f_n \tag{7.18}$$

we find the inverse transformation

$$f_n = \frac{1}{N}\sum_{j=0}^{N-1} \widetilde{f}_{\omega_j} \mathrm{e}^{\mathrm{i}\omega_j t_n} = \frac{1}{N}\sum_{j=0}^{N-1} \widetilde{f}_{\omega_j} \mathrm{e}^{\mathrm{i}\frac{2\pi}{N}nj}. \tag{7.19}$$

7.2.1 Trigonometric Interpolation

The last equation can be interpreted as an interpolation of the function $f(t)$ at the sampling points t_n by a linear combination of trigonometric functions

$$f(t) = \frac{1}{N}\sum_{j=0}^{N-1} \widetilde{f}_{\omega_j} \left(\mathrm{e}^{\mathrm{i}\frac{2\pi}{T}t}\right)^j \tag{7.20}$$

which is a polynomial of

$$q = \mathrm{e}^{\mathrm{i}\frac{2\pi}{T}t}. \tag{7.21}$$

Since

$$\mathrm{e}^{-\mathrm{i}\omega_j t_n} = \mathrm{e}^{-\mathrm{i}\frac{2\pi}{N}jn} = \mathrm{e}^{\mathrm{i}\frac{2\pi}{N}(N-j)n} = \mathrm{e}^{\mathrm{i}\omega_{N-j}t_n} \tag{7.22}$$

the frequencies ω_j and ω_{N-j} are equivalent (Fig. 7.1)

$$\widetilde{f}_{\omega_{N-j}} = \sum_{n=0}^{N-1} f_n \mathrm{e}^{-\mathrm{i}\frac{2\pi}{N}(N-j)n} = \sum_{n=0}^{N-1} f_n \mathrm{e}^{\mathrm{i}\frac{2\pi}{N}jn} = \widetilde{f}_{\omega_{-j}}. \tag{7.23}$$

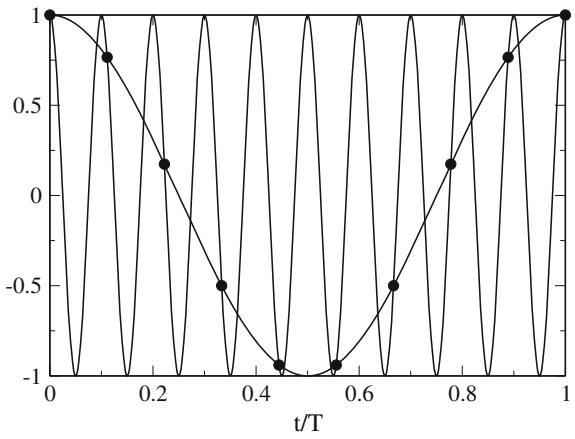


Fig. 7.1 (Equivalence of ω_1 and ω_{N-1}) The two functions $\cos\omega t$ and $\cos(N-1)\omega t$ have the same values at the sample points t_n but are very different in between

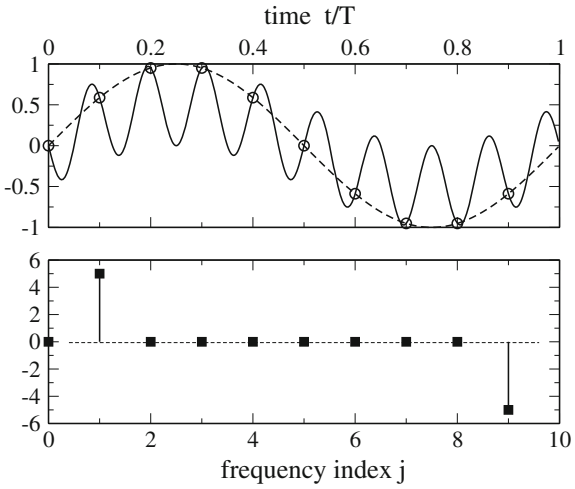


Fig. 7.2 (Trigonometric interpolation) For trigonometric interpolation the high frequencies have to be replaced by the corresponding negative frequencies to provide meaningful results between the sampling points. The *circles* show sampling points which are fitted using only positive frequencies (*full curve*) or replacing the unphysical high frequency by its negative counterpart (*broken curve*). The *squares* show the calculated Fourier spectrum. See also Problem 7.1

If we use trigonometric interpolation to approximate $f(t)$ between the grid points, the two frequencies are no longer equivalent and we have to restrict the frequency range to avoid unphysical high frequency components (Fig. 7.2):

$$-\frac{2\pi}{T}\frac{N-1}{2} \le \omega_j \le \frac{2\pi}{T}\frac{N-1}{2} \qquad N\ odd$$
$$-\frac{2\pi}{T}\frac{N}{2} \le \omega_j \le \frac{2\pi}{T}\ \ \frac{N}{2}-1 \qquad N\ even. \tag{7.24}$$

The interpolating function (N even) is

$$f(t) = \frac{1}{N}\sum_{j=-\frac{N}{2}}^{\frac{N}{2}-1} f_{\omega_j} \mathrm{e}^{\mathrm{i}\omega_j t} \quad \text{even } N \tag{7.25}$$

$$f(t) = \frac{1}{N}\sum_{j=-\frac{N-1}{2}}^{\frac{N-1}{2}} f_{\omega_j} \mathrm{e}^{\mathrm{i}\omega_j t} \quad \text{odd } N. \tag{7.26}$$

The maximum frequency is

$$\omega_{\max} = \frac{2\pi}{T}\frac{N}{2} \tag{7.27}$$

and hence

$$f_{\max} = \frac{1}{2\pi}\omega_{\max} = \frac{N}{2T} = \frac{f_s}{2}. \tag{7.28}$$

This is known as the sampling theorem which states that the sampling frequency f_s must be larger than twice the maximum frequency present in the signal.

7.2.2 Real Valued Functions

For a real valued function

$$f_n = f_n^* \tag{7.29}$$

and hence

$$\widetilde{f}^*_{\omega_j} = \left(\sum_{n=0}^{N-1} f_n \mathrm{e}^{-\mathrm{i}\omega_j t_n}\right)^* = \sum_{n=0}^{N-1} f_n \mathrm{e}^{\mathrm{i}\omega_j t_n} = \widetilde{f}_{\omega_{-j}}. \tag{7.30}$$

Here it is sufficient to calculate the sums for $j = 0, \ldots N/2$. If the function is real valued and also even

$$f_{-n} = f_n \tag{7.31}$$

$$\widetilde{f}_{\omega_j} = \sum_{n=0}^{N-1} f_{-n} \mathrm{e}^{-\mathrm{i}\omega_j t_n} = \sum_{n=0}^{N-1} f_n \mathrm{e}^{-\mathrm{i}(-\omega_j) t_n} = \widetilde{f}_{\omega_{-j}} \tag{7.32}$$

and the Fourier sum (7.19) turns into a cosine sum

$$f_n = \frac{1}{2M-1}\tilde{f}_{\omega_0} + \frac{2}{2M-1}\sum_{j=1}^{M-1} \widetilde{f}_{\omega_j} \cos\left(\frac{2\pi}{2M-1} jn\right) \quad \text{odd } N = 2M-1 \tag{7.33}$$

$$f_n = \frac{1}{2M}\tilde{f}_{\omega_0} + \frac{1}{M}\sum_{j=1}^{M-1} \widetilde{f}_{\omega_j} \cos\left(\frac{\pi}{M} jn\right) + \frac{1}{2M}\tilde{f}_{\omega_M} \cos(n\pi) \quad \text{even } N = 2M \tag{7.34}$$

which correspond to two out of eight different versions [70] of the discrete cosine transformation [71, 72].

Equation 7.34 can be used to define the interpolating function

$$f(t) = \frac{1}{2M}\tilde{f}_{\omega_0} + \frac{1}{M}\sum_{j=1}^{M-1} f_{\omega_j}\cos\left(\omega_j t\right) + \frac{1}{2M}\tilde{f}_{\omega_M}\cos\left(\frac{2\pi M}{T}t\right). \tag{7.35}$$

The real valued Fourier coefficients are given by

$$\widetilde{f}_{\omega_j} = f_0 + 2\sum_{n=1}^{M-1} f_n\cos(\omega_j t_n) \quad \text{odd } N = 2M-1 \tag{7.36}$$

$$f_{\omega_j} = f_0 + 2\sum_{n=1}^{M-1} f_n\cos(\omega_j t_n) + f_M\cos(j\pi) \quad \text{even } N = 2M. \tag{7.37}$$

7.2.3 Approximate Continuous Fourier Transformation

We continue the function $f(t)$ periodically by setting

$$f_N = f_0 \tag{7.38}$$

and write

$$\widetilde{f}_{\omega_j} = \sum_{n=0}^{N-1} f_n \mathrm{e}^{-\mathrm{i}\omega_j n} = \frac{1}{2}f_0 + \mathrm{e}^{-\mathrm{i}\omega_j} f_1 + \cdots \mathrm{e}^{-\mathrm{i}\omega_j(N-1)} f_{N-1} + \frac{1}{2}f_N. \tag{7.39}$$

Comparing with the trapezoidal rule (4.13) for the integral

$$\begin{aligned}\int_0^T \mathrm{e}^{-\mathrm{i}\omega_j t} f(t)dt \approx \frac{T}{N}\Big[&\frac{1}{2}\mathrm{e}^{-\mathrm{i}\omega_j 0} f(0) + \mathrm{e}^{-\mathrm{i}\omega_j \frac{T}{N}} f\left(\frac{T}{N}\right) \\ &+ \cdots + \mathrm{e}^{-\mathrm{i}\omega_j \frac{T}{N}(N-1)} f\left(\frac{T}{N}(N-1)\right) + \tfrac{1}{2}f(T)\Big]\end{aligned} \tag{7.40}$$

we find

$$\hat{f}(\omega_j) = \frac{1}{T}\int_0^T \mathrm{e}^{-\mathrm{i}\omega_j t} f(t)dt \approx \frac{1}{N}\widetilde{f}_{\omega_j} \tag{7.41}$$

which shows that the discrete Fourier transformation is an approximation to the Fourier series of a periodic function with period T which coincides with $f(t)$ in the interval $0 < t < T$. The range of the integral can be formally extended to $\pm\infty$ by introducing a windowing function

$$W(t) = \begin{cases} 1 & for \quad 0 < t < T \\ 0 & else \end{cases} . \tag{7.42}$$

The discrete Fourier transformation approximates the continuous Fourier transformation but windowing leads to a broadening of the spectrum (see p. 145). For practical purposes smoother windowing functions are used like a triangular window or one of the following [73]:

$$W(t_n) = \mathrm{e}^{-\frac{1}{2}\left(\frac{n-(N-1)/2}{\sigma(N-1)/2}\right)^2} \quad \sigma \leq 0.5 \quad \text{Gaussian window}$$
$$W(t_n) = 0.53836 - 0.46164 \cos\left(\frac{2\pi n}{N-1}\right) \quad \text{Hamming window}$$
$$W(t_n) = 0.5 \left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right) \quad \text{Hann window.}$$

7.3 Fourier Transform Algorithms

Straight Forward evaluation of the sum

$$\widetilde{f}_{\omega_j} = \sum_{n=0}^{N-1} \cos\left(\frac{2\pi}{N} jn\right) f_n + \mathrm{i} \sin\left(\frac{2\pi}{N} jn\right) f_n \tag{7.43}$$

needs $O(N^2)$ additions, multiplications and trigonometric functions.

7.3.1 Goertzel's Algorithm

Goertzel's method [74] is very useful if not the whole Fourier spectrum is needed but only some of the Fourier components, for instance to demodulate a frequency shift key signal or the dial tones which are used in telephony.

The Fourier transform can be written as

$$\sum_{n=0}^{N-1} f_n \mathrm{e}^{-\mathrm{i}\frac{2\pi}{N} jn} = f_0 + \mathrm{e}^{-\frac{2\pi \mathrm{i}}{N} j} \left(f_1 + \mathrm{e}^{-\frac{2\pi \mathrm{i}}{N} j} f_2 \ldots \left(f_{N-2} + \mathrm{e}^{-\frac{2\pi \mathrm{i}}{N} j} f_{N-1} \right) \ldots \right) \tag{7.44}$$

which can be evaluated recursively

$$\begin{aligned} y_{N-1} &= f_{N-1} \\ y_n &= f_n + \mathrm{e}^{-\frac{2\pi \mathrm{i}}{N} j} y_{n+1} \quad n = N-2, \ldots 0 \end{aligned} \tag{7.45}$$

to give the result

$$\hat{f}_{\omega_j} = y_0. \tag{7.46}$$

Equation (7.45) is a simple discrete filter function. Its transmission function is obtained by application of the z-transform [75]

$$u(z) = \sum_{n=0}^{\infty} u_n z^{-n} \tag{7.47}$$

(the discrete version of the Laplace transform) which yields

$$y(z) = \frac{f(z)}{1 - z\mathrm{e}^{-\frac{2\pi \mathrm{i}}{N} j}}. \tag{7.48}$$

One disadvantage of this method is that it uses complex numbers. This can be avoided by the following more complicated recursion

$$\begin{aligned} u_{N+1} &= u_N = 0 \\ u_n &= f_n + 2u_{n+1} \cos \tfrac{2\pi}{N} k - u_{n+2} \text{ for } n = N-1, \ldots 0 \end{aligned} \tag{7.49}$$

with the transmission function

$$\begin{aligned} \frac{u(z)}{f(z)} &= \frac{1}{1 - z\left(\mathrm{e}^{\frac{2\pi \mathrm{i}}{N} j} + \mathrm{e}^{-\frac{2\pi \mathrm{i}}{N} j}\right) + z^2} \\ &= \frac{1}{1 - z\mathrm{e}^{-\frac{2\pi \mathrm{i}}{N} j}} \frac{1}{1 - z\mathrm{e}^{\frac{2\pi \mathrm{i}}{N} j}}. \end{aligned} \tag{7.50}$$

A second filter removes one factor in the denominator

$$\frac{y(z)}{u(z)} = 1 - z\mathrm{e}^{\frac{2\pi \mathrm{i}}{N} j} \tag{7.51}$$

which in the time domain corresponds to the simple expression

$$y_n = u_n - \mathrm{e}^{\frac{2\pi \mathrm{i}}{N} j} u_{n+1}.$$

The overall filter function finally again is (7.48).

$$\frac{y(z)}{f(z)} = \frac{1}{1 - z\mathrm{e}^{-\frac{2\pi \mathrm{i}}{N} j}}. \tag{7.52}$$

Hence the Fourier component of **f** is given by

$$\hat{f}_{\omega_j} = y_0 = u_0 - \mathrm{e}^{\frac{2\pi \mathrm{i}}{N} j} u_1. \tag{7.53}$$

The order of the iteration (7.44) can be reversed by writing

$$\hat{f}_{\omega_j} = f_0 \ldots \mathrm{e}^{\frac{2\pi \mathrm{i}}{N}(N-1)} f_{N-1} = \mathrm{e}^{-\frac{2\pi \mathrm{i}}{N} j(N-1)} \left(f_0 \mathrm{e}^{\frac{2\pi \mathrm{i}}{N} j(N-1)} \ldots f_{N-1} \right) \tag{7.54}$$

which is very useful for real time filter applications.

7.3.2 *Fast Fourier Transformation*

If the number of samples is $N = 2^p$, the Fourier transformation can be performed very efficiently by this method.[2] The phase factor

$$\mathrm{e}^{-\mathrm{i}\frac{2\pi}{N} jm} = W_N^{jm} \tag{7.55}$$

can take only N different values. The number of trigonometric functions can be reduced by reordering the sum. Starting from a sum with N samples

$$F_N(f_0 \ldots f_{N-1}) = \sum_{n=0}^{N-1} f_n W_N^{jn} \tag{7.56}$$

we separate even and odd powers of the unit root

$$\begin{aligned} F_N(f_0 \ldots f_{N-1}) &= \sum_{m=0}^{\frac{N}{2}-1} f_{2m} W_N^{j2m} + \sum_{m=0}^{\frac{N}{2}-1} f_{2m+1} W_N^{j(2m+1)} \\ &= \sum_{m=0}^{\frac{N}{2}-1} f_{2m} \mathrm{e}^{-\mathrm{i}\frac{2\pi}{N/2} jm} + W_N^j \sum_{m=0}^{\frac{N}{2}-1} f_{2m+1} \mathrm{e}^{-\mathrm{i}\frac{2\pi}{N/2} jm} \\ &= F_{N/2}(f_0, f_2 \ldots f_{N-2}) + W_N^j F_{N/2}(f_1, f_3 \ldots f_{N-1}). \end{aligned} \tag{7.57}$$

This division is repeated until only sums with one summand remain

$$F_1(f_n) = f_n. \tag{7.58}$$

[2] There exist several Fast Fourier Transformation algorithms [76, 77]. We consider only the simplest one here [78].

For example, consider the case $N = 8$:

$$\begin{aligned} F_8(f_0 \dots f_7) &= F_4(f_0 f_2 f_4 f_6) + W_8^j F_4(f_1 f_3 f_5 f_7) \\ &\quad - - - \\ F_4(f_0 f_2 f_4 f_6) &= F_2(f_0 f_4) + W_4^j F_2(f_2 f_6) \\ F_4(f_1 f_3 f_5 f_7) &= F_2(f_1 f_5) + W_4^j F_2(f_3 f_7) \\ &\quad - - - \\ F_2(f_0 f_4) &= f_0 + W_2^j f_4 \\ F_2(f_2 f_6) &= f_2 + W_2^j f_6 \\ F_2(f_1 f_5) &= f_1 + W_2^j f_5 \\ F_2(f_3 f_7) &= f_3 + W_2^j f_7. \end{aligned}$$

Expansion gives

$$\begin{aligned} F_8 &= f_0 + W_2^j f_4 + W_4^j f_2 + W_4^j W_2^j f_6 \\ &+ W_8^j f_1 + W_8^j W_2^j f_5 + W_8^j W_4^j f_3 + W_8^j W_4^j W_2^j f_7. \end{aligned} \tag{7.59}$$

Generally a summand of the Fourier sum can be written using the binary representation of n

$$n = \quad l_i \quad l_i = 1, 2, 4, 8 \dots \tag{7.60}$$

in the following way:

$$f_n \mathrm{e}^{-\mathrm{i}\frac{2\pi}{N} jn} = f_n \mathrm{e}^{-\mathrm{i}\frac{2\pi}{N}(l_1 + l_2 + \cdots) j} = f_n W_{N/l_1}^j W_{N/l_2}^j \dots . \tag{7.61}$$

The function values are reordered according to the following algorithm
(i) count from 0 to N-1 using binary numbers $m = 000, 001, 010, \dots$
(ii) bit reversal gives the binary numbers $n = 000, 100, 010, \dots$
(iii) store f_n at the position m. This will be denoted as $s_m = f_n$
As an example for N=8 the function values are in the order

$$\begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{pmatrix} = \begin{pmatrix} f_0 \\ f_4 \\ f_2 \\ f_6 \\ f_1 \\ f_5 \\ f_3 \\ f_7 \end{pmatrix}. \tag{7.62}$$

Now calculate sums with two summands. Since W_2^j can take only two different values

$$W_2^j = \begin{cases} 1 \; for \; j = 0, 2, 4, 6 \\ -1 \; for \; j = 1, 3, 5, 7 \end{cases} \tag{7.63}$$

a total of 8 sums have to be calculated which can be stored again in the same workspace:

$$\begin{pmatrix} f_0 + f_4 \\ f_0 - f_4 \\ f_2 + f_6 \\ f_2 - f_6 \\ f_1 + f_5 \\ f_1 - f_5 \\ f_3 + f_7 \\ f_3 - f_7 \end{pmatrix} = \begin{pmatrix} s_0 + W_2^0 s_1 \\ s_0 + W_2^1 s_1 \\ s_2 + W_2^2 s_3 \\ s_2 + W_2^3 s_3 \\ s_4 + W_2^4 s_5 \\ s_4 + W_2^5 s_5 \\ s_6 + W_2^6 s_7 \\ s_6 + W_2^7 s_7 \end{pmatrix}. \tag{7.64}$$

Next calculate sums with four summands. W_4^j can take one of four values

$$W_4^j = \begin{cases} 1 \quad for \quad j = 0, 4 \\ -1 \quad for \quad j = 2, 6 \\ W_4 \quad for \quad j = 1, 5 \\ -W_4 \quad for \quad j = 3, 7 \end{cases}. \tag{7.65}$$

The following combinations are needed:

$$\begin{pmatrix} f_0 + f_4 + (f_2 + f_6) \\ f_0 + f_4 - (f_2 + f_6) \\ (f_0 - f_4) + W_4(f_2 - f_6) \\ (f_0 - f_4) - W_4(f_2 - f_6) \\ f_1 + f_5 + (f_3 + f_7) \\ f_1 + f_5 - (f_3 + f_7) \\ (f_1 - f_5) \pm W_4(f_3 - f_7) \\ (f_1 - f_5) \pm W_4(f_3 - f_7) \end{pmatrix} = \begin{pmatrix} s_0 + W_4^0 s_2 \\ s_1 + W_4^1 s_3 \\ s_0 + W_4^2 s_2 \\ s_1 + W_4^3 s_3 \\ s_4 + W_4^4 s_6 \\ s_5 + W_4^5 s_7 \\ s_4 + W_4^6 s_6 \\ s_5 + W_4^7 s_7 \end{pmatrix}. \tag{7.66}$$

The next step gives the sums with eight summands. With

$$W_8^j = \begin{cases} 1 \quad j = 0 \\ W_8 \quad j = 1 \\ W_8^2 \quad j = 2 \\ W_8^3 \quad j = 3 \\ -1 \quad j = 4 \\ -W_8 \quad j = 5 \\ -W_8^2 \quad j = 6 \\ -W_8^3 \quad j = 7 \end{cases} \tag{7.67}$$

we calculate

$$\begin{pmatrix} f_0+f_4+(f_2+f_6)+(f_1+f_5+(f_3+f_7)) \\ f_0+f_4-(f_2+f_6)+W_8(f_1+f_5-(f_3+f_7)) \\ (f_0-f_4)+W_4(f_2-f_6)+W_8^2(f_1-f_5)\pm W_4(f_3-f_7) \\ (f_0-f_4)-W_4(f_2-f_6)+W_8^3((f_1-f_5)\pm W_4(f_3-f_7)) \\ f_0+f_4+(f_2+f_6)-(f_1+f_5+(f_3+f_7)) \\ f_0+f_4-(f_2+f_6)-W_8(f_1+f_5-(f_3+f_7)) \\ (f_0-f_4)+W_4(f_2-f_6)-W_8^2((f_1-f_5)\pm W_4(f_3-f_7)) \\ (f_0-f_4)-W_4(f_2-f_6)-W_8^3((f_1-f_5)\pm W_4(f_3-f_7)) \end{pmatrix} = \begin{pmatrix} s_0+W_8^0 s_4 \\ s_1+W_8^1 s_5 \\ s_2+W_8^2 s_6 \\ s_3+W_8^3 s_7 \\ s_0+W_8^4 s_4 \\ s_1+W_8^5 s_5 \\ s_2+W_8^6 s_6 \\ s_3+W_8^7 s_7 \end{pmatrix} \tag{7.68}$$

which is the final result.

The following shows a simple Fast Fourier Transformation algorithm. The number of trigonometric function evaluations can be reduced but this reduces the readability. At the beginning $Data[k]$ are the input data in bit reversed order.

```
size:=2
first:=0
While first < Number_of_Samples do begin
  for n:=0 to size/2-1 do begin
    j:=first+n
    k:=j+size/2-1
    T:=exp(-2*Pi*i*n/Number_of_Samples)*Data[k]
    Data[j]:=Data[j]+T
    Data[k]:=Data[k]-T
  end;
  first:=first*2
  size:=size*2
end;
```

Problems

Problem 7.1 Discrete Fourier Transformation

In this computer experiment for a given set of input samples

$$f_n = f(n\frac{T}{N}) \quad n = 0 \ldots N-1 \tag{7.69}$$

- the Fourier coefficients

$$\widetilde{f}_{\omega_j} = \sum_{n=0}^{N-1} f_n e^{-i\omega_j t_n} \quad \omega_j = \frac{2\pi}{T} j, \quad j = 0 \ldots N-1 \tag{7.70}$$

are calculated with Görtzel's method 7.3.1.

- The results from the inverse transformation

$$f_n = \frac{1}{N} \sum_{j=0}^{N-1} \widetilde{f}_{\omega_j} e^{i \frac{2\pi}{N} nj} \tag{7.71}$$

are compared with the original function values $f(t_n)$.

- The Fourier sum is used for trigonometric interpolation with only positive frequencies

$$f(t) = \frac{1}{N} \sum_{j=0}^{N-1} \widetilde{f}_{\omega_j} \left(e^{i \frac{2\pi}{T} t} \right)^j . \tag{7.72}$$

- Finally the unphysical high frequencies are replaced by negative frequencies (7.24). The results can be studied for several kinds of input data.

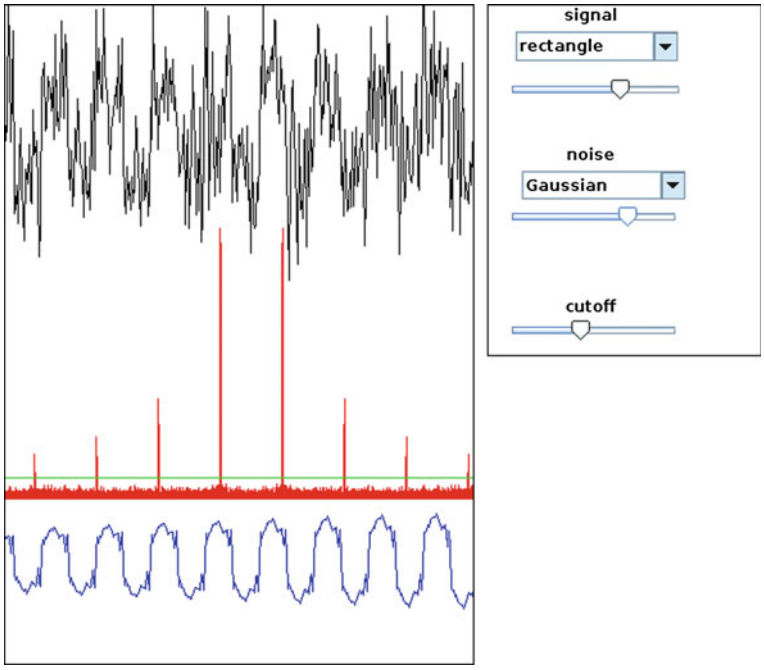


Fig. 7.3 (Screenshot from computer exercise 7.2) ***Top*** The input signal is rectangular with Gaussian noise. ***Middle*** The Components of the Fourier spectrum (*red*) below the threshold (*green line*) are dropped. ***Bottom*** the filtered signal is reconstructed (*blue*)

Problem 7.2 Noise Filter

This computer experiment demonstrates a nonlinear filter.
First a noisy input signal is generated.
The signal can be chosen as

- monochromatic $\sin(\omega t)$
- the sum of two monochromatic signals $a_1 \sin \omega_1 t + a_2 \sin \omega_2 t$
- a rectangular signal with many harmonic frequencies $\text{sign}(\sin \omega t)$

Different kinds of white noise can be added

- dichotomous ± 1
- constant probability density in the range $[-1, 1]$
- Gaussian probability density

The amplitudes of signal and noise can be varied. All Fourier components are removed which are below a threshold value and the filtered signal is calculated by inverse Fourier transformation. Figure 7.3 shows a screenshot from the program.

Chapter 8
Time-Frequency Analysis

Fourier-analysis provides a description of a given data set in terms of monochromatic oscillations without any time information. It is thus mostly useful for stationary signals. If the spectrum changes in time it is desirable to obtain information about the time at which certain frequencies appear. This can be achieved by applying Fourier analysis to a short slice of the data (short time Fourier analysis) which is shifted along the time axis. The frequency resolution is the same for all frequencies and therefore it can be difficult to find a compromise between time and frequency resolution. Wavelet analysis uses a frequency dependent window and keeps the relative frequency resolution constant. This is achieved by scaling and shifting a prototype wavelet - the so called mother wavelet. Depending on the application wavelets can be more general and need not be sinusoidal or even continuous functions. Multiresolution analysis provides orthonormal wavelet bases which simplify the wavelet analysis. The fast wavelet transform connects a set of sampled data with its wavelet coefficients and is very useful for processing audio and image data.

8.1 Short Time Fourier Transform (STFT)

Fourier analysis transforms a function in the time domain $f(t)$ into its spectrum

$$\widetilde{f}(\omega) = \mathcal{F}[f](\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) \mathrm{e}^{-\mathrm{i}\omega t} dt \tag{8.1}$$

thereby losing all time information. Short time Fourier analysis applies a windowing function[1] (p. 133) e.g. a simple rectangle (Fig. 8.1)[2]

[1] Also known as apodization function or tapering function.

[2] There are two different definitions of the sinc function in the literature.

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_8

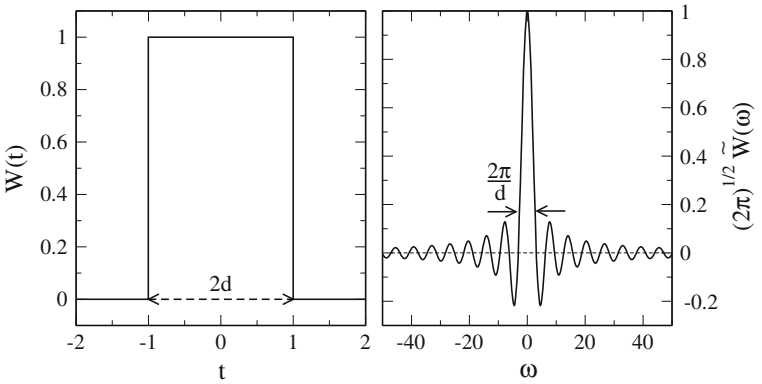


Fig. 8.1 (Rectangular window) The rectangular (uniform) window and its Fourier transform are shown for $d = 1$

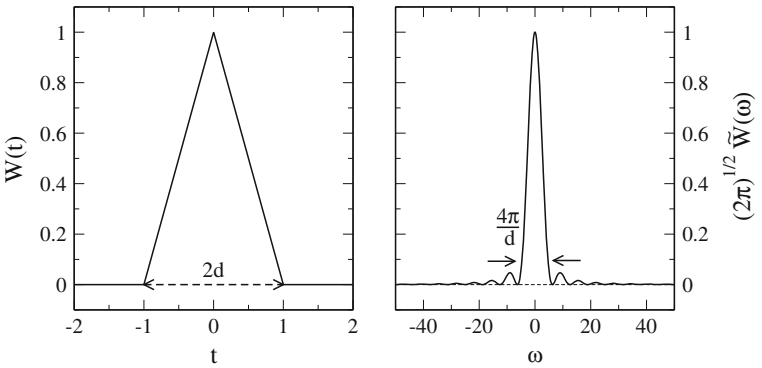


Fig. 8.2 (Triangular window) The *triangular* (Bartlett) window and its Fourier transform are shown for $d = 1$

$$W_R(t) = \begin{array}{c} 1 \text{ for } |t| \le d \\ 0 \text{ else} \end{array} \tag{8.2}$$

$$\tilde{W_R}(\omega) = \frac{2d}{\sqrt{2\pi}} \frac{\sin \omega d}{\omega d} = \frac{2d}{\sqrt{2\pi}} \text{sinc}(\omega d) \tag{8.3}$$

or triangle (Fig. 8.2)

$$W_{Tr}(t) = \begin{array}{c} 1 - \frac{|t|}{d} \quad \text{for } t \le d \\ 0 \text{ else} \end{array} \tag{8.4}$$

$$\tilde{W}_{Tr}(\omega) = \frac{d}{\sqrt{2\pi}} \frac{2(1 - \cos \omega d)}{\omega^2 d^2} == \frac{d}{\sqrt{2\pi}} \quad \text{sinc} \quad \frac{\omega d}{2} \quad ^2 . \tag{8.5}$$

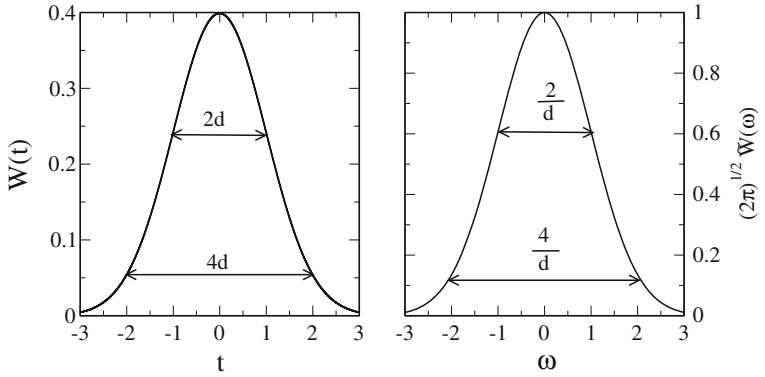


Fig. 8.3 (Gaussian window) The Gaussian window and its Fourier transform are shown for $d = 1$

A smoother window is the Gaussian (p. 192) (Fig. 8.3)

$$W_G(t) = \frac{1}{d\sqrt{2\pi}} \exp\left\{-\frac{t^2}{2d^2}\right\} \tag{8.6}$$

with

$$\tilde{W}_G(\omega) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{\omega^2 d^2}{2}\right\}. \tag{8.7}$$

For the Gaussian window the standard deviations[3]

$$\sigma_t = d \quad \sigma_\omega = \frac{1}{d} \tag{8.8}$$

obey the uncertainty relation[4]

$$\sigma_t \sigma_\omega = 1. \tag{8.9}$$

Since the Gaussian extends to infinity, it has to be cut off for practical calculations.

Quite popular are the Hann(ing) and Hamming windows (Fig. 8.4)

$$W_{Hann}(t) = \cos^2\left(\frac{\pi t}{2d}\right) = \left(\frac{1}{2} + \frac{1}{2}\cos\frac{\pi t}{d}\right) \tag{8.10}$$

[3]Here we use the definition $\sigma^2 = \int_{-\infty}^{\infty} dt\, W(t)t^2$. If instead $\sigma^2 = \int_{-\infty}^{\infty} dt\, |W(t)|^2 t^2$ is used then $\sigma_t = \frac{d}{\sqrt{2}}$ and $\sigma_\omega = \frac{1}{\sqrt{2}d}$.

[4]For a Gaussian the time-bandwidth product is minimal.

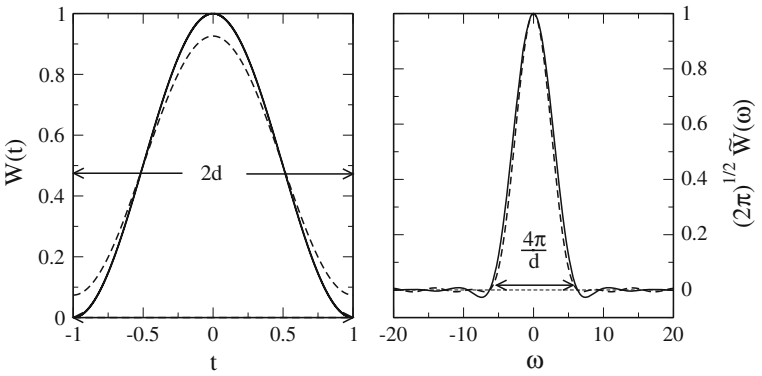


Fig. 8.4 (Hann and Hamming window) The Hann (*full curves*) and Hamming (*dashed curves*) windows together with their Fourier transforms are shown for $\Delta t = 1$. The Hamming window is optimized to reduce the side lobes in the spectrum

$$\tilde{W}_{Hann}(\omega) = \frac{d}{\sqrt{2\pi}} \frac{\mathrm{sinc}\omega d}{1 - \frac{\omega^2 d^2}{\pi^2}} \tag{8.11}$$

$$W_{Hamm}(t) = \frac{27}{54} + \frac{23}{54} \cos \frac{\pi t}{d} \tag{8.12}$$

$$\tilde{W}_{Hamm}(\omega) = \frac{d}{\sqrt{2\pi}} \frac{1 - \frac{4}{27} \frac{\omega^2 d^2}{\pi^2}}{1 - \frac{\omega^2 d^2}{\pi^2}} \mathrm{sinc}\omega d. \tag{8.13}$$

For a general real valued function

$$\tilde{W}(\omega)^* = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} W(t) \mathrm{e}^{-\mathrm{i}\omega t} dt \Big)^* = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} W(t) \mathrm{e}^{\mathrm{i}\omega t} dt = \tilde{W}(-\omega) \tag{8.14}$$

and for an even function

$$\tilde{W}(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} W(t) \mathrm{e}^{-\mathrm{i}\omega t} dt = \frac{1}{\sqrt{2\pi}} \int_{\infty}^{-\infty} W(t) \mathrm{e}^{\mathrm{i}\omega t} d(-t) = \tilde{W}(-\omega). \tag{8.15}$$

If $W(t)$ is real and even than this holds also for its Fourier transform

$$\tilde{W}(-\omega) = \tilde{W}(\omega) = \tilde{W}(\omega)^* = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} W(t) \cos(\omega t) dt. \tag{8.16}$$

The short time Fourier Transform

$$X(t_0, \omega) = \mathcal{F}[W^*(t - t_0) f(t)](\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} W^*(t - t_0) f(t) \mathrm{e}^{-\mathrm{i}\omega t} dt \tag{8.17}$$

depends on two variables t_0 and ω. Since it has the form of a convolution integral it becomes a product in Fourier space, where

$$\begin{aligned}
&\frac{1}{\sqrt{2\pi}} \int dt_0 \mathrm{e}^{-\mathrm{i}\omega_0 t_0} X(t_0, \omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dt_0 \mathrm{e}^{-\mathrm{i}\omega_0 t_0} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dt \; W^*(t - t_0) f(t) \mathrm{e}^{-\mathrm{i}\omega t} \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} dt \, d(t - t_0) \mathrm{e}^{\mathrm{i}\omega_0 (t - t_0)} \; W^*(t - t_0) f(t) \mathrm{e}^{-\mathrm{i}\omega_0 t} \mathrm{e}^{-\mathrm{i}\omega t} \\
&= \frac{1}{2\pi} \left(\int_{-\infty}^{\infty} d(t - t_0) \mathrm{e}^{-\mathrm{i}\omega_0 (t - t_0)} W(t - t_0) \int_{-\infty}^{\infty} \right)^* dt \;\; f(t) \mathrm{e}^{-\mathrm{i}\omega_0 t} \mathrm{e}^{-\mathrm{i}\omega t} \\
&= \tilde{W}^*(\omega_0) \tilde{f}(\omega + \omega_0).
\end{aligned} \tag{8.18}$$

For a real valued an even windowing function like the Gaussian (8.6) the STFT can therefore be calculated from

$$X(t_0, \omega) = \frac{1}{\sqrt{2\pi}} \int d\omega_0 \mathrm{e}^{\mathrm{i}\omega_0 t_0} \tilde{W}(\omega_0) \tilde{f}(\omega_0 + \omega). \tag{8.19}$$

Alternatively, the STFT can be formulated as

$$X(t_0, \omega) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{\mathrm{i}\omega t_0} \int_{-\infty}^{\infty} f(t) \; W^*(t - t_0) \mathrm{e}^{-\mathrm{i}\omega (t - t_0)} dt \tag{8.20}$$

$$= \frac{1}{\sqrt{2\pi}} \mathrm{e}^{\mathrm{i}\omega t_0} \int_{-\infty}^{\infty} f(t) \; \Omega^*(t - t_0) dt$$

which involves a convolution of $f(t)$ with the wave packet (Fig. 8.5)

$$\Omega(t - t_0) = W(t - t_0) \mathrm{e}^{\mathrm{i}\omega (t - t_0)} \tag{8.21}$$

which is localized around t_0. In frequency space the wave packet becomes a band pass filter

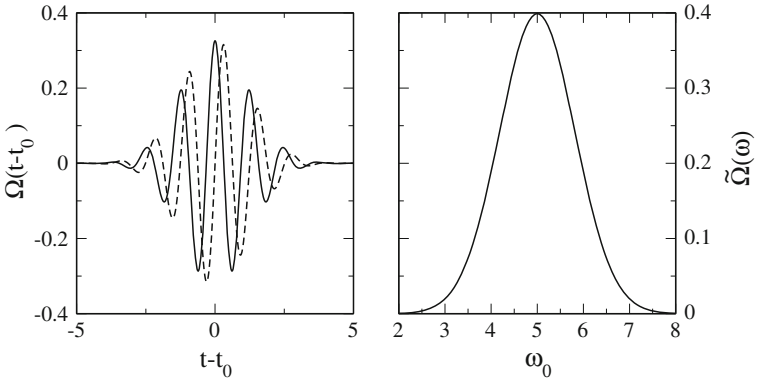


Fig. 8.5 (Gabor wave packet) ***Left*** Real (*full curve*) and imaginary (*dashed curve*) part of the wave packet (8.21) are shown for a Gaussian windowing function with $\omega = 5$ and $2d^2 = 3$. ***Right*** In the frequency domain the wave packet acts as a bandpass filter at $\omega_0 = \omega$

$$\begin{aligned}
\tilde{\Omega}(\omega_0) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dt \mathrm{e}^{-\mathrm{i}\omega_0 t} W(t) \mathrm{e}^{\mathrm{i}\omega t} \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dt \mathrm{e}^{-\mathrm{i}(\omega_0-\omega)t} W(t) \\
&= \tilde{W}(\omega_0 - \omega)
\end{aligned}$$

$$\begin{aligned}
&\frac{1}{2\pi} \int_{-\infty}^{\infty} dt_0 \mathrm{e}^{-\mathrm{i}\omega_0 t_0} \int_{-\infty}^{\infty} f(t)\, \Omega^*(t - t_0) dt \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} dt \int_{-\infty}^{\infty} d(t - t_0)\, f(t) \mathrm{e}^{-\mathrm{i}\omega_0 t}\, \Omega^*(t - t_0) \mathrm{e}^{\mathrm{i}\omega_0 (t-t_0)} \\
&= \tilde{f}(\omega_0) \tilde{\Omega}^*(\omega_0)
\end{aligned} \tag{8.22}$$

$$\begin{aligned}
X(t_0, \omega) &= \frac{1}{\sqrt{2\pi}} \mathrm{e}^{\mathrm{i}\omega t_0} \int_{-\infty}^{\infty} d\omega_0 \mathrm{e}^{\mathrm{i}\omega_0 t_0} \tilde{f}(\omega_0) \tilde{\Omega}^*(\omega_0) \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} d\omega_0 \mathrm{e}^{\mathrm{i}(\omega+\omega_0)t_0} \tilde{f}(\omega_0) \tilde{\Omega}^*(\omega_0) \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} d\omega_0 \mathrm{e}^{\mathrm{i}\omega t_0} \tilde{f}(\omega_0 - \omega) \tilde{\Omega}^*(\omega_0 - \omega).
\end{aligned}$$

The STFT can be inverted by

$$\int d\omega X(t_0,\omega)\mathrm{e}^{\mathrm{i}\omega t_0} = \frac{1}{\sqrt{2\pi}}\int dt \int d\omega \mathrm{e}^{\mathrm{i}\omega t_0} W^*(t-t_0) f(t)\mathrm{e}^{-\mathrm{i}\omega t}$$
$$= \frac{1}{\sqrt{2\pi}}\int dt W^*(t-t_0) f(t) 2\pi\delta(t-t_0) = \sqrt{2\pi} W^*(0) f(t_0) \tag{8.23}$$

or alternatively

$$\int dt_0 \int d\omega X(t_0,\omega)\mathrm{e}^{\mathrm{i}\omega t} = \int dt_0 \int d\omega \frac{1}{\sqrt{2\pi}}\mathrm{e}^{\mathrm{i}\omega t}\int_{-\infty}^{\infty} W^*(t'-t_0) f(t')\mathrm{e}^{-\mathrm{i}\omega t'} dt'$$
$$= \int dt_0 \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} 2\pi\delta(t-t') W^*(t'-t_0) f(t') dt'$$
$$= \int dt_0 \frac{1}{\sqrt{2\pi}} 2\pi W^*(t-t_0) f(t) = \sqrt{2\pi} f(t)\int W^*(t') dt'. \tag{8.24}$$

STFT with a Gaussian window is also known as Gabor transform [79] which is conventionally defined as

$$\mathcal{G}[f](t_0,\omega) = \int_{-\infty}^{\infty} dt\, \mathrm{e}^{-\alpha\pi(t-t_0)^2}\mathrm{e}^{-\mathrm{i}\omega t} f(t). \tag{8.25}$$

Example: Spectrogram

The STFT is often used to analyze audio signals. Let us consider as a simple example a monochromatic signal, which is switched on at time $t = 0$

$$f(t) = \begin{cases} 0 & t < 0 \\ \sin(\omega_s t) & t \geq 0. \end{cases} \tag{8.26}$$

Using a Gaussian window, the Fourier transform can be calculated explicitly (an algebra program is very helpful)

$$X(t_0,\omega) = \frac{1}{2\pi\Delta t}\int_0^{\infty} dt\, \mathrm{e}^{-\mathrm{i}\omega t}\mathrm{e}^{-(t-t_0)^2/2d^2}\sin(\omega_s t)$$
$$= -\frac{\mathrm{i}}{4\sqrt{2\pi}}\mathrm{e}^{-\mathrm{i}t_0(\omega-\omega_s)}\mathrm{e}^{-d^2(\omega-\omega_s)^2/2}\left(\mathrm{erf}\left(\frac{\mathrm{i}\Delta t^2(\omega-\omega_s)-t_0}{\sqrt{2}d}\right)-1\right)$$
$$+\frac{\mathrm{i}}{4\sqrt{2\pi}}\mathrm{e}^{-\mathrm{i}t_0(\omega+\omega_s)}\mathrm{e}^{-d^2(\omega+\omega_s)^2/2}\left(\mathrm{erf}\left(\frac{\mathrm{i}d^2(\omega+\omega_s)-t_0}{\sqrt{2}d}\right)-1\right). \tag{8.27}$$

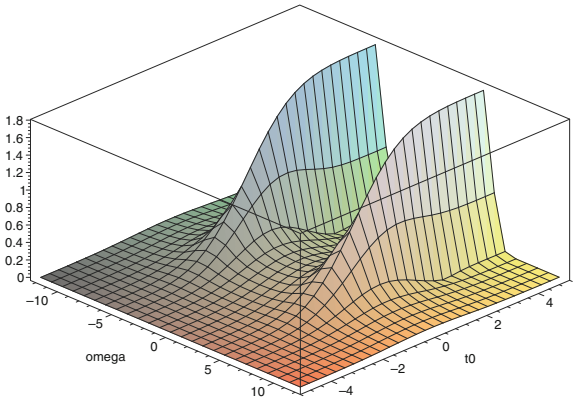


Fig. 8.6 (3-d spectrogram) The squared magnitude of the STFT (8.27) is shown for $\omega_s = 5$ and $2d = 1$

There are two contributions since the real function $f(t)$ contains oscillations with $\pm\omega_s$. The squared magnitude[5] $|X(t_0, \omega)|^2$ is shown as a 3-d spectrogram in Fig. 8.6. The width of the window determines the resolution both in time and frequency. Neglecting interference terms, at resonance $\omega = \omega_s$ the squared magnitude rises according to

$$|X(t_0, \omega_s)|^2 = \frac{1}{32\pi} \quad \text{erf} \frac{t_0}{\sqrt{2}d} - 1 \quad ^2 \tag{8.28}$$

and reaches its stationary value within a time of $\approx 2d$, whereas in the stationary state at $t_0 \gg d$, the dependency on the frequency mismatch $\Delta\omega = \omega - \omega_s$ is given by a Gaussian with a width of $2\sigma_\omega = \sqrt{2}/d$. The dependence of time and frequency resolution on the window width is shown qualitatively by 2-dimensional spectrograms in Fig. 8.7.

8.2 Discrete Short Time Fourier Transform

The continuous STFT is very redundant and not useful for the analysis of data which are sampled at discrete times. Therefore we introduce a series of overlapping windows centered at equidistant times $t_n = n\Delta t$ (Fig. 8.8)

$$W_n(t) = W(n\Delta t - t). \tag{8.29}$$

[5] This is a measure of the spectral power distribution.

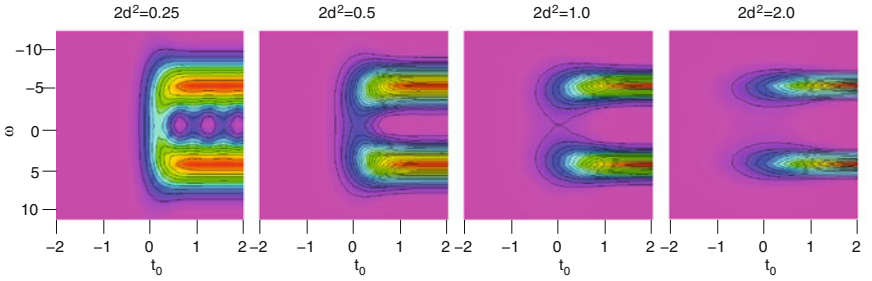


Fig. 8.7 (Spectrograms with different window width Δt) The squared magnitude of the STFT (8.27) is shown for $\omega_s = 5$ and $2d^2 = 0.25, 0.5, 1.0, 2.0$. For larger values of the time window d the frequency resolution becomes higher but the time resolution lower

Assuming that the windowing function $W_n(t) = 0$ outside the interval $[t_n - d, t_n + d]$ we apply (7.5) and expand $W_n(t)\, f(t)$ inside the interval as a Fourier series

$$g_n(t) = W_n(t)\, f(t) = \sum_{m=-\infty}^{\infty} \mathrm{e}^{\mathrm{i}\omega_m t}\, \hat{g}_{nm} \text{ with } \omega_m = m\frac{\pi}{d} \quad |t - t_n| \le d. \tag{8.30}$$

We extend this expression to all times t by introducing the characteristic function of the interval

$$\chi_n(t) = \begin{cases} 1 & \text{for } |t - t_n| \le d \\ 0 & \text{else} \end{cases} \tag{8.31}$$

$$g_n(t) = W_n(t)\, f(t) = \chi_n(t) \sum_{m=-\infty}^{\infty} \mathrm{e}^{\mathrm{i}\omega_m t}\, \hat{g}_{nm}. \tag{8.32}$$

The Fourier coefficients, given by the integral

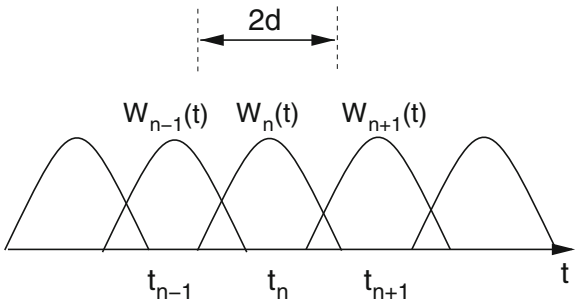


Fig. 8.8 (Discrete STFT) Assuming that the windowing function $W_n(t) = 0$ outside the interval $[t_n - d, t_n + d]$ we apply (7.5) and expand $W_n(t)\, f(t)$ inside the interval as a Fourier series

$$\hat{g}_{nm} = \frac{1}{2d} \int_{t_n-d}^{t_n+d} W(t-t_n) f(t) \mathrm{e}^{-\mathrm{i}\omega_m t} dt \tag{8.33}$$

obviously correspond to the STFT at times t_n and frequencies ω_m

$$X(t_n, \omega_m) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} W(t-t_n) f(t) \mathrm{e}^{-\mathrm{i}\omega_m t} dt = \frac{1}{\sqrt{2\pi}} \int_{t_n-d}^{t_n+d} W(t-t_n) f(t) \mathrm{e}^{-\mathrm{i}\omega_m t} dt$$
$$= \frac{2d}{\sqrt{2\pi}} \hat{g}_{nm}. \tag{8.34}$$

If the windows are dense enough such that there union spans all times, the signal can be reconstructed by summation

$$g_n(t) = f(t) \sum_n W_n(t) = \sum_{nm} \chi_n(t) \mathrm{e}^{\mathrm{i}\omega_m t} \hat{g}_{nm}. \tag{8.35}$$

This expression simplifies, if

$$\sum_n W_n(t) = \text{const} \tag{8.36}$$

which is e.g. the case for triangular windows as well as the Hann and Hamming windows with $\Delta t = d$ (Fig. 8.9).

For practical applications, we assume that the function $g(t)$ has been sampled at N equidistant times within the interval $[t_n - d, t_n + d]$

$$\tau_{n,s} = t_n - d + s\frac{2d}{N} \quad s = 0, 1, \ldots N-1 \tag{8.37}$$

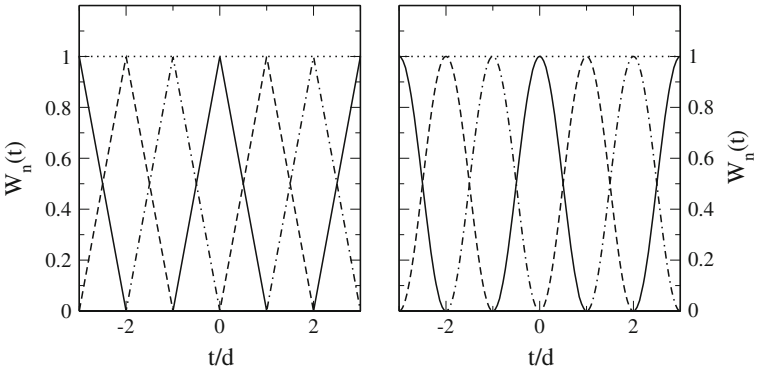


Fig. 8.9 (Window functions with constant sum) For the triangular (***Left***) and the Hann and Hamming (***Right***) window the sum $\sum_n W_n(t)$ becomes constant (*dotted lines*) if the windows are shifted by half their width $\Delta t = d$

$$\omega_j = \frac{\pi}{d} j, \quad \omega_j(\tau_{n,s} - t_n + d) = js\frac{2\pi}{N} \tag{8.38}$$

and apply the discrete Fourier transformation method (p. 131)

$$\widetilde{g}_{n,\omega_j} = \sum_{s=0}^{N-1} g_{n,s} \mathrm{e}^{-\mathrm{i}\omega_j(\tau_{ns} - t_n + d)} = \sum_{s=0}^{N-1} g_{n,s} \mathrm{e}^{-\mathrm{i}js\frac{2\pi}{N}} \tag{8.39}$$

$$\frac{1}{N}\sum_{j=0}^{N-1} \tilde{g}_{n,\omega_j} \mathrm{e}^{\mathrm{i}js\frac{2\pi}{N}} = \frac{1}{N}\sum_{j=0}^{N-1}\sum_{s=0}^{N-1} g_{n,s'} \mathrm{e}^{-\mathrm{i}js'\frac{2\pi}{N}} \mathrm{e}^{\mathrm{i}js\frac{2\pi}{N}} = g_{n,s}. \tag{8.40}$$

Example: FM Signal

Figures 8.10 and 8.11 show the STFT analysis of a frequency modulated signal

$$f(t) = \sin \Phi(t) = \sin\left(\omega_0 t + \frac{a\omega_0}{\omega_1}(1 - \cos\omega_1 t)\right) \tag{8.41}$$

with a momentaneous frequency of

$$\omega(t) = \frac{\partial \Phi}{\partial t} = \omega_0(1 + a\sin\omega_1 t) \tag{8.42}$$

for carrier frequency $\frac{\omega_0}{2\pi} = 10\,\mathrm{kHz}$, modulation frequency $\frac{\omega_1}{2\pi} = 25\,\mathrm{Hz}(100\,\mathrm{Hz})$ and modulation depth $a = 0.3$.

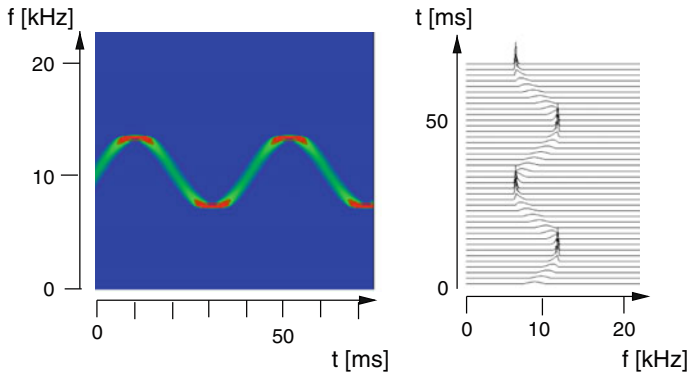


Fig. 8.10 (STFT analysis of a FM signal) The figure shows screenshots from Problem 8.1. ***Left*** spectrogram ***Right*** STFT spectra. Sampling frequency is 44100 Hz, number of samples 512, Hann windows are used with a shift of 8 samples (0.18 ms) between neighbor windows. 6 ms time resolution and 1.1 kHz frequency resolution are sufficient to resolve the 25 Hz modulation. The time dependent spectra have their smallest width at the stationary points of the momentaneous frequency

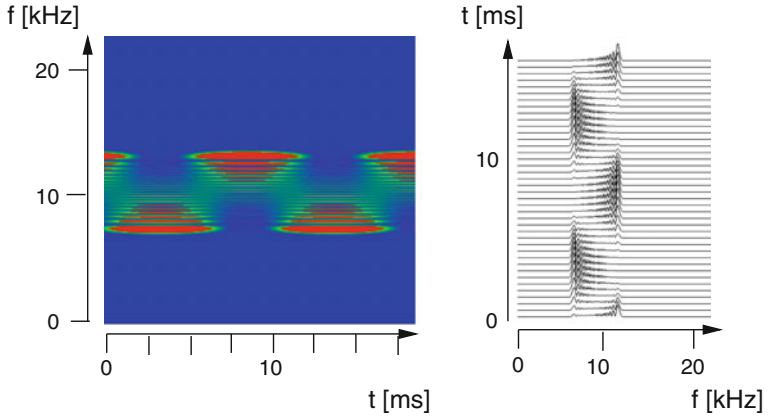


Fig. 8.11 (STFT analysis of a FM signal) The figure shows screenshots from Problem 8.1. ***Left*** spectrogram ***Right*** STFT spectra. Sampling frequency is 44100 Hz, number of samples 512, Hann windows are used with a shift of 2 samples (0.045 ms) between neighbor windows. 6 ms time resolution and 1.1 kHz frequency resolution are not sufficient to resolve the 100 Hz modulation. Only minimum and maximum frequencies are observed

8.3 Gabor Expansion

For the special case of rectangular windows with distance $\Delta t = 2d$[6]

$$W_n(t) = W_R(t - 2d\,n) = \chi_n(t) \tag{8.43}$$

$$\sum_n W_n(t) = 1 = \text{const} \tag{8.44}$$

we have

$$f(t) = \sum_n g_n(t) \tag{8.45}$$

$$= \sum_n \sum_{m=-\infty}^{\infty} \chi_n(t) \mathrm{e}^{\mathrm{i}\omega_m t} \hat{g}_{nm}. \tag{8.46}$$

This equation expands $f(t)$ and its Fourier transform as linear combinations of elementary "signals" which are located at t_n in time and ω_m in frequency

$$h_{n,m} = \chi_n \mathrm{e}^{\mathrm{i}\omega_m t}. \tag{8.47}$$

[6]For simplicity, we do not normalize the window here.

$$\tilde{h}_{n,m} = \frac{2d}{\sqrt{2\pi}} \mathrm{e}^{-\mathrm{i}t_n(\omega-\omega_m)} \operatorname{sinc} d(\omega - \omega_m) \tag{8.48}$$

$$f(t) = \sum_{nm} h_{n,m}\hat{g}_{n,m} \quad \tilde{f}(\omega) = \sum_{nm} \tilde{h}_{nm}\hat{g}_{nm}. \tag{8.49}$$

A similar expansion is obtained if we use rectangular windows in Fourier space with width and distance $\Delta\omega$ [80]

$$\tilde{h}_{nm}(\omega) = \chi_{\mathrm{m}} \mathrm{e}^{-\mathrm{i}t_n(\omega-\omega_m)}$$

$$h_{nm}(t) = \frac{2\Delta\omega}{\sqrt{2\pi}} \operatorname{sinc}((t - t_n)\Delta\omega)\mathrm{e}^{\mathrm{i}\omega_m t}$$

and sample the spectrum at times

$$t_n = n\frac{\pi}{\Delta\omega} \tag{8.50}$$

to obtain

$$\tilde{f}(\omega) = \sum \chi_m(\omega)\tilde{f}_m(\omega) = \sum_{n=-\infty}^{\infty} \chi_m(\omega)\mathrm{e}^{\mathrm{i}\omega t_n}\,\hat{f}_{nm} \tag{8.51}$$

$$\hat{f}_{nm} = \frac{1}{2\Delta\omega}\int_{\omega_m-\Delta\omega}^{\omega_m+\Delta\omega} \tilde{f}(\omega)\mathrm{e}^{\mathrm{i}\omega t_n} d\omega. \tag{8.52}$$

Gabor [79] discussed an expansion with Gaussian signals (Fig. 8.5). In general, however, the elementary signals are not orthogonal which makes the determination of the coefficients $a_{n,m}$ complicated. Bastiaans [80, 81] introduced another auxiliary set of elementary signals

$$\gamma_{n,m} = \gamma(t - n\Delta t)\mathrm{e}^{\mathrm{i}\omega_m t} \tag{8.53}$$

which are biorthogonal, i.e.

$$\int \gamma^*_{n',m'}(t)h_{n,m}(t)dt = \delta_{n,n'}\delta_{m,m'} \tag{8.54}$$

and allow the calculation of the Gabor expansion coefficients from a scalar product

$$\int \gamma^*_{n,m}(t)f(t)dt = \sum_{nm}\int a_{nm}\gamma^*_{n,m}h_{nm}(t)dt = a_{nm}. \tag{8.55}$$

Determination of γ for a given windowing function can be simplified by application of the Zak transform [82]. Discrete versions of the Gabor transform [83] are popular in signal, speech and image processing.

8.4 Wavelet Analysis

The STFT method uses constant frequency and time resolution. Therefore the lowest frequency of interest determines the minimum width of the window whereas at higher frequencies shorter time windows could be more appropriate to increase time resolution while keeping the relative uncertainty in frequency constant (Fig. 8.12). This is the basic idea of the wavelet transform. Whereas STFT uses wave packets of the form (8.21)

$$\Omega_{t_0,\omega}(t) = W(t - t_0)\mathrm{e}^{\mathrm{i}\omega(t-t_0)} \tag{8.56}$$

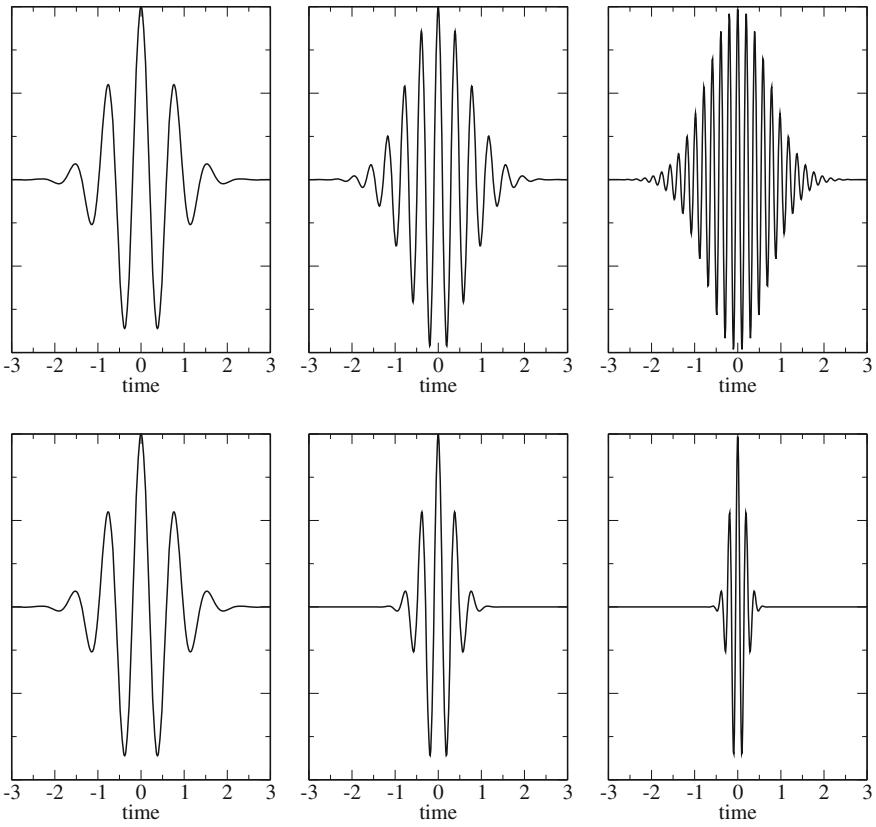


Fig. 8.12 (Morlet wavelets and STFT wave packets) ***Top*** STFT uses the same window for all frequencies ***Bottom*** wavelets use a variable window width to keep the form of the wave packet and the relative frequency resolution constant (only the real part is shown)

where only the oscillating part is scaled with frequency ω, wavelets scale the whole function like in

$$\Omega_{t_0,s}(t) = W\left(\frac{t-t_0}{s}\right) \mathrm{e}^{\mathrm{i}\omega_0(t-t_0)/s} \tag{8.57}$$

or, more generally

$$\Omega_{t_0,s}(t) = \frac{1}{\sqrt{|s|}}\Psi\left(\frac{t-t_0}{s}\right) \tag{8.58}$$

$$\begin{aligned}\tilde{\Omega}(\omega) &= \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} \mathrm{e}^{-\mathrm{i}\omega t}\frac{1}{\sqrt{|s|}}\Psi\left(\frac{t-t_0}{s}\right)dt \\ &= \frac{1}{\sqrt{2\pi}}\int_{-\infty \mathrm{sign}\, s}^{\infty \mathrm{sign}\, s} \mathrm{e}^{-\mathrm{i}\omega(st'+t_0)}\frac{1}{\sqrt{|s|}}\Psi(t')d(st'+t_0) \\ &= \sqrt{|s|}\mathrm{e}^{-\mathrm{i}\omega t_0}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}\mathrm{e}^{-\mathrm{i}\omega s t'}\Psi(t')dt' = \sqrt{|s|}\mathrm{e}^{-\mathrm{i}\omega t_0}\tilde{\Psi}(s\omega)\end{aligned} \tag{8.59}$$

where a whole family of wavelets is derived from the "mother wavelet" $\Psi(t)$ by shifting and rescaling. The prefactor has been introduced to keep the norm invariant

$$\begin{aligned}&\int |\Omega_{t_0,s}(t)|^2 dt = \frac{1}{|s|}\int |\Psi\left(\frac{t-t_0}{s}\right)|^2 dt \\ &= \frac{1}{|s|}\int_{-\infty \mathrm{sign}s}^{\infty \mathrm{sign}s} |\Psi(t')|^2 d(st'+t_0) = \int |\Psi(t')|^2 dt'.\end{aligned} \tag{8.60}$$

Closely related to the short time Fourier analysis is the Morlet (or Gabor) wavelet, which is also very useful in quantum physics [84]. It is defined as[7]

$$\Psi(t) = W_G(t)\mathrm{e}^{\mathrm{i}\omega_0 t} = \frac{1}{\pi^{1/4}\sqrt{d}}\exp\left\{-\frac{t^2}{2d^2}\right\}\mathrm{e}^{\mathrm{i}\omega_0 t} \tag{8.61}$$

$$\tilde{\Psi}(\omega) = \tilde{W}_G(\omega-\omega_0) = \frac{\sqrt{d}}{\pi^{1/4}}\exp\left\{-\frac{d^2}{2}(\omega-\omega_0)^2\right\}. \tag{8.62}$$

The similarity of a signal $f(t)$ to a wavelet with scale s centered at t_0 is measured by the correlation integral

$$C(t_0,s) = \int_{-\infty}^{\infty} f(t)\Omega^*_{t_0,s}(t)dt = \frac{1}{\sqrt{|s|}}\int_{-\infty}^{\infty}\Psi^*\left(\frac{t-t_0}{s}\right)f(t)dt \tag{8.63}$$

[7] The conventional normalization is $\int dt|\Psi(t)|^2 = 1$.

which becomes a product after Fourier transformation with respect to t_0

$$\begin{aligned}
\tilde{C}(\omega, s) &= \frac{1}{\sqrt{2\pi}} \quad C(t_0, s)\mathrm{e}^{-\mathrm{i}\omega t_0} dt_0 \\
&= \frac{1}{\sqrt{2\pi}} \quad dt_0 \mathrm{e}^{-\mathrm{i}\omega t_0} \quad dt \frac{1}{2\pi} \quad d\omega' \quad \overline{|s|}\tilde{\Psi}^*(s\omega')\mathrm{e}^{-\mathrm{i}\omega'(t-t_0)} \quad d\omega'' \tilde{f}(\omega'')\mathrm{e}^{\mathrm{i}\omega'' t} \\
&= \overline{2\pi|s|} \quad dt \quad d\omega' \tilde{\Psi}^*(s\omega')\mathrm{e}^{-\mathrm{i}\omega' t} \quad d\omega'' \tilde{f}(\omega'')\mathrm{e}^{\mathrm{i}\omega'' t} \delta(\omega - \omega')\delta(\omega' - \omega'') \\
&= \overline{2\pi|s|} \quad d\omega' \tilde{\Psi}^*(s\omega')\mathrm{e}^{-\mathrm{i}\omega' t} \tilde{f}(\omega')\mathrm{e}^{\mathrm{i}\omega' t} \\
&= \overline{2\pi|s|}\tilde{\Psi}^*(s\omega)\tilde{f}(\omega). \qquad (8.64)
\end{aligned}$$

For the Morlet wavelet this becomes

$$\begin{aligned}
\tilde{C}(\omega, s) &= \pi^{1/4} \quad \overline{2|s|d} \exp\left\{-\frac{d^2}{2}(s\omega - \omega_0)^2 \quad \tilde{f}(\omega)\right. \\
&= \pi^{1/4} \quad \overline{2|s|d} \exp\left\{-\frac{(sd)^2}{2} \quad \omega - \frac{\omega_0}{s}\right)^2 \quad \tilde{f}(\omega) \qquad (8.65)
\end{aligned}$$

i.e. $\tilde{C}(\omega, s)$ averages the spectrum $\tilde{f}$ over a range with a width of $\sigma_\omega = 1/sd$ around $\omega = \omega_0/s$ and a constant ratio

$$\frac{\sigma_\omega}{\omega} = \frac{1}{\omega_0 d}. \qquad (8.66)$$

8.5 Wavelet Synthesis

For data processing it is necessary to reconstruct the data from the wavelet coefficients $C(t_0 s)$. This can be achieved with the help of the integral[8]

$$\begin{aligned}
&\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{s^2} C(t_0, s)\Omega_{t_0,s}(t) dt_0 ds = \quad \frac{1}{s^2} dt_0 ds \frac{1}{\sqrt{|s|}} \Psi \quad \frac{t - t_0}{s} \quad C(t_0, s) \\
&= \frac{1}{\sqrt{2\pi}} \quad \frac{1}{s^2} ds \quad dt_0 \quad d\omega \mathrm{e}^{\mathrm{i}\omega(t-t_0)/s} \tilde{\Psi}(\omega) \quad d\omega' \mathrm{e}^{\mathrm{i}\omega' t_0} \tilde{\Psi}^*(s\omega)\tilde{f}(\omega) \\
&= \frac{1}{\sqrt{2\pi}} \quad \frac{1}{s^2} ds \quad dt_0 \quad s d\omega'' \mathrm{e}^{\mathrm{i}\omega''(t-t_0)} \tilde{\Psi}(\omega'' s) \quad d\omega' \mathrm{e}^{\mathrm{i}\omega' t_0} \tilde{\Psi}^*(s\omega')\tilde{f}(\omega') \\
&= \frac{1}{\sqrt{2\pi}} \quad \frac{1}{s} ds \quad d\omega'' \mathrm{e}^{\mathrm{i}\omega'' t} \tilde{\Psi}(\omega'' s) \quad d\omega' \tilde{\Psi}^*(s\omega')\tilde{f}(\omega') 2\pi\delta(\omega' - \omega'') \\
&= \sqrt{2\pi} \quad \frac{1}{s} ds \quad s d\omega'' \mathrm{e}^{\mathrm{i}\omega'' t} \tilde{\Psi}(\omega'' s)\Psi(s\omega'')\tilde{f}(\omega'') \\
&= \sqrt{2\pi} \quad \frac{1}{s} ds \quad d\omega'' \mathrm{e}^{\mathrm{i}\omega'' t} \tilde{\Psi}(\omega'' s)\tilde{\Psi}^*(s\omega'')\tilde{f}(\omega''). \qquad (8.67)
\end{aligned}$$

[8] A more rigorous treatment introducing the concept of frames in Hilbert space can be found in [85].

If the admissibility condition is fulfilled, which states that the integral

$$C_\Psi = \int_{-\infty}^{\infty} \frac{|\tilde{\Psi}(\omega)|^2}{\omega} d\omega < \infty \tag{8.68}$$

exists and is final, then

$$\int_{-\infty}^{\infty} \frac{1}{s} ds \tilde{\Psi}(\omega s)\tilde{\Psi}^*(\omega s) = \int_{-\infty}^{\infty} \frac{1}{\omega'} d\omega' \tilde{\Psi}(\omega')\tilde{\Psi}^*(\omega') = C_\Psi$$

and we obtain

$$f(t) = \frac{1}{2\pi C_\Psi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{s^2} C(t_0, s) \Omega_{t_0,s}(t) dt_0 ds.$$

The admissibility condition implies that $\tilde{\Psi}(0) = 0$ and thus $\int dt \Psi(t) = 0$. Hence, the Morlet wavelet (8.61) has to be modified[9]

$$\Psi(t) = \frac{1}{\pi^{1/4}\sqrt{d}} N_d \exp\left\{-\frac{t^2}{2d^2}\right\} \left[e^{i\omega_0 t} - \exp\left\{-\frac{\omega_0^2 d^2}{2}\right\}\right] \tag{8.69}$$

$$\tilde{\Psi}(\omega) = \frac{\sqrt{d}}{\pi^{1/4}} N_d \left[\exp\left\{-\frac{d^2}{2}(\omega - \omega_0)^2\right\} - \exp\left\{-\frac{d^2}{2}(\omega^2 + \omega_0^2)\right\}\right] \tag{8.70}$$

$$N_d = \left[\left(1 + \exp\left\{-\omega^2 d^2\right\} - 2\exp\left\{-\frac{3}{4}\omega^2 d^2\right\}\right)\right]^{-1/2}. \tag{8.71}$$

Another popular (continuous) wavelet is the "Mexican hat" (also known as Ricker wavelet or Marr wavelet) which is given by the normalized negative second derivative of a Gaussian (Fig. 8.13)

$$\Psi(t) = \frac{2}{\pi^{1/4}\sqrt{3d}} \left(1 - \frac{t^2}{d^2}\right) \exp\left\{-\frac{t^2}{2d^2}\right\} \tag{8.72}$$

$$\tilde{\Psi}(\omega) = \frac{2\sqrt{d}}{\pi^{1/4}\sqrt{3}} \omega^2 \exp\left\{-\frac{\omega^2 d^2}{2}\right\}. \tag{8.73}$$

Example: Wavelet Analysis of a Nonstationary Signal

The following example shows screen shots from Problem 8.2. The signal consists of two sweeps with linearly increasing frequency of the form

$$f_{1,2}(t) = \sin\left[\omega_{1,2} t + \frac{\alpha_{1,2}}{2} t^2\right] \tag{8.74}$$

[9] This correction is often neglected, if the width is large.

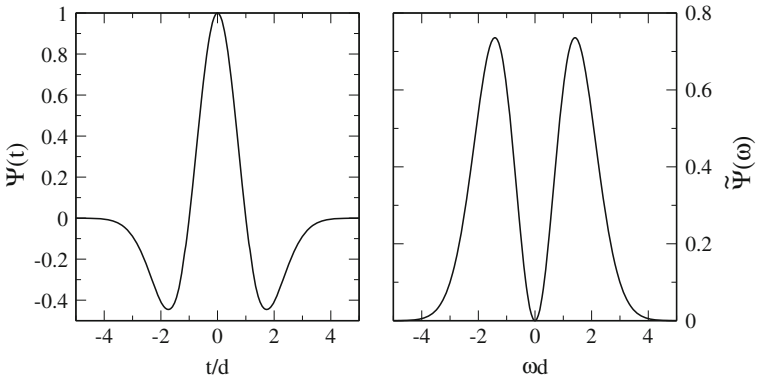


Fig. 8.13 (Mexican hat wavelet) ***Left*** The mexican hat wavelet is essentially the second derivative of a Gaussian. ***Right*** Its Fourier transform is a band pass filter around $\omega_{max} = \pm 2/d$

and another component which switches between a 5 kHz oscillation and the sum of a 300 Hz and a 20 kHz oscillation at a rate of 20 Hz

$$f_3(t) = \begin{array}{cc} \sin(\omega_{20kHz}t) + \sin(\omega_{300Hz}t) & \text{if } \sin(\omega_{20Hz}t) < 0 \\ \sin(\omega_{5kHz}t) & else. \end{array} \tag{8.75}$$

The signal is sampled with a rate of 44 kHz and analyzed with Morlet wavelets over 6 octaves (Fig. 8.14). The parameter d of the mother wavelet (8.61) determines frequency and time resolution. The frequency ω_0 of the mother wavelet is taken as the Nyquist frequency which is half the sampling rate. The convolution with the daughter wavelets

$$\Psi_{m,n}(t) = \frac{1}{\sqrt{s_m}} \Psi \quad \frac{t - t_n}{s_m} \tag{8.76}$$

is calculated at 400 times with a step size of 0.726 ms (corresponding to 32 samples)

$$t_n = t_0 + n\Delta t \tag{8.77}$$

and for 300 different values of the scaling parameter

$$s_m = 1.015^m. \tag{8.78}$$

In a logarithmic plot, the relative frequency uncertainty has the same size for all stationary signals.

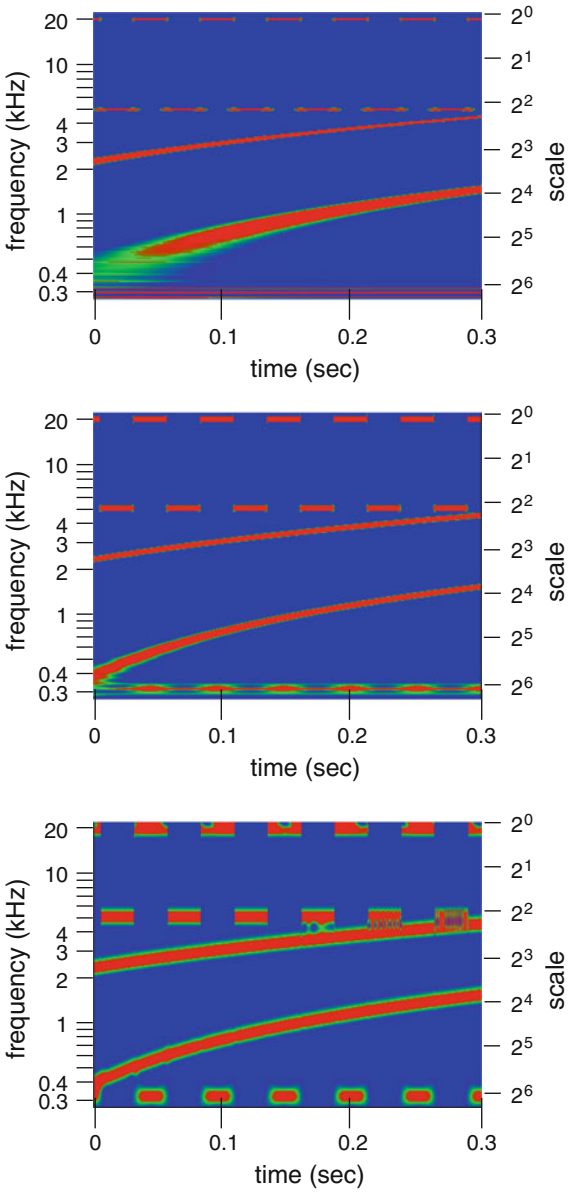


Fig. 8.14 (Wavelet analysis) ***Top*** for $d = 1$ ms the frequency resolution is high for the stationary parts of the signal. Time resolution is low. ***Middle*** for $d = 0.25$ ms the pulsating component at 300 Hz can be resolved but time resolution is still poor. ***Bottom*** For $d = 0.0625$ ms time resolution is sufficient to show all the modulations while frequency resolution is rather poor

8.6 Discrete Wavelet Transform and Multiresolution Analysis

The continuous wavelet transform is very redundant and time consuming. Multiresolution analysis provides a way to define a discrete set of orthogonal wavelets, for which the wavelet transform can be calculated very efficiently from a scalar product. A discrete wavelet transform uses discrete values of shift and scaling parameters

$$s = a^{-m} \quad t_0 = na^{-m}b \tag{8.79}$$

to define the daughter wavelets[10]

$$\Psi_{m,n}(t) = a^{m/2}\Psi\left(a^m t - nb\right) . \tag{8.80}$$

For integer a, in most cases $a = 2$, this equation defines wavelets of a multiresolution analysis (Fig. 8.15) where m corresponds to the resolution 2^m.

8.6.1 Scaling Function and Multiresolution Approximation

At the basic resolution 2^0 the function $f(t)$ is approximated as a linear combination

$$f(t) \approx f(t)^{(0)} = \sum_n f_{0,n}\Phi_{0,n}(t) \tag{8.81}$$

of a scaling function and its translations,

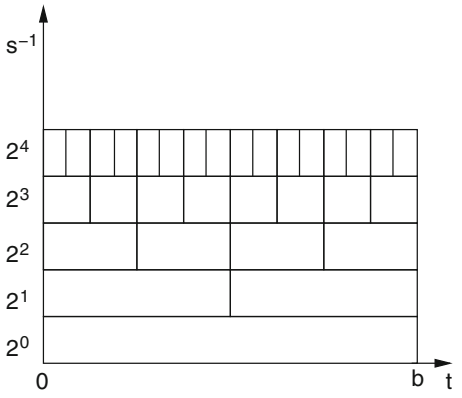


Fig. 8.15 (Multiresolution analysis) Data are analyzed with decreasing time window $\Delta t = b/2^m$

[10] Equation 8.76, in contrast, describes the continuous wavelet transform, which has to be discretized for numerical calculations.

$$\Phi_{0,n} = \Phi(t - nb) \quad , n = 0, \pm 1 \ldots \tag{8.82}$$

which is chosen [86, 87] such, that the $\Phi_{0,n}$ form an orthonormal basis of the space of linear combinations

$$V_0 = \text{span}\{\Phi_{0,n}, n = 0, \pm 1, \ldots\} \tag{8.83}$$

$$\int \Phi^*_{0,n}(t)\Phi_{0,n'}(t)dt = \delta_{n,n'}. \tag{8.84}$$

The best approximation is found by minimizing the norm

$$\begin{aligned} ||f(t) - \sum_n f_{0,n}\Phi_{0n}(t)||^2 &= \int (f^*(t) - \sum_n f^*_{0,n}\Phi^*_{0n}(t))(f(t) - \sum_{n'} f_{0,n'}\Phi_{0n'}(t))dt \\ &= \int |f(t)|^2 dt - \sum_{n'} f_{0n'} \int f^*(t)\Phi_{0n'}(t)dt - \sum_n f^*_{0n}(t) \int \Phi^*_{0n}(t) f(t)dt + \sum_n |f_{0n}|^2 \end{aligned} \tag{8.85}$$

hence by choosing

$$f_{0n} = \int \Phi^*_{0n}(t) f(t)dt \tag{8.86}$$

i.e., the orthogonal projection of $f(t)$ onto V_0. Approximation at the higher resolution 2^m similarly is given by linear combination

$$f(t) \approx f(t)^{(m)} = \sum_n f_{m,n}\Phi_{m,n}(t) \tag{8.87}$$

of the scaled functions

$$\Phi_{m,n} = 2^{m/2}\Phi(2^m t - nb) \tag{8.88}$$

which form an orthonormal basis for the space

$$V_m = \text{span}\{\Phi_{m,n}, n = 0, \pm 1, \ldots\} \tag{8.89}$$

since

$$\begin{aligned} &\int \Phi^*_{m,n}(t)\Phi_{m,n'}(t)dt = 2^m \int \Phi^*(2^m t - nb)\Phi(2^m t - n'b)dt \\ &= 2^m \int \Phi^*(t' - nb)\Phi(t' - n'b)\frac{dt'}{2^m} = \delta_{n,n'}. \end{aligned} \tag{8.90}$$

The sequence of spaces V_m is called a multiresolution approximation to the space of square integrable functions $L^2(\mathbb{R})$, if [86]

$$\text{(i)} \quad \cdots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \ldots \tag{8.91}$$

$$\text{(ii)} \quad \bigcup_{m=-\infty}^{\infty} V_m \quad \text{is dense in } L^2(\mathbb{R}) \tag{8.92}$$

$$\text{(iii)} \quad \bigcap_{m=-\infty}^{\infty} V_m = \{0\}. \tag{8.93}$$

Property (ii) has as a consequence, that the approximations $f^{(m)}(t)$ converge to $f(t)$ for large m. Hence, due to orthonormality

$$f^{(m)}(t) = \sum_n \Phi_{m,n}(t) \int_{-\infty}^{\infty} \Phi^*_{m,n}(t') f(t') dt' \to f(t) \tag{8.94}$$

and the projection operator onto V_m

$$P_m = \sum_n \Phi_{m,n}(t) \Phi^*_{m,n}(t') = 2^m \sum_n \Phi(2^m t - nb) \Phi^*(2^m t' - nb) \to 1 \tag{8.95}$$

converges to the unit operator. Now, with $a > 0$ choose the function

$$f_a(t) = \begin{cases} 1 \text{ if } -a \le x \le a \\ 0 \text{ else.} \end{cases} \tag{8.96}$$

Then,

$$\begin{aligned} (P_m f_{ab})(t) &= \sum_n \Phi(2^m t - nb) \int_{-a}^{a} 2^m dt' \Phi^*(2^m t' - nb) \\ &= \sum_n \Phi(2^m t - nb) \int_{-2^m a}^{2^m a} \Phi^*(t' - nb) dt'. \end{aligned} \tag{8.97}$$

For large a, the integrals become more and more independent on n, and

$$\int_{-\infty}^{\infty} \Phi^*(t') dt' \sum_n \Phi(2^m t - nb) \to 1. \tag{8.98}$$

Now we integrate the sum over one period $0 \le t \le 2^{-m} b$ and find

$$\int_0^{2^{-m} b} \sum_n \Phi(2^m t - nb) dt = \sum_n \int_{-nb}^{(1-n)b} \Phi(t)\, 2^{-m} dt = 2^{-m} \int_{-\infty}^{\infty} \Phi(t) dt \tag{8.99}$$

and therefore

$$\left(\int_{-\infty}^{\infty} \Phi^*(t')dt'\right)\left(\int_{-\infty}^{\infty} \Phi(t')dt'\right) = b \tag{8.100}$$

or

$$\left|\int_{-\infty}^{\infty} \Phi(t')dt'\right| = \sqrt{b} \tag{8.101}$$

as well as

$$|\tilde{\Phi}(0)| = \frac{1}{\sqrt{2\pi}}\left|\int_{-\infty}^{\infty} \Phi(t')dt'\right| = \sqrt{\frac{b}{2\pi}}. \tag{8.102}$$

Fourier transformation of (8.84) gives

$$\begin{aligned}
\delta_{nn'} &= \int \Phi^*(t-nb)\Phi(t-n'b)dt \\
&= \frac{1}{2\pi}\int dt \int \tilde{\Phi}^*(\omega)\mathrm{e}^{-\mathrm{i}\omega(t-nb)}d\omega \int \tilde{\Phi}(\omega')\mathrm{e}^{\mathrm{i}\omega'(t-n'b)}d\omega' \\
&= \int d\omega d\omega' \tilde{\Phi}^*(\omega)\tilde{\Phi}(\omega')\mathrm{e}^{\mathrm{i}(\omega n-\omega' n')b}\delta(\omega-\omega') = \int d\omega |\tilde{\Phi}(\omega)|^2 \mathrm{e}^{\mathrm{i}\omega(n-n')b} \\
&= \sum_{j=-\infty}^{\infty}\int_{2\pi j/b}^{2\pi(j+1)/b} d\omega |\tilde{\Phi}(\omega)|^2 \mathrm{e}^{\mathrm{i}\omega(n-n')b} = \int_0^{2\pi/b} d\omega \sum_{j=-\infty}^{\infty} |\tilde{\Phi}(\omega+2\pi j/b)|^2 \mathrm{e}^{-\mathrm{i}\omega\Delta nb} \\
&= \int_0^{2\pi/b} d\omega F(\omega)\mathrm{e}^{-\mathrm{i}\omega\Delta nb}.
\end{aligned} \tag{8.103}$$

$F(\omega)$ is periodic with period $\Omega_0 = 2\pi/b$ and can be represented as a Fourier sum

$$F(\omega) = \sum_{n=-\infty}^{\infty} F_n \mathrm{e}^{\mathrm{i}2\pi n\omega/\Omega_0} = \sum_{n=-\infty}^{\infty} F_n \mathrm{e}^{\mathrm{i}nb\omega} \tag{8.104}$$

where the Fourier coefficients

$$F_n = \frac{1}{\Omega_0}\int_0^{\Omega_0} F(\omega)\mathrm{e}^{-\mathrm{i}2\pi n\omega/\Omega_0} = \frac{b}{2\pi}\int_0^{2\pi/b} F(\omega)\mathrm{e}^{-\mathrm{i}nb\omega} \tag{8.105}$$

are found from comparison with (8.103)

$$F_n = \frac{b}{2\pi}\delta_{n,0}. \tag{8.106}$$

Finally, evaluation of the Fourier sum (8.104) gives

$$F(\omega) = \sum_j |\tilde{\Phi}(\omega + j\Omega_0)|^2 = \frac{1}{\Omega_0} \tag{8.107}$$

which is the equivalent of the orthonormality of Φ_{0n} in Fourier space.

Equation 8.91 implies that $\Phi_{m,n}$ can be represented as linear combination of the $\Phi_{m+1,n}$. Starting from

$$\Phi(t) = \Phi_{0,0}(t) = \sum_n h_n \Phi_{1,n}(t) = \sqrt{2} \sum_n h_n \Phi(2t - nb) \tag{8.108}$$

scaling and translation gives

$$\Phi_{m,n}(t) = \sum_{n'} h_{n'-2n} \Phi_{m+1,n'}(t). \tag{8.109}$$

Fourier transformation of (8.88) gives

$$\tilde{\Phi}_{m,n}(\omega) = \mathrm{e}^{-2n\pi \mathrm{i}\omega/\Omega_m} \tilde{\Phi}_{m,0}(\omega) = \frac{1}{\sqrt{2^m}} \mathrm{e}^{-2n\pi \mathrm{i}\omega/\Omega_m} \tilde{\Phi}(\omega/2^m) \tag{8.110}$$

$$\tilde{\Phi}_{m+1,n}(\omega) = \frac{1}{\sqrt{2}} \tilde{\Phi}_{mn}(\omega/2) \tag{8.111}$$

with

$$\Omega_m = 2^m \frac{2\pi}{b} = 2^m \Omega_0 \tag{8.112}$$

and (8.108) becomes

$$\tilde{\Phi}(\omega) = \sum_n \frac{h_n}{\sqrt{2}} \mathrm{e}^{-2n\pi \mathrm{i}\omega/\Omega_1} \tilde{\Phi}(\omega/2^m) = M_0(\omega/2) \tilde{\Phi}(\omega/2) \tag{8.113}$$

where

$$M_0(\omega/2) = \sum_n \frac{h_n}{\sqrt{2}} \mathrm{e}^{-2n\pi \mathrm{i}(\omega/2)/\Omega_0} \tag{8.114}$$

is Ω_0-periodic. Similarly, we find

$$\tilde{\Phi}_{m0}(\omega) = \sum_n h_n \tilde{\Phi}_{m+1n}(\omega) = \sum_n \frac{h_n}{\sqrt{2^{m+1}}} \mathrm{e}^{-2n\pi \mathrm{i}(\omega/2^{m+1})/\Omega_0} \tilde{\Phi}(\omega/2^{m+1})$$
$$= \frac{1}{\sqrt{2^m}} M_0(\omega/2^{m+1}) \tilde{\Phi}(\omega/2^{m+1}). \tag{8.115}$$

Equation 8.113 can be iterated to obtain

$$\tilde{\Phi}(\omega) = M_0(\omega/2)\tilde{\Phi}(\omega/2) = M_0(\omega/2)M_0(\omega/4)\tilde{\Phi}(\omega/4) = \ldots$$
$$= \prod_{j=1}^{\infty} M_0(\omega/2^j)\tilde{\Phi}(0) = \prod_{j=1}^{\infty} M_0(\omega/2^j)\sqrt{\frac{b}{2\pi}}. \tag{8.116}$$

This equation shows that knowledge of M_0 is sufficient to determine the scaling function (see also p. 182).

From the orthogonality condition (8.107) we obtain

$$\frac{1}{\Omega_0} = \sum_j |\tilde{\Phi}(\omega + j\Omega_0)|^2 = \sum_j \left| M_0\left(\omega/2 + j\frac{\Omega}{2}\right)\right|^2 \left|\tilde{\Phi}\left(\omega/2 + j\frac{\Omega}{2}\right)\right|^2$$
$$= \sum_j |M_0(\omega/2 + j\Omega_0)|^2 |\tilde{\Phi}(\omega/2 + j\Omega_0)|^2$$
$$+ \sum_j \left| M_0\left(\omega/2 + \left(j + \frac{1}{2}\right)\Omega_0\right)\right|^2 \left|\tilde{\Phi}\left(\omega/2 + \left(j + \frac{1}{2}\right)\Omega_0\right)\right|^2$$
$$= |M_0(\omega/2)|^2 \sum_j |\tilde{\Phi}(\omega/2 + j\Omega_0)|^2$$
$$+ \left| M_0\left(\omega/2 + \frac{\Omega_0}{2}\right)\right|^2 \sum_j |\tilde{\Phi}((\omega + \Omega_0)/2 + j\Omega_0)|^2$$
$$= \frac{1}{\Omega_0}\left[|M_0(\omega/2)|^2 + \left|M_0\left(\omega/2 + \frac{\Omega_0}{2}\right)\right|^2\right]. \tag{8.117}$$

Example: Rectangular Scaling Function

The simplest example of a scaling function is the rectangular function

$$\Phi(t) = \begin{cases} \frac{1}{\sqrt{b}} & \text{for } \left|t - \frac{b}{2}\right| \leq \frac{b}{2} \\ 0 & \text{else} \end{cases} \tag{8.118}$$

with the scaled and translated functions

$$\Phi_{0,n}(t) = \Phi(t - nb) = \begin{array}{ll} \frac{1}{\sqrt{b}} & \text{for } \left|t - \left(n + \frac{1}{2}\right. \right. b\right| \le \frac{b}{2} \\ 0 & \text{else} \end{array} \tag{8.119}$$

$$\Phi_{1,n}(t) = \sqrt{2}\Phi(2t - nb) = \begin{array}{ll} \frac{1}{\sqrt{b/2}} & \text{for } \left|t - \left(n + \frac{1}{2}\right. \right. \left.\left.\frac{b}{2}\right| \le \frac{b}{4} \\ 0 & \text{else} \end{array} \tag{8.120}$$

$$\vdots$$

$$\Phi_{m,n}(t) = \sqrt{2^m}\Phi(2^m t - nb) = \begin{array}{ll} \frac{1}{\sqrt{b/2^m}} & \text{for } \left|t - \left(n + \frac{1}{2}\right. \right. \left.\left.\frac{b}{2^m}\right| < \frac{b}{2^{m+1}} \\ 0 & \text{else} \end{array} . \tag{8.121}$$

Obviously, the $\Phi_{mn}(t)$ for fixed m are orthonormal and can be represented as linear combination

$$\Phi_{m,n}(t) = \frac{1}{\sqrt{2}}\Phi_{m+1,2n}(t) + \frac{1}{\sqrt{2}}\Phi_{m+1,2n+1}(t). \tag{8.122}$$

V_m is the space of functions which are piecewise constant on intervals $|t - (n + 1/2)b/2^m| < b/2^{m+1}$. Figure 8.16 shows the approximation of the parabola $f(t) = t^2$ by functions in $V_0 \ldots V_3$.

The Fourier transform of the scaling function is

$$\tilde{\Phi}(\omega) = \frac{1}{\sqrt{2\pi}} \frac{2\sin\frac{\omega b}{2}}{\omega\sqrt{b}} \mathrm{e}^{-\mathrm{i}\omega b/2} = \frac{\sqrt{b}}{\sqrt{2\pi}} \mathrm{sinc}\left(\frac{\omega b}{2}\right) \mathrm{e}^{-\mathrm{i}\omega b/2} \tag{8.123}$$

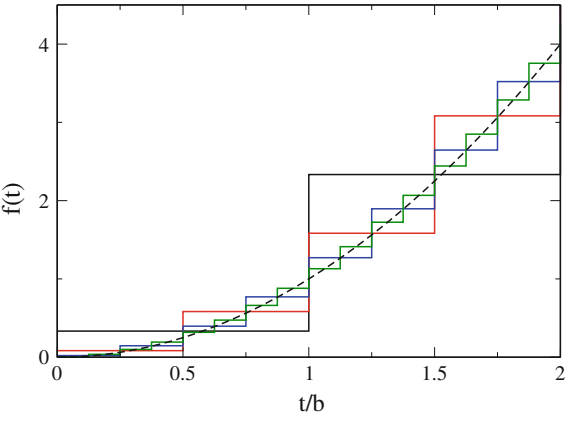


Fig. 8.16 (Approximation by piecewise constant functions) The parabola $f(t) = t^2$ (*dashed curve*) is approximated by linear combination of orthonormal rectangular functions (8.121) $f_m(t) = \sum_n \Phi_{mn}(t) \int_{-\infty}^{\infty} \Phi^*_{mn}(t) f(t) dt$ for $m = 0$ (*black*) $m = 1$ (*red*) $m = 2$ (*blue*) $m = 3$ (*green*)

and from

$$\tilde{\Phi}(\omega) = \frac{1}{\sqrt{2\pi}} \frac{2\left[2\sin\left(\frac{\omega b}{4}\right)\cos\left(\frac{\omega b}{4}\right)\right]}{\omega\sqrt{b}} \mathrm{e}^{-\mathrm{i}\omega b/2} = \tilde{\Phi}\left(\frac{\omega b}{2}\right)\cos\left(\frac{\omega b}{4}\right)\mathrm{e}^{-\mathrm{i}\omega b/4} \tag{8.124}$$

we find

$$M_0\left(\frac{\omega}{2}\right) = \cos\left(\frac{\omega b}{4}\right)\mathrm{e}^{-\mathrm{i}\omega b/4}. \tag{8.125}$$

8.6.2 Construction of an Orthonormal Wavelet Basis

The approximation $f^{(m+1)}(t)$ contains more details than $f^{(m)}(t)$. We would like to extract these details by dividing the space

$$V_{m+1} = V_m + W_m \tag{8.126}$$

into the sum of V_m and an orthogonal complement $W_m \perp V_m$. The approximation $f^{(m+1)}(t)$ then can be divide into the approximation $f^{(m)}$ plus the projection onto W_m, which provides the details. In the following we will construct an orthonormal basis of W_m in terms of wavelet functions $\Psi(t)$ which have the properties

$$\text{(i)} \quad \Psi \in V_{m+1} \tag{8.127}$$

or

$$\Psi = \sum_n C_n \Phi_{m+1,n} \tag{8.128}$$

and

$$\text{(ii)} \quad \Psi \perp V_m \tag{8.129}$$

or

$$\int_{-\infty}^{\infty} \Psi^*(t)\Phi_{mn}(t)dt = 0 \quad \forall n \tag{8.130}$$

which is equivalent to

$$\int_{-\infty}^{\infty} \tilde{\Psi}^*(\omega)\tilde{\Phi}_{mn}(\omega)d\omega = 0 \quad \forall n. \tag{8.131}$$

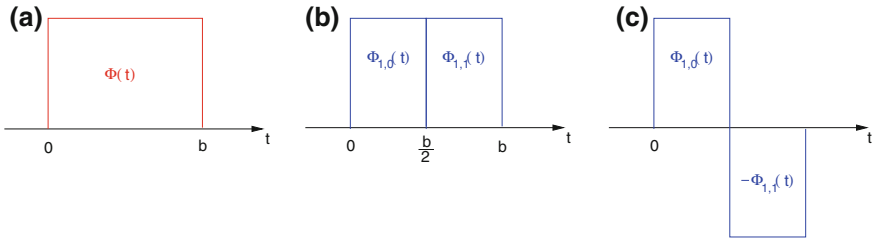


Fig. 8.17 (Haar wavelet) The rectangular scaling function (**a**) can be written as a linear combination of translated scaling functions at the next higher resolution (**b**). This is also the case for the wavelet function (**c**) which is orthogonal to the scaling function

Example: Haar Wavelet

With the rectangular scaling function

$$\Phi(t) = \begin{array}{c} 1 \text{ if } 0 \le x \le 1 \\ 0 \text{ else} \end{array} \tag{8.132}$$

the Haar wavelet [88] (Fig. 8.17)

$$\Psi(t) = \frac{1}{\sqrt{2}}\Phi_{1,0}(t) - \frac{1}{\sqrt{2}}\Phi_{1,0}(t) \tag{8.133}$$

is a linear combination of the translated functions $\Phi_{1,n}$ and orthogonal to all $\Phi_{0,n}$. The family of scaled and translated daughter wavelets

$$\Psi_{m,n}(t) = \frac{1}{\sqrt{2}}\Phi_{m,n}(t) - \frac{1}{\sqrt{2}}\Phi_{m,n+1}(t)$$

obeys

$$\Psi_{m,n} \in V_{m+1} \quad \Psi_{m,n} \perp V_m. \tag{8.134}$$

Orthogonality Condition

After Fourier transformation, (8.131) becomes

$$\begin{aligned} 0 = & \int \tilde{\Psi}^*(\omega) \mathrm{e}^{-n\mathrm{i}\omega 2\pi/\Omega_m} \tilde{\Phi}_{m0}(\omega) d\omega \\ = & \sum_j \int_{j\Omega_m}^{(j+1)\Omega_m} \tilde{\Psi}^*(\omega) \mathrm{e}^{-n\mathrm{i}\omega 2\pi/\Omega_m} \tilde{\Phi}_{m0}(\omega) d\omega \end{aligned}$$

$$
\begin{aligned}
&= \sum_j \int_0^{\Omega_m} \tilde{\Psi}^*(\omega + j\Omega_m) \mathrm{e}^{-n\mathrm{i}(\omega + j2\pi/t_m)t_m} \tilde{\Phi}_{m0}(\omega + j\Omega_m) d\omega \\
&= \int_0^{\Omega_m} \mathrm{e}^{-n\mathrm{i}\omega 2\pi/\Omega_m} \sum_j \tilde{\Psi}^*(\omega + j\Omega_m) \tilde{\Phi}_{m0}(\omega + j\Omega_m) d\omega \\
&= \int_0^{\Omega_m} \mathrm{e}^{-n\mathrm{i}\omega 2\pi/\Omega_m} G(\omega) d\omega = \Omega_m \hat{G}(t_n).
\end{aligned} \tag{8.135}
$$

This expression looks like the Fourier coefficient of an Ω_m-periodic function with the Fourier sum (7.5 with ω and t exchanged)

$$
G(\omega) = \sum_{n=-\infty}^{\infty} \mathrm{e}^{\mathrm{i}t_n\omega} \hat{G}(t_n) \text{ with } t_n = n\frac{2\pi}{\Omega_m}. \tag{8.136}
$$

But, since $\hat{G}(t) = 0$, we obtain the orthogonality condition

$$
\sum_j \tilde{\Psi}^*(\omega + j\Omega_m) \tilde{\Phi}_{m0}(\omega + j\Omega_m) = 0. \tag{8.137}
$$

Construction of the Wavelet

Now, Ψ and Φ_{m0} both are in V_{m+1}, therefore (8.113)

$$
\tilde{\Phi}_{m0} = M_{m0}(\omega/2^{m+1}) \tilde{\Phi}(\omega/2^{m+1}) \tag{8.138}
$$

$$
\tilde{\Psi} = M_{\Psi}(\omega/2^{m+1}) \tilde{\Phi}(\omega/2^{m+1}) \tag{8.139}
$$

where $M_{m,0}$ and M_{Ψ} are Ω_0-periodic.

Hence, from (8.137)

$$
\begin{aligned}
0 &= \sum_j M_{\Psi}^*((\omega + j\Omega_m)/2^{m+1}) M_{m0}((\omega + j\Omega_m)/2^{m+1}) |\tilde{\Phi}((\omega + j\Omega_m)/2^{m+1})|^2 \\
&= \sum_j M_{\Psi}^*(\omega/2^{m+1} + j\Omega_0/2) M_{m0}(\omega/2^{m+1} + j\Omega_0/2) |\tilde{\Phi}(\omega/2^{m+1} + j\Omega_0/2)|^2 \\
&= \sum_{j\,even} M_{\Psi}^*(\omega/2^{m+1}) M_{m0}(\omega/2^{m+1}2) |\tilde{\Phi}(\omega/2^{m+1} + \Omega_0 j/2)|^2 \\
&+ \sum_{j\,odd} M_{\Psi}^*\left(\omega/2^{m+1} + \frac{\Omega_0}{2}\right) M_{m0}\left(\omega/2^{m+1} + \frac{\Omega_0}{2}\right) |\tilde{\Phi}(\omega/2^{m+1} + \Omega_0 j/2)|^2 \\
&= M_{\Psi}^*(\omega/2^{m+1}) M_{m0}(\omega/2^{m+1}2) \sum_{j\,even} |\tilde{\Phi}(\omega/2^{m+1} + \Omega_0 j/2)|^2 \\
&+ M_{\Psi}^*\left(\omega/2^{m+1} + \frac{\Omega_0}{2}\right) M_{m0}\left(\omega/2^{m+1} + \frac{\Omega_0}{2}\right) \sum_{j\,odd} |\tilde{\Phi}(\omega/2^{m+1} + \Omega_0 j/2)|^2.
\end{aligned} \tag{8.140}
$$

From orthogonality of $\Phi_{m+1,n}$

$$\sum_j |\tilde{\Phi}(\omega/2^{m+1} + j\Omega_0)|^2 = \frac{1}{\Omega_0} \tag{8.141}$$

we see that both sums have the same value

$$\sum_{j\,even} |\tilde{\Phi}(\omega/2^{m+1} + \Omega_0 j/2)|^2 = \sum_k |\tilde{\Phi}(\omega/2^{m+1} + k\Omega_0)|^2 = \frac{1}{\Omega_0} \tag{8.142}$$

$$\sum_{j\,odd} |\tilde{\Phi}(\omega/2^{m+1} + \Omega_0 j/2)|^2 = \sum_k |\tilde{\Phi}(\omega/2^{m+1} + k\Omega_0 + \Omega_0/2)|^2 = \frac{1}{\Omega_0} \tag{8.143}$$

and therefore

$$M_\Psi^*(\omega/2^{m+1}) M_{m0}(\omega/2^{m+1}2) + M_\Psi^* \quad \omega/2^{m+1} + \frac{\Omega_0}{2} \quad M_{m0} \quad \omega/2^{m+1} + \frac{\Omega_0}{2} \quad = 0 \tag{8.144}$$

which can be satisfied by choosing [86]

$$M_\Psi(\omega/2^{m+1}) = M_{m0}^* \quad \omega/2^{m+1} + \frac{\Omega_0}{2} \quad \mathrm{e}^{\mathrm{i}\omega 2\pi/\Omega_{m+1}} \tag{8.145}$$

which implies

$$\begin{aligned} &M_\Psi \quad \omega/2^{m+1} + \frac{\Omega_0}{2} \quad = M_{m0}^*(\omega/2^{m+1} + \Omega_0)\mathrm{e}^{\mathrm{i}(\omega+\Omega_{m+1}/2)2\pi/\Omega_{m+1}} \\ &= -M_{m0}^*(\omega/2^{m+1})\mathrm{e}^{\mathrm{i}\omega 2\pi/\Omega_{m+1}}. \end{aligned} \tag{8.146}$$

Hence we obtain the solution

$$\begin{aligned} \tilde{\Psi}_m(\omega) &= \mathrm{e}^{\mathrm{i}\omega 2\pi/\Omega_{m+1}} M_{m0}^* \quad \omega/2^{m+1} + \frac{\Omega_0}{2} \quad \tilde{\Phi}(\omega/2^{m+1}) \\ &= \sum_{n'} \frac{h_{n'}^*}{\sqrt{2^{m+1}}} \mathrm{e}^{\mathrm{i}n'\pi} \mathrm{e}^{\mathrm{i}(n'+1)\omega 2\pi/\Omega_{m+1}} \tilde{\Phi}(\omega/2^{m+1}) \end{aligned} \tag{8.147}$$

which becomes in the time domain

$$\Psi_m(t) = \sum_{n'} \frac{h^*_{n'}}{\sqrt{2^{m+1}}}(-1)^{n'} \frac{1}{\sqrt{2\pi}} \int d\omega \mathrm{e}^{\mathrm{i}\omega t} \mathrm{e}^{\mathrm{i}(n'+1)\omega 2\pi/\Omega_{m+1}} \tilde{\Phi}(\omega/2^{m+1})$$

$$= \sum_{n'} (-1)^{n'} h^*_{n'} \sqrt{2^{m+1}} \Phi(2^{m+1}t + (n'+1)b)$$

$$= \sum_{n'} (-1)^{n'} h^*_{n'} \Phi_{m+1,-n'-1}(t) \tag{8.148}$$

$$= \sum_{n} (-1)^{-n-1} h^*_{-n-1} \Phi_{m+1,n}(t). \tag{8.149}$$

From the orthogonality condition (8.107) we obtain

$$\sum_j |\tilde{\Psi}_0(\omega + j\Omega_0)|^2 = \sum \left| M_{00}\left(\omega/2 + (j+1)\frac{\Omega_0}{2}\right)\right|^2 \left|\tilde{\Phi}\left(\omega/2 + j\frac{\Omega_0}{2}\right)\right|^2$$

$$= \sum \left| M_{00}\left(\omega/2 + (2j+1)\frac{\Omega_0}{2}\right)\right|^2 \left|\tilde{\Phi}\left(\omega/2 + 2j\frac{\Omega_0}{2}\right)\right|^2$$

$$+ \sum \left| M_{00}\left(\omega/2 + (2j+2)\frac{\Omega_0}{2}\right)\right|^2 \left|\tilde{\Phi}\left(\omega/2 + (2j+1)\frac{\Omega_0}{2}\right)\right|^2$$

$$= \left| M_{00}\left(\omega/2 + \frac{\Omega_0}{2}\right)\right|^2 \sum |\tilde{\Phi}(\omega/2 + j\Omega_0)|^2$$

$$+ |M_{00}(\omega/2)|^2 \sum |\tilde{\Phi}((\omega+\Omega_0)/2 + j\Omega_0)|^2$$

$$= \frac{1}{\Omega_0}\left(\left| M_{00}\left(\omega/2 + \frac{\Omega_0}{2}\right)\right|^2 + |M_{00}(\omega/2)|^2 \right). \tag{8.150}$$

But, since the scaling function obeys the orthonormality condition (8.107),

$$\frac{1}{\Omega_0} = \sum |\tilde{\Phi}(\omega + j\Omega_0)|^2 = \sum \left| M_{00}\left(\omega/2 + j\frac{\Omega}{2}\right)\right|^2 \left|\tilde{\Phi}\left(\omega/2 + j\frac{\Omega}{2}\right)\right|^2$$

$$= \sum |M_{00}(\omega/2 + j\Omega_0)|^2 |\tilde{\Phi}(\omega/2 + j\Omega_0)|^2$$

$$+ \sum \left| M_{00}\left(\omega/2 + \left(j + \frac{1}{2}\right)\Omega_0\right)\right|^2 \left|\tilde{\Phi}\left(\omega/2 + \left(j + \frac{1}{2}\right)\Omega_0\right)\right|^2$$

$$= |M_{00}(\omega/2)|^2 \sum |\tilde{\Phi}(\omega/2 + j\Omega_0)|^2$$

$$+ \left| M_{00}\left(\omega/2 + \frac{\Omega_0}{2}\right)\right|^2 \sum |\tilde{\Phi}((\omega+\Omega_0)/2 + j\Omega_0)|^2$$

$$= \frac{1}{\Omega_0}\left[|M_{00}(\omega/2)|^2 + \left| M_{00}\left(\omega/2 + \frac{\Omega_0}{2}\right)\right|^2 \right] \tag{8.151}$$

hence the wavelet also fulfills the orthonormality condition

$$\sum_j |\tilde{\Psi}_0(\omega + j\Omega_0)|^2 = \frac{1}{\Omega_0}. \tag{8.152}$$

Therefore the translated wavelet functions

$$\Psi_{mn}(t) = \Psi_m(t - n2^{-m}b) = \sum_{n'} (-1)^{-n'-1} h^*_{-n'-1} \sqrt{2^{m+1}} \Phi(2^{m+1}(t - n2^{-m}b) - n'b)$$
$$= \sum_{n'} (-1)^{-n'-1} h^*_{-n'-1} \Phi_{m+1,2n+n'}(t) \tag{8.153}$$

are orthonormal

$$\int \Psi^*_{mn}(t)\Psi_{mn'}(t)dt = \delta_{n.n'}. \tag{8.154}$$

Wavelets for different resolution m are orthogonal since they are by construction in orthogonal spaces. The $\Psi_{mn}(t)$ with $m, n = -\infty \ldots \infty$ provide an orthonormal basis of

$$L^2(\mathbb{R}) = \sum_{m=-\infty}^{\infty} W_m. \tag{8.155}$$

Alternatively, (8.155) is replaced by

$$L^2(\mathbb{R}) = V_0 + \sum_{m=0}^{\infty} W_m \tag{8.156}$$

which is more useful for practical applications with limited total observation time. According to (8.156), starting from a basic approximation in V_0, more and more details are added to obtain approximations with increasing accuracy.

Example: Meyer Wavelet

Meyer introduced the first non trivial wavelet (Fig. 8.18) which, in contrast to the Haar wavelet is differentiable [89, 90]. It was originally defined by its scaling function in Fourier space[11] (here, $b = 1$)

$$\tilde{\Phi}(\omega) = \begin{cases} \frac{1}{\sqrt{2\pi}} & \text{if } \omega \le \frac{2\pi}{3} \\ \frac{1}{\sqrt{2\pi}} \cos\left[\frac{\pi}{2}\left(\frac{3|\omega|}{2\pi} - 1\right)\right] & \text{if } \frac{2\pi}{3} < |\omega| < \frac{4\pi}{3} \\ 0 & \text{if } |\omega| > \frac{4\pi}{3} \end{cases} \tag{8.157}$$

from which the mother wavelet can be derived

[11] There are different variants of the Meyer wavelet in the literature.

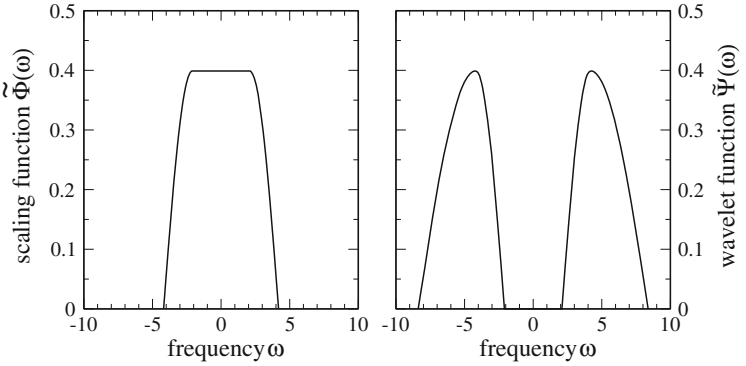


Fig. 8.18 (Meyer wavelet in frequency space) ***Left*** scaling function ***Right*** magnitude of the wavelet function

$$\tilde{\Psi}(\omega) = \begin{cases} \frac{1}{\sqrt{2\pi}} \sin\left(\frac{\pi}{2}\left(\frac{3|\omega|}{2\pi} - 1\right)\right) \mathrm{e}^{\mathrm{i}\omega/2} & \text{if } \frac{2\pi}{3} \leq |\omega| \leq \frac{4\pi}{3} \\ \frac{1}{\sqrt{2\pi}} \cos\left(\frac{\pi}{2}\left(\frac{3|\omega|}{4\pi} - 1\right)\right) \mathrm{e}^{\mathrm{i}\omega/2} & \text{if } \frac{4\pi}{3} \leq |\omega| \leq \frac{8\pi}{3} \\ 0 & \text{else.} \end{cases} \tag{8.158}$$

Explicit expressions in the time domain (Fig. 8.19) were given in 2015 [91]

$$\Phi(t) = \begin{cases} \frac{2}{3} + \frac{4}{3\pi} & \text{if } t = 0 \\ \frac{\sin\frac{2\pi}{3}t + \frac{4}{3}t\cos\frac{4\pi}{3}t}{\pi t - \frac{16\pi}{9}t^3} & \text{else} \end{cases} \tag{8.159}$$

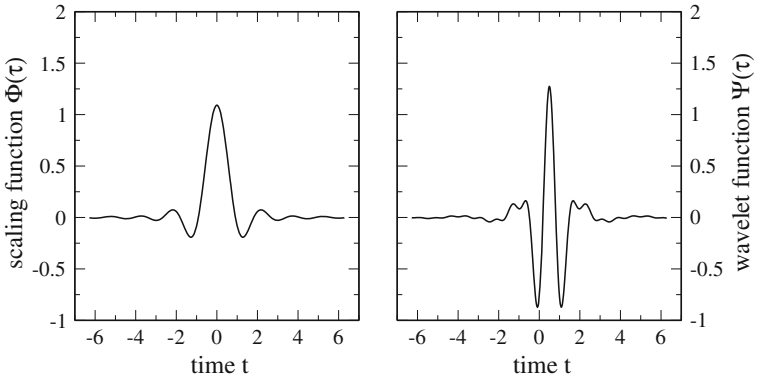


Fig. 8.19 (Meyer wavelet in the time domain) ***Left*** scaling function ***Right*** wavelet function

$$\Psi(t) = \frac{\frac{4}{3\pi}\left(t-\frac{1}{2}\right.\ \cos\ \frac{2\pi}{3}\left(t-\frac{1}{2}\right.\ -\frac{1}{\pi}\sin\ \frac{4\pi}{3}\left(t-\frac{1}{2}\right.}{\left(t-\frac{1}{2}\right.\ -\frac{16}{9}\left(t-\frac{1}{2}\right)^3} + \frac{\frac{8}{3\pi}\left(t-\frac{1}{2}\right.\ \cos\ \frac{8\pi}{3}\left(t-\frac{1}{2}\right.\ +\frac{1}{\pi}\sin\ \frac{4\pi}{3}\left(t-\frac{1}{2}\right.}{\left(t-\frac{1}{2}\right.\ -\frac{64}{9}\left(t-\frac{1}{2}\right)^3}. \tag{8.160}$$

8.7 Discrete Data and Fast Wavelet Transform

Mallet's algorithm [87] starts with function values

$$f_n = f(n\Delta t_s) \tag{8.161}$$

sampled at multiples of

$$\Delta t_s = 1/f_s = b/2^{m_{max}}. \tag{8.162}$$

We do not really approximate the function but from the series of sample values we construct the linear combination

$$\sum_n f_n \Phi_{m_{max},n}(t) \tag{8.163}$$

which is an element of

$$V_{m_{max}} = V_0 + \sum_{m=0}^{m_{max}-1} W_m \tag{8.164}$$

and can therefore be represented as a coarse approximation in V_0 and a series of details with increasing resolution

$$\sum_n f_n \Phi_{m_{max},n}(t) = \sum_n c_n \Phi_{0,n}(t) + \sum_{m=0}^{m_{max}-1} \sum_n d_{mn} \Psi_{mn}(t). \tag{8.165}$$

8.7.1 Recursive Wavelet Transformation

The approximation coefficients c_n and detail coefficients d_{mn} are determined recursively which avoids the calculation of scalar products.

Starting with

$$c_{m_{max},n} = f_n \tag{8.166}$$

the details are extracted by expanding

$$\sum_n c_{m_{max},n}\Phi_{m_{max},n}(t) = \sum_n c_{m_{max}-1,n}\Phi_{m_{max}-1,n}(t) + \sum_n d_{m_{max}-1,n}\Psi_{m_{max}-1,n}(t). \tag{8.167}$$

Due to orthogonality, the coefficients at the next lower resolution can be determined from

$$c_{m_{max}-1,n'} = \sum_n c_{m_{max},n} < \Phi_{m_{max}-1,n'}|\Phi_{m_{max},n} >= \sum_n c_{m_{max},n}h^*_{n-2n'}$$
$$= \sum_n c_{m_{max},n+2n'}h^*_n \tag{8.168}$$

$$d_{m_{max}-1,n'} = \sum_n c_{m_{max},n} < \Psi_{m_{max}-1,n'}|\Phi_{m_{max},n} >= \sum_n c_{m_{max},n}(-1)^{n-1}h_{2n'-n-1} \tag{8.169}$$

which can be written as

$$d_{m_{max}-1,n'} = \sum_n c_{m_{max},n}g^*_{n-2n'} = \sum_n c_{m_{max},n+2n'}g^*_n \quad \text{with } g^*_n = (-1)^{n-1}h_{-n-1}. \tag{8.170}$$

Iterating this recursion allows the calculation of the wavelet coefficients even without explicit knowledge of the scaling and wavelet functions. Equations (8.168) and (8.170) have the form of discrete digital filter functions with subsequent downsampling by a factor of two (dropping samples with odd n').[12] This can be seen by defining the down sampled coefficients

$$c^{\downarrow}_{n'/2} = \sum_n c_n h^*_{n-n'} \tag{8.171}$$

$$d^{\downarrow}_{n'/2} = \sum c_n(-1)^{n-1}h_{n'-n-1} \tag{8.172}$$

and applying the z-transform to (8.168) and (8.170). For the approximation filter we obtain

[12] For the more general class of bi-orthogonal wavelets, a different filter pair is used for reconstruction.

$$f_c(z) = \sum_{n'=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} h_n^* c_{n+n'} z^{-n'}$$

$$= \sum_n h_n^* z^n \sum_{n'} c_{n+n'} z^{-n'-n} = \sum_n h_n z^{-n} \sum_{n'}^{*} c_{n'} z^{-n'} = h^*(z)\, c(z) \tag{8.173}$$

hence in frequency space the signal is multiplied with the filter function

$$h(\mathrm{e}^{\mathrm{i}\omega\Delta t}) = \sum_n h_n \mathrm{e}^{-n\mathrm{i}\omega\Delta t} = \sqrt{2} M_0(\omega). \tag{8.174}$$

Similar we obtain for the detail filter

$$f_d(z) = \sum_{nn'} c_n (-1)^{n-1} h_{n'-n-1} z^{-n'} = \sum_{nn'} c_n z^{-n} (-1)^{n-1} h_{n'-n-1} z^{n-n'}. \tag{8.175}$$

Since only even values of n' are relevant, we may change the sign by $(-1)^{n'}$ to obtain

$$\sum c_n z^{-n} (-1)^{n'-n-1} h_{n'-n-1} z^{n-n'} = z^* h(-z)\, c(z) = g^*(z)\, c(z) \tag{8.176}$$

where

$$g(z) = \sum_n (-1)^{n-1} h_{-n-1}^* z^{-n} = \sum_n (-1)^{-n-2} h_n^* z^{n+1} = z \sum_n h_n^* (-z)^n$$

$$= z \left(\sum_n h_n (-z)^{-n} \right)^* = z h^*(-z). \tag{8.177}$$

8.7.2 Example: Haar Wavelet

For the Haar wavelet with[13]

$$h_0 = h_1 = \frac{1}{\sqrt{2}} \quad h_n = 0 \text{ else} \tag{8.178}$$

$$g_{-1} = \frac{1}{\sqrt{2}} \quad g_{-2} = -\frac{1}{\sqrt{2}} \quad g_n = 0 \text{ else} \tag{8.179}$$

[13]The standard form of the Haar wavelet with $g_0 = 1/\sqrt{2}$, $g_1 = -1/\sqrt{2}$ differs from (8.179) by a shift and time reversal. The resulting wavelet basis, however, is the same.

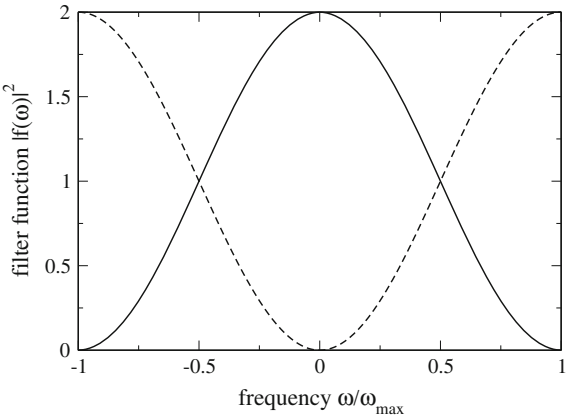


Fig. 8.20 Haar filter pair

we obtain the filter functions

$$h(z) = \frac{1}{\sqrt{2}}\left(1 + \frac{1}{z}\right) \quad g(z) = \frac{1}{\sqrt{2}}(z - z^2). \tag{8.180}$$

On the unit circle,

$$|h(\mathrm{e}^{\mathrm{i}\omega\Delta t})|^2 = 1 + \cos\omega\Delta t \tag{8.181}$$

$$|g(\mathrm{e}^{\mathrm{i}\omega\Delta t})|^2 = 1 - \cos\omega\Delta t \tag{8.182}$$

which describes a low and a high pass forming a so called quadrature mirror filter pair (Fig. 8.20) [92].

8.7.3 Signal Reconstruction

The wavelet transformation can be inverted using the expansion

$$\sum_n c_{m,n}\Phi_{m,n}(t) = \sum_n c_{m-1,n}\Phi_{m-1,n}(t) + \sum_n d_{m-1,n}\Psi_{m-1,n}(t) \tag{8.183}$$

where the coefficients at the higher level of approximation are obtained from

$$\begin{aligned} c_{m,n'} &= \sum_n c_{m-1,n} < \Phi_{m,n'}|\Phi_{m-1,n} > + \sum_n d_{m-1,n} < \Phi_{m,n'}|\Psi_{m-1,n} > \\ &= \sum_n c_{m-1,n} h_{n'-2n} + \sum_n d_{m-1,n}(-1)^{n'-1} h^*_{2n-n'-1} \\ &= \sum_n c_{m-1,n} h_{n'-2n} + \sum_n d_{m-1,n}(-1)^{n'-1} h^*_{2n-n'-1}. \end{aligned} \tag{8.184}$$

This can be formulated as upsampling and subsequent filtering. Formally, we insert zeros and define the up sampled coefficients

$$c^{\uparrow}_{2n} = c_{m-1,n} \quad c^{\uparrow}_{2n+1} = 0 \tag{8.185}$$

$$d^{\uparrow}_{2n} = d_{m-1,n} \quad d^{\uparrow}_{2n+1} = 0. \tag{8.186}$$

Then,

$$\sum_n c_{m-1,n} h_{n'-2n} = \sum_n c^{\uparrow}_{2n} h_{n'-2n} = \sum_n c^{\uparrow}_n h_{n'-n} \tag{8.187}$$

$$\sum_n d_{m-1,n} (-1)^{n'-1} h^*_{2n-n'-1} = (-1)^{n'-1} \sum_n d^{\uparrow}_{2n} h^*_{2n-n'-1}$$

$$= (-1)^{n'-1} \sum_n d^{\uparrow}_n h^*_{n-n'-1} = \sum (-1)^n d_n g_{n'-n} \tag{8.188}$$

where due to (8.186) the alternating sign can be omitted. Z-transformation then gives

$$\sum_{n,n'} c_n h_{n'-n} z^{-n'} = \sum_{n,n'} c_n z^{-n} h_{n'-n} z^{n-n'} = h(z)\, c(z) \tag{8.189}$$

$$\sum_{nn'} d_n g_{n'-n} = g(z) d(z) = z h^*(-z)\, d(z). \tag{8.190}$$

8.7.4 Example: Analysis with Compactly Supported Wavelets

Wavelet analysis has become quite popular for processing of audio and image data. In Problem 8.3 we use Daubechies wavelets [93] to analyze a complex audio signal consisting of a mixture of short tones, sweeps and noise (Figs. 8.23, 8.24). Daubechies satisfies (8.117) by taking

$$M_0(\omega/2) = \frac{1}{2}\left(1 + \mathrm{e}^{-\mathrm{i}\omega/2}\right)^N Q(\mathrm{e}^{-\mathrm{i}\omega/2}) \tag{8.191}$$

with a trigonometric polynomial Q. This leads to a class of compactly supported orthonormal wavelet bases, which for N = 1 include the Haar wavelet as the simplest member. For N = 2,

$$M_0(\omega/2) = \frac{1}{2}\left(1 + \mathrm{e}^{-\mathrm{i}\omega/2}\right)^2 \frac{1}{2}\left(1 + \sqrt{3} + \left(1 - \sqrt{3}\right)\mathrm{e}^{-\mathrm{i}\omega/2}\right) \tag{8.192}$$

$$= \frac{1}{8}\left[\left(1+\sqrt{3}\right) + \left(3+\sqrt{3}\right)\mathrm{e}^{-\mathrm{i}\omega/2} + \left(3-\sqrt{3}\right)\mathrm{e}^{-2\mathrm{i}\omega/2} + \left(1-\sqrt{3}\right)\mathrm{e}^{-3\mathrm{i}\omega/2}\right] \tag{8.193}$$

with the four nonzero scaling parameters

$$h_0 = \frac{\sqrt{2}}{8}(1+\sqrt{3}) \approx 0.48296 \tag{8.194}$$

$$h_1 = \frac{\sqrt{2}}{8}(3+\sqrt{3}) \approx 0.83652 \tag{8.195}$$

$$h_2 = \frac{\sqrt{2}}{8}(3-\sqrt{3}) \approx 0.22414 \tag{8.196}$$

$$h_3 = \frac{\sqrt{2}}{8}(1-\sqrt{3}) \approx -0.12941. \tag{8.197}$$

This defines the wavelet basis which is known as Daubechies 2. There are no analytic expressions for the scaling and wavelet functions available. They can be calculated numerically from the infinite product (8.116) or a corresponding (infinitely) nested convolution in real space. Figures 8.21 and 8.22 show the fast convergence.

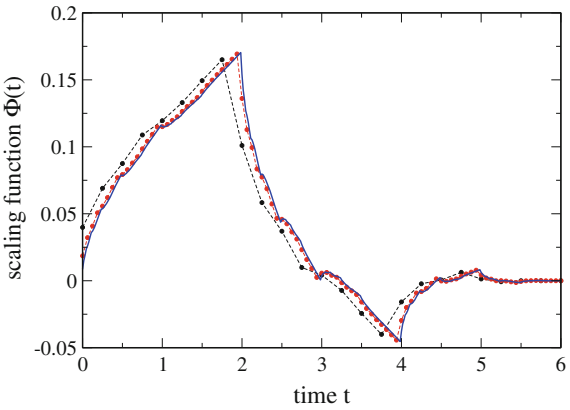


Fig. 8.21 (Daubechies 2 scaling function) The scaling function is calculated numerically in the time domain from the Fourier transform of (8.116) with a finite number of factors. The *blue curve* shows the result for $j_{max} = 7$, *red dots* show results for $j_{max} = 5$, *black dots* for $j_{max} = 3$. Delta functions are replaced by rectangular functions of equal area

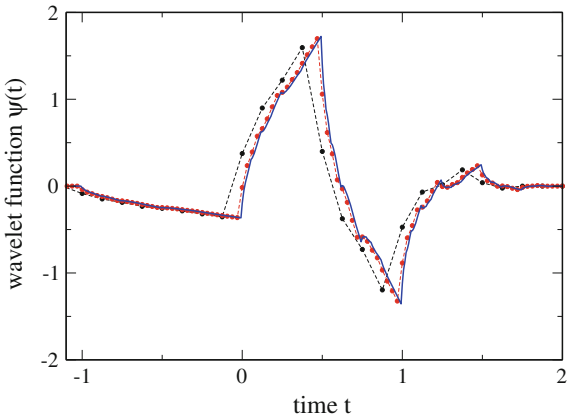


Fig. 8.22 (Daubechies 2 wavelet function) The wavelet function is calculated numerically in the time domain from the Fourier transform of (8.116) and (8.147) with a finite number of factors. The *blue curve* shows the result for $j_{max} = 7$, *red dots* show results for $j_{max} = 5$, *black dots* for $j_{max} = 3$. Delta functions are replaced by rectangular functions of equal area

Problems

Problem 8.1 Short Time Fourier Transformation

In this computer experiment STFT analysis of a frequency modulated signal

$$f(t) = \sin \Phi(t) = \sin \quad \omega_0 t + \frac{a\omega_0}{\omega_1}(1 - \cos \omega_1 t) \tag{8.198}$$

with a momentaneous frequency of

$$\omega(t) = \frac{\partial \Phi}{\partial t} = \omega_0(1 + a \sin \omega_1 t) \tag{8.199}$$

is performed and shown as a spectrogram (Figs. 8.10, 8.11). Sampling frequency is 44100 Hz, number of samples 512.

You can vary the carrier frequency ω_0, modulation frequency ω_1 and depth a as well as the distance between the windows. Study time and frequency resolution

Problem 8.2 Wavelet Analysis of a Nonstationary Signal

In this computer experiment, a complex signal is analyzed with Morlet wavelets over 6 octaves (Fig. 8.14). The signal is sampled with a rate of 44 kHz. The parameter d of the mother wavelet (8.61) determines frequency and time resolution. The frequency ω_0 of the mother wavelet is taken as the Nyquist frequency which is half the sampling rate. The convolution with the daughter wavelets (8.76) is calculated at 400 times with a step size of 0.726 ms (corresponding to 32 samples)

$$t_n = t_0 + n\Delta t \tag{8.200}$$

and for 300 different values of the scaling parameter

$$s_m = 1.015^m. \tag{8.201}$$

The signal consists of two sweeps with linearly increasing frequency of the form

$$f_{1,2}(t) = \sin\left[\omega_{1,2}t + \frac{\alpha_{1,2}}{2}t^2\right] \tag{8.202}$$

and another component which switches between a 5 kHz oscillation and the sum of a 300 Hz and a 20 kHz oscillation at a rate of 20 Hz

$$f_3(t) = \begin{cases} \sin(\omega_{20kHz}t) + \sin(\omega_{300Hz}t) & \text{if } \sin(\omega_{20Hz}t) < 0 \\ \sin(\omega_{5kHz}t) & else. \end{cases} \tag{8.203}$$

Study time and frequency resolution as a function of d

Problem 8.3 Discrete Wavelet Transformation

In this computer experiment the discrete wavelet transformation is applied to a complex audio signal. You can switch on and off different components like sweeps, dial tones and noise. The wavelet coefficients and the reconstructed signals are shown. (see Figs. 8.23, 8.24).

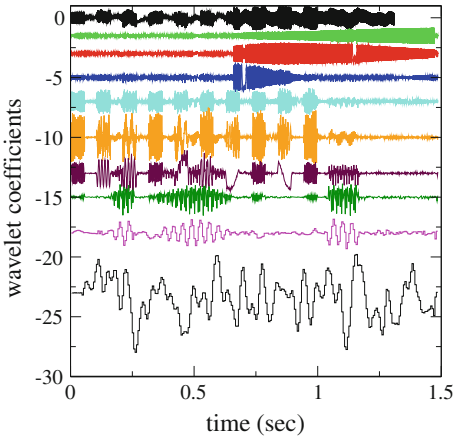


Fig. 8.23 (Wavelet coefficients of a complex audio signal) From ***Top*** to ***Bottom*** The *black curve* shows the input signal. The finest details in *light green*, *red* and *blue* correspond to a high frequency sweep from 5000–15000 Hz starting at 0.7 s plus some time dependent noise. *Cyan*, *orange* and *maroon* represent a sequence of dial tones around 1000 Hz, *dark green* and *magenta* show the signature of several rectangular 100 Hz bursts with many harmonics. The *black curve* at the ***Bottom*** shows the coefficients of the coarse approximation, which essentially describes random low frequency fluctuations. The *curves* are vertically shifted relative to each other

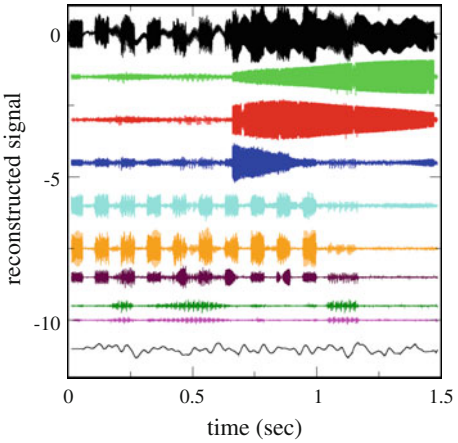


Fig. 8.24 (Wavelet reconstruction) The different contributions to the signal are reconstructed from the wavelet coefficients. Color code as in Fig. 8.23. The original signal (***Top*** *black curve*) is exactly the sum of the coarse approximation (***Bottom*** *black curve*) and all details (*colored curves*). The *curves* are vertically shifted relative to each other

Chapter 9
Random Numbers and Monte-Carlo Methods

Many-body problems often involve the calculation of integrals of very high dimension which can not be treated by standard methods. For the calculation of thermodynamic averages Monte Carlo methods [94–97] *are very useful which sample the integration volume at randomly chosen points. In this chapter we discuss algorithms for the generation of pseudo-random numbers with given probability distribution which are essential for all Monte Carlo methods. We show how the efficiency of Monte Carlo integration can be improved by sampling preferentially the important configurations. Finally the famous Metropolis algorithm is applied to classical many-particle systems and nonlinear optimization problems.*

9.1 Some Basic Statistics

In the following we discuss some important concepts which are used to analyze experimental data sets [98]. Repeated measurements of some observable usually give slightly different results due to fluctuations of the observable in time and errors of the measurement process. The distribution of the measured data is described by a probability distribution, which in many cases approximates a simple mathematical form like the Gaussian normal distribution. The moments of the probability density give important information about the statistical properties, especially the mean and the standard deviation of the distribution. If the errors of different measurements are uncorrelated, the average value of a larger number of measurements is a good approximation to the "exact" value.

9.1.1 Probability Density and Cumulative Probability Distribution

Consider an observable ξ, which is measured in a real or a computer experiment. Repeated measurements give a statistical distribution of values.

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_9

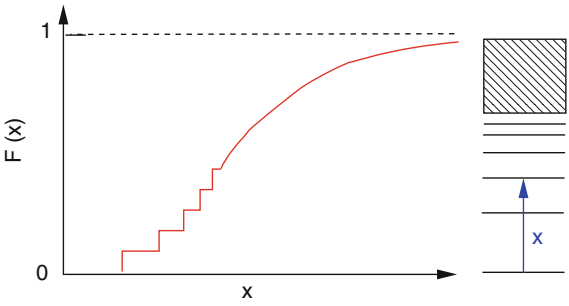


Fig. 9.1 (Cumulative probability distribution of transition energies) The figure shows schematically the distribution of transition energies for an atom which has a discrete and a continuous part

The cumulative probability distribution (Fig. 9.1) is given by the function

$$F(x) = P\{\xi \leq x\} \tag{9.1}$$

and has the following properties:

- $F(x)$ is monotonously increasing
- $F(-\infty) = 0, F(\infty) = 1$
- $F(x)$ can be discontinuous (if there are discrete values of ξ)

The probability to measure a value in the interval $x_1 < \xi \leq x_2$ is

$$P(x_1 < \xi \leq x_2) = F(x_2) - F(x_1). \tag{9.2}$$

The height of a jump gives the probability of a discrete value

$$P(\xi = x_0) = F(x_0 + 0) - F(x_0 - 0). \tag{9.3}$$

In regions where $F(x)$ is continuous, the probability density can be defined as

$$f(x_0) = F'(x_0) = \lim_{\Delta x \to 0} \frac{1}{\Delta x} P(x_0 < \xi \leq x_0 + \Delta x). \tag{9.4}$$

9.1.2 Histogram

From an experiment $F(x)$ cannot be determined directly. Instead a finite number N of values x_i are measured. By

$$Z_N(x)$$

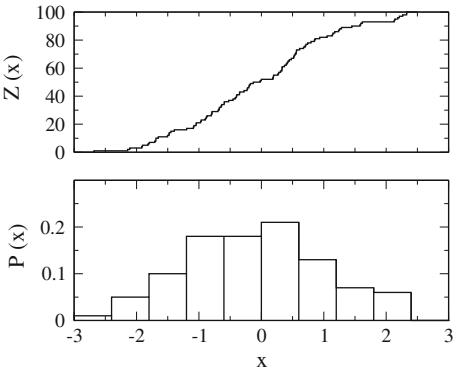


Fig. 9.2 (Histogram) The cumulative distribution of 100 Gaussian random numbers is shown together with a histogram with bin width $\Delta x = 0.6$

we denote the number of measurements with $x_i \leq x$. The cumulative probability distribution is the limit

$$F(x) = \lim_{N\to\infty} \frac{1}{N} Z_N(x). \tag{9.5}$$

A histogram (Fig. 9.2) counts the number of measured values which are in the interval $x_i < x \leq x_{i+1}$:

$$\frac{1}{N}(Z_N(x_{i+1}) - Z_N(x_i)) \approx F(x_{i+1}) - F(x_i) = P(x_i < \xi \leq x_{i+1}). \tag{9.6}$$

Contrary to $Z_N(x)$ itself, the histogram depends on the choice of the intervals.

9.1.3 Expectation Values and Moments

The expectation value of the random variable ξ is defined by

$$E[\xi] = \int_{-\infty}^{\infty} x dF(x) = \lim_{a\to-\infty, b\to\infty} \int_a^b x dF(x) \tag{9.7}$$

with the Riemann-Stieltjes-Integral [99]

$$\int_a^b x dF(x) = \lim_{N\to\infty} \sum_{i=1}^{N} x_i (F(x_i) - F(x_{i-1}))|_{x_i = a + \frac{b-a}{N} i}. \tag{9.8}$$

Higher moments are defined as

$$E[\xi^k] = \int_{-\infty}^{\infty} x^k dF(x) \tag{9.9}$$

if these integrals exist. Most important are the expectation value

$$\overline{x} = E[\xi] \tag{9.10}$$

and the variance, which results from the first two moments

$$\sigma^2 = \int_{-\infty}^{\infty} (x - \overline{x})^2 dF = \int x^2 dF + \int \overline{x}^2 dF - 2\overline{x} \int x dF$$
$$= E[\xi^2] - (E[\xi])^2. \tag{9.11}$$

The standard deviation σ is a measure of the width of the distribution. The expectation value of a function $\varphi(x)$ is defined by

$$E[\varphi(x)] = \int_{-\infty}^{\infty} \varphi(x) dF(x). \tag{9.12}$$

For continuous $F(x)$ we have with $\mathrm{d}F(x) = f(x)\mathrm{d}x$ the ordinary integral

$$E[\xi^k] = \int_{-\infty}^{\infty} x^k f(x) dx \tag{9.13}$$

$$E[\varphi(x)] = \int_{-\infty}^{\infty} \varphi(x) f(x) dx \tag{9.14}$$

whereas for a pure step function $F(x)$ (only discrete values x_i are observed with probabilities $p(x_i) = F(x_i + 0) - F(x_i - 0)$)

$$E[\xi^k] = \sum x_i^k p(x_i) \tag{9.15}$$

$$E[\varphi(x)] = \sum \varphi(x_i) p(x_i). \tag{9.16}$$

9.1.4 Example: Fair Die

When a six-sided fair die is rolled, each of its sides shows up with the same probability of 1/6. The cumulative probability distribution $F(x)$ is a pure step function (Fig. 9.3) and

$$\overline{x} = \int_{-\infty}^{\infty} x dF = \sum_{i=1}^{6} x_i (F(x_i + 0) - F(x_i - 0)) = \frac{1}{6} \sum_{i=1}^{6} x_i = \frac{21}{6} = 3.5 \tag{9.17}$$

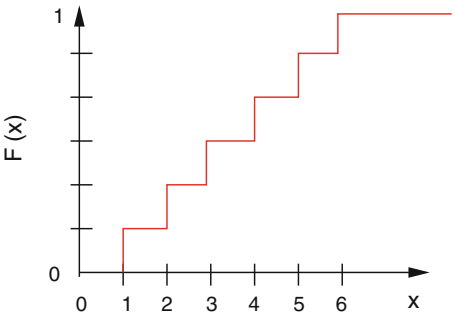


Fig. 9.3 Cumulative probability distribution of a fair die

$$\overline{x^2} = \sum_{i=1}^{6} x_i^2 (F(x_i + 0) - F(x_i - 0)) = \frac{1}{6} \sum_{i=1}^{6} x_i^2 = \frac{91}{6} = 15.1666\cdots \tag{9.18}$$

$$\sigma = \sqrt{\overline{x^2} - \overline{x}^2} = 2.9. \tag{9.19}$$

9.1.5 Normal Distribution

The Gaussian normal distribution is defined by the cumulative probability distribution

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \mathrm{e}^{-t^2/2} dt \tag{9.20}$$

and the probability density

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-x^2/2} \tag{9.21}$$

with the properties

$$\int_{-\infty}^{\infty} \varphi(x) dx = \Phi(\infty) = 1 \tag{9.22}$$

$$\overline{x} = \int_{-\infty}^{\infty} x \varphi(x) dx = 0 \tag{9.23}$$

$$\sigma^2 = \overline{x^2} = \int_{-\infty}^{\infty} x^2 \varphi(x) dx = 1. \tag{9.24}$$

Since $\Phi(0) = \frac{1}{2}$ and with the definition

$$\Phi_0(x) = \frac{1}{\sqrt{2\pi}} \int_0^x \mathrm{e}^{-t^2/2} dt \tag{9.25}$$

we have

$$\Phi(x) = \frac{1}{2} + \Phi_0(x) \tag{9.26}$$

which can be expressed in terms of the error function[1]

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \mathrm{e}^{-t^2 dt} dt = 2\Phi_0(\sqrt{2}x) \tag{9.27}$$

as

$$\Phi_0(x) = \frac{1}{2}\mathrm{erf}(\frac{x}{\sqrt{2}}). \tag{9.28}$$

A general Gaussian distribution with mean value $\overline{x}$ and standard deviation σ has the probability distribution

$$\varphi_{\overline{x},\sigma} = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x'-\overline{x})^2}{2\sigma^2}\right) \tag{9.29}$$

and the cumulative distribution

$$\Phi_{\overline{x},\sigma}(x) = \Phi\left(\frac{x-\overline{x}}{\sigma}\right) = \int_{-\infty}^{x} dx' \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x'-\overline{x})^2}{2\sigma^2}\right) \tag{9.30}$$

$$= \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{x-\overline{x}}{\sigma\sqrt{2}}\right)\right). \tag{9.31}$$

9.1.6 Multivariate Distributions

Consider now two quantities which are measured simultaneously. ξ and η are the corresponding random variables. The cumulative distribution function is

$$F(x, y) = P(\xi \leq x \quad \text{and} \quad \eta \leq y). \tag{9.32}$$

[1] erf(x) is an intrinsic function in FORTRAN or C.

Expectation values are defined as

$$E[\varphi(x,y)] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \varphi(x,y) d^2F(x,y). \tag{9.33}$$

For continuous $F(x,y)$ the probability density is

$$f(x,y) = \frac{\partial^2 F}{\partial x \partial y} \tag{9.34}$$

and the expectation value is simply

$$E[\varphi(x,y)] \int_{-\infty}^{\infty} dx \int_{-\infty}^{\infty} dy \varphi(x,y) f(x,y). \tag{9.35}$$

The moments of the distribution are the expectation values

$$M_{k,l} = E[\xi^k \eta^l]. \tag{9.36}$$

Most important are the averages

$$\overline{x} = E[\xi] \quad \overline{y} = E[\eta] \tag{9.37}$$

and the covariance matrix

$$\begin{pmatrix} E[(\xi-\overline{x})^2] & E[(\xi-\overline{x})(\eta-\overline{y})] \\ E[(\xi-\overline{x})(\eta-\overline{y})] & E[(\eta-\overline{y})^2] \end{pmatrix} = \begin{pmatrix} \overline{x^2}-\overline{x}^2 & \overline{xy}-\overline{x}\,\overline{y} \\ \overline{xy}-\overline{x}\,\overline{y} & \overline{y^2}-\overline{y}^2 \end{pmatrix}. \tag{9.38}$$

The correlation coefficient is defined as

$$\rho = \frac{\overline{xy}-\overline{x}\,\overline{y}}{\sqrt{\left(\overline{x^2}-\overline{x}^2\right)\left(\overline{y^2}-\overline{y}^2\right)}}. \tag{9.39}$$

If there is no correlation then $\rho = 0$ and $F(x,y) = F_1(x)F_2(y)$.

9.1.7 Central Limit Theorem

Consider N independent random variables ξ_i with the same cumulative distribution function $F(x)$, for which $E[\xi] = 0$ and $E[\xi^2] = 1$. Define a new random variable

$$\eta_N = \frac{\xi_1 + \xi_2 + \cdots \xi_N}{\sqrt{N}} \tag{9.40}$$

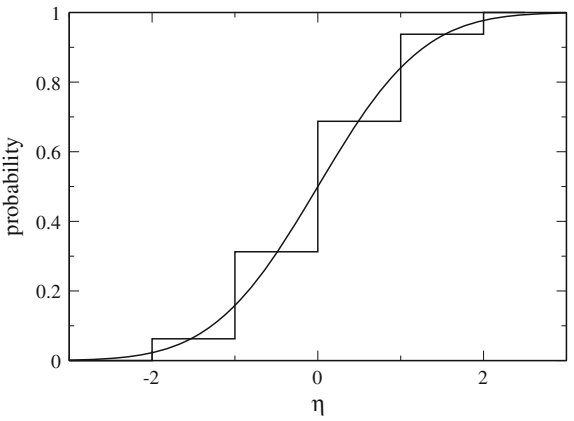


Fig. 9.4 (Central limit theorem) The cumulative distribution function of η (9.42) is shown for $N = 4$ and compared to the normal distribution (9.20)

with the cumulative distribution function $F_N(x)$. In the limit $N \to \infty$ this distribution approaches (Fig. 9.4) a cumulative normal distribution [100]

$$\lim_{N\to\infty} F_N(x) = \Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \mathrm{e}^{-t^2/2} dt. \tag{9.41}$$

9.1.8 Example: Binomial Distribution

Toss a coin N times giving $\xi_i = 1$ (heads) or $\xi_i = -1$ (tails) with equal probability $P = \frac{1}{2}$. Then $E[\xi_i] = 0$ and $E[\xi_i^2] = 1$. The distribution of

$$\eta = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} \xi_i \tag{9.42}$$

can be derived from the binomial distribution

$$1 = \left[\frac{1}{2} + \left(-\frac{1}{2}\right)\right]^N = 2^{-N} \sum_{p=0}^{N} (-1)^{N-p} \binom{N}{N-p} \tag{9.43}$$

where p counts the number of tosses with $\xi = +1$. Since

$$n = p \cdot 1 + (N - p) \cdot (-1) = 2p - N \in [-N, N] \tag{9.44}$$

the probability of finding $\eta = \frac{n}{\sqrt{N}}$ is given by the binomial coefficient

$$P(\eta = \frac{2p - N}{\sqrt{N}}) = 2^{-N} \begin{pmatrix} N \\ N - p \end{pmatrix} \tag{9.45}$$

or

$$P(\eta = \frac{n}{\sqrt{N}}) = 2^{-N} \begin{pmatrix} N \\ \frac{N-n}{2} \end{pmatrix}. \tag{9.46}$$

9.1.9 Average of Repeated Measurements

A quantity X is measured N times. The results $X_1 \cdots X_N$ are independent random numbers with the same distribution function $f(X_i)$. Their expectation value is the exact value $E[X_i] = \int dX_i \, X_i f(X_i) = X$ and the standard deviation due to measurement uncertainties is $\sigma_X = \sqrt{E[X_i^2] - X^2}$. The new random variables

$$\xi_i = \frac{X_i - X}{\sigma_X} \tag{9.47}$$

have zero mean

$$E[\xi_i] = \frac{E[X_i] - X}{\sigma_X} = 0 \tag{9.48}$$

and unit standard deviation

$$\sigma_\xi^2 = E[\xi_i^2] - E[\xi_i]^2 = E\left[\frac{X_i^2 + X^2 - 2XX_i}{\sigma_X^2}\right] = \frac{E[X_i^2] - X^2}{\sigma_X^2} = 1. \tag{9.49}$$

Hence the quantity

$$\eta = \frac{\sum_1^N \xi_i}{\sqrt{N}} = \frac{\sum_1^N X_i - NX}{\sqrt{N}\sigma_X} = \frac{\sqrt{N}}{\sigma_X}(\overline{X} - X) \tag{9.50}$$

obeys a normal distribution

$$f(\eta) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\eta^2/2}. \tag{9.51}$$

From

$$f(\overline{X})d\overline{X} = f(\eta)d\eta = f(\eta(\overline{X}))\frac{\sqrt{N}}{\sigma_X}d\overline{X} \tag{9.52}$$

we obtain

$$f(\overline{X}) = \frac{\sqrt{N}}{\sqrt{2\pi}\sigma_X}\exp\left\{-\frac{N}{2\sigma_X^2}(\overline{X}-X)^2\right\}. \tag{9.53}$$

The average of N measurements obeys a Gaussian distribution around the exact value X with a reduced standard deviation of

$$\sigma_{\overline{X}} = \frac{\sigma_X}{\sqrt{N}}. \tag{9.54}$$

9.2 Random Numbers

True random numbers of high quality can be generated using physical effects like thermal noise in a diode or atmospheric noise [101]. Computers very often make use of pseudo random numbers which have comparable statistical properties but are not totally unpredictable. For cryptographic purposes sophisticated algorithms are available which are slow but cryptographically secure, e.g. the **Yarrow** [102] and **Fortuna** [103] algorithms. In computational physics, usually simpler methods are sufficient which are not cryptographically secure, but pass important statistical tests like Marsaglia's DIEHARD collection [104, 105] and TestU01 [106, 107]. Most methods use an iterated function (Sect. 22.1). A set of numbers $\mathcal{Z}$ (e.g. 32-bit integers) is mapped onto itself by an invertible function $f(r)$ and, starting from a random seed number $r_0 \in \mathcal{Z}$, the sequence

$$r_{i+1} = f(r_i) \tag{9.55}$$

is calculated to provide a series of pseudo random numbers [105]. Using 32-bit integers there are 2^{32} different numbers, hence the period cannot exceed 2^{32}. The method can be improved by taking $\mathcal{Z}$ to be the set of m-tuples of 32-bit integers $r = \{z_1, z_2 \ldots z_m\}$ and $f(r)$ a function that converts one m-tuple into another. An m-tuple of successive function values defines the iteration

$$r_i = \{z_i, z_{i-1}, \ldots z_{i-m+1}\} \tag{9.56}$$

$$r_{i+1} = \{z_{i+1}, z_i, \ldots z_{i-m+2}\} = \{f(z_i, \ldots z_{i-m+1}), z_i, \ldots z_{i-m+2}\}. \tag{9.57}$$

Table 9.1 Addition modulo 2

$0+0=0$
$1+0=1$
$0+1=1$
$1+1=0$

Using 32-bit integers, (9.57) has a maximum period of 2^{32m} (Example: for $m=2$ and generating 10^6 numbers per second the period is 584942 years). For the initial seed, here m independent random numbers have to be provided.

The special case of a lagged RNG simply uses

$$r_{i+1} = \{z_{i+1}, z_i, \ldots z_{i-m+2}\} = \{f(z_{i-m+1}), z_i, \ldots z_{i-m+2}\}. \tag{9.58}$$

Popular kinds of functions $f(r)$ include linear congruent mappings, xorshift, lagged Fibonacci, multiply with carry (MWC), complimentary multiply with carry (CMWC) methods and combinations of these like the famous Mersenne Twister [108] and KISS [105] algorithms. We discuss briefly some important principles.

9.2.1 Linear Congruent Mapping (LC)

A simple algorithm, mainly of historical importance due to some well known problems [109], is the linear congruent mapping

$$r_{i+1} = (ar_i + c) \bmod b \tag{9.59}$$

with multiplier a and base b which is usually taken to be $b = 2^{32}$ for 32-bit integers since this can be implemented most easily. The maximum period is given by b.

9.2.2 Xorshift

A 32-Bit integer[2] can be viewed as a vector $\mathbf{r} = (b_0, b_1 \ldots b_{31})$ of elements b_i in the field $\mathcal{F}_2 = \{0, 1\}$. Addition of two such vectors (modulo 2) can be implemented with the exclusive-or operation as can be seen from comparison with the table (Table 9.1).

An invertible linear transformation of the vector $\mathbf{r}$ can be described by multiplication with a nonsingular 32×32 matrix T

$$f(\mathbf{r}) = \mathbf{r}T. \tag{9.60}$$

[2]This method can be easily extended to 64-Bit integers.

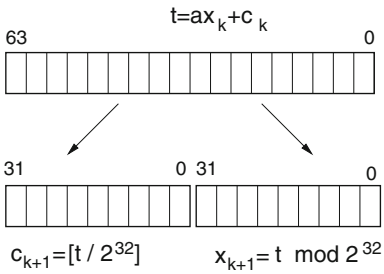


Fig. 9.5 Multiply with carry method using 64-bit integer arithmetic

To simplify the numerical calculation, Marsaglia [105] considers matrices of the special form[3]

$$T = (1 + L^a)(1 + R^b)(1 + L^c) \tag{9.61}$$

where L (R) is a matrix that produces a left (right) shift by one. For properly chosen numbers a, b, c the matrix T is of order $2^{32} - 1$ and the random numbers have the maximum possible period. There are many possible choices, one of them leads to the sequence

$$\begin{aligned} y &= y \text{ xor } (y \ll 13) \\ y &= y \text{ xor}(y \gg 17) \\ y &= y \text{ xor } (y \ll 5). \end{aligned} \tag{9.62}$$

9.2.3 Multiply with Carry (MWC)

This method is quite similar to the linear congruent mapping. However, instead of the constant c in (9.59) a varying carry is used.

For base $b = 2^{32}$ and multiplier $a = 698769069$ consider pairs of integers $r = [x, c]$ with $0 \le c < a$, $0 \le x < b$ excluding $[0, 0]$and $[a - 1, b - 1]$ and the iteration function[4]

$$f([x, c]) = [ax + c \bmod b, (ax + c)/b]. \tag{9.63}$$

Starting with a random seed $[x_0, c_0]$ the sequence $[x_k, c_k] = f([x_{k-1}, c_{k-1}])$ has a period of about 2^{60} [105]. If one calculates $t = ax_k + c_k$ in 64 bits, then for $b = 2^{32}$,$c_{k+1}$is given by the top 32 bits and $x_{k+1}$by the bottom 32 bits (Fig. 9.5).

[3] At least three factors are necessary for 32 and 64-Bit integers.

[4] Using integer arithmetics.

9.2.4 Complementary Multiply with Carry (CMWC)

The simple MWC method has some inherent problems which can be overcome by a slight modification. First, the base is taken to be $b = 2^{32} - 1$ and second the iteration is changed to use the $(b - 1)$-complement

$$x_k = (b - 1) - (ax_{k-1} + c_{k-1}) \bmod b. \tag{9.64}$$

This method can provide random numbers which pass many tests and have very large periods.

9.2.5 Random Numbers with Given Distribution

Assume we have a program that generates random numbers in the interval [0,1] like in C:

$$rand()/(double)RAND\_MAX.$$

The corresponding cumulative distribution function is

$$F_0(x) = \begin{cases} 0 & for \quad x < 0 \\ x & for \quad 0 \le x \le 1 \\ 1 & for \quad x > 1 \end{cases}. \tag{9.65}$$

Random numbers with cumulative distribution $F(x)$ can be obtained as follows:

choose a RN $r \in [0, 1]$ with $P(r \le x) = F_0(x)$
let $\xi = F^{-1}(r)$

$F(x)$ increases monotonously and therefore

$$P(\xi \le x) = P(F(\xi) \le F(x)) = P(r \le F(x)) = F_0(F(x)) \tag{9.66}$$

but since $0 \le F(x) \le 1$ we have

$$P(\xi \le x) = F(x). \tag{9.67}$$

This method of course is applicable only if F^{-1} can be expressed analytically.

9.2.6 Examples

9.2.6.1 Fair Die

A six-sided fair die can be simulated as follows:

choose a random number $r \in [0, 1]$

$$\text{Let } \xi = F^{-1}(r) = \begin{cases} 1 & for \quad 0 \le r < \frac{1}{6} \\ 2 & for \quad \frac{1}{6} \le r < \frac{2}{6} \\ 3 & for \quad \frac{2}{6} \le r < \frac{3}{6} \\ 4 & for \quad \frac{3}{6} \le r < \frac{4}{6} \\ 5 & for \quad \frac{4}{6} \le r < \frac{5}{6} \\ 6 & for \quad \frac{5}{6} \le r < 1 \end{cases}$$

9.2.6.2 Exponential Distribution

The cumulative distribution function

$$F(x) = 1 - \mathrm{e}^{-x/\lambda} \tag{9.68}$$

which corresponds to the exponential probability density

$$f(x) = \frac{1}{\lambda}\mathrm{e}^{-x/\lambda} \tag{9.69}$$

can be inverted by solving

$$r = 1 - \mathrm{e}^{-x/\lambda} \tag{9.70}$$

for x:

choose a random number $r \in [0, 1]$
Let $x = F^{-1}(r) = -\lambda \ln(1 - r)$.

9.2.6.3 Random Points on the Unit Sphere

We consider the surface element

$$\frac{1}{4\pi}R^2 d\varphi \sin\theta d\theta. \tag{9.71}$$

Our aim is to generate points on the unit sphere (θ, φ) with the probability density

$$f(\theta,\varphi)d\varphi d\theta = \frac{1}{4\pi}d\varphi \sin\theta d\theta = -\frac{1}{4\pi}d\varphi d\cos\theta. \tag{9.72}$$

The corresponding cumulative distribution is

$$F(\theta,\varphi) = -\frac{1}{4\pi}\int_1^{\cos\theta} d\cos\theta \int_0^{\varphi} d\varphi = \frac{\varphi}{2\pi}\frac{1-\cos\theta}{2} = F_\varphi F_\theta. \tag{9.73}$$

Since this factorizes, the two angles can be determined independently:

choose a first random number $r_1 \in [0, 1]$
Let $\varphi = F_\varphi^{-1}(r_1) = 2\pi r_1$
choose a second random number $r_2 \in [0, 1]$
Let $\theta = F_\theta^{-1}(r_2) = \arccos(1 - 2r_2)$

9.2.6.4 Gaussian Distribution (Box Muller)

For a Gaussian distribution the inverse F^{-1} has no simple analytical form. The famous Box Muller method [110] is based on a 2-dimensional normal distribution with probability density

$$f(x,y) = \frac{1}{2\pi}\exp\left\{-\frac{x^2+y^2}{2}\right\} \tag{9.74}$$

which reads in polar coordinates

$$f(x,y)dxdy = f_p(\rho,\varphi)d\rho d\varphi \frac{1}{2\pi}\mathrm{e}^{-\rho^2/2}\rho d\rho d\varphi. \tag{9.75}$$

Hence

$$f_p(\rho,\varphi) = \frac{1}{2\pi}\rho\mathrm{e}^{-\rho^2/2} \tag{9.76}$$

and the cumulative distribution factorizes:

$$F_p(\rho,\varphi) = \frac{1}{2\pi}\varphi\cdot\int_0^{\rho}\rho'\mathrm{e}^{-\rho'^2/2}d\rho' = \frac{\varphi}{2\pi}(1-\mathrm{e}^{-\rho^2}) = F_\varphi(\varphi)F_\rho(\rho). \tag{9.77}$$

The inverse of F_ρ is

$$\rho = \sqrt{-\ln(1-r)} \tag{9.78}$$

and the following algorithm generates Gaussian random numbers:

$r1 = RN \in [0, 1]$
$r2 = RN \in [0, 1]$
$\rho = \sqrt{-\ln(1 - r_1)}$
$\varphi = 2\pi r_2$
$x = \rho \cos \varphi.$

9.3 Monte-Carlo Integration

Physical problems often involve high dimensional integrals (for instance path integrals, thermodynamic averages) which cannot be evaluated by standard methods. Here Monte Carlo methods can be very useful. Let us start with a very basic example.

9.3.1 Numerical Calculation of π

The area of a unit circle ($r = 1$) is given by $r^2\pi = \pi$. Hence π can be calculated by numerical integration. We use the following algorithm:

choose N points randomly in the first quadrant, for instance N independent pairs $x, y \in [0, 1]$
Calculate $r^2 = x^2 + y^2$
Count the number of points within the circle, i.e. the number of points $Z(r^2 \leq 1)$.
$\frac{\pi}{4}$ is approximately given by $\frac{Z(r^2 \leq 1)}{N}$

The result converges rather slowly (Figs. 9.6, 9.7).

9.3.2 Calculation of an Integral

Let ξ be a random variable in the interval $[a, b]$ with the distribution

$$P(x < \xi \leq x + dx) = f(x)dx = \begin{cases} \frac{1}{b-a} & for \quad x \in [a, b] \\ 0 & else \end{cases} . \tag{9.79}$$

The expectation value of a function $g(x)$ is

$$E[g(x)] = \int_{-\infty}^{\infty} g(x)f(x)dx = \int_a^b g(x)dx \tag{9.80}$$

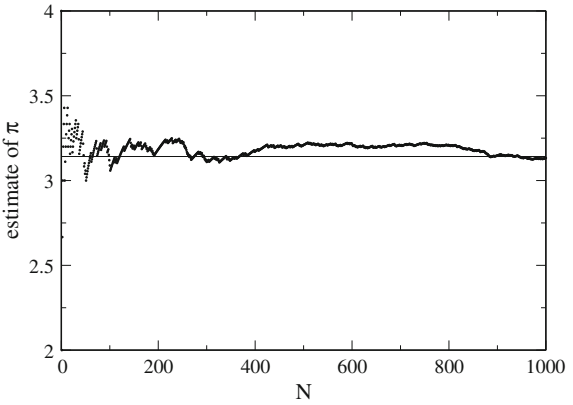


Fig. 9.6 Convergence of the numerical integration

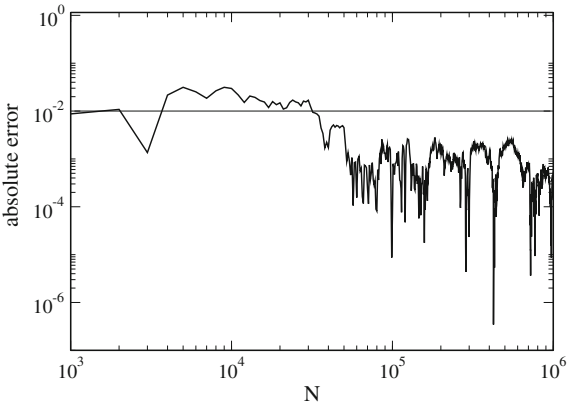


Fig. 9.7 Error of the numerical integration

hence the average of N randomly taken function values approximates the integral

$$\int_a^b g(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} g(\xi_i) = \overline{g(\xi)}. \tag{9.81}$$

To estimate the error we consider the new random variable

$$\gamma = \frac{1}{N}\sum_{i=1}^{N} g(\xi). \tag{9.82}$$

Its average is

$$\overline{\gamma} = E[\gamma] = \frac{1}{N}\sum_{i=1}^{N} E[g(x)] = E[g(x)] = \int_a^b g(x)dx \tag{9.83}$$

and the variance follows from

$$\sigma_\gamma^2 = E[(\gamma - \overline{\gamma})^2] = E[(\frac{1}{N}\sum g(\xi_i) - \overline{\gamma})^2] = E[(\frac{1}{N}\sum (g(\xi_i) - \overline{\gamma}))^2] \quad (9.84)$$

$$= \frac{1}{N^2} E[\sum (g(\xi_i) - \overline{\gamma})^2] = \frac{1}{N}(\overline{g(\xi)^2} - \overline{g(\xi)}^2) = \frac{1}{N}\sigma_{g(\xi)}^2. \quad (9.85)$$

The width of the distribution and hence the uncertainty falls off as $1/\sqrt{N}$.

9.3.3 More General Random Numbers

Consider now random numbers $\xi \in [a, b]$ with arbitrary (but within $[a, b]$ not vanishing) probability density $f(x)$. The integral is approximated by

$$\frac{1}{N}\sum_{i=1}^{N} \frac{g(\xi_i)}{f(\xi_i)} = E\left[\frac{g(x)}{f(x)}\right] = \int_a^b \frac{g(x)}{f(x)} f(x)dx = \int_a^b g(x)dx. \quad (9.86)$$

The new random variable

$$\tau = \frac{1}{N}\sum_{i=1}^{N} \frac{g(\xi_i)}{f(\xi_i)} \quad (9.87)$$

according to (9.85) has a standard deviation given by

$$\sigma_\tau = \frac{1}{\sqrt{N}}\sigma(\frac{g(\xi)}{f(\xi)}) \quad (9.88)$$

which can be reduced by choosing f similar to g. Then preferentially ξ are generated in regions where the integrand is large (importance sampling).

9.3.4 Configuration Integrals

Consider a system which is described by a *ndim* dimensional configuration space $q_1 \ldots q_{ndim}$ where a certain configuration has the normalized probability density

$$\varrho(q_1, \ldots q_{ndim}) \quad (9.89)$$

$$\int \ldots \int \varrho(q_1, \ldots q_{ndim}) dq^{ndim} = 1. \quad (9.90)$$

The average of an observable $A(q_1 \dots q_{ndim})$ has the form

$$< A >= \int \dots \int A(q_1 \dots q_{ndim}) \varrho(q_1, \dots q_{ndim}) dq^{ndim} \tag{9.91}$$

which will be calculated by MC integration.

Classical Thermodynamic Averages

Consider a classical N particle system with potential energy

$$V(q_1 \dots q_{3N}). \tag{9.92}$$

The probability of a certain configuration is given by its normalized Boltzmann-factor

$$\varrho(q_1 \dots q_{3N}) = \frac{\mathrm{e}^{-\beta V(q_1 \dots q_{3N})}}{\int dq^{3N} \mathrm{e}^{-\beta V(q_1 \dots q_{3N})}} \tag{9.93}$$

and the thermal average of some observable quantity $A(q_1 \dots q_{3N})$ is given by the configuration integral

$$< A >= \int A(q_1 \dots q_{ndim}) \varrho(q_1 \dots q_{3N}) dq^{3N}$$

$$= \frac{\int dq^{3N} A(q_1 \dots q_{ndim}) \mathrm{e}^{-\beta V(q_1 \dots q_{3N})}}{\int dq^{3N} \mathrm{e}^{-\beta V(q_1 \dots q_{3N})}}. \tag{9.94}$$

Variational Quantum Monte Carlo method

Consider a quantum mechanical N particle system with Hamiltonian

$$H = T + V(q_1 \dots q_{3N}). \tag{9.95}$$

According to Ritz's variational principle, the ground state energy is a lower bound to the energy expectation value of any trial wavefunction

$$E_V = \frac{< \Psi_{trial} | H | \Psi_{trial} >}{< \Psi_{trial} | \Psi_{trial} >} \geq E_0. \tag{9.96}$$

Energy and wavefunction of the ground state can be approximated by minimizing the energy of the trial wavefunction, which is rewritten in the form

$$E_V = \frac{\int \Psi^*_{trial}(q_1 \dots q_{3N}) H \Psi_{trial}(q_1 \dots q_{3N}) dq^{3N}}{\int |\Psi_{trial}(q_1 \dots q_{3N})|^2 dq^{3N}}$$

$$= \frac{\int \varrho(q_1 \dots q_{3N}) E_L(q_1 \dots q_{3N}) dq^{3N}}{\int \varrho(q_1 \dots q_{3N}) dq^{3N}} \tag{9.97}$$

with the probability density

$$\varrho(q_1 \dots q_{3N}) = |\Psi_{trial}(q_1 \dots q_{3N})|^2 \tag{9.98}$$

and the so called local energy

$$E_L = \frac{H \Psi_{trial}(q_1 \dots q_{3N})}{\Psi_{trial}(q_1 \dots q_{3N})}. \tag{9.99}$$

9.3.5 Simple Sampling

Let ξ be a random variable which is equally distributed over the range $q_{\min} \cdots q_{\max}$, i.e. a probability distribution

$$P(\xi \in [q, q + dq]) = f(q) dq \tag{9.100}$$

$$f(q) = \begin{cases} \frac{1}{q_{\max} - q_{\min}} & q \in [q_{\min}, q_{\max}] \\ 0 \text{ else} \end{cases} \tag{9.101}$$

$$\int f(q) dq = 1. \tag{9.102}$$

Repeatedly choose *ndim* random numbers $\xi_1^{(m)}, \dots \xi_{ndim}^{(m)}$ and calculate the expectation value

$$E\left(A(\xi_1 \cdots \xi_{ndim}) \varrho(\xi_1, \dots \xi_{ndim})\right) = \lim_{M \to \infty} \frac{1}{M} \sum_{m=1}^{M} A(\xi_1^{(m)} \dots \xi_{ndim}^{(m)}) \varrho(\xi_1^{(m)} \dots \xi_{ndim}^{(m)})$$

$$= \int A(q_1 \dots q_{ndim}) \varrho(q_1 \dots q_{ndim}) f(q_1) \cdots f(q_{ndim}) dq_1 \cdots dq_{ndim}$$

$$= \frac{1}{(q_{\max} - q_{\min})^{ndim}} \int_{q_{\min}}^{q_{\max}} \cdots \int_{q_{\min}}^{q_{\max}} A(q_1 \dots q_{ndim}) \varrho(q_1 \dots q_{ndim}) dq^{ndim}.$$

Hence

$$\frac{E\left(A(\xi_1 \cdots \xi_{ndim})\varrho(\xi_1, \ldots \xi_{ndim})\right)}{E\left(\varrho(\xi_1, \ldots \xi_{ndim})\right)}$$

$$= \frac{\int_{q_{\min}}^{q_{\max}} \cdots \int_{q_{\min}}^{q_{\max}} A(q_1 \ldots q_{ndim})\varrho(q_1 \ldots q_{ndim})dq^{ndim}}{\int_{q_{\min}}^{q_{\max}} \cdots \int_{q_{\min}}^{q_{\max}} \varrho(q_1 \ldots q_{ndim})dq^{ndim}} \approx < A > . \tag{9.103}$$

Each set of random numbers $\xi_1 \ldots \xi_{ndim}$ defines one sample configuration. The average over a large number M of samples gives an approximation to the average $< A >$, if the range of the q_i is sufficiently large. However, many of the samples will have small weight and contribute only little.

9.3.6 Importance Sampling

Let us try to sample preferentially the most important configurations. Choose the distribution function as

$$f(q_1 \cdots q_{ndim}) = \varrho(q_1 \ldots q_{ndim}). \tag{9.104}$$

The expectation value of A now directly approximates the configurational average

$$E\left(A(\xi_1 \cdots \xi_{ndim})\right) = \lim_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} A(\xi_1^{(m)} \ldots \xi_{ndim}^{(m)})$$

$$= \int A(q_1 \ldots q_{ndim})\varrho(q_1 \ldots q_{ndim})dq^{ndim} = < A > . \tag{9.105}$$

9.3.7 Metropolis Algorithm

The algorithm by Metropolis [111] can be used to select the necessary configurations. Starting from an initial configuration $\mathbf{q}_0 = (q_1^{(0)} \cdots q_{3N}^{(0)})$ a chain of configurations is generated. Each configuration depends only on its predecessor, hence the configurations form a Markov chain.

The transition probabilities

$$W_{i\to j} = P(\mathbf{q}_i \to \mathbf{q}_j) \tag{9.106}$$

are chosen to fulfill the condition of detailed balance (Fig. 9.8)

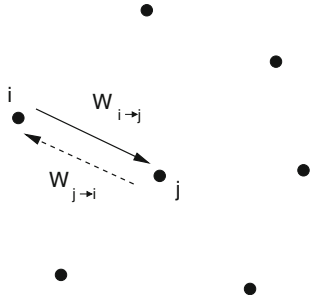


Fig. 9.8 Principle of detailed balance

$$\frac{W_{i\to j}}{W_{j\to i}} = \frac{\varrho(\mathbf{q}_j)}{\varrho(\mathbf{q}_i)}. \tag{9.107}$$

This is a sufficient condition that the configurations are generated with probabilities given by their Boltzmann factors. This can be seen from consideration of an ensemble of such Markov chains: Let $N_n(\mathbf{q}_i)$ denote the number of chains which are in the configuration $\mathbf{q}_i$ after n steps. The changes during the following step are

$$\Delta N(\mathbf{q}_i) = N_{n+1}(\mathbf{q}_i) - N_n(\mathbf{q}_i) = \sum_{\mathbf{q}_j \in conf.} N_n(\mathbf{q}_j) W_{j\to i} - N_n(\mathbf{q}_i) W_{i\to j}. \tag{9.108}$$

In equilibrium

$$N_{eq}(\mathbf{q}_i) = N_0 \varrho(\mathbf{q}_i) \tag{9.109}$$

and the changes (9.108) vanish:

$$\begin{aligned}\Delta N(\mathbf{q}_i) &= N_0 \sum_{\mathbf{q}_j} \varrho(\mathbf{q}_j) W_{j\to i} - \varrho(\mathbf{q}_i) W_{i\to j} \\ &= N_0 \sum_{\mathbf{q}_j} \varrho(\mathbf{q}_j) W_{j\to i} - \varrho(\mathbf{q}_i) \left[W_{j\to i} \frac{\varrho(\mathbf{q}_j)}{\varrho(\mathbf{q}_i)} \right] \\ &= 0. \end{aligned} \tag{9.110}$$

A solution of

$$\Delta N(\mathbf{q}_i) = \sum_{\mathbf{q}_j \in conf.} N_n(\mathbf{q}_j) W_{j\to i} - N_n(\mathbf{q}_i) W_{i\to j} = 0 \tag{9.111}$$

corresponds to a zero eigenvalue of the system of equations

$$\sum_{\mathbf{q}_j} N(\mathbf{q}_j) W_{j\to i} - N(\mathbf{q}_i) \sum_{\mathbf{q}_j} W_{i\to j} = \lambda N(\mathbf{q}_i). \tag{9.112}$$

One solution of this eigenvalue equation is given by

$$\frac{N_{eq}(\mathbf{q}_j)}{N_{eq}(\mathbf{q}_i)} = \frac{\varrho(\mathbf{q}_j)}{\varrho(\mathbf{q}_i)}. \tag{9.113}$$

However, there may be other solutions. For instance if not all configurations are connected by possible transitions and some isolated configurations are occupied initially.

Metropolis Algorithm

This famous algorithm consists of the following steps:
(a) choose a new configuration randomly (trial step) with probability

$$T(\mathbf{q}_i \to \mathbf{q}_{trial}) = T(\mathbf{q}_{trial} \to \mathbf{q}_i)$$

(b) calculate

$$R = \frac{\varrho(\mathbf{q}_{trial})}{\varrho(\mathbf{q}_i)}$$

(c) if $R \geq 1$ the trial step is accepted $\mathbf{q}_{i+1} = \mathbf{q}_{trial}$
(d) if $R < 1$ the trial step is accepted only with probability R. choose a random number $\xi \in [0, 1]$ and the next configuration according to

$$\mathbf{q}_{i+1} = \begin{cases} \mathbf{q}_{trial} & if \quad \xi < R \\ \mathbf{q}_i & if \quad \xi \geq R. \end{cases}$$

The transition probability is the product

$$W_{i\to j} = T_{i\to j} A_{i\to j} \tag{9.114}$$

of the probability $T_{i\to j}$ to select $i \to j$ as a trial step and the probability $A_{i\to j}$ to accept the trial step. Now we have

$$\begin{array}{l} for\ R \geq 1 \ \to \ A_{i\to j} = 1,\ A_{j\to i} = R^{-1} \\ for\ R < 1 \ \to \ A_{i\to j} = R,\ A_{j\to i} = 1 \end{array}. \tag{9.115}$$

Since $T_{i\to j} = T_{j\to i}$, in both cases

$$\frac{N_{eq}(\mathbf{q}_j)}{N_{eq}(\mathbf{q}_i)} = \frac{W_{i\to j}}{W_{j\to i}} = \frac{A_{i\to j}}{A_{j\to i}} = R = \frac{\varrho(\mathbf{q}_j)}{\varrho(\mathbf{q}_i)}. \tag{9.116}$$

The size of the trial steps has to be adjusted to produce a reasonable acceptance ratio of

$$\frac{N_{accepted}}{N_{rejected}} \approx 1. \tag{9.117}$$

Multiple Walkers

To scan the relevant configurations more completely and reduce correlation between the samples, usually a large number of "walkers" is used (e.g. several hundred) which, starting from different initial conditions, represent independent Markov chains. This also offers a simple possibility for parallelization.

Problems

Problem 9.1 Central Limit Theorem

This computer experiment draws a histogram for the random variable τ, which is calculated from N random numbers $\xi_1 \cdots \xi_N$:

$$\tau = \frac{\sum_{i=1}^{N} \xi_i}{\sqrt{N}}. \tag{9.118}$$

The ξ_i are random numbers with zero mean and unit variance and can be chosen as

- $\xi_i = \pm 1$ (coin tossing)
- Gaussian random numbers

Investigate how a Gaussian distribution is approached for large N.

Problem 9.2 Nonlinear Optimization

MC methods can be used for nonlinear optimization (Traveling salesman problem, structure optimization etc.) [112]. Consider an energy function depending on many coordinates

$$E(q_1, q_2 \cdots q_N). \tag{9.119}$$

Introduce a fictitious temperature T and generate configurations with probabilities

$$P(q_1 \cdots q_N) = \frac{1}{Z} \mathrm{e}^{-E(q_1 \cdots q_N)/T}. \tag{9.120}$$

Slow cooling drives the system into a local minimum. By repeated heating and cooling other local minima can be reached (simulated annealing)

In this computer experiment we try to find the shortest path which visits each of N up to 50 given points. The fictitious Boltzmann factor for a path with total length L is

$$P(L) = e^{-L/T}. \tag{9.121}$$

Starting from an initial path $S = (i_1, i_2, \cdots i_N)$ $n < 5$ and p are chosen randomly and a new path $S' = (i_1, \cdots i_{p-1}, i_{p+n}, \cdots i_p, i_{p+n+1}, \cdots i_N)$ is generated by reverting the sub-path

$$i_p \cdots i_{p+n} \rightarrow i_{p+n} \cdots i_p.$$

Start at high temperature $T > L$ and cool down slowly.

Chapter 10
Eigenvalue Problems

Eigenvalue problems are omnipresent in physics. Important examples are the time independent Schrödinger equation in a finite orthogonal basis (Chap. 10)

$$\sum_{j=1}^{M} < \phi_{j'}|H|\phi_j > C_j = EC_{j'} \tag{10.1}$$

or the harmonic motion of a molecule around its equilibrium structure (Sect. 15.4.1)

$$\omega^2 m_i(\xi_i - \xi_i^{eq}) = \sum_j \frac{\partial^2 U}{\partial \xi_i \partial \xi_j}(\xi_j - \xi_j^{eq}). \tag{10.2}$$

Most important are ordinary eigenvalue problems,[1] *which involve the solution of a homogeneous system of linear equations*

$$\sum_{j=1}^{N} a_{ij}x_j = \lambda x_i \tag{10.3}$$

with a Hermitian (or symmetric, if real) matrix [113]

$$a_{ji} = a_{ij}^*. \tag{10.4}$$

The couple $(\lambda, \boldsymbol{x})$ *consisting of an eigenvector* $\boldsymbol{x}$ *and the corresponding eigenvalue* λ *is called an eigenpair.*

[1] *We do not consider general eigenvalue problems here.*

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_10

Matrices of small dimension can be diagonalized directly by determining the roots of the characteristic polynomial and solving a homogeneous system of linear equations. The Jacobi method uses successive rotations to diagonalize a matrix with a unitary transformation. A very popular method for not too large symmetric matrices reduces the matrix to tridiagonal form which can be diagonalized efficiently with the QL algorithm. Some special tridiagonal matrices can be diagonalized analytically. Special algorithms are available for matrices of very large dimension, for instance the famous Lanczos method.

10.1 Direct Solution

For matrices of very small dimension (2, 3) the determinant

$$\det \left| a_{ij} - \lambda \delta_{ij} \right| = 0 \tag{10.5}$$

can be written explicitly as a polynomial of λ. The roots of this polynomial are the eigenvalues. The eigenvectors are given by the system of equations

$$\sum_j (a_{ij} - \lambda \delta_{ij}) u_j = 0. \tag{10.6}$$

10.2 Jacobi Method

Any symmetric 2×2 matrix

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix} \tag{10.7}$$

can be diagonalized by a rotation of the coordinate system. Rotation by the angle φ corresponds to an orthogonal transformation with the rotation matrix

$$R_\varphi = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix}. \tag{10.8}$$

In the following we use the abbreviations

$$c = \cos\varphi, \quad s = \sin\varphi, \quad t = \tan\varphi \tag{10.9}$$

The transformed matrix is

$$RAR^{-1} = \begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{pmatrix} \begin{pmatrix} c & s \\ -s & c \end{pmatrix}$$
$$= \begin{pmatrix} c^2 a_{11} + s^2 a_{22} - 2csa_{12} & cs(a_{11} - a_{22}) + (c^2 - s^2)a_{12} \\ cs(a_{11} - a_{22}) + (c^2 - s^2)a_{12} & s^2 a_{11} + c^2 a_{22} + 2csa_{12} \end{pmatrix}. \tag{10.10}$$

It is diagonal if

$$0 = cs(a_{11} - a_{22}) + (c^2 - s^2)a_{12} = \frac{a_{11} - a_{22}}{2} \sin(2\varphi) + a_{12} \cos(2\varphi) \tag{10.11}$$

or

$$\tan(2\varphi) = \frac{2a_{12}}{a_{22} - a_{11}}. \tag{10.12}$$

Calculation of φ is not necessary since only its cosine and sine appear in (10.10). From [113]

$$\frac{1 - t^2}{t} = \frac{c^2 - s^2}{2cs} = \cot(2\varphi) = \frac{a_{22} - a_{11}}{2a_{12}} \tag{10.13}$$

we see that t is a root of

$$t^2 + \frac{a_{22} - a_{11}}{a_{12}} t - 1 = 0 \tag{10.14}$$

hence

$$t = -\frac{a_{22} - a_{11}}{2a_{12}} \pm \sqrt{1 + \left(\frac{a_{22} - a_{11}}{2a_{12}}\right)^2} = \frac{1}{\frac{a_{22} - a_{11}}{2a_{12}} \pm \sqrt{1 + \left(\frac{a_{22} - a_{11}}{2a_{12}}\right)^2}}. \tag{10.15}$$

For reasons of convergence [113] the solution with smaller magnitude is chosen which can be written as

$$t = \frac{\text{sign}\left(\frac{a_{22} - a_{11}}{2a_{12}}\right)}{\left|\frac{a_{22} - a_{11}}{2a_{12}}\right| + \sqrt{1 + \left(\frac{a_{22} - a_{11}}{2a_{12}}\right)^2}}. \tag{10.16}$$

Again for reasons of convergence the smaller solution φ is preferred and therefore we take

$$c = \frac{1}{\sqrt{1+t^2}} \quad s = \frac{t}{\sqrt{1+t^2}}. \tag{10.17}$$

The diagonal elements of the transformed matrix are

$$\tilde{a}_{11} = c^2 a_{11} + s^2 a_{22} - 2csa_{12} \tag{10.18}$$

$$\tilde{a}_{22} = s^2 a_{11} + c^2 a_{22} + 2csa_{12}. \tag{10.19}$$

The trace of the matrix is invariant

$$\tilde{a}_{11} + \tilde{a}_{22} = a_{11} + a_{22} \tag{10.20}$$

whereas the difference of the diagonal elements is

$$\begin{aligned} \tilde{a}_{11} - \tilde{a}_{22} &= (c^2 - s^2)(a_{11} - a_{22}) - 4csa_{12} = \frac{1-t^2}{1+t^2}(a_{11} - a_{22}) - 4\frac{a_{12}t}{1+t^2} \\ &= (a_{11} - a_{22}) + (-a_{12}\frac{1-t^2}{t})\frac{-2t^2}{1+t^2} - 4\frac{a_{12}t}{1+t^2} = (a_{11} - a_{22}) - 2ta_{12} \end{aligned} \tag{10.21}$$

and the transformed matrix has the simple form

$$\begin{pmatrix} a_{11} - a_{12}t & \\ & a_{22} + a_{12}t \end{pmatrix}. \tag{10.22}$$

For larger dimension $N > 2$ the Jacobi method uses the following algorithm:

(1) look for the dominant non-diagonal element $\max_{i \neq j} |a_{ij}|$

(2) Perform a rotation in the (ij)-plane to cancel the element $\tilde{a}_{ij}$ of the transformed matrix $\tilde{A} = R^{(ij)} \cdot A \cdot R^{(ij)-1}$. The corresponding rotation matrix has the form

$$R^{(ij)} = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & c & & s & & \\ & & & \ddots & & & \\ & & -s & & c & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{pmatrix}. \tag{10.23}$$

(3) repeat (1–2) until convergence (if possible).

The sequence of Jacobi rotations gives the over all transformation

$$RAR^{-1} = \cdots R_2 R_1 A R_1^{-1} R_2^{-1} \cdots = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_N \end{pmatrix}. \tag{10.24}$$

Hence

$$AR^{-1} = R^{-1} \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_N \end{pmatrix} \tag{10.25}$$

and the column vectors of $R^{-1} = (\mathbf{v}_1, \mathbf{v}_2 \cdots \mathbf{v}_N)$ are the eigenvectors of A:

$$A\,(\mathbf{v}_1, \mathbf{v}_2 \cdots \mathbf{v}_N) = (\lambda_1 \mathbf{v}_1, \lambda_2 \mathbf{v}_2, \cdots \lambda_N \mathbf{v}_N)\,. \tag{10.26}$$

10.3 Tridiagonal Matrices

A tridiagonal matrix has nonzero elements only in the main diagonal and the first diagonal above and below. Many algorithms simplify significantly when applied to tridiagonal matrices.

10.3.1 Characteristic Polynomial of a Tridiagonal Matrix

The characteristic polynomial of a tridiagonal matrix

$$P_A(\lambda) = \det \begin{vmatrix} a_{11} - \lambda & a_{12} & & \\ a_{21} & a_{22} - \lambda & & \\ & & \ddots & a_{N-1N} \\ & & a_{NN-1} & a_{NN} - \lambda \end{vmatrix} \tag{10.27}$$

can be calculated recursively:

$$\begin{aligned} P_0 &= 1 \\ P_1(\lambda) &= a_{11} - \lambda \\ P_2(\lambda) &= (a_{22} - \lambda) P_1(\lambda) - a_{12} a_{21} \\ &\vdots \\ P_N(\lambda) &= (a_{NN} - \lambda) P_{N-1}(\lambda) - a_{N,N-1} a_{N-1,N} P_{N-2}(\lambda). \end{aligned} \tag{10.28}$$

10.3.2 Special Tridiagonal Matrices

Certain classes of tridiagonal matrices can be diagonalized exactly [114–116].

10.3.2.1 Discretized Second Derivatives

Discretization of a second derivative involves, under Dirichlet boundary conditions $f(x_0) = f(x_{N+1}) = 0$, the differentiation matrix (Sect. 20.2)

$$M = \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix}. \tag{10.29}$$

Its eigenvectors have the form

$$\boldsymbol{f} = \begin{pmatrix} f_1 \\ \vdots \\ f_n \\ \vdots \\ f_N \end{pmatrix} = \begin{pmatrix} \sin k \\ \vdots \\ \sin(nk) \\ \vdots \\ \sin(Nk) \end{pmatrix}. \tag{10.30}$$

This can be seen by inserting (10.30) into the n-th line of the eigenvalue (10.31)

$$M\boldsymbol{f} = \lambda \boldsymbol{f} \tag{10.31}$$

$$\begin{aligned} (M\boldsymbol{f})_n &= (\sin((n-1)k) + \sin((n+1)k) - 2\sin(nk)) \\ &= 2\sin(nk)\,(\cos(k) - 1) = \lambda\,(\boldsymbol{f})_n \end{aligned} \tag{10.32}$$

with the eigenvalue

$$\lambda = 2\,(\cos k - 1) = -4\sin^2\left(\frac{k}{2}\right). \tag{10.33}$$

The first line of the eigenvalue (10.31) reads

$$\begin{aligned} (M\boldsymbol{f})_1 &= (-2\sin(k) + \sin(2k)) \\ &= 2\sin(k)(\cos(k) - 1) = \lambda(\boldsymbol{f})_n \end{aligned} \tag{10.34}$$

and from the last line we have

$$\begin{aligned}(M\boldsymbol{f})_N &= (-2\sin(Nk) + \sin([N-1]k)) \\ &= \lambda(\boldsymbol{f})_N = 2(\cos(k) - 1)\sin(Nk)\end{aligned} \tag{10.35}$$

which holds if

$$\sin((N-1)k) = 2\sin(Nk)\cos(k). \tag{10.36}$$

This simplifies to

$$\begin{aligned}&\sin(Nk)\cos(k) - \cos(Nk)\sin(k) = 2\sin(Nk)\cos(k) \\ &\sin(Nk)\cos(k) + \cos(Nk)\sin(k) = 0 \\ &\sin((N+1)k) = 0.\end{aligned} \tag{10.37}$$

Hence the possible values of k are

$$k = \frac{\pi}{(N+1)} l \text{ with } l = 1, 2, \cdots N \tag{10.38}$$

and the eigenvectors are explicitly (Fig. 10.1)

$$\boldsymbol{f} = \begin{pmatrix} \sin\left(\frac{\pi}{N+1} l\right) \\ \vdots \\ \sin\left(\frac{\pi}{N+1} l\, n\right) \\ \vdots \\ \sin\left(\frac{\pi}{N+1} l\, N\right) \end{pmatrix}. \tag{10.39}$$

For Neumann boundary conditions $\frac{\partial f}{\partial x}(x_1) = \frac{\partial f}{\partial x}(x_N) = 0$ the matrix is slightly different (Sect. 20.2)

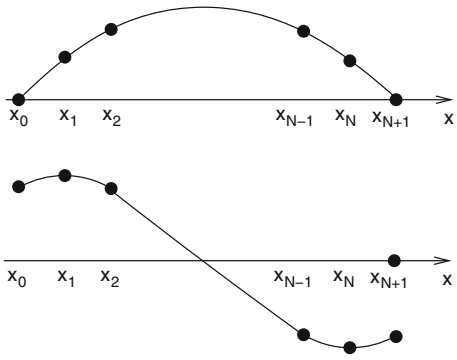


Fig. 10.1 (Lowest eigenvector) ***Top*** for fixed boundaries $f_n = \sin(nk)$ which is zero at the additional points x_0, x_{N+1}. ***Bottom*** for open boundaries $f_n = \cos((n-1)k)$ with horizontal tangent at x_1, x_N due to the boundary conditions $f_2 = f_0, f_{N-1} = f_{N+1}$

$$M = \begin{pmatrix} -2 & 2 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 2 & -2 \end{pmatrix}. \tag{10.40}$$

Its eigenvalues are also given by the expression (10.33). To obtain the eigenvectors, we try a more general ansatz with a phase shift

$$\boldsymbol{f} = \begin{pmatrix} \sin \Phi_1 \\ \vdots \\ \sin(\Phi_1 + (n-1)k) \\ \vdots \\ \sin(\Phi_1 + (N-1)k) \end{pmatrix}. \tag{10.41}$$

Obviously

$$\begin{aligned} &\sin(\Phi_1 + (n-1)k - k) + \sin(\Phi_1 + (n-1)k + k) - 2\sin(\Phi_1 + (n-1)k) \\ &= 2\,(\cos k - 1)\sin(\Phi_1 + (n-1)k). \end{aligned} \tag{10.42}$$

The first and last lines of the eigenvalue equation give

$$\begin{aligned} 0 &= -2\sin(\Phi_1) + 2\sin(\Phi_1 + k) - 2(\cos k - 1)\sin(\Phi_1) \\ &= 2\cos\Phi_1 \sin k \end{aligned} \tag{10.43}$$

and

$$\begin{aligned} 0 &= -2\sin(\Phi_1 + (N-1)k) + 2\sin(\Phi_1 + (N-1)k - k) \\ &\quad - 2(\cos k - 1)\sin(\Phi_1 + (N-1)k) = 2\cos(\Phi_1 + (N-1)k)\sin k \end{aligned} \tag{10.44}$$

which is solved by

$$\Phi_1 = \frac{\pi}{2} \quad k = \frac{\pi}{N-1} l, \quad l = 1, 2 \ldots N \tag{10.45}$$

hence finally the eigenvector is (Fig. 10.1)

$$\boldsymbol{f} = \begin{pmatrix} 1 \\ \vdots \\ \cos\left(\frac{n-1}{N-1}\pi l\right) \\ \vdots \\ (-1)^l \end{pmatrix}. \tag{10.46}$$

Even simpler is the case of the corresponding cyclic tridiagonal matrix

$$M = \begin{pmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 1 & & & 1 & -2 \end{pmatrix} \tag{10.47}$$

which has eigenvectors

$$f = \begin{pmatrix} \mathrm{e}^{\mathrm{i}k} \\ \vdots \\ \mathrm{e}^{\mathrm{i}nk} \\ \vdots \\ \mathrm{e}^{\mathrm{i}Nk} \end{pmatrix} \tag{10.48}$$

and eigenvalues

$$\lambda = -2 + \mathrm{e}^{-\mathrm{i}k} + \mathrm{e}^{\mathrm{i}k} = 2\left(\cos(k) - 1\right) = -4\sin^2\left(\frac{k}{2}\right) \tag{10.49}$$

where the possible $k-$ values again follow from the first and last line

$$-2\mathrm{e}^{\mathrm{i}k} + \mathrm{e}^{\mathrm{i}2k} + \mathrm{e}^{\mathrm{i}Nk} = \left(-2 + \mathrm{e}^{-\mathrm{i}k} + \mathrm{e}^{\mathrm{i}k}\right)\mathrm{e}^{\mathrm{i}k} \tag{10.50}$$

$$\mathrm{e}^{\mathrm{i}k} + \mathrm{e}^{\mathrm{i}(N-1)k} - 2\mathrm{e}^{\mathrm{i}Nk} = \left(-2 + \mathrm{e}^{-\mathrm{i}k} + \mathrm{e}^{\mathrm{i}k}\right)\mathrm{e}^{\mathrm{i}Nk} \tag{10.51}$$

which both lead to

$$\mathrm{e}^{\mathrm{i}Nk} = 1 \tag{10.52}$$

$$k = \frac{2\pi}{N}l, \quad l = 0, 1, \cdots N-1. \tag{10.53}$$

10.3.2.2 Discretized First Derivatives

Using symmetric differences to discretize a first derivative in one dimension leads to the matrix[2]

[2]This matrix is skew symmetric, hence $\mathrm{i}T$ is Hermitian and has real eigenvalues $\mathrm{i}\lambda$.

$$D = \begin{pmatrix} 1 & & & & \\ -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ & & & & -1 \end{pmatrix}. \tag{10.54}$$

The characteristic polynomial of the Hermitian matrix iD is given by the recursion Sect. 10.3.1

$$\begin{aligned} P_0 &= 1 \\ P_1 &= -\lambda \\ &\vdots \\ P_N &= -\lambda P_{N-1} - P_{N-2} \end{aligned} \tag{10.55}$$

which after the substitution $x = -\lambda/2$ is exactly the recursion for the Chebyshev polynomial of the second kind $U_N(x)$. Hence the eigenvalues of D are given by the roots x_k of $U_N(x)$ as

$$\lambda_D = 2\mathrm{i}x_k = 2\mathrm{i}\cos\left(\frac{k\pi}{N+1}\right) \quad k = 1, 2 \ldots N. \tag{10.56}$$

The eigenvalues of the corresponding cyclic tridiagonal matrix

$$D = \begin{pmatrix} 1 & & & & -1 \\ -1 & 1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ 1 & & & -1 & \end{pmatrix} \tag{10.57}$$

are easy to find. Inserting the ansatz for the eigenvector

$$\begin{pmatrix} \exp \mathrm{i}k \\ \vdots \\ \exp \mathrm{i}Nk \end{pmatrix} \tag{10.58}$$

we find the eigenvalues

$$\mathrm{e}^{\mathrm{i}(m+1)k} - \mathrm{e}^{\mathrm{i}(m-1)k} = \lambda \mathrm{e}^{\mathrm{i}mk} \tag{10.59}$$

$$\lambda = 2\mathrm{i}\sin k \tag{10.60}$$

and from the first and last equation

$$1 = \mathrm{e}^{\mathrm{i}Nk} \tag{10.61}$$

$$\mathrm{e}^{\mathrm{i}k} = \mathrm{e}^{\mathrm{i}(N+1)k} \tag{10.62}$$

the possible k-values

$$k = \frac{2\pi}{N} l, \quad l = 0, 1, \ldots N-1. \tag{10.63}$$

10.4 Reduction to a Tridiagonal Matrix

Eigenproblem algorithms work especially efficient if the matrix is first transformed to tridiagonal form (for real symmetric matrices, upper Hessian form for real non-symmetric matrices) which can be achieved by a series of Householder transformations (5.56)

$$A' = PAP \quad \text{with} \quad P = P^T = 1 - 2\frac{\boldsymbol{u}\boldsymbol{u}^T}{|\boldsymbol{u}|^2}. \tag{10.64}$$

The following orthogonal transformation P_1 brings the first row and column to tridiagonal form. We divide the matrix A according to

$$A = \begin{pmatrix} a_{11} & \boldsymbol{\alpha}^T \\ \boldsymbol{\alpha} & A_{rest} \end{pmatrix} \tag{10.65}$$

with the $(N-1)$- dimensional vector

$$\boldsymbol{\alpha} = \begin{pmatrix} a_{12} \\ \vdots \\ a_{1n} \end{pmatrix}.$$

Now let

$$\boldsymbol{u} = \begin{pmatrix} 0 \\ a_{12} + \lambda \\ \vdots \\ a_{1N} \end{pmatrix} = \begin{pmatrix} 0 \\ \boldsymbol{\alpha} \end{pmatrix} + \lambda \boldsymbol{e}^{(2)} \quad \text{with} \quad \boldsymbol{e}^{(2)} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{10.66}$$

Then

$$|\boldsymbol{u}|^2 = |\boldsymbol{\alpha}|^2 + \lambda^2 + 2\lambda a_{12} \tag{10.67}$$

and

$$\boldsymbol{u}^T \begin{pmatrix} a_{11} \\ \boldsymbol{\alpha} \end{pmatrix} = |\alpha|^2 + \lambda a_{12}. \tag{10.68}$$

The first row of A is transformed by multiplication with P_1 according to

$$P_1 \begin{pmatrix} a_{11} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} a_{11} \\ \boldsymbol{\alpha} \end{pmatrix} - 2\frac{|\alpha|^2 + \lambda a_{12}}{|\boldsymbol{\alpha}|^2 + \lambda^2 + 2\lambda a_{12}} \left[\begin{pmatrix} 0 \\ \boldsymbol{\alpha} \end{pmatrix} + \lambda \boldsymbol{e}^{(2)} \right]. \tag{10.69}$$

The elements number $3 \ldots N$ are eliminated if we choose[3]

$$\lambda = \pm|\alpha| \tag{10.70}$$

because then

$$2\frac{|\alpha|^2 + \lambda a_{12}}{|\boldsymbol{\alpha}|^2 + \lambda^2 + 2\lambda a_{12}} = 2\frac{|\alpha|^2 \pm |\alpha| a_{12}}{|\alpha|^2 + |\alpha|^2 \pm 2|\alpha| a_{12}} = 1 \tag{10.71}$$

and

$$P_1 \begin{pmatrix} a_{11} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} a_{11} \\ \boldsymbol{\alpha} \end{pmatrix} - \begin{pmatrix} 0 \\ \boldsymbol{\alpha} \end{pmatrix} - \lambda \boldsymbol{e}^{(2)} = \begin{pmatrix} a_{11} \\ \mp|\alpha| \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{10.72}$$

Finally we have

$$A^{(2)} = P_1 A P_1 = \begin{pmatrix} a_{11} & a_{12}^{(2)} & 0 & \cdots & 0 \\ a_{12}^{(2)} & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2N}^{(2)} \\ 0 & a_{23}^{(2)} & \ddots & & a_{3N}^{(2)} \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & a_{2N}^{(2)} & a_{3N}^{(2)} & \cdots & a_{NN}^{(2)} \end{pmatrix} \tag{10.73}$$

as desired.

[3] To avoid numerical extinction we choose the sign to be that of a_{12}.

For the next step we choose

$$\boldsymbol{\alpha} = \begin{pmatrix} a_{22}^{(2)} \\ \vdots \\ a_{2N}^{(2)} \end{pmatrix}, \quad \boldsymbol{u} = \begin{pmatrix} 0 \\ 0 \\ \alpha \end{pmatrix} \pm |\alpha| \boldsymbol{e}^{(3)} \tag{10.74}$$

to eliminate the elements $a_{24} \ldots a_{2N}$. Note that P_2 does not change the first row and column of $A^{(2)}$ and therefore

$$A^{(3)} = P_2 A^{(2)} P_2 = \begin{pmatrix} a_{11} & a_{12}^{(2)} & 0 & \cdots\cdots & 0 \\ a_{12}^{(2)} & a_{22}^{(2)} & a_{23}^{(3)} & 0 \cdots & 0 \\ 0 & a_{23}^{(3)} & a_{33}^{(3)} & \cdots\cdots & a_{3N}^{(3)} \\ \vdots & 0 & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & a_{3N}^{(3)} & \cdots\cdots & a_{NN}^{(3)} \end{pmatrix}. \tag{10.75}$$

After $N-1$ transformations finally a tridiagonal matrix is obtained.

10.5 The Power Iteration Method

A real symmetric $N \times N$ matrix with (orthonormal) eigenvectors and eigenvalues[4]

$$A\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i \tag{10.76}$$

can be expanded as

$$A = \sum_i \boldsymbol{u}_i \lambda_i \boldsymbol{u}_i^T. \tag{10.77}$$

The sequence of powers

$$A^n = \sum_i \boldsymbol{u}_i \lambda_i^n \boldsymbol{u}_i^T \tag{10.78}$$

converges to

$$A^n \to \boldsymbol{u}_{max} \lambda_{max}^n \boldsymbol{u}_{max}^T$$

[4] We do not consider degenerate eigenvalues explicitly here.

where[5]

$$|\lambda_{max}| = \max. \tag{10.79}$$

Hence for any initial vector $\boldsymbol{v}_1$ (which is arbitrary but not perpendicular to $\boldsymbol{u}_{max}$) the sequence

$$\boldsymbol{v}_{n+1} = A\boldsymbol{v}_n \tag{10.80}$$

converges to a multiple of $\boldsymbol{u}_{max}$. To obtain all eigenvectors simultaneously, we could use a set of independent start vectors, e.g. the N unit vectors and iterate simultaneously for all of them

$$(\boldsymbol{v}_1^{(1)}, \ldots \boldsymbol{v}_1^{(N)}) = (\boldsymbol{e}_1, \ldots \boldsymbol{e}_N) = \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix} \tag{10.81}$$

$$(\boldsymbol{v}_2^{(1)}, \ldots \boldsymbol{v}_2^{(N)}) = A(\boldsymbol{v}_1^{(1)}, \ldots \boldsymbol{v}_1^{(N)}) = A. \tag{10.82}$$

Most probably, all column vectors of A then converge to multiples of the same eigenvector. To assure linear independence, an orthogonalization step has to follow each iteration. This can be done (QR decomposition, Sect. 5.2) by decomposing the matrix into the product of an upper triangular[6] matrix R and an orthogonal matrix $Q^T = Q^{-1}$ (Sect. 5.2)

$$A = QR. \tag{10.83}$$

For symmetric tridiagonal matrices this factorization can be efficiently realized by multiplication with a sequence of Givens rotation matrices which eliminate the off-diagonal elements in the lower part one by one[7]

$$Q = R_{\alpha_{N-1}}^{(N-1,N)} \ldots R_{\alpha_2}^{(2,3)} R_{\alpha_1}^{(1,2)} \tag{10.84}$$

beginning with

$$R_{\alpha_1}^{(1,2)} A = \begin{pmatrix} c & s & & & \\ -s & c & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & & & & \\ a_{12} & a_{22} & a_{23} & & & \\ & \ddots & \ddots & \ddots & & \\ & & a_{N-2,N-1} & a_{N-1,N-1} & a_{N-1,N} & \\ & & & a_{N-1,N} & a_{N,N} & \end{pmatrix}$$

[5] For simplicity we do not consider eigenvalues which are different but have the same absolute value.

[6] The equivalent QL method uses a lower triangular matrix.

[7] This is quite different from the Jacobi method since it is not an orthogonal transformation.

$$= \begin{pmatrix} ca_{11} + sa_{12} & ca_{12} + sa_{22} & sa_{23} & & \\ 0 & -sa_{12} + ca_{22} & ca_{23} & & \\ & a_{34} & a_{33} & a_{34} & \\ & & \ddots & \ddots & \ddots \\ & & & a_{N-1,N} & a_{N,N} \end{pmatrix} \tag{10.85}$$

where the rotation angle α_1 has to be chosen such that

$$\tan \alpha_1 = \frac{s}{c} = \frac{a_{12}}{a_{11}}. \tag{10.86}$$

Finally, this leads to a method known as orthogonal simultaneous power iteration

$$W^{(1)} = A = Q^{(1)} R^{(1)} \tag{10.87}$$

$$W^{(n+1)} = AQ^{(n)} \tag{10.88}$$

$$Q^{(n+1)} R^{(n+1)} = W^{(n+1)}. \tag{10.89}$$

This method calculates a sequence of orthogonal matrices $Q^{(n)}$ which converge to a set of independent eigenvectors. Moreover, from (10.88) and (10.89)

$$A = W^{(n+1)} Q^{(n)T} = Q^{(n+1)} R^{(n+1)} Q^{(n)T} \tag{10.90}$$

and therefore powers of A are given by

$$\begin{aligned} A^n &= \left(Q^{(n)} R^{(n)} Q^{(n-1)T}\right) \left(Q^{(n-1)} R^{(n-1)} Q^{(n-2)T}\right) \ldots \left(Q^{(2)} R^{(2)} Q^{(1)T}\right) \left(Q^{(1)} R^{(1)}\right) \\ &= Q^{(n)} R^{(n)} R^{(n-1)} \ldots R^{(1)}. \end{aligned} \tag{10.91}$$

The product of two upper triangular matrices is upper triangular again which can be seen from

$$(R^{(m)} R^{(n)})_{i,k} = \sum_{j:\, i \le j \le k} R^{(m)}_{i,j} R^{(n)}_{jk} = 0 \text{ if } i > k. \tag{10.92}$$

Therefore the QR decomposition of A^n is

$$A^n = Q^{(n)} \overline{R}^{(n)} \tag{10.93}$$

with

$$\overline{R}^{(n)} = R^{(n)} \ldots R^{(1)}. \tag{10.94}$$

To obtain other than the dominant eigenvalues, the inverse power iteration method with shift is useful. Consider the matrix

$$\tilde{A} = (A - \sigma)^{-1} \tag{10.95}$$

where μ is not an eigenvalue $\lambda_i$of A. Obviously it has the same eigenvectors as A and eigenvalues given by

$$\tilde{A}\boldsymbol{u}_i = \tilde{\lambda}_i \boldsymbol{u}_i = \frac{1}{\lambda_i - \sigma}\boldsymbol{u}_i. \tag{10.96}$$

Hence, if σ is close to λ_i, the power iteration method will converge to a multiple of $\boldsymbol{u}_i$. For practical calculations, an equivalent formulation of the power iteration method is used which is known as the QR (or QL) method.

10.6 The QR Algorithm

The QR algorithm [117] is an iterative algorithm. It uses a series of orthogonal transformations which conserve the eigenvalues. Starting from the decomposition of A

$$A = Q_1 R_1 \tag{10.97}$$

$$A_2 = R_1 Q_1 = Q_1^T A Q_1 \tag{10.98}$$

we iterate

$$\vdots$$

$$A_n = Q_n R_n \tag{10.99}$$

$$A_{n+1} = R_n Q_n = Q_n^T A_n Q_n. \tag{10.100}$$

From (10.99) and (10.100)

$$Q_{n+1} R_{n+1} = R_n Q_n$$

and the n-th power of A is

$$\begin{aligned} A^n &= AA \ldots A = Q_1 R_1 Q_1 R_1 \ldots Q_1 R_1 = Q_1 (Q_2 R_2 \ldots Q_2 R_2) R_1 \\ &= Q_1 Q_2 (Q_3 R_3 \ldots Q_3 R_3) R_2 R_1 \cdots = \overline{Q}_n \overline{R}_n \end{aligned} \tag{10.101}$$

$$\overline{Q}_n = Q_1 \ldots Q_n \quad \overline{R}_n = R_n \ldots R_1. \tag{10.102}$$

But since QR decomposition is unique, comparison of (10.93) and (10.101) shows

$$\overline{Q}_n = Q^{(n)} \quad \overline{R}_n = \overline{R}^{(n)} \tag{10.103}$$

i.e. the column vectors of $\overline{Q}_n$ converge to a set of eigenvectors and the transformed matrix

$$A_{n+1} = Q_n^T A_n Q_n = Q_n^T Q_{n-1}^T A_{n-1} Q_{n-1} Q_n = \cdots = \overline{Q}_n^T A \overline{Q}_n \tag{10.104}$$

converges to a diagonal matrix. Now consider the inverse power

$$A^{-n} = \overline{R}_n^{-1} \overline{Q}_n^T. \tag{10.105}$$

The inverse of a symmetric matrix is also symmetric and

$$A^{-n} = \overline{Q}_n \left(\overline{R}_n^{-1}\right)^T \tag{10.106}$$

shows, that the QR algorithm uses the same orthogonal transformations as ordinary and also inverse power iteration. The inverse of an upper triangular matrix is also upper triangular but the transpose is lower triangular. Therefore we modify (10.106) by multiplying with a permutation matrix

$$P = \begin{pmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{pmatrix} \quad P^2 = 1 \tag{10.107}$$

from the right side, which reverses the order of the columns and

$$A^{-n} P = \overline{Q}_n P P \left(\overline{R}_n^{-1}\right)^T P = \tilde{Q}\,\tilde{R} \tag{10.108}$$

is the QR decomposition of $A^{-n}P$. This shows the close relationship between the QR algorithm[8] and the inverse power iteration method.

To improve convergence, a shift σ is introduced and the QR factorization applied to $A_n - \sigma$. The modified iteration then reads

$$A_n - \sigma = Q_n R_n \tag{10.109}$$

$$A_{n+1} = R_n Q_n + \sigma = Q_n^T (A_n - \sigma) Q_n + \sigma = Q_n^T A_n Q_n. \tag{10.110}$$

[8] Or the equivalent QL algorithm.

Symmetry and tridiagonal form are conserved by this algorithm. The simplest choice for the shift is to take a diagonal element $\sigma_m = a_{m,m} = (A_n)_{m,m}$ corresponding to the Rayleigh quotient method. An even more robust and very popular choice [113] is Wilkinson's shift

$$\sigma_m = a_{m,m} + \delta - \text{sign}(\delta)\sqrt{\delta^2 + a_{m,m-1}^2} \quad \delta = \frac{a_{m-1,m-1} - a_{m,m}}{2} \tag{10.111}$$

which is that eigenvalue of the matrix $\begin{pmatrix} a_{m-1,m-1} & a_{m-1,m} \\ a_{m-1,m} & a_{m,m} \end{pmatrix}$ which is closer to $a_{n,n}$. The calculation starts with σ_N and iterates until the off-diagonal element $a_{N-1,N}$ becomes sufficiently small.[9] Then the transformed matrix has the form

$$\begin{pmatrix} a_{11} & a_{12} & & & & \\ a_{12} & a_{22} & \ddots & & & \\ & \ddots & \ddots & a_{N-2,N-1} & & \\ & & a_{N-2,N-1} & a_{N-1,N-1} & 0 & \\ & & & 0 & a_{NN} & \end{pmatrix}. \tag{10.112}$$

Now the last column and row can be discarded (deflation method) and the next iteration performed with the shift σ_{N-1} on a tridiagonal matrix of dimension $N-1$. This procedure has to be repeated N times to obtain all eigenvalues. Convergence is usually cubic (or at least quadratic if there are degenerate eigenvalues).

10.7 Hermitian Matrices

In quantum mechanics often Hermitian matrices have to be diagonalized (which have real valued eigenvalues). To avoid complex arithmetics, an Hermitian eigenproblem can be replaced by a symmetric real valued problem of double dimension by introducing

$$B = \Re(A) \quad C = \Im(A) \quad \boldsymbol{x} = \boldsymbol{u} + \mathrm{i}\boldsymbol{v} \tag{10.113}$$

where, for Hermitian A

$$A = B + \mathrm{i}C = A^H = B^T - \mathrm{i}C^T \tag{10.114}$$

hence

$$B = B^T \quad C = -C^T \tag{10.115}$$

[9] For the QL method, it is numerically more efficient to start at the upper left corner of the matrix.

and the eigenvalue problem can be rewritten as

$$0 = (A\boldsymbol{x} - \lambda\boldsymbol{x}) = (B\boldsymbol{u} - C\boldsymbol{v} - \lambda\boldsymbol{u}) + \mathrm{i}(B\boldsymbol{v} + C\boldsymbol{u} - \lambda\boldsymbol{v}) \tag{10.116}$$

or finally in the real symmetric form

$$\begin{pmatrix} B & C^T \\ C & B \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{pmatrix} = \lambda \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{v} \end{pmatrix}. \tag{10.117}$$

Each eigenvalue of the N-dimensional Hermitian problem corresponds to two eigenvectors of the 2N-dimensional problem since for any solution of (10.117)

$$\begin{pmatrix} B & -C \\ C & B \end{pmatrix} \begin{pmatrix} -\boldsymbol{v} \\ \boldsymbol{u} \end{pmatrix} = \begin{pmatrix} -B\boldsymbol{v} - C\boldsymbol{u} \\ -C\boldsymbol{v} + B\boldsymbol{u} \end{pmatrix} = \lambda \begin{pmatrix} -\boldsymbol{v} \\ \boldsymbol{u} \end{pmatrix} \tag{10.118}$$

provides a different solution, while the complex vectors $\boldsymbol{u} + \mathrm{i}\boldsymbol{v}$ and $\mathrm{i}(\boldsymbol{u} + \mathrm{i}\boldsymbol{v}) = -\boldsymbol{v} + \mathrm{i}\boldsymbol{u}$ only differ by a phase factor.

10.8 Large Matrices

Many problems in computational physics involve very large matrices, for which standard methods are not applicable. It might be even difficult or impossible to keep the full matrix in memory. Here methods are used which only involve the product of the matrix with a vector which can be computed on the fly. Krylov methods are very similar to power iteration but diagonalize only the projection of the matrix onto a Krylov space of much smaller dimension $n \ll N$ which is constructed by multiplying a normalized start vector $\boldsymbol{q}_1$ repeatedly with A

$$K_n(A, \boldsymbol{q}_1) = \mathrm{span}\{\boldsymbol{q}_1, A\boldsymbol{q}_1, A^2\boldsymbol{q}_1, \ldots A^{n-1}\boldsymbol{q}_1\}. \tag{10.119}$$

We use the Arnoldi method (Sect. 5.6.5) to construct an orthonormalized basis of this space. For a symmetric matrix this simplifies to a three-term recursion also known as symmetric Lanczos algorithm [118]. Applying the Arnoldi method

$$h_{j,n} = (\boldsymbol{q}_j^T A \boldsymbol{q}_n) \quad j \leq n \tag{10.120}$$

$$\tilde{\boldsymbol{q}}_{n+1} = A\boldsymbol{q}_n - \sum_{j=1}^{n} h_{jn}\boldsymbol{q}_j \tag{10.121}$$

$$h_{n+1,n} = |\tilde{\boldsymbol{q}}_{n+1}| \quad \boldsymbol{q}_{n+1} = \frac{\tilde{\boldsymbol{q}}_{n+1}}{h_{n+1,n}} \tag{10.122}$$

to a symmetric matrix A, we find

$$h_{n+1,n}^2 = \boldsymbol{q}_n^T A^2 \boldsymbol{q}_n - 2\sum_j (\boldsymbol{q}_n^T A \boldsymbol{q}_j)(\boldsymbol{q}_j^T A \boldsymbol{q}_n) + \sum_{jj'} (\boldsymbol{q}_j^T A \boldsymbol{q}_n)(\boldsymbol{q}_{j'}^T A \boldsymbol{q}_n)\delta_{jj'}$$
$$= \boldsymbol{q}_n^T A \left[h_{n+1,n}\boldsymbol{q}_{n+1} + \sum_j h_{jn}\boldsymbol{q}_j \right] - \sum_j h_{jn}^2 = h_{n+1,n}h_{n,n+1} \tag{10.123}$$

hence

$$h_{n+1,n} = h_{n,n+1}. \tag{10.124}$$

Furthermore,

$$h_{n-2,n} = \boldsymbol{q}_{n-2}^T A \boldsymbol{q}_n = \boldsymbol{q}_{n-2}^T A \frac{1}{h_{n,n-1}} \left[A\boldsymbol{q}_{n-1} - \sum_{j=1}^{n-1} h_{jn-1}\boldsymbol{q}_j \right]$$
$$= \frac{1}{h_{n.n-1}} \left[-\sum_{j=1}^{n-2} h_{jn-1}h_{j,n-2} - h_{n-1,n-1}h_{n-2,n-1} + \boldsymbol{q}_{n-1}^T A \left(h_{n-1,n-2}\boldsymbol{q}_{n-1} + \sum_{j=1}^{n-2} h_{jn-2}\boldsymbol{q}_j \right) \right]$$
$$= \frac{1}{h_{n.n-1}} \left[-\sum_{j=1}^{n-2} h_{jn-1}h_{j,n-2} - h_{n-1,n-1}h_{n-2,n-1} + h_{n-1,n-1}h_{n-1,n-2} + \sum_{j=1}^{n-2} h_{jn-1}h_{jn-2} \right] = 0 \tag{10.125}$$

and similar for $s > 2$

$$h_{n-s,n} = \boldsymbol{q}_{n-s}^T A \boldsymbol{q}_n = \boldsymbol{q}_{n-s}^T A \frac{1}{h_{n,n-1}} \left[A\boldsymbol{q}_{n-1} - \sum_{j=1}^{n-1} h_{jn-1}\boldsymbol{q}_j \right]$$
$$= \frac{1}{h_{n.n-1}} \left[-\sum_{j=1}^{n-2} h_{jn-1}h_{j,n-s} - h_{n-1,n-1}h_{n-s,n-1} + \boldsymbol{q}_{n-1}^T A \left(h_{n-s+1,n-s}\boldsymbol{q}_{n-s+1} + \sum_{j=1}^{n-s} h_{jn-s}\boldsymbol{q}_j \right) \right]$$
$$= \frac{1}{h_{n.n-1}} \left[-\sum_{j=1}^{n-2} h_{jn-1}h_{j,n-s} - h_{n-1,n-1}h_{n-s,n-1} + h_{n-s+1,n-s}h_{n-1,n-s+1} + \sum_{j=1}^{n-s} h_{jn-1}h_{jn-s} \right]$$
$$= \frac{1}{h_{n,n-1}} \left[-\sum_{j=n-s+1}^{n-2} h_{j,n-1}h_{j,n-s} - h_{n-1,n-1}h_{n-s,n-1} + h_{n-s,n-s+1}h_{n-s+1,n-1} \right] \tag{10.126}$$

$$h_{n-s,n} = \frac{1}{h_{n,n-1}} \left[-\sum_{j=n-s+2}^{n-2} h_{j,n-1}h_{j,n-s} - h_{n-1,n-1}h_{n-s,n-1} \right]. \tag{10.127}$$

Starting from (10.125) for $s = 2$ we increment s repeatedly and find

$$h_{n-2,n} = h_{n-3,n} = \ldots h_{1,n} = 0 \tag{10.128}$$

since (10.127) only involves smaller values of s, for which (10.128) already has been shown. The Arnoldi decomposition produces an upper Hessenberg matrix Sect. 5.6.5,

$$U_n = (\boldsymbol{q}_1, \ldots \boldsymbol{q}_n) \tag{10.129}$$

$$AU_n = U_{n+1}H = (U_n, \boldsymbol{q}_{n+1}) \begin{pmatrix} H_n \\ h_{n+1,n}\boldsymbol{e}_n^T \end{pmatrix} = U_n H_n + h_{n+1}\boldsymbol{q}_{n+1}\boldsymbol{e}_n^T \tag{10.130}$$

which for symmetric A becomes tridiagonal

$$H = \begin{pmatrix} h_{11} & h_{12} & \ldots & h_{1n} \\ h_{21} & h_{22} & \ldots & h_{2n} \\ & h_{32} & \ddots & \\ & & \ddots & h_{nn} \\ & & & h_{n+1,n} \end{pmatrix} = \begin{pmatrix} a_1 & b_1 & & & \\ b_1 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & b_{n-2} & a_{n-1} & b_{n-1} \\ & & & b_{n-1} & a_n \\ & & & & b_n \end{pmatrix} = \begin{pmatrix} T \\ b_n\boldsymbol{e}_n^T \end{pmatrix} \tag{10.131}$$

with a symmetric tridiagonal matrix T, which is the desired projection of A into the Krylov space K_n

$$\begin{aligned} U_n^T A U_n &= U_n^T U_{n+1} H = U_n^T \ (U_n, \boldsymbol{q}_{n+1}) \begin{pmatrix} T \\ b_n\boldsymbol{e}_n^T \end{pmatrix} \\ &= (E_n, 0) \begin{pmatrix} T \\ b_n\boldsymbol{e}_n^T \end{pmatrix} = T. \end{aligned} \tag{10.132}$$

For an eigenpair $(\lambda, \boldsymbol{v})$ of T

$$\begin{aligned} A(U_n\boldsymbol{v}) &= U_{n+1}H\boldsymbol{v} = \left(U_n, \boldsymbol{q}_{n+1}\right) \begin{pmatrix} T \\ b_n\boldsymbol{e}_n^T \end{pmatrix} \boldsymbol{v} \\ &= \left(U_n T + b_n\boldsymbol{q}_{n+1}\boldsymbol{e}_n^T\right) \boldsymbol{v} = \lambda(U_n\boldsymbol{v}) + b_n\boldsymbol{q}_{n+1}\boldsymbol{e}_n^T\boldsymbol{v}. \end{aligned} \tag{10.133}$$

Hence, an approximate eigenpair of A is given by the Ritz pair $(\lambda, U_n\boldsymbol{v})$ and the error can be estimated from the residual norm

$$\frac{|(A-\lambda)U_n\boldsymbol{v}|}{|\boldsymbol{v}|} = |b_n||\boldsymbol{e}_n^T\boldsymbol{v}|. \tag{10.134}$$

Due to numerical errors, orthogonality of the Lanczos vectors $\boldsymbol{q}_n$ can get lost and reorthogonalization is necessary [119, 120]. If this takes to much time or if memory limits do not allow to store enough Lanczos vectors, the procedure has to be restarted with a new initial vector which is usually taken as a linear combination of selected eigenvectors which have already been found [121, 122]. Furthermore, special care has to be taken to determine possible degeneracies of the eigenvalues.

10.9 Non-symmetric Matrices

Eigenvalue problems with non-symmetric matrices are more complicated. Left and right eigenvectors have to be distinguished and the eigenvalues can be complex valued even if the matrix is real. The QR method [117] is applicable also to a non-symmetric matrix but very expensive unless the matrix is first brought to upper triangular (instead of tridiagonal) form, which can be achieved by a series of similarity transformations with Householder reflections (5.2.2). The implicit QR method with double shift avoids complex arithmetics by treating pairs of complex conjugated eigenvalues simultaneously. For very large matrices the Arnoldi method brings a non-symmetric matrix to upper Hessenberg form, which provides the projection onto the Krylov space as an upper triangular matrix.

Problems

Problem 10.1 Computer Experiment: Disorder in a Tight-Binding Model

We consider a two-dimensional lattice of interacting particles. Pairs of nearest neighbors have an interaction V and the diagonal energies are chosen from a Gaussian distribution

$$P(E) = \frac{1}{\Delta\sqrt{2\pi}} e^{-E^2/2\Delta^2}. \tag{10.135}$$

The wave function of the system is given by a linear combination

$$\psi = \sum_{ij} C_{ij}\psi_{ij} \tag{10.136}$$

where on each particle (i, j) one basis function ψ_{ij} is located. The nonzero elements of the interaction matrix are given by

$$H(ij|ij) = E_{ij} \tag{10.137}$$

$$H(ij|i \pm 1, j) = H(ij|i, j \pm 1) = V. \tag{10.138}$$

The Matrix H is numerically diagonalized and the amplitudes C_{ij} of the lowest state are shown as circles located at the grid points. As a measure of the degree of localization the quantity

$$\sum_{ij} |C_{ij}|^4 \tag{10.139}$$

is evaluated. Explore the influence of coupling V and disorder Δ.

Chapter 11
Data Fitting

Often a set of data points has to be fitted by a continuous function, either to obtain approximate function values in between the data points or to describe a functional relationship between two or more variables by a smooth curve, i.e. to fit a certain model to the data. If uncertainties of the data are negligibly small, an exact fit is possible, for instance with polynomials, spline functions or trigonometric functions (Chap. 2). If the uncertainties are considerable, a curve has to be constructed that fits the data points approximately. Consider a two-dimensional data set

$$(x_i, y_i) \quad i = 1 \ldots m \tag{11.1}$$

and a model function

$$f(x, a_1 \ldots a_n) \quad m \geq n \tag{11.2}$$

which depends on the variable x and $n \leq m$ additional parameters a_j. The errors of the fitting procedure are given by the residuals

$$r_i = y_i - f(x_i, a_1 \ldots a_n). \tag{11.3}$$

The parameters a_j have to be determined such, that the overall error is minimized, which in most practical cases is measured by the mean square difference[1]

$$S_{sd}(a_1 \ldots a_n) = \frac{1}{m} \sum_{i=1}^{m} r_i^2. \tag{11.4}$$

The optimal parameters are determined by solving the system of normal equations. If the model function depends linearly on the parameters, orthogonalization offers a numerically more stable method. The dimensionality of a data matrix can be reduced

[1] Minimization of the sum of absolute errors $\sum |r_i|$ is much more complicated.

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_11

with the help of singular value decomposition, which allows to approximate a matrix by another matrix of lower rank and is also useful for linear regression, especially if the columns of the data matrix are linearly dependent.

11.1 Least Square Fit

A (local) minimum of (11.4) corresponds to a stationary point with zero gradient. For n model parameters there are n, generally nonlinear, equations which have to be solved [123]. From the general condition

$$\frac{\partial S_{sd}}{\partial a_j} = 0 \quad j = 1 \ldots n \tag{11.5}$$

we find

$$\sum_{i=1}^{m} r_i \frac{\partial f(x_i, a_1 \ldots a_n)}{\partial a_j} = 0 \tag{11.6}$$

which can be solved with the methods discussed in Chap. 6. For instance, the Newton–Raphson method starts from a suitable initial guess of parameters

$$(a_1^0 \ldots a_n^0) \tag{11.7}$$

and tries to improve the fit iteratively by making small changes to the parameters

$$a_j^{s+1} = a_j^s + \Delta a_j^s. \tag{11.8}$$

The changes Δa_j^s are determined approximately by expanding the model function

$$f(x_i, a_1^{s+1} \ldots a_n^{s+1}) = f(x_i, a_1^s \ldots a_n^s) + \sum_{j=1}^{n} \frac{\partial f(x_i, a_1^s \ldots a_n^s)}{\partial a_j} \Delta a_j^s + \ldots \tag{11.9}$$

to approximate the new residuals

$$r_i^{s+1} = r_i^s - \sum_{j=1}^{n} \frac{\partial f(x_i, a_1^s \ldots a_m^s)}{\partial a_j} \Delta a_j^s \tag{11.10}$$

and the derivatives

$$\frac{\partial r_i^s}{\partial a_j} = -\frac{\partial f(x_i, a_1^s \ldots a_m^s)}{\partial a_j}. \tag{11.11}$$

Equation (11.6) now becomes

$$\sum_{i=1}^{m}\left(r_i^s - \sum_{j=1}^{n}\frac{\partial f(x_i)}{\partial a_j}\Delta a_j^s\right)\frac{\partial f(x_i)}{\partial a_k} \tag{11.12}$$

which is a system of n (usually overdetermined) linear equations for the Δa_j, the so-called normal equations:

$$\sum_{i=1}^{m}\sum_{j=1}^{n}\frac{\partial f(x_i)}{\partial a_j}\frac{\partial f(x_i)}{\partial a_k}\Delta a_j^s = \sum_{i=1}^{m} r_i^s\frac{\partial f(x_i)}{\partial a_k}. \tag{11.13}$$

With the definition

$$A_{kj} = \frac{1}{m}\sum_{i=1}^{m}\frac{\partial f(x_i)}{\partial a_k}\frac{\partial f(x_i)}{\partial a_j} \tag{11.14}$$

$$b_k = \frac{1}{m}\sum_{i=1}^{m} y_i\frac{\partial f(x_i)}{\partial a_k} \tag{11.15}$$

the normal equations can be written as

$$\sum_{j=1}^{n} A_{kj}\Delta a_j = b_k. \tag{11.16}$$

11.1.1 Linear Least Square Fit

Especially important are model functions which depend linearly on all parameters (Fig. 11.1 shows an example which is discussed in problem 11.1)

$$f(x, a_1 \ldots a_n) = \sum_{j=1}^{n} a_j f_j(x). \tag{11.17}$$

The derivatives are

$$\frac{\partial f(x_i)}{\partial a_j} = f_j(x_i) \tag{11.18}$$

and the minimum of (11.4) is given by the solution of the normal equations

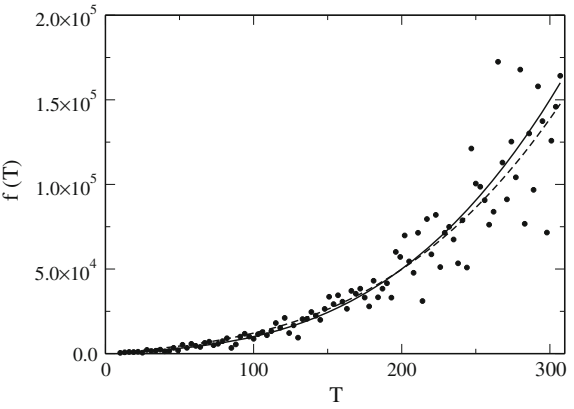


Fig. 11.1 (Least *square* fit) The polynomial $C(T) = aT + bT^3$ (*full curve*) is fitted to a set of data points which are distributed randomly around the "exact" values $C(T) = a_0T + b_0T^3$ (*dashed curve*). For more details see problem 11.1

$$\frac{1}{m}\sum_{j=1}^{n}\sum_{i=1}^{m} f_k(x_i)f_j(x_i)a_j = \frac{1}{m}\sum_{i=1}^{m} y_i f_k(x_i) \tag{11.19}$$

which for a linear fit problem become

$$\sum_{j=1}^{n} A_{kj}a_j = b_k \tag{11.20}$$

with

$$A_{kj} = \frac{1}{m}\sum_{i=1}^{m} f_k(x_i)f_j(x_i) \tag{11.21}$$

$$b_k = \frac{1}{m}\sum_{i=1}^{m} y_i f_k(x_i). \tag{11.22}$$

Example: Linear Regression

For a linear fit function

$$f(x) = a_0 + a_1 x \tag{11.23}$$

the mean square difference is

$$S_{sd} = \frac{1}{m}\sum_{i=1}^{m}(y_i - a_0 - a_1 x_i)^2 \tag{11.24}$$

and we have to solve the equations

$$0 = \frac{\partial S_{sd}}{\partial a_0} = \frac{1}{m}\sum_{i=1}^{m}(y_i - a_0 - a_1 x_i) = \overline{y} - a_0 - a_1\overline{x}$$

$$0 = \frac{\partial S_{sd}}{\partial a_1} = \frac{1}{m}\sum_{i=1}^{m}(y_i - a_0 - a_1 x_i)x_i = \overline{xy} - a_0\overline{x} - a_1\overline{x^2} \tag{11.25}$$

which can be done here with determinants

$$a_0 = \frac{\begin{vmatrix} \overline{y} & \overline{x} \\ \overline{xy} & \overline{x^2} \end{vmatrix}}{\begin{vmatrix} 1 & \overline{x} \\ \overline{x} & \overline{x^2} \end{vmatrix}} = \frac{\overline{y}\,\overline{x^2} - \overline{x}\,\overline{xy}}{\overline{x^2} - \overline{x}^2} \tag{11.26}$$

$$a_1 = \frac{\begin{vmatrix} 1 & \overline{y} \\ \overline{x} & \overline{xy} \end{vmatrix}}{\begin{vmatrix} 1 & \overline{x} \\ \overline{x} & \overline{x^2} \end{vmatrix}} = \frac{\overline{xy} - \overline{x}\,\overline{y}}{\overline{x^2} - \overline{x}^2}. \tag{11.27}$$

11.1.2 Linear Least Square Fit with Orthogonalization

With the definitions

$$\mathbf{x} = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} \tag{11.28}$$

and the $m \times n$ matrix

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} = \begin{pmatrix} f_1(x_1) & \cdots & f_n(x_1) \\ \vdots & \ddots & \vdots \\ f_1(x_m) & \cdots & f_n(x_m) \end{pmatrix} \tag{11.29}$$

the linear least square fit problem (11.20) can be formulated as a search for the minimum of

$$|A\mathbf{x} - \mathbf{b}| = \sqrt{(A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b})}. \tag{11.30}$$

In the last section we calculated the gradient

$$\frac{\partial |A\mathbf{x} - \mathbf{b}|^2}{\partial \mathbf{x}} = A^T(A\mathbf{x} - \mathbf{b}) + (A\mathbf{x} - \mathbf{b})^T A = 2A^T A\mathbf{x} - 2A^T\mathbf{b} \tag{11.31}$$

and solved the normal equations

$$A^T A\mathbf{x} = A^T\mathbf{b}. \tag{11.32}$$

This method can become numerically unstable. Alternatively we use orthogonalization of the n column vectors $\mathbf{a}_k$ of A to have

$$A = (\mathbf{a}_1 \cdots \mathbf{a}_n) = (\mathbf{q}_1 \cdots \mathbf{q}_n) \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{pmatrix} \tag{11.33}$$

where $\mathbf{a}_k$ and $\mathbf{q}_k$ are now vectors of dimension m. Since the $\mathbf{q}_k$ are orthonormal $\mathbf{q}_i^T\mathbf{q}_k = \delta_{ik}$ we have

$$\begin{pmatrix} \mathbf{q}_1^T \\ \vdots \\ \mathbf{q}_n^T \end{pmatrix} A = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & \cdots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{pmatrix}. \tag{11.34}$$

The $\mathbf{q}_k$ can be augmented by another $(m-n)$ vectors to provide an orthonormal basis of $\mathbb{R}^m$. These will not be needed explicitly. They are orthogonal to the first n vectors and hence to the column vectors of A. All vectors $\mathbf{q}_k$ together form an orthogonal matrix

$$Q = \begin{pmatrix} \mathbf{q}_1 & \cdots & \mathbf{q}_n & \mathbf{q}_{n+1} & \cdots & \mathbf{q}_m \end{pmatrix} \tag{11.35}$$

and we can define the transformation of the matrix A:

$$\widetilde{A} = \begin{pmatrix} \mathbf{q}_1^T \\ \vdots \\ \mathbf{q}_n^T \\ \mathbf{q}_{n+1}^T \\ \vdots \\ \mathbf{q}_m^T \end{pmatrix} (\mathbf{a}_1 \cdots \mathbf{a}_n) = Q^T A = \begin{pmatrix} R \\ 0 \end{pmatrix} \quad R = \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ & \ddots & \vdots \\ & & r_{nn} \end{pmatrix}. \tag{11.36}$$

The vector $\mathbf{b}$ transforms as

$$\tilde{\mathbf{b}} = Q^T\mathbf{b} = \begin{pmatrix} \mathbf{b}_u \\ \mathbf{b}_l \end{pmatrix} \quad \mathbf{b}_u = \begin{pmatrix} \mathbf{q}_1^T \\ \vdots \\ \mathbf{q}_n^T \end{pmatrix} \mathbf{b} \quad \mathbf{b}_l = \begin{pmatrix} \mathbf{q}_{n+1}^T \\ \vdots \\ \mathbf{q}_m^T \end{pmatrix} \mathbf{b}. \tag{11.37}$$

Since the norm of a vector is not changed by unitary transformations

$$|\mathbf{b} - A\mathbf{x}| = \sqrt{(\mathbf{b}_u - R\mathbf{x})^2 + \mathbf{b}_l^2} \tag{11.38}$$

which is minimized if

$$R\mathbf{x} = \mathbf{b}_u. \tag{11.39}$$

The error of the fit is given by

$$|\mathbf{b} - A\mathbf{x}| = |\mathbf{b}_l|. \tag{11.40}$$

Example: Linear Regression

Consider again the fit function

$$f(x) = a_0 + a_1 x \tag{11.41}$$

for the measured data (x_i, y_i). The fit problem is to determine

$$\left| \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_m \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} - \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} \right| = \min. \tag{11.42}$$

Orthogonalization of the column vectors

$$\mathbf{a}_1 = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \quad \mathbf{a}_2 = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} \tag{11.43}$$

with the Schmidt method gives:

$$r_{11} = \sqrt{m} \tag{11.44}$$

$$\mathbf{q}_1 = \begin{pmatrix} \frac{1}{\sqrt{m}} \\ \vdots \\ \frac{1}{\sqrt{m}} \end{pmatrix} \tag{11.45}$$

$$r_{12} = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} x_i = \sqrt{m}\,\overline{x} \tag{11.46}$$

$$\mathbf{b}_2 = (x_i - \overline{x}) \tag{11.47}$$

$$r_{22} = \sqrt{\sum (x_i - \overline{x})^2} = \sqrt{m}\sigma_x \tag{11.48}$$

$$\mathbf{q}_2 = (\frac{x_i - \overline{x}}{\sqrt{m}\sigma_x}). \tag{11.49}$$

Transformation of the right hand side gives

$$\begin{pmatrix} \mathbf{q}_1^T \\ \mathbf{q}_2^T \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} \sqrt{m}\,\overline{y} \\ \sqrt{m}\frac{\overline{xy} - \overline{x}\,\overline{y}}{\sigma_x} \end{pmatrix} \tag{11.50}$$

and we have to solve the system of linear equations

$$R\mathbf{x} = \begin{pmatrix} \sqrt{m} & \sqrt{m}\,\overline{x} \\ 0 & \sqrt{m}\sigma \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} \sqrt{m}\,\overline{y} \\ \sqrt{m}\frac{\overline{xy} - \overline{x}\,\overline{y}}{\sigma_x} \end{pmatrix}. \tag{11.51}$$

The solution

$$a_1 = \frac{\overline{xy} - \overline{x}\,\overline{y}}{\overline{(x - \overline{x})^2}} \tag{11.52}$$

$$a_0 = \overline{y} - \overline{x}a_1 = \frac{\overline{y}\,\overline{x^2} - \overline{x}\,\overline{xy}}{\overline{(x - \overline{x})^2}} \tag{11.53}$$

coincides with the earlier results since

$$\overline{(x - \overline{x})^2} = \overline{x^2} - \overline{x}^2. \tag{11.54}$$

11.2 Singular Value Decomposition

Computational physics often has to deal with large amounts of data. Singular value decomposition is a very useful tool to reduce redundancies and to extract the most important information from data. It has been used for instance for image compression [124], it is very useful to extract the essential dynamics from molecular dynamics simulations [125, 126] and it is an essential tool of Bio-informatics [127].

11.2.1 Full Singular Value Decomposition

For $m \geq n$,[2] any real[3] $m \times n$ matrix A of rank $r \leq n$ can be decomposed into a product

$$A = U\, \Sigma\, V^T \tag{11.55}$$

$$\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} = \begin{pmatrix} u_{11} & \dots & u_{1m} \\ \vdots & \ddots & \vdots \\ u_{m1} & \dots & u_{mm} \end{pmatrix} \begin{pmatrix} s_1 & & \\ & \ddots & \\ & & s_n \\ 0 & \dots & 0 \end{pmatrix} \begin{pmatrix} v_{11} & \dots & v_{n1} \\ \vdots & \ddots & \vdots \\ v_{1n} & \dots & v_{nn} \end{pmatrix} \tag{11.56}$$

where U is a $m \times m$ orthogonal matrix, Σ is a $m \times n$ matrix, in which the upper part is a $n \times n$ diagonal matrix and V is an orthogonal $n \times n$ matrix.

The diagonal elements s_i are called singular values. Conventionally, they are sorted in descending order and the last $n - r$ of them are zero. For a square $n \times n$ matrix singular value decomposition (11.56) is equivalent to diagonalization

$$A = USU^T. \tag{11.57}$$

11.2.2 Reduced Singular Value Decomposition

We write

$$U = (U_n, U_{m-n}) \tag{11.58}$$

with the $m \times n$ matrix U_n and the $m \times (m - n)$ matrix U_{m-n} and

$$\Sigma = \begin{pmatrix} S \\ 0 \end{pmatrix} \tag{11.59}$$

with the diagonal $n \times n$ matrix S. The singular value decomposition then becomes

$$A = (U_n, U_{m-n}) \begin{pmatrix} S \\ 0 \end{pmatrix} V^T = U_n S V^T \tag{11.60}$$

which is known as reduced singular value decomposition. U_n (usually simply denoted by U) is not unitary but its column vectors, called the left singular vectors, are orthonormal

[2] Otherwise consider the transpose matrix.

[3] Generalization to complex matrices is straightforward.

$$\sum_{i=1}^{m} u_{i,r} u_{i,s} = \delta_{r,s} \tag{11.61}$$

as well as the column vectors of V which are called the right singular vectors

$$\sum_{i=1}^{n} v_{i,r} v_{i,s} = \delta_{r,s}. \tag{11.62}$$

Hence the products

$$U_n^T U_n = V^T V = E_n \tag{11.63}$$

give the $n \times n$ unit matrix.

In principle, U and V can be obtained from diagonalization of $A^T A$ and AA^T, since

$$A^T A = (V \Sigma^T U^T)(U \Sigma V^T) = V(S, 0) \begin{pmatrix} S \\ 0 \end{pmatrix} V^T = V S^2 V^T \tag{11.64}$$

$$AA^T = (U \Sigma V^T)(V \Sigma^T U^T) = U \begin{pmatrix} S \\ 0 \end{pmatrix} (S, 0) U^T = U_n S^2 U_n^T. \tag{11.65}$$

However, calculation of U by diagonalization is very inefficient, since usually only the first n rows are needed (i.e. U_n). To perform a reduced singular value decomposition, we first diagonalize

$$A^T A = V D V^T \tag{11.66}$$

which has positive eigenvalues $d_i \geq 0$, sorted in descending order and obtain the singular values

$$S = D^{1/2} = \begin{pmatrix} \sqrt{d_1} & & \\ & \ddots & \\ & & \sqrt{d_n} \end{pmatrix}. \tag{11.67}$$

Now we determine a matrix U such, that

$$A = USV^T \tag{11.68}$$

or, since V is unitary

$$Y = AV = US. \tag{11.69}$$

The last $n-r$ singular values are zero if $r < n$. Therefore we partition the matrices (indices denote the number of rows)

$$\left(Y_r \; 0_{n-r}\right) = \left(U_r \; U_{n-r}\right) \begin{pmatrix} S_r \\ 0_{n-r} \end{pmatrix} = \left(U_r S_r \; 0\right). \tag{11.70}$$

We retain only the first r columns and obtain a system of equations

$$\begin{pmatrix} y_{11} & \dots & y_{1r} \\ \vdots & \ddots & \vdots \\ y_{m1} & \dots & y_{mr} \end{pmatrix} = \begin{pmatrix} u_{11} & \dots & u_{1r} \\ \vdots & \ddots & \vdots \\ u_{m1} & \dots & u_{mr} \end{pmatrix} \begin{pmatrix} s_1 & & \\ & \ddots & \\ & & s_r \end{pmatrix} \tag{11.71}$$

which can be easily solved to give the first r rows of U

$$\begin{pmatrix} u_{11} & \dots & u_{1r} \\ \vdots & \ddots & \vdots \\ u_{m1} & \dots & u_{mr} \end{pmatrix} = \begin{pmatrix} y_{11} & \dots & y_{1n} \\ \vdots & \ddots & \vdots \\ y_{m1} & \dots & y_{mn} \end{pmatrix} \begin{pmatrix} s_1^{-1} & & \\ & \ddots & \\ & & s_r^{-1} \end{pmatrix}. \tag{11.72}$$

The remaining $n-r$ column vectors of U have to be orthogonal to the first r columns but are otherwise arbitrary. They can be obtained for instance by the Gram Schmidt method.

For larger matrices direct decomposition algorithms are available, for instance [128], which is based on a reduction to bidiagonal form and a variant of the *QL* algorithm as first introduced by Golub and Kahan [129].

11.2.3 Low Rank Matrix Approximation

Component-wise (11.60) reads

$$a_{i,j} = \sum_{k=1}^{r} u_{i,k} s_k v_{j,k}. \tag{11.73}$$

Approximations to A of lower rank are obtained by reducing the sum to only the largest singular values (the smaller singular values are replaced by zero). It can be shown [130] that the matrix of rank $l \leq r$

$$a_{i,j}^{(l)} = \sum_{k=1}^{l} u_{i,k} s_k v_{j,k} \tag{11.74}$$

is the rank-l matrix which minimizes

$$\sum_{i,j} |a_{i,j} - a_{i,j}^{(l)}|^2. \tag{11.75}$$

If only the largest singular value is taken into account, A is approximated by the rank-1 matrix

$$a_{i,j}^{(1)} = s_1 u_{i,1} v_{j,1}. \tag{11.76}$$

As an example, consider a $m \times n$ matrix

$$A = \begin{pmatrix} x_1(t_1) & \dots & x_n(t_1) \\ \vdots & & \vdots \\ x_1(t_m) & \dots & x_n(t_m) \end{pmatrix} \tag{11.77}$$

which contains the values of certain quantities $x_1 \dots x_n$ observed at different times $t_1 \dots t_m$. For convenience, we assume that the average values have been subtracted, such that $\sum_{j=1}^{m} x_i = 0$. Approximation (11.76) reduces the dimensionality to 1, i.e. a linear relation between the data. The i-th row of A,

$$\begin{pmatrix} x_1(t_i) & \dots & x_n(t_i) \end{pmatrix} \tag{11.78}$$

is approximated by

$$s_1 \, u_{i,1} \begin{pmatrix} v_{1,1} & \dots & v_{n,1} \end{pmatrix} \tag{11.79}$$

which describes a direct proportionality of different observables

$$\frac{1}{v_{j,1}} x_j(t_i) = \frac{1}{v_{k,1}} x_k(t_i). \tag{11.80}$$

According to (11.75) this linear relation minimizes the mean square distance between the data points (11.78) and their approximation (11.79).

Example: Linear approximation [131]

Consider the data matrix

$$A^T = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2.5 & 3.9 & 3.5 & 4.0 \end{pmatrix}. \tag{11.81}$$

First subtract the row averages

$$\overline{x} = 3 \quad \overline{y} = 2.98 \tag{11.82}$$

to obtain

$$A^T = \begin{pmatrix} -2 & -1 & 0 & 1 & 2 \\ -1.98 & -0.48 & 0.92 & 0.52 & 1.02 \end{pmatrix}. \tag{11.83}$$

Diagonalization of

$$A^T A = \begin{pmatrix} 10.00 & 7.00 \\ 7.00 & 6.308 \end{pmatrix} \tag{11.84}$$

gives the eigenvalues

$$d_1 = 15.393 \quad d_2 = 0.915 \tag{11.85}$$

and the eigenvectors

$$V = \begin{pmatrix} 0.792 & -0.610 \\ 0.610 & -0.792 \end{pmatrix}. \tag{11.86}$$

Since there are no zero singular values we find

$$U = AVS^{-1} = \begin{pmatrix} -0.181 & -0.380 \\ -0.070 & 0.252 \\ 0.036 & 0.797 \\ 0.072 & -0.217 \\ 0.143 & -0.451 \end{pmatrix}. \tag{11.87}$$

This gives the decomposition[4]

$$A = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 \end{pmatrix} \begin{pmatrix} s_1 & \\ & s_2 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \end{pmatrix} = s_1 \mathbf{u}_1 \mathbf{v}_1^T + s_2 \mathbf{u}_2 \mathbf{v}_2^T$$

$$= \begin{pmatrix} -2.212 & -1.704 \\ -0.860 & -0.662 \\ 0.445 & 0.343 \\ 0.879 & 0.677 \\ 1.748 & 1.347 \end{pmatrix} + \begin{pmatrix} 0.212 & -0.276 \\ -0.140 & 0.182 \\ -0.445 & 0.577 \\ 0.121 & -0.157 \\ 0.252 & -0.327 \end{pmatrix}. \tag{11.88}$$

If we neglect the second contribution corresponding to the small singular value s_2 we have an approximation of the data matrix by a rank $-$ 1 matrix. The column vectors of the data matrix, denoted as $\mathbf{x}$ and $\mathbf{y}$, are approximated by

[4] $\mathbf{u}_i \mathbf{v}_i^T$ is the outer or matrix product of two vectors.

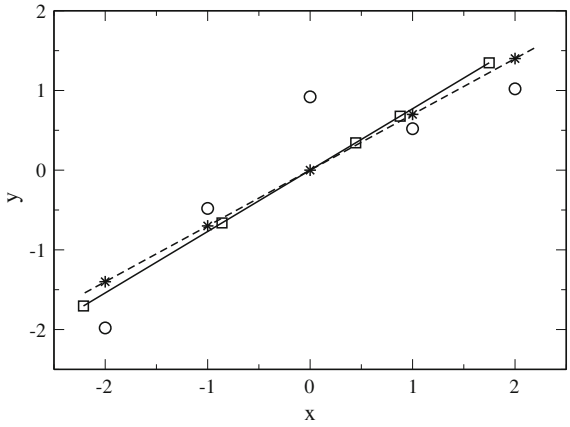


Fig. 11.2 (Linear approximation by singular value decomposition) The data set (11.81) is shown as *circles*. The linear approximation which is obtained by retaining only the dominant singular value is shown by the *squares* and the full line. It minimizes the mean *square* distance to the data points. *Stars* and the *dashed line* show the approximation by linear regression, which minimizes the mean *square* distance in *vertical* direction

$$\mathbf{x} = s_1 v_{11} \mathbf{u}_1 \quad \mathbf{y} = s_1 v_{21} \mathbf{u}_1 \tag{11.89}$$

which describes a proportionality between **x** and **y** (Fig. 11.2).

11.2.4 Linear Least Square Fit with Singular Value Decomposition

The singular value decomposition can be used for linear regression [131]. Consider a set of data, which have to be fitted to a linear function

$$y = c_0 + c_1 x_1 \cdots + c_n x_n \tag{11.90}$$

with the residual

$$r_i = c_0 + c_1 x_{i,1} \cdots + c_n x_{i,n} - y_i. \tag{11.91}$$

Let us subtract the averages

$$r_i - \overline{r} = c_1 (x_{i,1} - \overline{x}_1) \cdots + c_n (x_{i,n} - \overline{x}_n) - (y_i - \overline{y}) \tag{11.92}$$

which we write in matrix notation as

$$\begin{pmatrix} r_1 - \overline{r} \\ \vdots \\ r_m - \overline{r} \end{pmatrix} = \begin{pmatrix} x_{1,1} - \overline{x}_1 \ \ldots \ x_{1,n} - \overline{x}_n \\ \vdots \\ x_{m,1} - \overline{x}_1 \ \ldots \ x_{m,n} - \overline{x}_n \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} - \begin{pmatrix} y_1 - \overline{y} \\ \vdots \\ y_m - \overline{y} \end{pmatrix} \tag{11.93}$$

or shorter

$$\mathbf{r} = X\mathbf{c} - \mathbf{y}. \tag{11.94}$$

Now let us insert the full decomposition of X

$$\mathbf{r} = U\Sigma V^T\mathbf{c} - \mathbf{y}. \tag{11.95}$$

Since U is orthogonal

$$U^T\mathbf{r} = \Sigma V^T\mathbf{c} - U^T\mathbf{y} = \Sigma\mathbf{a} - \mathbf{b} \tag{11.96}$$

where we introduce the abbreviations

$$\mathbf{a} = V^T\mathbf{c} \quad \mathbf{b} = U^T\mathbf{y}. \tag{11.97}$$

The sum of squared residuals has the form

$$\begin{aligned} |\mathbf{r}|^2 = |U^T\mathbf{r}|^2 &= \left| \begin{pmatrix} S_r & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{a}_r \\ \mathbf{a}_{n-r} \end{pmatrix} - \begin{pmatrix} \mathbf{b}_r \\ \mathbf{b}_{n-r} \end{pmatrix} \right|^2 \\ &= |S_r\mathbf{a}_r - \mathbf{b}_r|^2 + \mathbf{b}_{n-r}^2 \leq |S_r\mathbf{a}_r - \mathbf{b}_r|^2. \end{aligned} \tag{11.98}$$

Hence $\mathbf{a}_{n-r}$ is arbitrary and one minimum of S_D is given by

$$\mathbf{a}_r = S_r^{-1}\mathbf{b}_r \quad \mathbf{a}_{n-r} = 0 \tag{11.99}$$

which can be written more compactly as

$$\mathbf{a} = \Sigma^+\mathbf{b} \tag{11.100}$$

with the Moore-Penrose pseudoinverse [132] of Σ

$$\Sigma^+ = \begin{pmatrix} s_1^{-1} & & \\ & \ddots & \\ & & s_r^{-1} \\ & & 0 \end{pmatrix}. \tag{11.101}$$

Finally we have

$$\mathbf{c} = V\Sigma^{+}U^{T}\mathbf{y} = X^{+}\mathbf{y} \tag{11.102}$$

where

$$X^{+} = V\Sigma^{+}U^{T} \tag{11.103}$$

is the Moore-Penrose pseudoinverse of X.

Example

The following data matrix has rank 2

$$X = \begin{pmatrix} -3 & -4 & -5 \\ -2 & -3 & -4 \\ 0 & 0 & 0 \\ 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} 1.0 \\ 1.1 \\ 0 \\ -1.0 \\ -1.1 \end{pmatrix}. \tag{11.104}$$

A solution to the linear fit problem is given by

$$\mathbf{c} = X^{+}\mathbf{y} = \begin{pmatrix} -0.917 & 1.167 & 0 & -1.167 & 0.917 \\ -0.167 & 0.167 & 0 & -0.167 & 0.167 \\ 0.583 & -0.833 & 0 & 0.833 & -0.583 \end{pmatrix} \begin{pmatrix} 1.0 \\ 1.1 \\ 0 \\ -1.0 \\ -1.1 \end{pmatrix} = \begin{pmatrix} 0.525 \\ 0.000 \\ -0.525 \end{pmatrix}. \tag{11.105}$$

The fit function is

$$y = 0.525(x_1 - x_3) \tag{11.106}$$

and the residuals are

$$X\mathbf{c} - \mathbf{y} = \begin{pmatrix} 0.05 \\ -0.05 \\ 0 \\ -0.05 \\ 0.05 \end{pmatrix}. \tag{11.107}$$

11.2.5 Singular and Underdetermined Linear Systems of Equations

SVD is also very useful to solve linear systems with a singular or almost singular matrix. Consider a system

$$\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \tag{11.108}$$

with $n > m$, i.e. more unknowns than equations. SVD transforms this system into

$$\begin{pmatrix} u_{11} & \dots & u_{1m} \\ \vdots & \ddots & \vdots \\ u_{m1} & \dots & u_{mm} \end{pmatrix} \begin{pmatrix} s_1 & & 0 \dots 0 \\ & \ddots & \vdots \\ & s_m & 0 \dots 0 \end{pmatrix} \begin{pmatrix} v_{11} & \dots & v_{n1} \\ \vdots & \ddots & \vdots \\ v_{1n} & \dots & v_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}. \tag{11.109}$$

Substituting

$$\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} v_{11} & \dots & v_{n1} \\ \vdots & \ddots & \vdots \\ v_{1n} & \dots & v_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \text{ and } \begin{pmatrix} c_1 \\ \vdots \\ c_m \end{pmatrix} = \begin{pmatrix} u_{11} & \dots & u_{1m} \\ \vdots & \ddots & \vdots \\ u_{m1} & \dots & u_{mm} \end{pmatrix}^{-1} \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \tag{11.110}$$

it remains to solve

$$\begin{pmatrix} s_1 & & 0 \dots 0 \\ & \ddots & \vdots \\ & s_m & 0 \dots 0 \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} c_1 \\ \vdots \\ c_m \end{pmatrix}. \tag{11.111}$$

For $y_1 \dots y_m$ the solution is

$$y_i = s_i^{-1} c_i \quad i = 1, \dots m \tag{11.112}$$

whereas $y_{m+1} \dots y_n$ are arbitrary and parametrize the solution manifold. Back substitution gives

$$\mathbf{x} = \mathbf{x}_p + \mathbf{z} \tag{11.113}$$

with the particular solution

$$\mathbf{x}_p = \begin{pmatrix} v_{11} & \dots & v_{1m} \\ \vdots & \ddots & \vdots \\ v_{n1} & \dots & v_{nm} \end{pmatrix} \begin{pmatrix} s_1^{-1} & & \\ & \ddots & \\ & & s_m^{-1} \end{pmatrix} \begin{pmatrix} u_{11} & \dots & u_{1m} \\ \vdots & \ddots & \vdots \\ u_{m1} & \dots & u_{mm} \end{pmatrix}^{-1} \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \tag{11.114}$$

and

$$\mathbf{z} = \begin{pmatrix} v_{1,m+1} & \dots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{n,m+11} & \dots & v_{nn} \end{pmatrix} \begin{pmatrix} y_{m+1} \\ \vdots \\ y_n \end{pmatrix} \tag{11.115}$$

which is in the nullspace of A

$$A\mathbf{z} = U\Sigma V^T V \begin{pmatrix} 0 \\ y_{m+1} \\ \vdots \\ y_n \end{pmatrix} = U \begin{pmatrix} S & 0 \end{pmatrix} \begin{pmatrix} 0 \\ y_{m+1} \\ \vdots \\ y_n \end{pmatrix} = 0. \tag{11.116}$$

If $m - r$ singular values are zero (or if the smallest singular values are set to zero) (11.111) becomes

$$\begin{pmatrix} s_1 & & & 0 \dots 0 \\ & \ddots & & \vdots \\ & & s_r & 0 \dots 0 \\ 0 & \dots & 0 & 0 \dots 0 \\ \vdots & & \vdots & \vdots \\ 0 & \dots & 0 & 0 \dots 0 \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} c_1 \\ \vdots \\ c_m \end{pmatrix} \tag{11.117}$$

which gives on the one hand

$$y_i = s_i^{-1} c_i \quad i = 1 \dots r \tag{11.118}$$

$$y_i = \text{arbitrary} \quad i = r + 1 \dots m \tag{11.119}$$

but also requires

$$c_i = 0 \quad i = r + 1 \dots m. \tag{11.120}$$

If this condition is not fulfilled, the equations are contradictory and no solution exists (e.g. if two rows of A are the same but the corresponding elements of **b** are different).

Problems

Problem 11.1 Least Square Fit

At temperatures far below Debye and Fermi temperatures the specific heat of a metal contains contributions from electrons and lattice vibrations and can be described by

$$C(T) = aT + bT^3. \tag{11.121}$$

The computer experiment generates data

$$T_j = T_0 + j\Delta t \tag{11.122}$$

$$C_j = (a_0 T_j + b_0 T_j^3)(1 + \varepsilon_j) \tag{11.123}$$

with relative error

$$\varepsilon_j = \varepsilon\, \xi_j. \tag{11.124}$$

Random numbers ξ_j are taken from a Gaussian normal distribution function (Sect. 9.2.6).
The fit parameters a, b are determined from minimization of the sum of squares

$$S = \frac{1}{n} \sum_{j=1}^{n} (C_j - aT_i - bT_i^3)^2. \tag{11.125}$$

Compare the "true values" a_0, b_0 with the fitted values a, b.

Chapter 12
Discretization of Differential Equations

Many processes in science and technology can be described by differential equations involving the rate of changes in time or space of a continuous variable, the unknown function. While the simplest differential equations can be solved exactly, a numerical treatment is necessary in most cases and the equations have to be discretized to turn them into a finite system of equations which can be solved by computers [133–135]. *In this chapter we discuss different methods to discretize differential equations. The simplest approach is the method of finite differences, which replaces the differential quotients by difference quotients (Chap.* 3*). It is often used for the discretization of time. Finite difference methods for the space variables work best on a regular grid. Finite volume methods are very popular in computational fluid dynamics. They take averages over small control volumes and can be easily used with irregular grids. Finite differences and finite volumes belong to the general class of finite element methods which are prominent in the engineering sciences and use an expansion in piecewise polynomials with small support. Spectral methods, on the other hand, expand the solution as a linear combination of global basis functions like polynomials or trigonometric functions. A general concept for the discretization of differential equations is the method of weighted residuals which minimizes the weighted residual of a numerical solution. Most popular is Galerkin's method which uses the expansion functions also as weight functions. Simpler are the point collocation and subdomain collocation methods which fulfill the differential equation only at certain points or averaged over certain control volumes. More demanding is the least-squares method which has become popular in computational fluid dynamics and computational electrodynamics. The least-square integral provides a measure for the quality of the solution which can be used for adaptive grid size control.*

If the Green's function is available for a problem, the method of boundary elements is an interesting alternative. It reduces the dimensionality and is, for instance, very popular in chemical physics to solve the Poisson–Boltzmann equation.

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_12

12.1 Classification of Differential Equations

An ordinary differential equation (ODE) is a differential equation for a function of one single variable, like Newton's law for the motion of a body under the influence of a force field

$$m\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathbf{x}(t) = \mathbf{F}(\mathbf{x}, t), \tag{12.1}$$

a typical initial value problem where the solution in the domain $t_0 \leq t \leq T$ is determined by position and velocity at the initial time

$$\mathbf{x}(t = t_0) = \mathbf{x}_0 \quad \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t = t_0) = \mathbf{v}_0. \tag{12.2}$$

Such equations of motion are discussed in Chap. 13. They also appear if the spatial derivatives of a partial differential equation have been discretized. Usually this kind of equation is solved by numerical integration over finite time steps $\Delta t = t_{n+1} - t_n$. Boundary value problems, on the other hand, require certain boundary conditions[1] to be fulfilled, for instance the linearized Poisson–Boltzmann equation in one dimension (Chap. 18).

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}\Phi - \kappa^2\Phi = -\frac{1}{\varepsilon}\rho(x) \tag{12.3}$$

where the value of the potential is prescribed on the boundary of the domain $x_0 \leq x \leq x_1$

$$\Phi(x_0) = \Phi_0 \quad \Phi(x_1) = \Phi_1. \tag{12.4}$$

Partial differential equations (PDE) finally involve partial derivatives with respect to at least two different variables, in many cases time and spatial coordinates.

Linear Second Order PDE

A very important class are second order linear partial differential equations of the general form

$$\left[\sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}\frac{\partial^2}{\partial x_i \partial x_j} + \sum_{i=1}^{N} b_i\frac{\partial}{\partial x_i} + c\right] f(x_1 \ldots x_N) + d = 0 \tag{12.5}$$

[1] Dirichlet b.c concern the function values, Neumann b.c. the derivative, Robin b.c. a linear combination of both, Cauchy b.c the function value and the normal derivative and mixed b.c. have different character on different parts of the boundary.

where the coefficients a_{ij}, b_i, c, d are functions of the variables $x_1 \ldots x_N$ but do not depend on the function f itself. The equation is classified according to the eigenvalues of the coefficient matrix a_{ij} as [136]

Elliptical

If all eigenvalues are positive or all eigenvalues are negative, like for the Poisson equation (Chap. 18)

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right) \Phi(x, y, z) = -\frac{1}{\varepsilon}\varrho(x, y, z), \tag{12.6}$$

Hyperbolic

If one eigenvalue is negative and all the other eigenvalues are positive or vice versa, for example the wave equation in one spatial dimension (Chap. 20).

$$\frac{\partial^2}{\partial t^2} f - c^2 \frac{\partial^2}{\partial x^2} f = 0, \tag{12.7}$$

Parabolic

If at least one eigenvalue is zero, like for the diffusion equation (Chap. 21)

$$\frac{\partial}{\partial t} f(x, y, z, t) - D\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right) f(x, y, z, t) = S(x, y, z, t), \tag{12.8}$$

Ultra-Hyperbolic

If there is no zero eigenvalue and more than one positive as well as more than one negative eigenvalue. Obviously the dimension then must be 4 at least.

Conservation Laws

One of the simplest first order partial differential equations is the 1D advection equation

$$\frac{\partial}{\partial t} f(x, t) + u \frac{\partial}{\partial x} f(x, t) = 0 \tag{12.9}$$

which describes transport of a conserved quantity with density f (for instance mass, number of particles, charge etc.) in a medium streaming with velocity u. This is a special case of the class of conservation laws (also called continuity equations)

$$\frac{\partial}{\partial t} f(\mathbf{x}, t) + \mathrm{div}\mathbf{J}(\mathbf{x}, t) = g(\mathbf{x}, t) \tag{12.10}$$

which are very common in physics. Here $\mathbf{J}$ describes the corresponding flux and g is an additional source (or sink) term. For instance the advection-diffusion equation

(also known as convection equation) has this form which describes quite general transport processes:

$$\frac{\partial}{\partial t}C = \text{ div } (D \text{ grad } C - \mathbf{u}C) + S(\mathbf{x}, t) = - \text{ div } \mathbf{J} + S(\mathbf{x}, t) \tag{12.11}$$

where one contribution to the flux

$$\mathbf{J} = -D \text{ grad } C + \mathbf{u}C \tag{12.12}$$

is proportional to the gradient of the concentration C (Fick's first law) and the second part depends on the velocity field $\mathbf{u}$ of a streaming medium. The source term S represents the effect of chemical reactions. Equation (12.11) is also similar to the drift-diffusion equation in semiconductor physics and closely related to the Navier Stokes equations which are based on the Cauchy momentum equation [137]

$$\varrho \frac{d\mathbf{u}}{dt} = \varrho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \text{ grad } \mathbf{u} \right) = \text{div}\sigma + \mathbf{f} \tag{12.13}$$

where σ denotes the stress tensor. Equation (12.10) is the strong or differential form of the conservation law. The requirements on the smoothness of the solution are reduced by using the integral form which is obtained with the help of Gauss' theorem

$$\int_V \left(\frac{\partial}{\partial t} f(\mathbf{x}, t) - g(\mathbf{x}, t) \right) dV + \oint_{\partial V} \mathbf{J}(\mathbf{x}, t) d\mathbf{A} = 0. \tag{12.14}$$

An alternative integral form results from Galerkin's [138] method of weighted residuals which introduces a weight function $w(\mathbf{x})$ and considers the equation

$$\int_V \left(\frac{\partial}{\partial t} f(\mathbf{x}, t) + \text{div}\mathbf{J}(\mathbf{x}, t) - g(\mathbf{x}, t) \right) w(\mathbf{x}) \, dV = 0 \tag{12.15}$$

or after applying Gauss' theorem

$$\int_V \left\{ \left(\frac{\partial}{\partial t} f(\mathbf{x}, t) - g(\mathbf{x}, t) \right) w(\mathbf{x}) - \mathbf{J}(\mathbf{x}, t) \text{ grad} w(\mathbf{x}) \right\} dV$$
$$+ \oint_{\partial V} w(\mathbf{x}) \mathbf{J}(\mathbf{x}, t) d\mathbf{A} = 0. \tag{12.16}$$

The so called weak form of the conservation law states that this equation holds for arbitrary weight functions w.

12.2 Finite Differences

The simplest method to discretize a differential equation is to introduce a grid of equidistant points and to discretize the differential operators by finite differences (FDM) as described in Chap. 3. For instance, in one dimension the first and second derivatives can be discretized by

$$x \to x_m = m\Delta x \quad m = 1 \ldots M \tag{12.17}$$

$$f(x) \to f_m = f(x_m) \quad m = 1 \ldots M \tag{12.18}$$

$$\frac{\partial f}{\partial x} \to \left(\frac{\partial}{\partial x} f\right)_m = \frac{f_{m+1} - f_m}{\Delta x} \text{ or } \left(\frac{\partial}{\partial x} f\right)_m = \frac{f_{m+1} - f_{m-1}}{2\Delta x} \tag{12.19}$$

$$\frac{\partial^2 f}{\partial x^2} \to \left(\frac{\partial^2}{\partial x^2} f\right)_m = \frac{f_{m+1} + f_{m-1} - 2f_m}{\Delta x^2} \quad . \tag{12.20}$$

These expressions are not well defined at the boundaries of the grid $m = 1, M$ unless the boundary conditions are taken into account. For instance, in case of a Dirichlet problem f_0 and f_{M+1} are given boundary values and

$$\left(\frac{\partial}{\partial x} f\right)_1 = \frac{f_2 - f_0}{2\Delta x} \quad \left(\frac{\partial^2}{\partial x^2} f\right)_1 = \frac{f_2 - 2f_1 + f_0}{\Delta x^2} \tag{12.21}$$

$$\left(\frac{\partial}{\partial x} f\right)_M = \frac{f_{M+1} - f_M}{\Delta x} \text{ or } \frac{f_{M+1} - f_{M-1}}{2\Delta x} \quad \left(\frac{\partial^2}{\partial x^2} f\right)_M = \frac{f_{M-1} - 2f_M + f_{M+1}}{\Delta x^2}. \tag{12.22}$$

Other kinds of boundary conditions can be treated in a similar way.

12.2.1 Finite Differences in Time

Time derivatives can be treated similarly using an independent time grid

$$t \to t_n = n\Delta t \quad n = 1 \ldots N \tag{12.23}$$

$$f(t, x) \to f_m^n = f(t_n, x_m) \tag{12.24}$$

and finite differences like the first order forward difference quotient

$$\frac{\partial f}{\partial t} \to \frac{f_m^{n+1} - f_m^n}{\Delta t} \tag{12.25}$$

or the symmetric difference quotient

$$\frac{\partial f}{\partial t} \to \frac{f_m^{n+1} - f_m^{n-1}}{2\Delta t} \tag{12.26}$$

to obtain a system of equations for the function values at the grid-points f_m^n. For instance for the diffusion equation in one spatial dimension

$$\frac{\partial f(x,t)}{\partial t} = D\frac{\partial^2}{\partial x^2} f(x,t) + S(x,t) \tag{12.27}$$

the simplest discretization is the FTCS (forward in time, centered in space) scheme

$$(f_m^{n+1} - f_m^n) = D\frac{\Delta t}{\Delta x^2}(f_{m+1}^n + f_{m-1}^n - 2f_m^n) + S_m^n \Delta t \tag{12.28}$$

which can be written in matrix notation as

$$\mathbf{f}_{n+1} - \mathbf{f}_n = D\frac{\Delta t}{\Delta x^2} M\mathbf{f}_n + \mathbf{S}_n \Delta t \tag{12.29}$$

with

$$\mathbf{f}_n = \begin{pmatrix} f_1^n \\ f_2^n \\ f_3^n \\ \vdots \\ f_M^n \end{pmatrix} \quad \text{and} \quad M = \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -2 \end{pmatrix}. \tag{12.30}$$

12.2.2 Stability Analysis

Fully discretized linear differential equations provide an iterative algorithm of the type[2]

$$\mathbf{f}_{n+1} = A\mathbf{f}_n + \mathbf{S}_n \Delta t \tag{12.31}$$

which propagates numerical errors according to

$$\mathbf{f}_{n+1} + \boldsymbol{\epsilon}_{n+1} = A(\mathbf{f}_n + \boldsymbol{\epsilon}_n) + \mathbf{S}_n \Delta t \tag{12.32}$$

[2]Differential equations which are higher order in time can be always brought to first order by introducing the time derivatives as additional variables.

$$\epsilon_{j+1} = A\epsilon_j. \tag{12.33}$$

Errors are amplified exponentially if the absolute value of at least one eigenvalue of A is larger than one. The algorithm is stable if all eigenvalues of A are smaller than one in absolute value (1.4). If the eigenvalue problem is difficult to solve, the von Neumann analysis is helpful which decomposes the errors into a Fourier series and considers the Fourier components individually by setting

$$\mathbf{f}_n = g^n(k) \begin{pmatrix} \mathrm{e}^{\mathrm{i}k} \\ \vdots \\ \mathrm{e}^{\mathrm{i}kM} \end{pmatrix} \tag{12.34}$$

and calculating the amplification factor

$$\left| \frac{f_m^{n+1}}{f_m^n} \right| = |g(k)| \,. \tag{12.35}$$

The algorithm is stable if $|g(k)| \leq 1$ for all k.

Example For the discretized diffusion equation (12.28) we find

$$g^{n+1}(k) = g^n(k) + 2D\frac{\Delta t}{\Delta x^2} g^n(k)\,(\cos k - 1) \tag{12.36}$$

$$g(k) = 1 + 2D\frac{\Delta t}{\Delta x^2}(\cos k - 1) = 1 - 4D\frac{\Delta t}{\Delta x^2}\sin^2\left(\frac{k}{2}\right) \tag{12.37}$$

$$1 - 4D\frac{\Delta t}{\Delta x^2} \leq g(k) \leq 1 \tag{12.38}$$

hence stability requires

$$D\frac{\Delta t}{\Delta x^2} \leq \frac{1}{2}. \tag{12.39}$$

12.2.3 Method of Lines

Alternatively time can be considered as a continuous variable. The discrete values of the function then are functions of time (so called lines)

$$f_m(t) \tag{12.40}$$

and a set of ordinary differential equations has to be solved. For instance for diffusion in one dimension (12.27) the equations

$$\frac{\mathrm{d}f_m}{\mathrm{d}t} = \frac{D}{h^2}\left(f_{m+1} + f_{m-1} - 2f_m\right) + S_m(t) \tag{12.41}$$

which can be written in matrix notation as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} f_1 \\ f_1 \\ f_2 \\ \vdots \\ f_M \end{pmatrix} = \frac{D}{\Delta x^2}\begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -2 \end{pmatrix}\begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_M \end{pmatrix} + \begin{pmatrix} S_1 + \frac{D}{h^2} f_0 \\ S_2 \\ S_3 \\ \vdots \\ S_M + \frac{D}{h^2} f_{M+1} \end{pmatrix} \tag{12.42}$$

or briefly

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{f}(t) = A\mathbf{f}(t) + \mathbf{S}(t). \tag{12.43}$$

Several methods to integrate such a semi-discretized equation will be discussed in Chap. 13. If eigenvectors and eigenvalues of A are easy available, an eigenvector expansion can be used.

12.2.4 Eigenvector Expansion

A homogeneous system

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{f}(t) = A\mathbf{f}(t) \tag{12.44}$$

where the matrix A is obtained from discretizing the spatial derivatives, can be solved by an eigenvector expansion. From the eigenvalue problem

$$A\mathbf{f} = \lambda\mathbf{f} \tag{12.45}$$

we obtain the eigenvalues λ and eigenvectors $\mathbf{f}_\lambda$ which provide the particular solutions:

$$\mathbf{f}(t) = e^{\lambda t}\,\mathbf{f}_\lambda \tag{12.46}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}(e^{\lambda t}\,\mathbf{f}_\lambda) = \lambda(e^{\lambda t}\,\mathbf{f}_\lambda) = A(e^{\lambda t}\,\mathbf{f}_\lambda). \tag{12.47}$$

These can be used to expand the general solution

$$\mathbf{f}(t) = \sum_{\lambda} C_\lambda e^{\lambda t} \mathbf{f}_\lambda. \tag{12.48}$$

The coefficients C_λ follow from the initial values by solving the linear equations

$$\mathbf{f}(t=0) = \sum_{\lambda} C_\lambda \mathbf{f}_\lambda. \tag{12.49}$$

If the differential equation is second order in time

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathbf{f}(t) = A\mathbf{f}(t) \tag{12.50}$$

the particular solutions are

$$\mathbf{f}(t) = e^{\pm t\sqrt{\lambda}}\,\mathbf{f}_\lambda \tag{12.51}$$

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}(e^{\pm t\sqrt{\lambda}}\,\mathbf{f}_\lambda) = \lambda(e^{\pm t\sqrt{\lambda}}\,\mathbf{f}_\lambda) = A(e^{\pm t\sqrt{\lambda}}\,\mathbf{f}_\lambda) \tag{12.52}$$

and the eigenvector expansion is

$$\mathbf{f}(t) = \sum_{\lambda} \left(C_{\lambda+} e^{t\sqrt{\lambda}} \; + \; C_{\lambda-} e^{-t\sqrt{\lambda}} \right) \mathbf{f}_\lambda. \tag{12.53}$$

The coefficients $C_{\lambda\pm}$ follow from the initial amplitudes and velocities

$$\begin{aligned} \mathbf{f}(t=0) &= \sum_{\lambda} (C_{\lambda+} + C_{\lambda-})\mathbf{f}_\lambda \\ \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{f}(t=0) &= \sum_{\lambda} \sqrt{\lambda}(C_{\lambda+} - C_{\lambda-})\mathbf{f}_\lambda. \end{aligned} \tag{12.54}$$

For a first order inhomogeneous system

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{f}(t) = A\mathbf{f}(t) + \mathbf{S}(t) \tag{12.55}$$

the expansion coefficients have to be time dependent

$$\mathbf{f}(t) = \sum_{\lambda} C_\lambda(t) e^{\lambda t} \mathbf{f}_\lambda \tag{12.56}$$

and satisfy

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{f}(t) - A\mathbf{f}(t) = \sum_{\lambda} \frac{\mathrm{d}C_\lambda}{\mathrm{d}t} \mathrm{e}^{\lambda t}\mathbf{f}_\lambda = \mathbf{S}(t). \tag{12.57}$$

After taking the scalar product with $\mathbf{f}_\mu$[3]

$$\frac{\mathrm{d}C_\mu}{\mathrm{d}t} = \mathrm{e}^{-\mu t}\left(\mathbf{f}_\mu \mathbf{S}(t)\right) \tag{12.58}$$

can be solved by a simple time integration. For a second order system

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathbf{f}(t) = A\mathbf{f}(t) + \mathbf{S}(t) \tag{12.59}$$

we introduce the first time derivative as a new variable

$$\mathbf{g} = \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{f} \tag{12.60}$$

to obtain a first order system of double dimension

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\mathbf{f}\\ \mathbf{g}\end{pmatrix} = \begin{pmatrix}0 & 1\\ A & 0\end{pmatrix}\begin{pmatrix}\mathbf{f}\\ \mathbf{g}\end{pmatrix} + \begin{pmatrix}\mathbf{S}\\ 0\end{pmatrix} \tag{12.61}$$

where eigenvectors and eigenvalues can be found from those of A (12.45)

$$\begin{pmatrix}0 & 1\\ A & 0\end{pmatrix}\begin{pmatrix}\mathbf{f}_\lambda\\ \pm\sqrt{\lambda}\mathbf{f}_\lambda\end{pmatrix} = \begin{pmatrix}\pm\sqrt{\lambda}\mathbf{f}_\lambda\\ \lambda\mathbf{f}_\lambda\end{pmatrix} = \pm\sqrt{\lambda}\begin{pmatrix}\mathbf{f}_\lambda\\ \pm\sqrt{\lambda}\mathbf{f}_\lambda\end{pmatrix} \tag{12.62}$$

$$\left(\pm\sqrt{\lambda}\mathbf{f}_\lambda^T\ \mathbf{f}_\lambda^T\right)\begin{pmatrix}0 & 1\\ A & 0\end{pmatrix} = \left(\lambda\mathbf{f}_\lambda^T\ \pm\sqrt{\lambda}\mathbf{f}_\lambda^T\right) = \pm\sqrt{\lambda}\left(\pm\sqrt{\lambda}\mathbf{f}_\lambda^T\ \mathbf{f}_\lambda^T\right). \tag{12.63}$$

Insertion of

$$\sum_{\lambda} C_{\lambda+}\mathrm{e}^{\sqrt{\lambda}t}\begin{pmatrix}\mathbf{f}_\lambda\\ \sqrt{\lambda}\mathbf{f}_\lambda\end{pmatrix} + C_{\lambda-}\mathrm{e}^{-\sqrt{\lambda}t}\begin{pmatrix}\mathbf{f}_\lambda\\ -\sqrt{\lambda}\mathbf{f}_\lambda\end{pmatrix}$$

gives

$$\sum_{\lambda} \frac{\mathrm{d}C_{\lambda+}}{\mathrm{d}t}\mathrm{e}^{\sqrt{\lambda}t}\begin{pmatrix}\mathbf{f}_\lambda\\ \sqrt{\lambda}\mathbf{f}_\lambda\end{pmatrix} + \frac{\mathrm{d}C_{\lambda-}}{\mathrm{d}t}\mathrm{e}^{\sqrt{\lambda}t}\begin{pmatrix}\mathbf{f}_\lambda\\ -\sqrt{\lambda}\mathbf{f}_\lambda\end{pmatrix} = \begin{pmatrix}\mathbf{S}(t)\\ 0\end{pmatrix} \tag{12.64}$$

[3] If A is not Hermitian we have to distinguish left- and right-eigenvectors.

and taking the scalar product with one of the left-eigenvectors we end up with

$$\frac{\mathrm{d}C_{\lambda+}}{\mathrm{d}t} = \frac{1}{2}\left(\mathbf{f}_\lambda \mathbf{S}(t)\right) \mathrm{e}^{-\sqrt{\lambda}t} \tag{12.65}$$

$$\frac{\mathrm{d}C_{\lambda-}}{\mathrm{d}t} = -\frac{1}{2}\left(\mathbf{f}_\lambda \mathbf{S}(t)\right) \mathrm{e}^{\sqrt{\lambda}t}. \tag{12.66}$$

12.3 Finite Volumes

Whereas the finite differences method uses function values

$$f_{i,j,k} = f(x_i, y_j, z_k) \tag{12.67}$$

at the grid points

$$\mathbf{r}_{ijk} = (x_i, y_j, z_k), \tag{12.68}$$

the finite volume method (FVM) [139] averages function values and derivatives over small control volumes V_r which are disjoint and span the domain V (Fig. 12.1)

$$V = \bigcup_r V_r \qquad V_r \bigcap V_{r'} = \emptyset \forall r \neq r'. \tag{12.69}$$

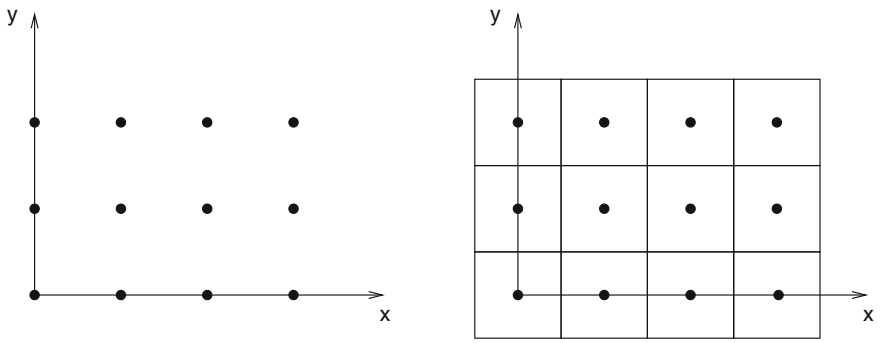


Fig. 12.1 (Finite volume method) The domain V is divided into small control volumes V_r, in the simplest case cubes around the grid points $\mathbf{r}_{ijk}$

The averages are

$$\overline{f_r} = \frac{1}{V_r} \int_{V_r} dV\, f(\mathbf{r}) \tag{12.70}$$

or in the simple case of cubic control volumes of equal size h^3

$$\overline{f}_{ijk} = \frac{1}{h^3} \int_{x_i - h/2}^{x_i + h/2} dx \int_{y_j - h/2}^{y_j + h/2} dy \int_{z_k - h/2}^{z_k + h/2} dz\, f(x, y, z). \tag{12.71}$$

Such average values have to be related to discrete function values by numerical integration (Chap. 4). The midpoint rule (4.17), for instance replaces the average by the central value

$$\overline{f}_{ijk} = f(x_i, y_j, z_k) + O(h^2) \tag{12.72}$$

whereas the trapezoidal rule (4.13) implies the average over the eight corners of the cube

$$\overline{f}_{ijk} = \frac{1}{8} \sum_{m,n,p=\pm 1} f(x_{i+m/2}, y_{j+n/2}, z_{k+p/2}) + O(h^2). \tag{12.73}$$

In (12.73) the function values refer to a dual grid (Fig. 12.2) [139] centered around the vertices of the original grid (12.68).

$$\mathbf{r}_{i+1/2, j+1/2, k+1/2} = \left(x_i + \frac{h}{2}, y_j + \frac{h}{2}, z_k + \frac{h}{2} \right). \tag{12.74}$$

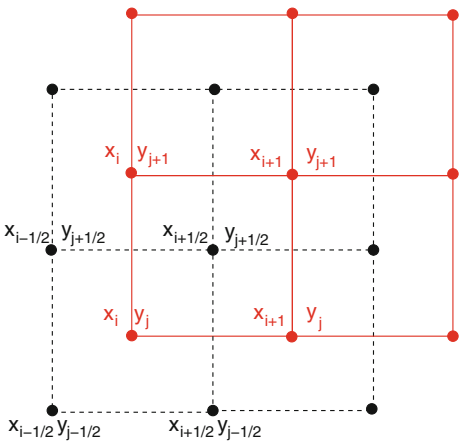


Fig. 12.2 (Dual grid) The dual grid (*black*) is centered around the vertices of the original grid (*red*)

The average gradient can be rewritten using the generalized Stokes' theorem as

$$\overline{\text{grad } f_{ijk}} = \frac{1}{V}\int_{V_{ijk}} dV \text{ grad } f(\mathbf{r}) = \oint_{\partial V_{ijk}} f(\mathbf{r})\, d\mathbf{A}. \tag{12.75}$$

For a cubic grid we have to integrate over the six faces of the control volume

$$\overline{\text{grad } f_{ijk}} = \frac{1}{h^3}\begin{pmatrix} \int_{z_k-h/2}^{z_k+h/2} dz \int_{y_j-h/2}^{y_j+h/2} dy \ \left(f(x_i+\frac{h}{2},y,z) - f(x_i-\frac{h}{2},y,z)\right) \\ \int_{z_k-h/2}^{z_k+h/2} dz \int_{x_i-h/2}^{x_i+h/2} dx \ \left(f(x_i,y+\frac{h}{2},z) - f(x_i,y-\frac{h}{2},z)\right) \\ \int_{x_i-h/2}^{x_i+h/2} dx \int_{y_j-h/2}^{y_j+h/2} dy \ \left(f(x_i,y,z+\frac{h}{2}) - f(x_i,y,z-\frac{h}{2})\right) \end{pmatrix}. \tag{12.76}$$

The integrals have to be evaluated numerically. Applying as the simplest approximation the midpoint rule (4.17)

$$\int_{x_i-h/2}^{x_i+h/2} dx \int_{y_j-h/2}^{y_j+h/2} dy\, f(x,y) = h^2\left(f(x_i,y_j) + O(h^2)\right) \tag{12.77}$$

this becomes

$$\overline{\text{grad } f_{ijk}} = \frac{1}{h}\begin{pmatrix} f(x_i+\frac{h}{2},y_j,z_k) - f(x_i-\frac{h}{2},y_j,z_k) \\ f(x_i,y_j+\frac{h}{2},z_k) - f(x_i,y_j-\frac{h}{2},z_k) \\ f(x_i,y_j,z_k+\frac{h}{2}) - f(x_i,y_j,z_k-\frac{h}{2}) \end{pmatrix} \tag{12.78}$$

which involves symmetric difference quotients. However, the function values in (12.78) refer neither to the original nor to the dual grid. Therefore we interpolate (Fig. 12.3).

$$f(x_i \pm \frac{h}{2}, y_j, z_k) \approx \frac{1}{2}\left(f(x_{i+1},y_j,z_i) + f(x_{i-1},y_j,z_i)\right) \tag{12.79}$$

$$\begin{aligned} &\frac{1}{2}\left(f(x_i+\frac{h}{2},y_j,z_k) - f(x_i-\frac{h}{2},y_j,z_k\right) \\ &\approx \frac{1}{2h}\left(f(x_{i+1},y_j,z_k) - f(x_{i-1},y_j,z_k\right) \end{aligned} \tag{12.80}$$

or

$$f\left(x_i \pm \frac{h}{2}, y_j, z_k\right) \approx \frac{1}{4}\sum_{m,n=\pm 1} f\left(x_i \pm \frac{h}{2}, y_j + m\frac{h}{2}, z_k + n\frac{h}{2}\right). \tag{12.81}$$

The finite volume method is capable of treating discontinuities and is very flexible concerning the size and shape of the control volumes.

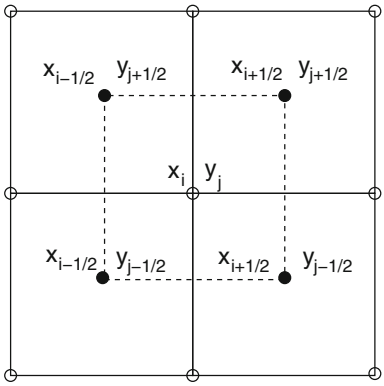


Fig. 12.3 (Interpolation between grid points) Interpolation is necessary to relate the averaged gradient (12.78) to the original or dual grid

12.3.1 Discretization of fluxes

Integration of (12.10) over a control volume and application of Gauss' theorem gives the integral form of the conservation law

$$\frac{1}{V}\oint \mathbf{J}d\mathbf{A} + \frac{\partial}{\partial t}\frac{1}{V}\int f\,dV = \frac{1}{V}\int g\,dV \tag{12.82}$$

which involves the flux $\mathbf{J}$ of some property like particle concentration, mass, energy or momentum density or the flux of an electromagnetic field. The total flux through a control volume is given by the surface integral

$$\Phi = \oint_{\partial V} \mathbf{J}d\mathbf{A} \tag{12.83}$$

which in the special case of a cubic volume element of size h^3 becomes the sum over the six faces of the cube (Fig. 12.4).

$$\begin{aligned}
\Phi &= \sum_{r=1}^{6}\int_{A_r}\mathbf{J}d\mathbf{A} \\
&= \int_{x_i-h/2}^{x_i+h/2} dx \int_{y_j-h/2}^{y_j+h/2} dy \left(J_z\left(x,y,z_k+\frac{h}{2}\right) - J_z\left(x,y,z_k-\frac{h}{2}\right)\right. \\
&\quad + \int_{x_i-h/2}^{x_i+h/2} dx \int_{z_k-h/2}^{z_k+h/2} dz \left(J_y\left(x,y_j+\frac{h}{2},z\right) - J_z\left(x,y_j-\frac{h}{2},z\right)\right. \\
&\quad + \int_{z_k-h/2}^{z_k+h/2} dz \int_{y_j-h/2}^{y_j+h/2} dy \left(J_x\left(x_i+\frac{h}{2},y,z\right) - J_z\left(x_i-\frac{h}{2},y,z\right)\right.
\end{aligned} \tag{12.84}$$

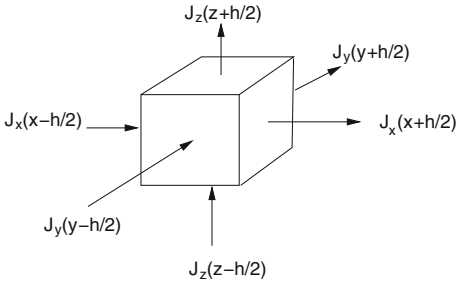


Fig. 12.4 Flux through a control volume

The surface integral can be evaluated numerically (Chap. 4). Using the midpoint approximation (12.77) we obtain

$$\frac{1}{V}\Phi(x_i, y_i, z_i) = \frac{1}{h}\left(J_z(x_i, y_j, z_{k+1/2} - J_z(x_i, y_j, z_{k-1/2})\right.$$
$$\left.+J_y(x_i y_{j+1/2}, z_k) - J_y(x_i, y_{j-1/2}, z_k) + J_x(x_{i+1/2}, y_j, z_k) - J_x(x_{i-1/2}, y_j, z_k)\right). \tag{12.85}$$

The trapezoidal rule (4.13) introduces an average over the four corners (Fig. 12.3)

$$\int_{x_i-h/2}^{x_i+h/2} dx \int_{y_j-h/2}^{y_j+h/2} dy\, f(x, y)$$
$$= h^2\left(\frac{1}{4}\sum_{m,n=\pm 1} f(x_{i+m/2}, y_{j+n/2}) + O(h^2)\right) \tag{12.86}$$

which replaces the flux values in (12.85) by the averages

$$J_x(x_{i\pm 1/2}, y_j, z_k) = \frac{1}{4}\sum_{m,n=\pm 1} J_z(x_{i\pm 1/2}, y_{j+m/2}, z_{k+n/2}) \tag{12.87}$$

$$J_y(x_i, y_{j\pm 1/2}, z_k) = \frac{1}{4}\sum_{m,n=\pm 1} J_z(x_{i+m/2}, y_{j\pm 1/2}, z_{k+n/2}) \tag{12.88}$$

$$J_z(x_i, y_j, z_{k\pm 1/2}) = \frac{1}{4}\sum_{m,n=\pm 1} J_z(x_{i+m/2}, y_{j+n/2}, z_{k\pm 1/2}). \tag{12.89}$$

One advantage of the finite volume method is that the flux is strictly conserved.

12.4 Weighted Residual Based Methods

A general method to discretize partial differential equations is to approximate the solution within a finite dimensional space of trial functions.[4] The partial differential equation is turned into a finite system of equations or a finite system of ordinary differential equations if time is treated as a continuous variable. This is the basis of spectral methods which make use of polynomials or Fourier series but also of the very successful finite element methods. Even finite difference methods and finite volume methods can be formulated as weighted residual based methods.

Consider a differential equation[5] on the domain V which is written symbolically with the differential operator $\mathcal{T}$

$$\mathcal{T}[u(\mathbf{r})] = f(\mathbf{r}) \quad \mathbf{r} \in V \tag{12.90}$$

and corresponding boundary conditions which are expressed with a boundary operator $\mathcal{B}$[6]

$$\mathcal{B}[u(\mathbf{r})] = g(\mathbf{r}) \quad \mathbf{r} \in \partial V. \tag{12.91}$$

The basic principle to obtain an approximate solution $\tilde{u}(\mathbf{r})$ is to choose a linear combination of expansion functions $N_i(\mathbf{r}) \quad i = 1 \ldots r$ as a trial function which fulfills the boundary conditions[7]

$$\tilde{u} = \sum_{i=1}^{r} u_i N_i(\mathbf{r}) \tag{12.92}$$

$$\mathcal{B}[\tilde{u}(\mathbf{r})] = g(\mathbf{r}). \tag{12.93}$$

In general (12.92) is not an exact solution and the residual

$$R(\mathbf{r}) = \mathcal{T}\left[\tilde{u}\right](\mathbf{r}) - f(\mathbf{r}) \tag{12.94}$$

will not be zero throughout the whole domain V. The function $\tilde{u}$ has to be determined such that the residual becomes "small" in a certain sense. To that end weight functions[8] $w_j \quad j = 1 \ldots r$ are chosen to define the weighted residuals

[4] Also called expansion functions.

[5] Generalization to systems of equations is straightforward.

[6] One or more linear differential operators, usually a combination of the function and its first derivatives.

[7] This requirement can be replaced by additional equations for the u_i, for instance with the tau method [140].

[8] Also called test functions.

$$R_j\left(u_1 \ldots u_r\right) = \int dV\, w_j(\mathbf{r}) \left(\mathcal{T}\left[\tilde{u}\right](\mathbf{r}) - f(\mathbf{r})\right). \tag{12.95}$$

The optimal parameters u_i are then obtained from the solution of the equations

$$R_j(u_1 \ldots u_r) = 0 \quad j = 1 \ldots r. \tag{12.96}$$

In the special case of a linear differential operator these equations are linear

$$\sum_{i=1}^{r} u_i \int dV\, w_j(\mathbf{r}) \mathcal{T}\left[N_i(\mathbf{r})\right] - \int dV\, w_j(\mathbf{r}) f(\mathbf{r}) = 0. \tag{12.97}$$

Several strategies are available to choose suitable weight functions.

12.4.1 Point Collocation Method

The collocation method uses the weight functions $w_j(\mathbf{r}) = \delta(\mathbf{r} - \mathbf{r}_j)$, with certain collocation points $\mathbf{r}_j \in V$. The approximation $\tilde{u}$ obeys the differential equation at the collocation points

$$0 = R_j = \mathcal{T}[\tilde{u}](\mathbf{r}_j) - f(\mathbf{r}_j) \tag{12.98}$$

and for a linear differential operator

$$0 = \sum_{i=1}^{r} u_i\, \mathcal{T}[N_i](\mathbf{r}_j) - f(\mathbf{r}_j). \tag{12.99}$$

The point collocation method is simple to use, especially for nonlinear problems. Instead of using trial functions satisfying the boundary conditions, extra collocation points on the boundary can be added (mixed collocation method).

12.4.2 Sub-domain Method

This approach uses weight functions which are the characteristic functions of a set of control volumes V_i which are disjoint and span the whole domain similar as for the finite volume method

$$V = \bigcup_j V_j \qquad V_j \bigcap V_{j'} = \emptyset \forall j \neq j' \tag{12.100}$$

$$w_j(\mathbf{r}) = \begin{cases} 1 & \mathbf{r} \in V_j \\ 0 & else. \end{cases} \tag{12.101}$$

The residuals then are integrals over the control volumes and

$$0 = R_j = \int_{V_j} dV \left(\mathcal{T}\left[\tilde{u}\right](\mathbf{r}) - f(\mathbf{r}) \right) \tag{12.102}$$

respectively

$$0 = \sum_i u_i \int_{V_j} dV \, \mathcal{T}\left[N_i\right](\mathbf{r}) - \int_{V_j} dV \, f(\mathbf{r}). \tag{12.103}$$

12.4.3 Least Squares Method

Least squares methods have become popular for first order systems of differential equations in computational fluid dynamics and computational electrodynamics [141, 142].

The L2-norm of the residual (12.94) is given by the integral

$$S = \int_V dV \, R(\mathbf{r})^2. \tag{12.104}$$

It is minimized by solving the equations

$$0 = \frac{\partial S}{\partial u_j} = 2 \int_V dV \, \frac{\partial R}{\partial u_j} \, R(\mathbf{r}) \tag{12.105}$$

which is equivalent to choosing the weight functions

$$w_j(\mathbf{r}) = \frac{\partial R}{\partial u_j} R(\mathbf{r}) = \frac{\partial}{\partial u_j} \mathcal{T} \left[\sum_i u_i N_i(\mathbf{r}) \right] \tag{12.106}$$

or for a linear differential operator simply

$$w_j(\mathbf{r}) = \mathcal{T}\left[N_j(\mathbf{r})\right]. \tag{12.107}$$

Advantages of the least squares method are that boundary conditions can be incorporated into the residual and that S provides a measure for the quality of the solution which can be used for adaptive grid size control. On the other hand S involves a differential operator of higher order and therefore much smoother trial functions are necessary.

12.4.4 Galerkin Method

Galerkin's widely used method [138, 143] chooses the basis functions as weight functions

$$w_j(\mathbf{r}) = N_j(\mathbf{r}) \tag{12.108}$$

and solves the following system of equations

$$\int dV\, N_j(\mathbf{r}) \mathcal{T}\left[\sum_i u_i N_i(\mathbf{r})\right] - \int dV\, N_j(\mathbf{r}) f(\mathbf{r}) = 0 \tag{12.109}$$

or in the simpler linear case

$$\sum u_i \int_V dV\, N_j(\mathbf{r}) \mathcal{T}[N_i(\mathbf{r})] = \int_V dV\, N_j(\mathbf{r}) f(\mathbf{r}). \tag{12.110}$$

12.5 Spectral and Pseudo-Spectral Methods

Spectral methods use basis functions which are nonzero over the whole domain, the trial functions being mostly polynomials or Fourier sums [144]. They can be used to solve ordinary as well as partial differential equations. The combination of a spectral method with the point collocation method is also known as pseudo-spectral method.

12.5.1 Fourier Pseudo-Spectral Methods

Linear differential operators become diagonal in Fourier space. Combination of Fourier series expansion and point collocation leads to equations involving a discrete Fourier transformation, which can be performed very efficiently with the Fast Fourier Transform methods.

For simplicity we consider only the one-dimensional case. We choose equidistant collocation points

$$x_m = m\Delta x \quad m = 0, 1 \ldots M-1 \tag{12.111}$$

and expansion functions

$$N_j(x) = \mathrm{e}^{\mathrm{i}k_j x} \quad k_j = \frac{2\pi}{M\Delta x} j \quad j = 0, 1 \ldots M-1. \tag{12.112}$$

For a linear differential operator

$$\mathcal{L}[\mathrm{e}^{\mathrm{i}k_j x}] = l(k_j)\mathrm{e}^{\mathrm{i}k_j x} \tag{12.113}$$

and the condition on the residual becomes

$$0 = R_m = \sum_{j=0}^{M-1} u_j\, l(k_j)\mathrm{e}^{\mathrm{i}k_j x_m} - f(x_m) \tag{12.114}$$

or

$$f(x_m) = \sum_{j=0}^{M-1} u_j\, l(k_j)\mathrm{e}^{\mathrm{i}2\pi mj/M} \tag{12.115}$$

which is nothing but a discrete Fourier back transformation (Sect. 7.2, 7.19) which can be inverted to give

$$u_j\, l(k_j) = \frac{1}{N}\sum_{m=0}^{M-1} f(x_m)\,\mathrm{e}^{-\mathrm{i}2\pi mj/M}. \tag{12.116}$$

Instead of exponential expansion functions, sine and cosine functions can be used to satisfy certain boundary conditions, for instance to solve the Poisson equation within a cube (Sect. 18.1.2).

12.5.2 Example: Polynomial Approximation

Let us consider the initial value problem (Fig. 12.5)

$$\frac{\mathrm{d}}{\mathrm{d}x}u(x) - u(x) = 0 \quad u(0) = 1 \quad \text{for } 0 \le x \le 1 \tag{12.117}$$

with the well known solution

$$u(x) = \mathrm{e}^x. \tag{12.118}$$

We choose a polynomial trial function with the proper initial value

$$\tilde{u}(x) = 1 + u_1 x + u_2 x^2. \tag{12.119}$$

The residual is

$$R(x) = u_1 + 2u_2 x - \left(1 + u_1 x + u_2 x^2\right) = (u_1 - 1) + (2u_2 - u_1)x - u_2 x^2. \tag{12.120}$$

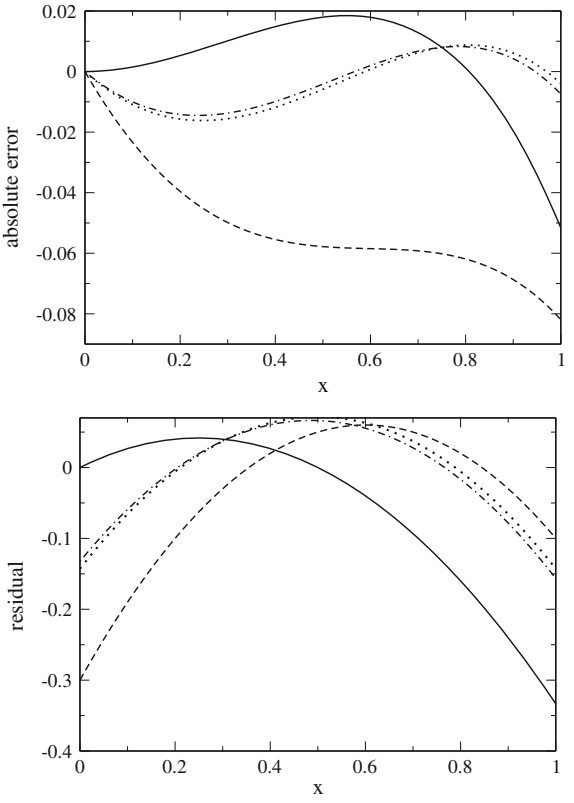


Fig. 12.5 (Approximate solution of a simple differential equation) The initial value problem $\frac{\mathrm{d}}{\mathrm{d}x}u(x) - u(x) = 0$ $u(0) = 1 \quad \text{for } 0 \le x \le 1$ is approximately solved with a polynomial trial function $\tilde{u}(x) = 1 + u_1 x + u_2 x^2$. The parameters $u_{1,2}$ are optimized with the method of weighted residuals using point collocation (*full curve*), sub-domain collocation (*dotted curve*), Galerkin's method (*dashed curve*) and least squares (*dash-dotted curve*). The absolute error $\tilde{u}(x) - \mathrm{e}^x$ (***Top***) and the residual $R(x) = \frac{\mathrm{d}}{\mathrm{d}x}\tilde{u}(x) - \tilde{u}(x) = (u_1 - 1) + (2u_2 - u_1)x - u_2 x^2$ both are smallest for the least squares and sub-domain collocation methods

12.5.2.1 Point Collocation Method

For our example we need two collocation points to obtain two equations for the two unknowns $u_{1,2}$. We choose $x_1 = 0, x_2 = \frac{1}{2}$. Then we have to solve the equations

$$R(x_1) = u_1 - 1 = 0 \tag{12.121}$$

$$R(x_2) = \frac{1}{2}u_1 + \frac{3}{4}u_2 - 1 = 0 \tag{12.122}$$

which gives

$$u_1 = 1 \quad u_2 = \frac{2}{3} \tag{12.123}$$

$$u_c = 1 + x + \frac{2}{3}x^2. \tag{12.124}$$

12.5.2.2 Sub-domain Method

We need two sub-domains to obtain two equations for the two unknowns $u_{1,2}$. We choose $V_1 = \{x, 0 < x < \frac{1}{2}\}$, $V_2 = \{x, \frac{1}{2} < x < 1\}$. Integration gives

$$R_1 = \frac{3}{8}u_1 + \frac{5}{24}u_2 - \frac{1}{2} = 0 \tag{12.125}$$

$$R_2 = \frac{1}{8}u_1 + \frac{11}{24}u_2 - \frac{1}{2} = 0 \tag{12.126}$$

$$u_1 = u_2 = \frac{6}{7} \tag{12.127}$$

$$u_{sdc} = 1 + \frac{6}{7}x + \frac{6}{7}x^2. \tag{12.128}$$

12.5.2.3 Galerkin Method

Galerkin's method uses the weight functions $w_1(x) = x$, $w_2(x) = x^2$. The equations

$$\int_0^1 dx\, w_1(x)R(x) = \frac{1}{6}u_1 + \frac{5}{12}u_2 - \frac{1}{2} = 0 \tag{12.129}$$

$$\int_0^1 dx\, w_2(x)R(x) = \frac{1}{12}u_1 + \frac{3}{10}u_2 - \frac{1}{3} = 0 \tag{12.130}$$

have the solution

$$u_1 = \frac{8}{11} \quad u_2 = \frac{10}{11} \tag{12.131}$$

$$u_G = 1 + \frac{8}{11}x + \frac{10}{11}x^2. \tag{12.132}$$

12.5.2.4 Least Squares Method

The integral of the squared residual

$$S = \int_0^1 dx\, R(x)^2 = 1 - u_1 - \frac{4}{3}u_2 + \frac{1}{3}u_1^2 + \frac{1}{2}u_1u_2 + \frac{8}{15}u_2^2 \tag{12.133}$$

is minimized by solving

$$\frac{\partial S}{\partial u_1} = \frac{2}{3}u_1 + \frac{1}{2}u_2 - 1 = 0 \tag{12.134}$$

$$\frac{\partial S}{\partial u_2} = \frac{1}{2}u_1 + \frac{16}{15}u_2 - \frac{4}{3} = 0 \tag{12.135}$$

which gives

$$u_1 = \frac{72}{83} \quad u_2 = \frac{70}{83} \tag{12.136}$$

$$u_{LS} = 1 + \frac{72}{83}x + \frac{70}{83}x^2. \tag{12.137}$$

12.6 Finite Elements

The method of finite elements (FEM) is a very flexible method to discretize partial differential equations [145, 146]. It is rather dominant in a variety of engineering sciences. Usually the expansion functions N_i are chosen to have compact support. The integration volume is divided into disjoint sub-volumes

$$V = \bigcup_{i=1}^{r} V_i \qquad V_i \bigcap V_{i'} = \emptyset \forall i \neq i'. \tag{12.138}$$

The $N_i(\mathbf{x})$ are piecewise continuous polynomials which are nonzero only inside V_i and a few neighbor cells.

12.6.1 One-Dimensional Elements

In one dimension the domain is an interval $V = \{x; a \leq x \leq b\}$ and the sub-volumes are small sub-intervals $V_i = \{x; x_i \leq x \leq x_{i+1}\}$. The one-dimensional mesh is the set of nodes $\{a = x_0, x_1 \ldots x_r = b\}$. Piecewise linear basis functions (Fig. 12.6) are in the 1-dimensional case given by

$$N_i(x) = \begin{cases} \frac{x_{i+1}-x}{x_{i+1}-x_i} & \text{for } x_i < x < x_{i+1} \\ \frac{x-x_{i-1}}{x_i-x_{i-1}} & \text{for } x_{i-1} < x < x_i \\ 0 \quad \text{else} \end{cases} \tag{12.139}$$

and the derivatives are (except at the nodes x_i)

$$N_i'(x) = \begin{cases} -\frac{1}{x_{i+1}-x_i} & \text{for } x_i < x < x_{i+1} \\ \frac{1}{x_i-x_{i-1}} & \text{for } x_{i-1} < x < x_i \\ 0 \quad \text{else} \end{cases} . \tag{12.140}$$

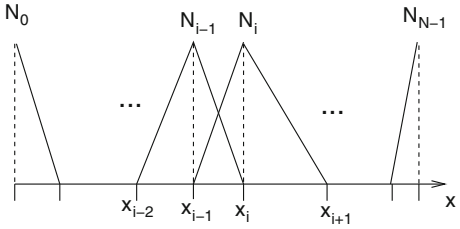


Fig. 12.6 (Finite elements in one dimension) The basis functions N_i are piecewise continuous polynomials and have compact support. In the simplest case they are composed of two linear functions over the sub-intervals $x_{i-1} \leq x \leq x_i$ and $x_i \leq x \leq x_{i+1}$

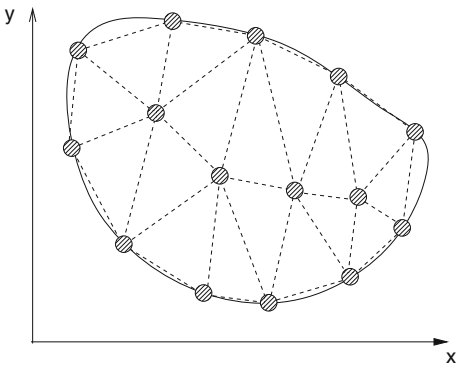


Fig. 12.7 (Triangulation of a two dimensional domain) A two-dimensional mesh is defined by a set of node points which can be regarded to form the vertices of a triangulation

12.6.2 Two-and Three-Dimensional Elements

In two dimensions the mesh is defined by a finite number of points $(x_i, y_i) \in V$ (the nodes of the mesh). There is considerable freedom in the choice of these points and they need not be equally spaced.

12.6.2.1 Triangulation

The nodes can be regarded as forming the vertices of a triangulation[9] of the domain V (Fig. 12.7).

The piecewise linear basis function in one dimension (12.139) can be generalized to the two-dimensional case by constructing functions $N_i(x, y)$ which are zero at all nodes except (x_i, y_i)

$$N_i(x_j, y_j) = \delta_{i,j}. \tag{12.141}$$

[9]The triangulation is not determined uniquely by the nodes.

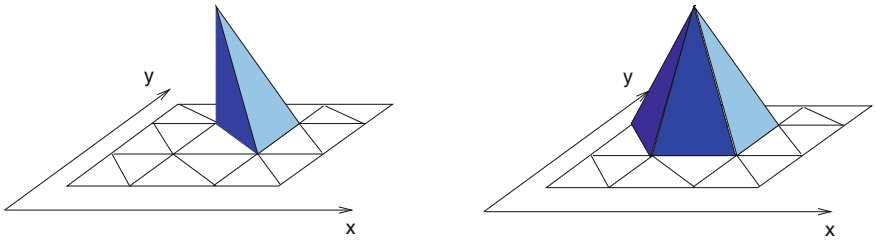


Fig. 12.8 (Finite elements in two dimensions) The simplest finite elements in two dimensions are piecewise linear functions $N_i(x, y)$ which are non-vanishing only at one node (x_i, y_i) (***Right side***). They can be constructed from small pyramids built upon one of the triangles that contains this node (***Left side***)

These functions are linear over each triangle which contains the vertex i and can be combined as the sum of small pyramids (Fig. 12.8). Let one of the triangles be denoted by its three vertices as T_{ijk}.[10] The corresponding linear function then is

$$n_{ijk}(x, y) = \alpha + \beta_x(x - x_i) + \beta_y(y - y_i) \tag{12.142}$$

where the coefficients follow from the conditions

$$n_{ijk}(x_i, y_i) = 1 \quad n_{ijk}(x_j, y_j) = n_{ijk}(x_k, y_k) = 0 \tag{12.143}$$

as

$$\alpha = 1 \quad \beta_x = \frac{y_j - y_k}{2A_{ijk}} \quad \beta_y = \frac{x_k - x_j}{2A_{ijk}} \tag{12.144}$$

with

$$A_{ijk} = \frac{1}{2} \det \begin{vmatrix} x_j - x_i & x_k - x_i \\ y_j - y_i & y_k - y_i \end{vmatrix} \tag{12.145}$$

which, apart from sign, is the area of the triangle T_{ijk}. The basis function N_i now is given by

$$N_i(x, y) = \begin{cases} n_{ijk}(x, y) & (x, y) \in T_{ijk} \\ 0 & \text{else} \end{cases} .$$

In three dimensions we consider tetrahedrons (Fig. 12.9) instead of triangles. The corresponding linear function of three arguments has the form

$$n_{i,j,k,l}(x, y, z) = \alpha + \beta_x(x - x_i) + \beta_y(y - y_i) + \beta_z(z - z_i) \tag{12.146}$$

[10] The order of the indices does matter.

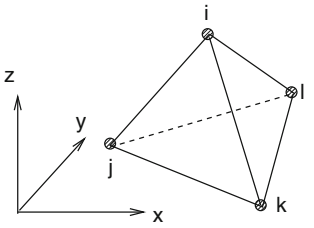


Fig. 12.9 (Tetrahedron) The tetrahedron is the three-dimensional case of an Euclidean simplex, i.e. the simplest polytop

and from the conditions $n_{i,j,k,l}(x_i, y_i, z_i) = 1$ and $n_{i,j,k,l} = 0$ on all other nodes we find (an algebra program is helpful at that point)

$$\begin{aligned}
\alpha &= 1 \\
\beta_x &= \frac{1}{6V_{ijkl}} \det \begin{vmatrix} y_k - y_j & y_l - y_j \\ z_k - z_j & z_l - z_j \end{vmatrix} \\
\beta_y &= \frac{1}{6V_{ijkl}} \det \begin{vmatrix} z_k - z_j & z_l - z_j \\ x_k - x_j & x_l - x_j \end{vmatrix} \\
\beta_z &= \frac{1}{6V_{ijkl}} \det \begin{vmatrix} x_k - x_j & x_l - x_j \\ y_k - y_j & y_l - y_j \end{vmatrix}
\end{aligned} \tag{12.147}$$

where V_{ijkl} is, apart from sign, the volume of the tetrahedron

$$V_{ijkl} = \frac{1}{6} \det \begin{vmatrix} x_j - x_i & x_k - x_i & x_l - x_i \\ y_j - y_i & y_k - y_i & y_l - y_i \\ z_j - z_i & z_k - z_i & z_l - z_i \end{vmatrix}. \tag{12.148}$$

12.6.2.2 Rectangular Elements

For a rectangular grid rectangular elements offer a practical alternative to triangles. Since equations for four nodes have to be fulfilled, the basic element needs four parameters, which is the case for a bilinear expression. Let us denote one of the rectangles which contains the vertex i as $R_{i,j,k,l}$. The other three edges are

$$(x_j, y_j) = (x_i + b_x, y_i) \quad (x_k, y_k) = (x_i, y_i + b_y) \quad (x_l, y_l) = (x_i + b_x, y_i + b_y) \tag{12.149}$$

where $b_x = \pm h_x, b_y = \pm h_y$ corresponding to the four rectangles with the common vertex i (Fig. 12.10).

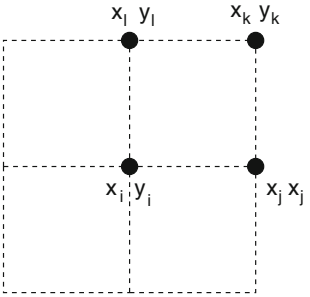


Fig. 12.10 (Rectangular elements around one vertex) The basis function N_i is a bilinear function on each of the four rectangles containing the vertex (x_i, y_i)

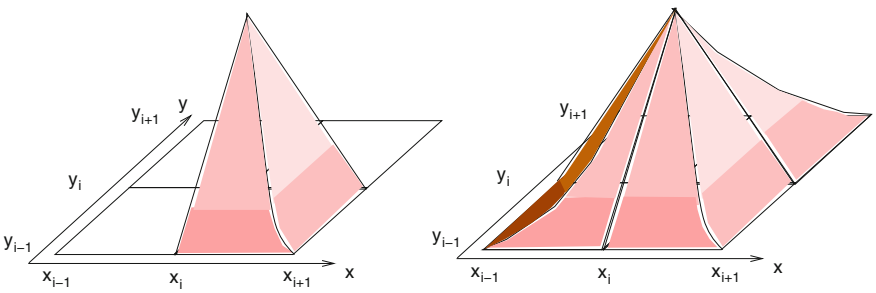


Fig. 12.11 (Bilinear elements on a rectangular grid) The basis functions $N_i(x, y)$ on a rectangular grid (***Right side***) are piecewise bilinear functions (***Left side***), which vanish at all nodes except (x_i, y_i)

The bilinear function (Fig. 12.11) corresponding to R_{ijkl} is

$$n_{i,j,k,l}(x, y) = \alpha + \beta(x - x_i) + \gamma(y - y_i) + \eta(x - x_i)(y - y_i). \tag{12.150}$$

It has to fulfill

$$n_{i,j,k,l}(x_i, y_i) = 1 \quad n_{i,j,k,l}(x_j, y_j) = n_{i,j,k,l}(x_k, y_k) = n_{i,j,k,l}(x_l, y_l) = 0 \tag{12.151}$$

from which we find

$$\alpha = 1 \quad \beta = -\frac{1}{b_x} \quad \gamma = -\frac{1}{b_y} \quad \eta = \frac{1}{b_x b_y} \tag{12.152}$$

$$n_{i.j.k.l}(x, y) = 1 - \frac{x - x_i}{b_x} - \frac{y - y_i}{b_y} + \frac{(x - x_i)}{b_x}\frac{(y - y_i)}{b_y}. \tag{12.153}$$

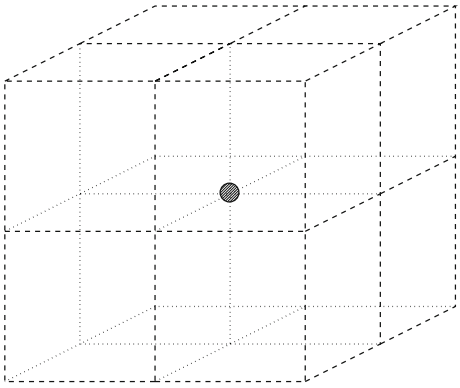


Fig. 12.12 (Three-dimensional rectangular grid) The basis function N_i is trilinear on each of the eight cuboids containing the vertex i. It vanishes on all nodes except (x_i, y_i, z_i)

The basis function centered at node i then is

$$N_i(x, y) = \begin{cases} n_{i,j,k,l}(x, y) & (x, y) \in R_{i,j,k,l} \\ 0 & \text{else} \end{cases} . \tag{12.154}$$

Generalization to a three dimensional grid is straightforward (Fig. 12.12). We denote one of the eight cuboids containing the node (x_i, y_i, z_i) as $C_{i,j_1 \ldots j_7}$ with $(x_{j_1}, y_{j_1}, z_{j_1}) = (x_i + b_x, y_i, z_i) \ldots (x_{j_7}, y_{j_7}, z_{j_7}) = (x_i + b_x, y_i + b_y, z_i + b_z)$. The corresponding trilinear function is

$$\begin{aligned} n_{i,j_1 \ldots j_7} &= 1 - \frac{x - x_i}{b_x} - \frac{y - y_i}{b_y} - \frac{z - z_i}{b_z} \\ &+ \frac{(x - x_i)}{b_x}\frac{(y - y_i)}{b_y} + \frac{(x - x_i)}{b_x}\frac{(z - z_i)}{b_z} + \frac{(z - z_i)}{b_z}\frac{(y - y_i)}{b_y} \\ &- \frac{(x - x_i)}{b_x}\frac{(y - y_i)}{b_y}\frac{(z - z_i)}{b_z}. \end{aligned} \tag{12.155}$$

12.6.3 One-Dimensional Galerkin FEM

As an example we consider the one dimensional linear differential equation (12.5)

$$\left(a\frac{\partial^2}{\partial x^2} + b\frac{\partial}{\partial x} + c\right) u(x) = f(x) \tag{12.156}$$

in the domain $0 \leq x \leq 1$ with boundary conditions

$$u(0) = u(1) = 0. \tag{12.157}$$

We use the basis functions from (12.139) on a one-dimensional grid with

$$x_{i+1} - x_i = h_i \tag{12.158}$$

and apply the Galerkin method [147]. The boundary conditions require

$$u_0 = u_{N-1} = 0. \tag{12.159}$$

The weighted residual is

$$0 = R_j = \sum_i u_i \int_0^1 dx\, N_j(x) \left(a \frac{\partial^2}{\partial x^2} + b \frac{\partial}{\partial x} + c \right) N_i(x) - \int_0^1 dx\, N_j(x) f(x). \tag{12.160}$$

First we integrate

$$\int_0^1 N_j(x) N_i(x)\, dx = \int_{x_{i-1}}^{x_{i+1}} N_j(x) N_i(x) dx = \begin{cases} \frac{h_i + h_{i-1}}{3} & j = i \\ \frac{h_i}{6} & j = i+1 \\ \frac{h_{i-1}}{6} & j = i-1 \\ 0 & |i-j| > 1 \end{cases}. \tag{12.161}$$

Integration of the first derivative gives

$$\int_0^1 dx\, N_j(x) N_i'(x) = \begin{cases} 0 & j = i \\ \frac{1}{2} & j = i-1 \\ -\frac{1}{2} & j = i+1 \\ 0 & \text{else} \end{cases}. \tag{12.162}$$

For the second derivative partial integration gives

$$\begin{aligned} &\int_0^1 dx\, N_j(x) \frac{\partial^2}{\partial x^2} N_i(x) \\ &= N_j(1) N_i'(1-\varepsilon) - N_j(0) N_i'(0+\varepsilon) - \int_0^1 dx\, N_j'(x) N_i'(x) \end{aligned} \tag{12.163}$$

where the first two summands are zero due to the boundary conditions. Since N_i and N_i' are nonzero only for $x_{i-1} < x < x_{i+1}$ we find

$$\int_0^1 dx\, N_j(x) \frac{\partial^2}{\partial x^2} N_i(x) = -\int_{x_{i-1}}^{x_{i+1}} dx\, N_j'(x)\, N_i'(x) = \begin{cases} \frac{1}{h_{i-1}} & j = i-1 \\ -\frac{1}{h_i} - \frac{1}{h_{i-1}} & i = j \\ \frac{1}{h_i} & j = i+1 \\ 0 & \text{else} \end{cases}. \tag{12.164}$$

Integration of the last term in (12.160) gives

$$\int_0^1 dx\, N_j(x) f(x) = \int_{x_{i-1}}^{x_{i+1}} dx\, N_j(x) f(x)$$
$$= \int_{x_{j-1}}^{x_j} dx\, \frac{x - x_{j-1}}{x_j - x_{j-1}} f(x) + \int_{x_j}^{x_{j+1}} dx\, \frac{x_{j+1} - x}{x_{j+1} - x_j} f(x). \tag{12.165}$$

Applying the trapezoidal rule[11] for both integrals we find

$$\int_{x_{j-1}}^{x_{j+1}} dx\, N_j(x) f(x) \approx f(x_j) \frac{h_j + h_{j-1}}{2}. \tag{12.166}$$

The discretized equation finally reads

$$a \left\{ \frac{1}{h_{j-1}} u_{j-1} - \left(\frac{1}{h_j} + \frac{1}{h_{j-1}} \right) u_j + \frac{1}{h_j} u_{j+1} \right\}$$
$$+ b \left\{ -\frac{1}{2} u_{j-1} + \frac{1}{2} u_{j+1} \right\}$$
$$+ c \left\{ \frac{h_{j-1}}{6} u_{j-1} + \frac{h_j + h_{j-1}}{3} u_j + \frac{h_j}{6} u_{j+1} \right\}$$
$$= f(x_j) \frac{h_j + h_{j-1}}{2} \tag{12.167}$$

which can be written in matrix notation as

$$A\mathbf{u} = B\mathbf{f} \tag{12.168}$$

where the matrix A is tridiagonal as a consequence of the compact support of the basis functions

$$A = a \begin{pmatrix} -\frac{1}{h_1} - \frac{1}{h_0}, & \frac{1}{h_1} & & & \\ \ddots & & & & \\ \frac{1}{h_{j-1}}, & -\frac{1}{h_j} - \frac{1}{h_{j-1}}, & \frac{1}{h_j} & & \\ & \ddots & & & \\ & & \frac{1}{h_{N-3}}, & -\frac{1}{h_{N-2}} - \frac{1}{h_{N-3}} & \end{pmatrix}$$

[11] Higher accuracy can be achieved, for instance, by Gaussian integration.

$$
+b\begin{pmatrix} 0, & \frac{1}{2} & & & \\ & \ddots & & & \\ & -\frac{1}{2}, & 0, & \frac{1}{2} & \\ & & & \ddots & \\ & & & -\frac{1}{2}, & 0 \end{pmatrix}
$$

$$
+c\begin{pmatrix} \frac{(h_1+h_0)}{3}, & \frac{h_1}{6} & & & \\ & \ddots & & & \\ & \frac{h_{j-1}}{6}, & \frac{(h_j+h_{j-1})}{3}, & \frac{h_j}{6} & \\ & & & \ddots & \\ & & & \frac{h_{N-3}}{6}, & \frac{(h_{N-2}+h_{N-3})}{3} \end{pmatrix}
$$

$$
B = \begin{pmatrix} \frac{h_0+h_1}{2} & & & & \\ & \ddots & & & \\ & & \frac{h_{j-1}+h_j}{2} & & \\ & & & \ddots & \\ & & & & \frac{h_{N-2}+h_{N-3}}{2} \end{pmatrix}. \tag{12.169}
$$

For equally spaced nodes $h_i = h_{i-1} = h$ and after division by h (12.169) reduces to a system of equations where the derivatives are replaced by finite differences (12.20)

$$
\left\{ a\frac{1}{h^2}M_2 + b\frac{1}{h}M_1 + cM_0 \right\} \mathbf{u} = \mathbf{f} \tag{12.170}
$$

with the so called consistent mass matrix

$$
M_0 = \begin{pmatrix} \ddots & & & & \\ \ddots & \ddots & \ddots & & \\ & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \\ & & \ddots & \ddots & \ddots \\ & & & & \ddots \end{pmatrix} \tag{12.171}
$$

and the derivative matrices

$$M_1 = \begin{pmatrix} \ddots & & & & \\ \ddots & \ddots & \ddots & \ddots & \\ & -\frac{1}{2} & 0 & \frac{1}{2} & \\ & & \ddots & \ddots & \ddots \\ & & & & \ddots \end{pmatrix} \quad M_2 = \begin{pmatrix} \ddots & & & & \\ \ddots & \ddots & \ddots & \ddots & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & & \ddots \end{pmatrix}. \tag{12.172}$$

The vector $\mathbf{u}$ is replaced by

$$M_0\mathbf{u} = \left[1 + \frac{1}{6}M_2\right]\mathbf{u}. \tag{12.173}$$

Within the framework of the finite differences method

$$u_j + \frac{1}{6}\left(u_{j-1} - 2u_j + u_{j+1}\right) = u_j + \frac{h^2}{6}\left(\frac{d^2u}{dx^2}\right)_j + O(h^4) \tag{12.174}$$

hence replacing it by u_j (this is called mass lumping) introduces an error of the order $O(h^2)$.

12.7 Boundary Element Method

The boundary element method (BEM) [148, 149] is a method for linear partial differential equations which can be brought into boundary integral form[12] like Laplace's equation (Chap. 18)[13]

$$-\triangle\Phi(\mathbf{r}) = 0 \tag{12.175}$$

for which the fundamental solution

$$\triangle G(\mathbf{r}, \mathbf{r}') = -\delta(\mathbf{r} - \mathbf{r}')$$

is given by

$$G(\mathbf{r} - \mathbf{r}') = \frac{1}{4\pi|\mathbf{r} - \mathbf{r}'|} \quad \text{in three dimensions} \tag{12.176}$$

[12] This is only possible if the fundamental solution or Green's function is available.

[13] The minus sign is traditionally used.

$$G(\mathbf{r}-\mathbf{r}') = \frac{1}{2\pi}\ln\frac{1}{|\mathbf{r}-\mathbf{r}'|} \quad \text{in two dimensions.} \tag{12.177}$$

We apply Gauss's theorem to the expression [150]

$$\begin{aligned}&\operatorname{div}\left[G(\mathbf{r}-\mathbf{r}')\operatorname{grad}(\Phi(\mathbf{r})) - \Phi(\mathbf{r})\operatorname{grad}(G(\mathbf{r}-\mathbf{r}'))\right]\\ &= -\Phi(\mathbf{r})\triangle(G(\mathbf{r}-\mathbf{r}')).\end{aligned} \tag{12.178}$$

Integration over a volume V gives

$$\begin{aligned}&\oint_{\partial V} dA\left(G(\mathbf{r}-\mathbf{r}')\frac{\partial}{\partial n}(\Phi(\mathbf{r})) - \Phi(\mathbf{r})\frac{\partial}{\partial n}(G(\mathbf{r}-\mathbf{r}'))\right)\\ &= -\int_V dV\ \left(\Phi(\mathbf{r})\triangle(G(\mathbf{r}-\mathbf{r}')) = \Phi(\mathbf{r}').\right.\end{aligned} \tag{12.179}$$

This integral equation determines the potential self-consistently by its value and normal derivative on the surface of the cavity. It can be solved numerically by dividing the surface into a finite number of boundary elements. The resulting system of linear equations often has smaller dimension than corresponding finite element approaches. However, the coefficient matrix is in general full and not necessarily symmetric.

Chapter 13
Equations of Motion

Simulation of a physical system means to calculate the time evolution of a model system in many cases. We consider a large class of models which can be described by a first order initial value problem

$$\frac{dY}{dt} = f(Y(t), t) \quad Y(t = 0) = Y_0 \tag{13.1}$$

where Y is the state vector (possibly of very high dimension) which contains all information about the system. Our goal is to calculate the time evolution of the state vector $Y(t)$ numerically. For obvious reasons this can be done only for a finite number of values of t and we have to introduce a grid of discrete times t_n which for simplicity are assumed to be equally spaced[1]*:*

$$t_{n+1} = t_n + \Delta t. \tag{13.2}$$

Advancing time by one step involves the calculation of the integral

$$Y(t_{n+1}) - Y(t_n) = \int_{t_n}^{t_{n+1}} f(Y(t'), t')dt' \tag{13.3}$$

which can be a formidable task since $f(Y(t), t)$ depends on time via the time dependence of all the elements of $Y(t)$. In this chapter we discuss several strategies for the time integration. The explicit Euler forward difference has low error order but is useful as a predictor step for implicit methods. A symmetric difference quotient is much more accurate. It can be used as the corrector step in combination with an explicit Euler predictor step and is often used for the time integration of partial differential equations. Methods with higher error order can be obtained from a Taylor series expansion, like the Nordsieck and Gear predictor-corrector methods which have been often applied in molecular dynamics calculations. Runge–Kutta methods are very important for ordinary differential equations. They are robust and allow an adaptive

[1] *Control of the step width will be discussed later.*

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_13

control of the step size. Very accurate results can be obtained for ordinary differential equations with extrapolation methods like the famous Gragg-Bulirsch-Stoer method. If the solution is smooth enough, multistep methods are applicable, which use information from several points. Most known are Adams-Bashforth–Moulton methods and Gear methods (also known as backward differentiation methods), which are especially useful for stiff problems. The class of Verlet methods has been developed for molecular dynamics calculations. They are symplectic and time reversible and conserve energy over long trajectories.

13.1 The State Vector

The state of a classical N-particle system is given by the position in phase space, or equivalently by specifying position and velocity for all the N particles

$$Y = (\mathbf{r}_1, \mathbf{v}_1, \ldots, \mathbf{r}_N, \mathbf{v}_N)\,. \tag{13.4}$$

The concept of a state vector is not restricted to a finite number of degrees of freedom. For instance a diffusive system can be described by the particle concentrations as a function of the coordinate, i.e. the elements of the state vector are now indexed by the continuous variable $\mathbf{x}$

$$Y = (c_1(\mathbf{x}), \ldots c_M(\mathbf{x}))\,. \tag{13.5}$$

Similarly, a quantum particle moving in an external potential can be described by the amplitude of the wave function

$$Y = (\Psi(\mathbf{x})). \tag{13.6}$$

Numerical treatment of continuous systems is not feasible since even the ultimate high end computer can only handle a finite number of data in finite time. Therefore discretization is necessary (Chap. 12), by introducing a spatial mesh (Sects. 12.2, 12.3, 12.6), which in the simplest case means a grid of equally spaced points

$$\mathbf{x}_{ijk} = (ih, jh, kh) \quad i = 1..i_{\max}, j = 1..j_{\max}, k = 1..k_{\max} \tag{13.7}$$

$$Y = \left(c_1(\mathbf{x}_{ijk}) \ldots c_M(\mathbf{x}_{ijk})\right) \tag{13.8}$$

$$Y = \left(\Psi(\mathbf{x}_{ijk})\right) \tag{13.9}$$

or by expanding the continuous function with respect to a finite set of basis functions (Sect. 12.5). The elements of the state vector then are the expansion coefficients

$$|\Psi> = \sum_{s=1}^{N} C_s |\Psi_s> \tag{13.10}$$

$$Y = (C_1, \ldots, C_N)\,. \tag{13.11}$$

If the density matrix formalism is used to take the average over a thermodynamic ensemble or to trace out the degrees of freedom of a heat bath, the state vector instead is composed of the elements of the density matrix

$$\rho = \sum_{s=1}^{N}\sum_{s'=1}^{N} \rho_{ss'} |\Psi_s><\Psi_{s'}| = \sum_{s=1}^{N}\sum_{s'=1}^{N} \overline{C^*_{s'} C_s} |\Psi_s><\Psi_{s'}| \tag{13.12}$$

$$Y = (\rho_{11} \cdots \rho_{1N}, \rho_{21} \cdots \rho_{2N}, \ldots, \rho_{N1} \cdots \rho_{NN})\,. \tag{13.13}$$

13.2 Time Evolution of the State Vector

We assume that all information about the system is included in the state vector. Then the simplest equation to describe the time evolution of the system gives the change of the state vector

$$\frac{\mathrm{d}Y}{\mathrm{d}t} = f(Y, t) \tag{13.14}$$

as a function of the state vector (or more generally a functional in the case of a continuous system). Explicit time dependence has to be considered for instance to describe the influence of an external time dependent field.

Some examples will show the universality of this equation of motion:

- N-particle system

The motion of N interacting particles is described by

$$\frac{\mathrm{d}Y}{\mathrm{d}t} = (\dot{\mathbf{r}}_1, \dot{\mathbf{v}}_1 \cdots) = (\mathbf{v}_1, \mathbf{a}_1 \cdots) \tag{13.15}$$

where the acceleration of a particle is given by the total force acting upon this particle and thus depends on all the coordinates and eventually time (velocity dependent forces could be also considered but are outside the scope of this book)

$$\mathbf{a}_i = \frac{\mathbf{F}_i(\mathbf{r}_1 \cdots \mathbf{r}_N, t)}{m_i}. \tag{13.16}$$

- Diffusion

Heat transport and other diffusive processes are described by the diffusion equation

$$\frac{\partial f}{\partial t} = D\Delta f + S(\mathbf{x}, t) \tag{13.17}$$

which in its simplest spatially discretized version for 1-dimensional diffusion reads

$$\frac{\partial f(\mathbf{x}_i)}{\partial t} = \frac{D}{\Delta x^2}\left(f(\mathbf{x}_{i+1}) + f(\mathbf{x}_{i-1}) - 2f(\mathbf{x}_i)\right) + S(\mathbf{x}_i, t). \tag{13.18}$$

- Waves

Consider the simple 1-dimensional wave equation

$$\frac{\partial^2 f}{\partial t^2} = c^2 \frac{\partial^2 f}{\partial x^2} \tag{13.19}$$

which by introducing the velocity $g(\mathbf{x}) = \frac{\partial}{\partial t} f(\mathbf{x})$ as an independent variable can be rewritten as

$$\frac{\partial}{\partial t}\left(f(\mathbf{x}), g(\mathbf{x})\right) = \left(g(\mathbf{x}), c^2 \frac{\partial^2}{\partial x^2} f(\mathbf{x})\right). \tag{13.20}$$

Discretization of space gives

$$\frac{\partial}{\partial t}\left(f(\mathbf{x}_i), g(\mathbf{x}_i)\right) = \left(g(\mathbf{x}_i), \frac{c^2}{\Delta x^2}\left(f(\mathbf{x}_{i+1}) + f(\mathbf{x}_{i-1}) - 2f(\mathbf{x}_i)\right)\right). \tag{13.21}$$

- two-state quantum system

The Schroedinger equation for a two level system (for instance a spin-1/2 particle in a magnetic field) reads

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} H_{11}(t)\ H_{12}(t) \\ H_{21}(t)\ H_{22}(t) \end{pmatrix}\begin{pmatrix} C_1 \\ C_2 \end{pmatrix}. \tag{13.22}$$

13.3 Explicit Forward Euler Method

The simplest method which is often discussed in elementary physics textbooks approximates the integrand by its value at the lower bound (Fig. 13.1):

$$Y(t_{n+1}) - Y(t_n) \approx f(Y(t_n), t_n)\Delta t. \tag{13.23}$$

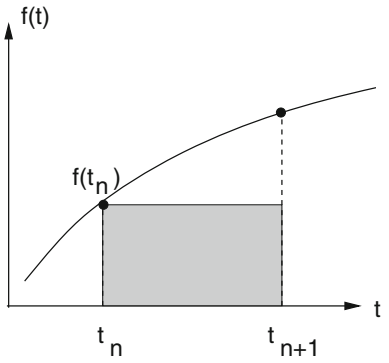


Fig. 13.1 Explicit Euler method

The truncation error can be estimated from a Taylor series expansion

$$Y(t_{n+1}) - Y(t_n) = \Delta t \frac{\mathrm{d}Y}{\mathrm{d}t}(t_n) + \frac{\Delta t^2}{2}\frac{\mathrm{d}^2 Y}{\mathrm{d}t^2}(t_n) + \cdots$$
$$= \Delta t f(Y(t_n), t_n) + O(\Delta t^2). \tag{13.24}$$

The explicit Euler method has several serious drawbacks

- low error order

Suppose you want to integrate from the initial time t_0 to the final time $t_0 + T$. For a time step of Δt you have to perform $N = T/\Delta t$ steps. Assuming comparable error contributions from all steps the global error scales as $N\Delta t^2 = O(\Delta t)$. The error gets smaller as the time step is reduced but it may be necessary to use very small Δt to obtain meaningful results.

- loss of orthogonality and normalization

The simple Euler method can produce systematic errors which are very inconvenient if you want, for instance, to calculate the orbits of a planetary system. This can be most easily seen from a very simple example. Try to integrate the following equation of motion (see Example 1.5 on p. 13):

$$\frac{\mathrm{d}z}{\mathrm{d}t} = \mathrm{i}\omega z. \tag{13.25}$$

The exact solution is obviously given by a circular orbit in the complex plane:

$$z = z_0 e^{\mathrm{i}\omega t} \tag{13.26}$$

$$|z| = |z_0| = \text{ const.} \tag{13.27}$$

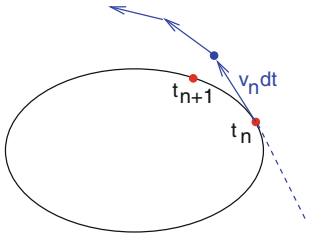


Fig. 13.2 Systematic errors of the Euler method

Application of the Euler method gives

$$z(t_{n+1}) = z(t_n) + \mathrm{i}\omega\Delta t\, z(t_n) = (1 + \mathrm{i}\omega\Delta t) z(t_n) \tag{13.28}$$

and you find immediately

$$|z(t_n)| = \sqrt{1 + \omega^2 \Delta t^2}\, |z(t_{n-1})| = \left(1 + \omega^2 \Delta t^2\right)^{n/2} |z(t_0)| \tag{13.29}$$

which shows that the radius increases continually even for the smallest time step possible (Fig. 13.2).

The same kind of error appears if you solve the Schroedinger equation for a particle in an external potential or if you calculate the rotational motion of a rigid body. For the N-body system it leads to a violation of the conservation of phase space volume. This can introduce an additional sensitivity of the calculated results to the initial conditions. Consider a harmonic oscillator with the equation of motion

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{pmatrix} x(t) \\ v(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ -\omega^2 x(t) \end{pmatrix}. \tag{13.30}$$

Application of the explicit Euler method gives

$$\begin{pmatrix} x(t + \Delta t) \\ v(t + \Delta t) \end{pmatrix} = \begin{pmatrix} x(t) \\ v(t) \end{pmatrix} + \begin{pmatrix} v(t) \\ -\omega^2 x(t) \end{pmatrix} \Delta t. \tag{13.31}$$

The change of the phase space volume (Fig. 13.3) is given by the Jacobi determinant

$$J = \left| \frac{\partial(x(t + \Delta t), v(t + \Delta t))}{\partial(x(t), v(t))} \right| = \begin{vmatrix} 1 & \Delta t \\ -\omega^2 \Delta t & 1 \end{vmatrix} = 1 + (\omega \Delta t)^2. \tag{13.32}$$

In this case the phase space volume increases continuously.

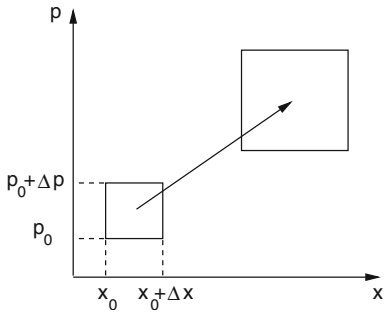


Fig. 13.3 Time evolution of the phase space volume

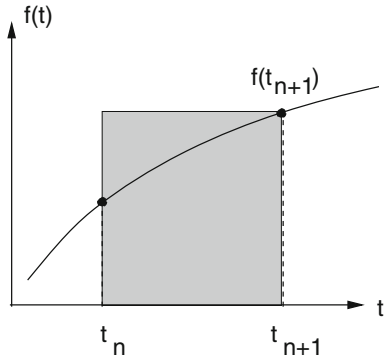


Fig. 13.4 Implicit backward Euler method

13.4 Implicit Backward Euler Method

Alternatively let us make a step backwards in time

$$Y(t_n) - Y(t_{n+1}) \approx -f(Y(t_{n+1}), t_{n+1})\Delta t \tag{13.33}$$

which can be written as (Fig. 13.4)

$$Y(t_{n+1}) \approx Y(t_n) + f(Y(t_{n+1}), t_{n+1})\Delta t. \tag{13.34}$$

Taylor series expansion gives

$$Y(t_n) = Y(t_{n+1}) - \frac{\mathrm{d}}{\mathrm{d}t}Y(t_{n+1})\Delta t + \frac{\mathrm{d}^2}{\mathrm{d}t^2}Y(t_{n+1})\frac{\Delta t^2}{2} + \cdots \tag{13.35}$$

which shows that the error order again is $O(\Delta t^2)$. The implicit method is sometimes used to avoid the inherent instability of the explicit method. For the examples in

Sect. 13.3 it shows the opposite behavior. The radius of the circular orbit as well as the phase space volume decrease in time. The gradient at future time has to be estimated before an implicit step can be performed.

13.5 Improved Euler Methods

The quality of the approximation can be improved significantly by employing the midpoint rule (Fig. 13.5)

$$Y(t_{n+1}) - Y(t_n) \approx f\left(Y\left(t + \frac{\Delta t}{2}\right), t_n + \frac{\Delta t}{2}\right)\Delta t. \tag{13.36}$$

The error is smaller by one order of Δt:

$$\begin{aligned} & Y(t_n) + f\left(Y\left(t + \frac{\Delta t}{2}\right), t_n + \frac{\Delta t}{2}\right)\Delta t \\ & = Y(t_n) + \left(\frac{\mathrm{d}Y}{\mathrm{d}t}(t_n) + \frac{\Delta t}{2}\frac{\mathrm{d}^2 Y}{\mathrm{d}t^2}(t_n) + \cdots\right)\Delta t \\ & = Y(t_n + \Delta t) + O(\Delta t^3). \end{aligned} \tag{13.37}$$

The future value $Y(t + \frac{\Delta t}{2})$ can be obtained by two different approaches:

- predictor-corrector method

Since $f(Y(t + \frac{\Delta t}{2}), t_n + \frac{\Delta t}{2})$ is multiplied with Δt, it is sufficient to use an approximation with lower error order. Even the explicit Euler step is sufficient. Together the following algorithm results:

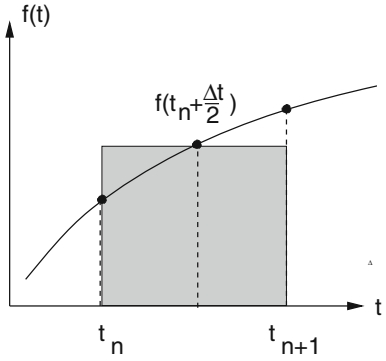


Fig. 13.5 Improved Euler method

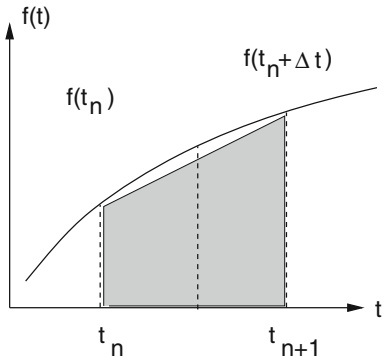


Fig. 13.6 Improved polygon (or Heun) method

$$\begin{aligned} &\text{predictor step:} \quad Y^{(p)} = Y(t_n) + \tfrac{\Delta t}{2} f(Y(t_n), t_n) \\ &\text{corrector step:} \quad Y(t_n + \Delta t) = Y(t_n) + \Delta t\, f(Y^{(p)}, t_n + \tfrac{\Delta t}{2}). \end{aligned} \tag{13.38}$$

- averaging (Heun method)

The average of $f(Y(t_n), t_n)$ and $f(Y(t_n + \Delta t), t + \Delta t)$ is another approximation to the midpoint value of comparable quality (Fig. 13.6).

Expansion around $t_n + \Delta t/2$ gives

$$\begin{aligned} &\frac{1}{2}\left(f(Y(t_n), t_n) + f(Y(t_n + \Delta t), t + \Delta t)\right) \\ &= f\left(Y\left(t_n + \frac{\Delta t}{2}\right), t_n + \frac{\Delta t}{2}\right) + O(\Delta t^2). \end{aligned} \tag{13.39}$$

Inserting the average in (13.36) gives the following algorithm, which is also known as improved polygon method and corresponds to the trapezoidal rule for the integral (4.13) or to a combination of explicit and implicit Euler step:

$$Y(t_n + \Delta t) = Y(t_n) + \frac{\Delta t}{2}\left(f(Y(t_n), t_n) + f(Y(t_n + \Delta t), t + \Delta t)\right). \tag{13.40}$$

In the special case of a linear function $f(Y(t), t) = F\,Y(t)$ (for instance rotational motion or diffusion) this can be solved formally by

$$Y(t_n + \Delta t) = \left(1 - \frac{\Delta t}{2} F\right)^{-1} \left(1 + \frac{\Delta t}{2} F\right) Y(t_n). \tag{13.41}$$

Numerically it is not necessary to perform the matrix inversion. Instead a linear system of equations is solved:

$$\left(1 - \frac{\Delta t}{2} F\right) Y(t_n + \Delta t) = \left(1 + \frac{\Delta t}{2} F\right) Y(t_n). \tag{13.42}$$

In certain cases the Heun method conserves the norm of the state vector, for instance if F has only imaginary eigenvalues (as for the 1-dimensional Schroedinger equation, see p. 526).

In the general case a predictor step has to be made to estimate the state vector at $t_n + \Delta t$ before the Heun expression (13.40) can be evaluated:

$$Y^{(p)} = Y(t_n) + \Delta t f(Y(t_n), t_n). \tag{13.43}$$

13.6 Taylor Series Methods

Higher order methods can be obtained from a Taylor series expansion

$$Y(t_n + \Delta t) = Y(t_n) + \Delta t f(Y(t_n), t_n) + \frac{\Delta t^2}{2} \frac{\mathrm{d}f(Y(t_n), t_n)}{\mathrm{d}t} + \cdots . \tag{13.44}$$

The total time derivative can be expressed as

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\partial f}{\partial Y}\frac{\mathrm{d}Y}{\mathrm{d}t} + \frac{\partial f}{\partial t} = f'f + \dot{f} \tag{13.45}$$

where the partial derivatives have been abbreviated in the usual way by $\frac{\partial f}{\partial t} = \dot{f}$ and $\frac{\partial f}{\partial Y} = f'$. Higher derivatives are given by

$$\frac{\mathrm{d}^2 f}{\mathrm{d}t^2} = f''f^2 + f'^2 f + 2\dot{f}'f + \ddot{f} \tag{13.46}$$

$$\frac{\mathrm{d}^3 f}{\mathrm{d}t^3} = \frac{\partial^3 f}{\partial t^3} + f'''f^3 + 3\dot{f}''f^2 + \ddot{f}f' + 3f''\dot{f}f + 3\dot{f}' + 4f''f'f^2 + 5\dot{f}'f'f + f'^3 f + f'^2\dot{f}. \tag{13.47}$$

13.6.1 Nordsieck Predictor-Corrector Method

Nordsieck [151] determines an interpolating polynomial of degree m. As variables he uses the 0th to mth derivatives[2] evaluated at the current time t, for instance for $m = 5$ he uses the variables

[2] In fact the derivatives of the interpolating polynomial which exist even if higher derivatives of f do not exist.

$$Y(t) \tag{13.48}$$

$$g(t) = \frac{\mathrm{d}}{\mathrm{d}t} Y(t) \tag{13.49}$$

$$a(t) = \frac{\Delta t}{2} \frac{\mathrm{d}^2}{\mathrm{d}t^2} Y(t) \tag{13.50}$$

$$b(t) = \frac{\Delta t^2}{6} \frac{\mathrm{d}^3}{\mathrm{d}t^3} Y(t) \tag{13.51}$$

$$c(t) = \frac{\Delta t^3}{24} \frac{\mathrm{d}^4}{\mathrm{d}t^4} Y(t) \tag{13.52}$$

$$d(t) = \frac{\Delta t^4}{120} \frac{\mathrm{d}^5}{\mathrm{d}t^5} Y(t). \tag{13.53}$$

Taylor expansion gives approximate values at $t + \Delta t$

$$\begin{aligned} Y(t + \Delta t) &= Y(t) + \Delta t \left[g(t) + a(t) + b(t) + c(t) + d(t) + e(t) \right] \\ &= Y^p(t + \Delta t) + e(t) \Delta t \end{aligned} \tag{13.54}$$

$$g(t+\Delta t) = g(t)+2a(t)+3b(t)+4c(t)+5d(t)+6e(t) = g^p(t+\Delta T)+6e(t) \tag{13.55}$$

$$a(t + \Delta t) = a(t) + 3b(t) + 6c(t) + 10d(t) + 15e(t) = a^p(t + \Delta t) + 15e(t) \tag{13.56}$$

$$b(t + \Delta t) = b(t) + 4c(t) + 10d(t) + 20e(t) = b^p(t + \Delta t) + 20e(t) \tag{13.57}$$

$$c(t + \Delta t) = c(t) + 5d(t) + 15e(t) = c^p(t + \Delta t) + 15e(t) \tag{13.58}$$

$$d(t + \Delta t) = d(t) + 6e(t) = d^p(t + \Delta t) + 6e(t) \tag{13.59}$$

where the next term of the Taylor series $e(t) = \frac{\Delta t^5}{6!} \frac{\mathrm{d}^6}{\mathrm{d}t^6} Y(t)$ has been introduced as an approximation to the truncation error of the predicted values Y^p, g^p, etc. It can be estimated from the second equation

$$e = \frac{1}{6} \left[f(Y^p(t + \Delta t), t + \Delta t) - g^p(t + \Delta t) \right] = \frac{1}{6} \delta f. \tag{13.60}$$

This predictor-corrector method turns out to be rather unstable. However, stability can be achieved by slightly modifying the coefficients of the corrector step. Nordsieck suggested to use

$$Y(t+\Delta t) = Y^p(t+\Delta t) + \frac{95}{288}\delta f \tag{13.61}$$

$$a(t+\Delta t) = a^p(t+\Delta t) + \frac{25}{24}\delta f \tag{13.62}$$

$$b(t+\Delta t) = b^p(t+\Delta t) + \frac{35}{72}\delta f \tag{13.63}$$

$$c(t+\Delta t) = c^p(t+\Delta t) + \frac{5}{48}\delta f \tag{13.64}$$

$$d(t+\Delta t) = d^p(t+\Delta t) + \frac{1}{120}\delta f\,. \tag{13.65}$$

13.6.2 Gear Predictor-Corrector Methods

Gear [152] designed special methods for molecular dynamics simulations (Chap. 15) where Newton's law (13.15) has to be solved numerically. He uses again a truncated Taylor expansion for the predictor step

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \mathbf{a}(t)\frac{\Delta t^2}{2} + \dot{\mathbf{a}}(t)\frac{\Delta t^3}{6} + \ddot{\mathbf{a}}(t)\frac{\Delta t^4}{24} + \cdots \tag{13.66}$$

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \mathbf{a}(t)\Delta t + \dot{\mathbf{a}}(t)\frac{\Delta t^2}{2} + \ddot{\mathbf{a}}(t)\frac{\Delta t^3}{6} + \cdots \tag{13.67}$$

$$\mathbf{a}(t+\Delta t) = \mathbf{a}(t) + \dot{\mathbf{a}}(t)\Delta t + \ddot{\mathbf{a}}(t)\frac{\Delta t^2}{2} + \cdots \tag{13.68}$$

$$\dot{\mathbf{a}}(t+\Delta t) = \dot{\mathbf{a}}(t) + \ddot{\mathbf{a}}(t)\Delta t + \cdots \tag{13.69}$$

$$\vdots$$

to calculate new coordinates etc. $\mathbf{r}^p_{n+1}, \mathbf{v}^p_{n+1}, \mathbf{a}^p_{n+1} \ldots$ (Fig. 13.7). The difference between the predicted acceleration and that calculated using the predicted coordinates

$$\delta\mathbf{a}_{n+1} = \mathbf{a}(\mathbf{r}^P_{n+1}, t+\Delta t) - \mathbf{a}^p_{n+1} \tag{13.70}$$

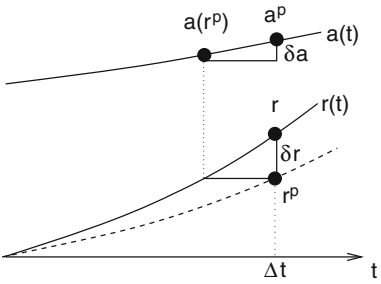


Fig. 13.7 (Gear Predictor Corrector Method) The difference between predicted acceleration $\mathbf{a}^p$ and acceleration calculated for the predicted coordinates $\mathbf{a}(\mathbf{r}^p)$ is used as a measure of the error to estimate the correction $\delta\mathbf{r}$

is then used as a measure of the error to correct the predicted values according to

$$\mathbf{r}_{n+1} = \mathbf{r}^p_{n+1} + c_1\delta\mathbf{a}_{n+1} \tag{13.71}$$

$$\mathbf{v}_{n+1} = \mathbf{v}^p_{n+1} + c_2\delta\mathbf{a}_{n+1} \tag{13.72}$$

$$\vdots$$

The coefficients c_i were determined to optimize stability and accuracy. For instance the fourth order Gear corrector reads

$$\mathbf{r}_{n+1} = \mathbf{r}^p_{n+1} + \frac{\Delta t^2}{12}\delta\mathbf{a}_{n+1} \tag{13.73}$$

$$\mathbf{v}_{n+1} = \mathbf{v}^p_{n+1} + \frac{5\Delta t}{12}\delta\mathbf{a}_{n+1} \tag{13.74}$$

$$\dot{\mathbf{a}}_{n+1} = \dot{\mathbf{a}}_n + \frac{1}{\Delta t}\delta\mathbf{a}_{n+1}. \tag{13.75}$$

Gear methods are generally not time reversible and show systematic energy drifts. A reversible symplectic predictor-corrector method has been presented recently by Martyna and Tuckerman [153].

13.7 Runge–Kutta Methods

If higher derivatives are not so easily available, they can be approximated by numerical differences. f is evaluated at several trial points and the results are combined to reproduce the Taylor series as close as possible [154].

13.7.1 Second Order Runge–Kutta Method

Let us begin with two function values. As common in the literature we will denote the function values as $K_1, K_2, \ldots$. From the gradient at time t_n

$$K_1 = f_n = f(Y(t_n), t_n) \tag{13.76}$$

we estimate the state vector at time $t_n + \Delta t$ as

$$Y(t_n + \Delta t) \approx \Delta t\, K_1. \tag{13.77}$$

The gradient at time $t_n + \Delta t$ is approximately

$$K_2 = f(Y(t_n) + \Delta t\, K_1, t_n + \Delta t) \tag{13.78}$$

which has the Taylor series expansion

$$K_2 = f_n + (\dot{f}_n + f_n' f_n)\Delta t + \cdots \tag{13.79}$$

and application of the trapezoidal rule (4.13) gives the 2nd order Runge–Kutta method

$$Y_{n+1} = Y_n + \frac{\Delta t}{2}(K_1 + K_2) \tag{13.80}$$

which in fact coincides with the improved Euler or Heun method. Taylor series expansion shows how the combination of K_1 and K_2 leads to an expression of higher error order:

$$\begin{aligned} Y_{n+1} &= Y_n + \frac{\Delta t}{2}(f_n + f_n + (\dot{f}_n + f_n' f_n)\Delta t + \cdots) \\ &= Y_n + f_n \Delta t + \frac{\mathrm{d}f_n}{\mathrm{d}t}\frac{\Delta t^2}{2} + \cdots . \end{aligned} \tag{13.81}$$

13.7.2 Third Order Runge–Kutta Method

The accuracy can be further improved by calculating one additional function value at mid-time. From (13.76) we estimate the gradient at mid-time by

$$\begin{aligned} K_2 &= f\left(Y(t) + \frac{\Delta t}{2}K_1, t + \frac{\Delta t}{2}\right) \\ &= f_n + (\dot{f}_n + f_n' f_n)\frac{\Delta t}{2} + (\ddot{f}_n + f_n'' f_n^2 + 2\dot{f}_n' f_n)\frac{\Delta t^2}{8} + \cdots . \end{aligned} \tag{13.82}$$

The gradient at time $t_n + \Delta t$ is then estimated as

$$\begin{aligned} K_3 &= f(Y(t_n) + \Delta t(2K_2 - K_1), t_n + \Delta t) \\ &= f_n + \dot{f}_n \Delta t + f_n'(2K_2 - K_1)\Delta t + \ddot{f}_n \frac{\Delta t^2}{2} \\ &+ f_n'' \frac{(2K_2 - K_1)^2 \Delta t^2}{2} + 2\dot{f}_n' \frac{(2K_2 - K_1)\Delta t^2}{2} + \cdots . \end{aligned} \tag{13.83}$$

Inserting the expansion (13.82) gives the leading terms

$$K_3 = f_n + (\dot{f}_n + f_n' f_n)\Delta t + (2f_n'^2 f_n + f_n'' f_n^2 + \ddot{f}_n + 2f_n' \dot{f}_n + 2\dot{f}_n^2) \frac{\Delta t^2}{2} + \cdots . \tag{13.84}$$

Applying Simpson's rule (4.14) we combine the three gradients to get the 3rd order Runge–Kutta method

$$Y_{n+1} = Y(t_n) + \frac{\Delta t}{6}(K_1 + 4K_2 + K_3) \tag{13.85}$$

where the Taylor series

$$\begin{aligned} Y_{n+1} &= Y(t_n) + \frac{\Delta t}{6}\left(6f_n + 3(\dot{f}_n + f_n f_n')\Delta t \right. \\ &\left. + (f_n'^2 f_n + f_n'' f_n^2 + 2\dot{f}_n' f_n + f_n + \dot{}_n f_n')\ddot{}\Delta t^2 + \cdot\cdot f\cdot\right) \\ &= Y(t_n + \Delta t) + O(\Delta t^4) \end{aligned} \tag{13.86}$$

recovers the exact Taylor series (13.44) including terms of order $O(\Delta t^3)$.

13.7.3 Fourth Order Runge–Kutta Method

The 4th order Runge–Kutta method (RK4) is often used because of its robustness and accuracy. It uses two different approximations for the midpoint

$$\begin{aligned} K_1 &= f(Y(t_n), t_n) \\ K_2 &= f\left(Y(t_n) + \frac{K_1}{2}\Delta t, t_n + \frac{\Delta t}{2}\right) \\ K_3 &= f\left(Y(t_n) + \frac{K_2}{2}\Delta t, t_n + \frac{\Delta t}{2}\right) \\ K_4 &= f(Y(t_n) + K_3 \Delta t, t_n + \Delta t) \end{aligned}$$

and Simpson's rule (4.14) to obtain

$$Y_{n+1} = Y(t_n) + \frac{\Delta t}{6}(K_1 + 2K_2 + 2K_3 + K_4) = Y(t_n + \Delta t) + O(\Delta t^5).$$

Expansion of the Taylor series is cumbersome but with the help of an algebra program one can easily check that the error is of order Δt^5.

13.8 Quality Control and Adaptive Step Size Control

For practical applications it is necessary to have an estimate for the local error and to adjust the step size properly. With the Runge Kutta method this can be achieved by a step doubling procedure. We calculate y_{n+2} first by two steps Δt and then by one step $2\Delta t$. This needs 11 function evaluations as compared to 8 for the smaller step size only (Fig. 13.8). For the 4th order method we estimate the following errors:

$$\Delta\left(Y_{n+2}^{(\Delta t)}\right) = 2a\Delta t^5 \tag{13.87}$$

$$\Delta\left(Y_{n+2}^{(2\Delta t)}\right) = a(2\Delta t)^5. \tag{13.88}$$

The local error can be estimated from

$$|Y_{n+2}^{(\Delta t)} - Y_{n+2}^{(2\Delta t)}| = 30|a|\Delta t^5$$

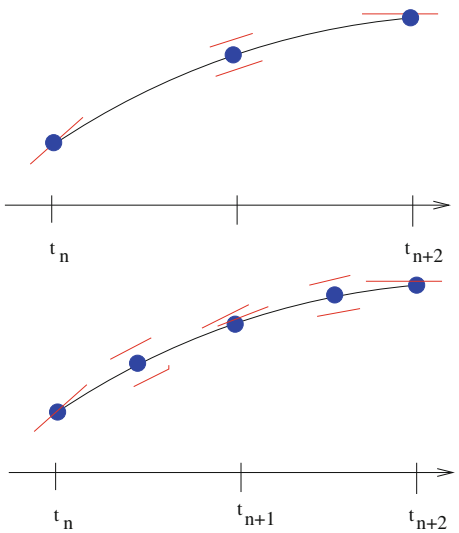


Fig. 13.8 Step doubling with the fourth order Runge–Kutta method

$$\Delta\left(Y_{n+1}^{(\Delta t)}\right) = a\Delta t^5 = \frac{|Y_{n+2}^{(\Delta t)} - Y_{n+2}^{(2\Delta t)}|}{30}.$$

The step size Δt can now be adjusted to keep the local error within the desired limits.

13.9 Extrapolation Methods

Application of the extrapolation method to calculate the integral $\int_{t_n}^{t_{n+1}} f(t)dt$ produces very accurate results but can also be time consuming. The famous Gragg-Bulirsch-Stoer method [2] starts from an explicit midpoint rule with a special starting procedure. The interval Δt is divided into a sequence of N sub-steps

$$h = \frac{\Delta t}{N}. \tag{13.89}$$

First a simple Euler step is performed

$$\begin{aligned} u_0 &= Y(t_n) \\ u_1 &= u_0 + hf(u_0, t_n) \end{aligned} \tag{13.90}$$

and then the midpoint rule is applied repeatedly to obtain

$$u_{j+1} = u_{j-1} + 2hf(u_j, t_n + jh) \qquad j = 1, 2 \ldots N - 1. \tag{13.91}$$

Gragg [155] introduced a smoothing procedure to remove oscillations of the leading error term by defining

$$v_j = \frac{1}{4}u_{j-1} + \frac{1}{2}u_j + \frac{1}{4}u_{j+1}. \tag{13.92}$$

He showed that both approximations (13.91, 13.92) have an asymptotic expansion in powers of h^2 and are therefore well suited for an extrapolation method. The modified midpoint method can be summarized as follows:

$$\begin{aligned} u_0 &= Y(t_n) \\ u_1 &= u_0 + hf(u_0, t_n) \\ u_{j+1} &= u_{j-1} + 2hf(u_j, t_n + jh) \quad j = 1, 2, \ldots N - 1 \\ Y(t_n + \Delta t) &\approx \frac{1}{2}\left(u_N + u_{N-1} + hf(u_N, t_n + \Delta t)\right). \end{aligned} \tag{13.93}$$

The number of sub-steps N is increased according to a sequence like

$$N = 2, 4, 6, 8, 12, 16, 24, 32, 48, 64 \ldots \quad N_j = 2N_{j-2} \quad \text{Bulirsch-Stoer sequence} \tag{13.94}$$

or

$$N = 2, 4, 6, 8, 10, 12 \ldots \quad N_j = 2j \quad \text{Deuflhard sequence.}$$

After each successive N is tried, a polynomial extrapolation is attempted. This extrapolation returns both the extrapolated values and an error estimate. If the error is still too large then N has to be increased further. A more detailed discussion can be found in [156, 157].

13.10 Linear Multistep Methods

All methods discussed so far evaluated one or more values of the gradient $f(Y(t), t)$ only within the interval $t_n \cdots t_n + \Delta t$. If the state vector changes sufficiently smooth then multistep methods can be applied. Linear multistep methods use a combination of function values Y_n and gradients f_n from several steps

$$Y_{n+1} = \sum_{j=1}^{k} \left(\alpha_j Y_{n-j+1} + \beta_j f_{n-j+1} \Delta t\right) + \beta_0 f_{n+1} \Delta t \tag{13.95}$$

where the coefficients α, β are determined such, that a polynomial of certain order r is integrated exactly. The method is explicit if $\beta_0 = 0$ and implicit otherwise. Multistep methods have a small local error and need fewer function evaluations. On the other hand, they have to be combined with other methods (like Runge–Kutta) to start and end properly and it can be rather complicated to change the step size during the calculation. Three families of linear multistep methods are commonly used: explicit Adams-Bashforth methods, implicit Adams-Moulton methods and backward differentiation formulas (also known as Gear formulas [158]).

13.10.1 Adams-Bashforth Methods

The explicit Adams-Bashforth method of order r uses the gradients from the last $r - 1$ steps (Fig. 13.9) to obtain the polynomial

$$p(t_n) = f(Y_n, t_n), \ldots p(t_{n-r+1}) = f(Y_{n-r+1}, t_{n-r+1}) \tag{13.96}$$

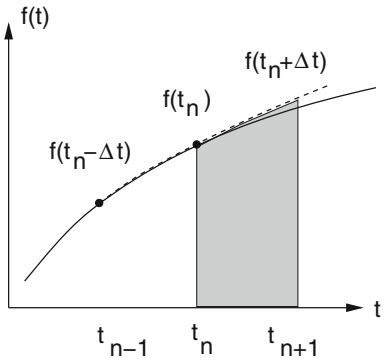


Fig. 13.9 Adams-Bashforth method

and to calculate the approximation

$$Y_{n+1} - Y_n \approx \int_{t_n}^{t_{n+1}} p(t)dt$$

which is generally a linear combination of $f_n \cdots f_{n-r+1}$. For example, the Adams-Bashforth formulas of order 2, 3, 4 are:

$$\begin{aligned}
Y_{n+1} - Y_n &= \frac{\Delta t}{2}(3f_n - f_{n-1}) + O(\Delta t^3) \\
Y_{n+1} - Y_n &= \frac{\Delta t}{12}(23f_n - 16f_{n-1} + 5f_{m-2}) + O(\Delta t^4) \\
Y_{n+1} - Y_n &= \frac{\Delta t}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) + O(\Delta t^5).
\end{aligned} \tag{13.97}$$

13.10.2 Adams-Moulton Methods

The implicit Adams-Moulton method also uses the yet not known value Y_{n+1} (Fig. 13.10) to obtain the polynomial

$$p(t_{n+1}) = f_{n+1}, \ldots p(t_{n-r+2}) = f_{n-r+2}. \tag{13.98}$$

The corresponding Adams-Moulton formulas of order 2 to 4 are:

$$\begin{aligned}
Y_{n+1} - Y_n &= \frac{\Delta t}{2}(f_{n+1} + f_n) + O(\Delta t^3) \\
Y_{n+1} - Y_n &= \frac{\Delta t}{12}(5f_{n+1} + 8f_n - f_{n-1}) + O(\Delta t^4) \qquad (13.99) \\
Y_{n+1} - Y_n &= \frac{\Delta t}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}) + O(\Delta t^5). \qquad (13.100)
\end{aligned}$$

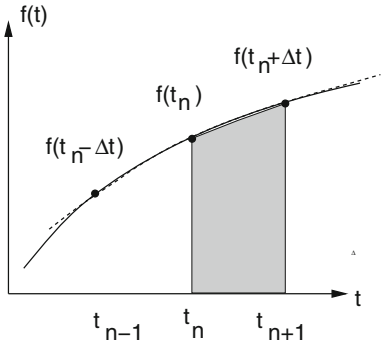


Fig. 13.10 Adams-Moulton method

13.10.3 Backward Differentiation (Gear) Methods

Gear methods [158] are implicit and usually combined with a modified Newton method. They make use of previous function values $Y_n, Y_{n-1} \ldots$ and the gradient f_{n+1} at time $t + \Delta t$. Only methods of order $r \leq 6$ are stable and useful. The general formula (13.95) is

$$Y_{n+1} = \sum_{j=1}^{r} \alpha_j Y_{n-j+1} + \beta_0 f_{n+1} \Delta t. \tag{13.101}$$

For $r = 1$ this becomes

$$Y_{n+1} = \alpha_1 Y_n + \beta_0 f_1 \Delta t \tag{13.102}$$

and all linear polynomials

$$p = p_0 + p_1(t - t_n), \quad \frac{\mathrm{d}p}{\mathrm{d}t} = p_1 \tag{13.103}$$

are integrated exactly if

$$p_0 + p_1 \Delta t = \alpha_1 p_0 + \beta_0 p_1 \tag{13.104}$$

which is the case for

$$\alpha_1 = 1, \quad \beta_0 = \Delta t. \tag{13.105}$$

Hence the first order Gear method is

$$Y_{n+1} = Y_n + f_{n+1} \Delta t + O(\Delta t^2) \tag{13.106}$$

which coincides with the implicit Euler method. The higher order stable Gear methods are given by

$$r = 2: \quad Y_{n+1} = \frac{4}{3}Y_n - \frac{1}{3}Y_{n-1} + \frac{2}{3}f_{n+1}\Delta t + O(\Delta t^3) \tag{13.107}$$

$$r = 3: \quad Y_{n+1} = \frac{18}{11}Y_n - \frac{9}{11}Y_{n-1} + \frac{2}{11}Y_{n-2} + \frac{6}{11}f_{n+1}\Delta t + O(\Delta t^4) \tag{13.108}$$

$$r = 4: \quad Y_{n+1} = \frac{48}{25}Y_n - \frac{36}{25}Y_{n-1} + \frac{16}{25}Y_{n-2} - \frac{3}{25}Y_{n-3} + \frac{12}{25}f_{n+1}\Delta t + O(\Delta t^5) \tag{13.109}$$

$$r = 5: \quad Y_{n+1} = \frac{300}{137}Y_n - \frac{300}{137}Y_{n-1} + \frac{200}{137}Y_{n-2} - \frac{75}{137}Y_{n-3} + \frac{12}{137}Y_{n-4} + \frac{60}{137}f_{n+1}\Delta t + O(\Delta t^6) \tag{13.110}$$

$$r = 6: \quad Y_{n+1} = \frac{120}{49}Y_n - \frac{150}{49}Y_{n-1} + \frac{400}{147}Y_{n-2} - \frac{75}{49}Y_{n-3} + \frac{24}{49}Y_{n-4} - \frac{10}{147}Y_{n-5} + \frac{20}{49}f_{n+1}\Delta t + O(\Delta t^7). \tag{13.111}$$

This class of algorithms is useful also for stiff problems (differential equations with strongly varying eigenvalues).

13.10.4 Predictor-Corrector Methods

The Adams-Bashforth–Moulton method combines the explicit method as a predictor step to calculate an estimate y^p_{n+1} with a corrector step using the implicit method of same order. The general class of linear multistep predictor corrector methods [159] uses a predictor step

$$Y^{(0)}_{n+1} = \sum_{j=1}^{k} \left(\alpha_j^{(p)} Y_{n-j+1} + \beta_j^{(p)} f_{n-j+1}\Delta t \right) \tag{13.112}$$

which is corrected using the formula

$$Y^{(1)}_{n+1} = \sum_{j=1}^{k} \left(\alpha_j^{(c)} Y_{n-j+1} + \beta_j^{(c)} f_{n-j+1}\Delta t \right) + \beta_0 f(Y^{(0)}_{n+1}, t_{n+1})\Delta t \tag{13.113}$$

and further iterations

$$Y_{n+1}^{(m+1)} = Y_{n+1}^{(m)} - \beta_0 \left[f(Y_{n+1}^{(m-1)}, t_{n+1}) - f(Y_{n+1}^{(m)}, t_{n+1}) \right] \Delta t \quad m = 1 \ldots M-1 \tag{13.114}$$

$$Y_{n+1} = Y_{n+1}^{(M)}, \quad \dot{Y}_{n+1} = f(Y_{n+1}^{(M-1)}, t_{n+1}). \tag{13.115}$$

The coefficients α, β have to be determined to optimize accuracy and stability.

13.11 Verlet Methods

For classical molecular dynamics simulations it is necessary to calculate very long trajectories. Here a family of symplectic methods often is used which conserve the phase space volume [160–165]. The equations of motion of a classical interacting N-body system are

$$m_i \ddot{\mathbf{x}}_i = F_i \tag{13.116}$$

where the force acting on atom i can be calculated once a specific force field is chosen. Let us write these equations as a system of first order differential equations

$$\begin{pmatrix} \dot{\mathbf{x}}_i \\ \dot{\mathbf{v}}_i \end{pmatrix} = \begin{pmatrix} \mathbf{v}_i \\ \mathbf{a}_i \end{pmatrix} \tag{13.117}$$

where $\mathbf{x}(t)$ and $\mathbf{v}(t)$ are functions of time and the forces $m\mathbf{a}(\mathbf{x}(t))$ are functions of the time dependent coordinates.

13.11.1 Liouville Equation

We rewrite (13.117) as

$$\begin{pmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{v}} \end{pmatrix} = \mathcal{L} \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} \tag{13.118}$$

where the Liouville operator $\mathcal{L}$ acts on the vector containing all coordinates and velocities:

$$\mathcal{L} \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} = \left(\mathbf{v} \frac{\partial}{\partial \mathbf{x}} + \mathbf{a} \frac{\partial}{\partial \mathbf{v}} \right) \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix}. \tag{13.119}$$

The Liouville equation (13.118) can be formally solved by

$$\begin{pmatrix} \mathbf{x}(t) \\ \mathbf{v}(t) \end{pmatrix} = e^{\mathcal{L}t} \begin{pmatrix} \mathbf{x}(0) \\ \mathbf{v}(0) \end{pmatrix}. \tag{13.120}$$

For a better understanding let us evaluate the first members of the Taylor series of the exponential:

$$\mathcal{L} \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} = \left(\mathbf{v} \frac{\partial}{\partial \mathbf{x}} + \mathbf{a} \frac{\partial}{\partial \mathbf{v}} \right) \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \mathbf{v} \\ \mathbf{a} \end{pmatrix} \tag{13.121}$$

$$\mathcal{L}^2 \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} = \left(\mathbf{v} \frac{\partial}{\partial \mathbf{x}} + \mathbf{a} \frac{\partial}{\partial \mathbf{v}} \right) \begin{pmatrix} \mathbf{v} \\ \mathbf{a}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \mathbf{a} \\ \mathbf{v} \frac{\partial}{\partial \mathbf{x}} \mathbf{a} \end{pmatrix} \tag{13.122}$$

$$\mathcal{L}^3 \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} = \left(\mathbf{v} \frac{\partial}{\partial \mathbf{x}} + \mathbf{a} \frac{\partial}{\partial \mathbf{v}} \right) \begin{pmatrix} \mathbf{a} \\ \mathbf{v} \frac{\partial}{\partial \mathbf{x}} \mathbf{a} \end{pmatrix} = \begin{pmatrix} \mathbf{v} \frac{\partial}{\partial \mathbf{x}} \mathbf{a} \\ \mathbf{a} \frac{\partial}{\partial \mathbf{x}} \mathbf{a} + \mathbf{v}\mathbf{v} \frac{\partial}{\partial \mathbf{x}} \frac{\partial}{\partial \mathbf{x}} \mathbf{a} \end{pmatrix}. \tag{13.123}$$

But since

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathbf{a}(\mathbf{x}(t)) = \mathbf{v} \frac{\partial}{\partial \mathbf{x}} \mathbf{a} \tag{13.124}$$

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2} \mathbf{a}(\mathbf{x}(t)) = \frac{\mathrm{d}}{\mathrm{d}t} \left(\mathbf{v} \frac{\partial}{\partial \mathbf{x}} \mathbf{a} \right) = \mathbf{a} \frac{\partial}{\partial \mathbf{x}} \mathbf{a} + \mathbf{v}\mathbf{v} \frac{\partial}{\partial \mathbf{x}} \frac{\partial}{\partial \mathbf{x}} \mathbf{a} \tag{13.125}$$

we recover

$$\left(1 + t\mathcal{L} + \frac{1}{2} t^2 \mathcal{L}^2 + \frac{1}{6} t^3 \mathcal{L}^3 + \cdots \right) \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \mathbf{x} + \mathbf{v}t + \frac{1}{2} t^2 \mathbf{a} + \frac{1}{6} t^3 \dot{\mathbf{a}} + \cdots \\ \mathbf{v} + \mathbf{a}t + \frac{1}{2} t^2 \dot{\mathbf{a}} + \frac{1}{6} t^3 \ddot{\mathbf{a}} + \cdots \end{pmatrix}. \tag{13.126}$$

13.11.2 Split Operator Approximation

We introduce a small time step $\Delta t = t/N$ and write

$$e^{\mathcal{L}t} = \left(e^{\mathcal{L}\Delta t} \right)^N. \tag{13.127}$$

For the small time step Δt the split-operator approximation can be used which approximately factorizes the exponential operator. For example, write the Liouville operator as the sum of two terms

$$\mathcal{L}_A = \mathbf{v}\frac{\partial}{\partial \mathbf{x}} \qquad \mathcal{L}_B = \mathbf{a}\frac{\partial}{\partial \mathbf{v}}$$

and make the approximation

$$e^{\mathcal{L}\Delta t} = e^{\mathcal{L}_A \Delta t} e^{\mathcal{L}_B \Delta t} + \cdots . \tag{13.128}$$

Each of the two factors simply shifts positions or velocities

$$e^{\mathcal{L}_A \Delta t}\begin{pmatrix}\mathbf{x}\\ \mathbf{v}\end{pmatrix} = \begin{pmatrix}\mathbf{x}+\mathbf{v}\Delta t\\ \mathbf{v}\end{pmatrix} \qquad e^{\mathcal{L}_B \Delta t}\begin{pmatrix}\mathbf{x}\\ \mathbf{v}\end{pmatrix} = \begin{pmatrix}\mathbf{x}\\ \mathbf{v}+\mathbf{a}\Delta t\end{pmatrix} \tag{13.129}$$

since these two steps correspond to either motion with constant velocities or constant coordinates and forces.

13.11.3 Position Verlet Method

Often the following approximation is used which is symmetrical in time

$$e^{\mathcal{L}\Delta t} = e^{\mathcal{L}_A \Delta t/2} e^{\mathcal{L}_B \Delta t} e^{\mathcal{L}_A \Delta t/2} + \cdots . \tag{13.130}$$

The corresponding algorithm is the so called position Verlet method (Fig. 13.11):

$$\mathbf{x}_{n+1/2} = \mathbf{x}_n + \mathbf{v}_n \frac{\Delta t}{2} \tag{13.131}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \mathbf{a}_{n+1/2}\Delta t = \mathbf{v}(t_n + \Delta t) + O(\Delta t^3) \tag{13.132}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_{n+1/2} + \mathbf{v}_{n+1}\frac{\Delta t}{2} = \mathbf{x}_n + \frac{\mathbf{v}_n + \mathbf{v}_{n+1}}{2}\Delta t = \mathbf{x}(t_n + \Delta t) + O(\Delta t^3). \tag{13.133}$$

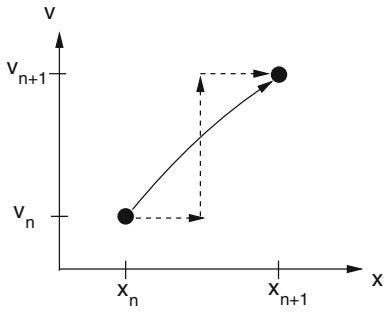


Fig. 13.11 (Position Verlet method) The exact integration path is approximated by two half-steps with constant velocities and one step with constant coordinates

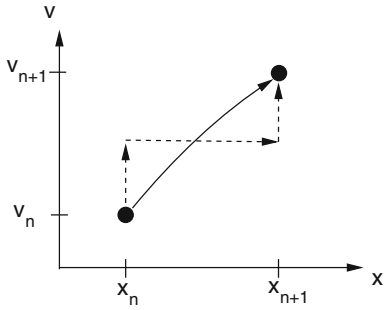


Fig. 13.12 (Velocity Verlet method) The exact integration path is approximated by two half-steps with constant coordinates and one step with constant velocities

13.11.4 Velocity Verlet Method

If we exchange operators A and B we have

$$e^{\mathcal{L}\Delta t} = e^{\mathcal{L}_B \Delta t/2} e^{\mathcal{L}_A \Delta t} e^{\mathcal{L}_B \Delta t/2} + \cdots \tag{13.134}$$

which produces the velocity Verlet algorithm (Fig. 13.12):

$$\mathbf{v}_{n+1/2} = \mathbf{v}_n + \mathbf{a}_n \frac{\Delta t}{2} \tag{13.135}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{v}_{n+1/2}\Delta t = \mathbf{x}_n + \mathbf{v}_n \Delta t + \mathbf{a}_n \frac{\Delta t^2}{2} = \mathbf{x}(t_n + \Delta t) + O(\Delta t^3) \tag{13.136}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_{n+1/2} + \mathbf{a}_{n+1}\frac{\Delta t}{2} = \mathbf{v}_n + \frac{\mathbf{a}_n + \mathbf{a}_{n+1}}{2}\Delta t = \mathbf{v}(t_n + \Delta t) + O(\Delta t^3). \tag{13.137}$$

13.11.5 Stoermer-Verlet Method

The velocity Verlet method is equivalent to Stoermer's version [166] of the Verlet method which is a two step method given by

$$\mathbf{x}_{n+1} = 2\mathbf{x}_n - \mathbf{x}_{n-1} + \mathbf{a}_n \Delta t^2 \tag{13.138}$$

$$\mathbf{v}_n = \frac{\mathbf{x}_{n+1} - \mathbf{x}_{n-1}}{2\Delta t}. \tag{13.139}$$

To show the equivalence we add two consecutive position vectors

$$\mathbf{x}_{n+2} + \mathbf{x}_{n+1} = 2\mathbf{x}_{n+1} + 2\mathbf{x}_n - \mathbf{x}_n - \mathbf{x}_{n-1} + (\mathbf{a}_{n+1} + \mathbf{a}_n)\Delta t^2 \tag{13.140}$$

which simplifies to

$$\mathbf{x}_{n+2} - \mathbf{x}_n - (\mathbf{x}_{n+1} - \mathbf{x}_n) = (\mathbf{a}_{n+1} + \mathbf{a}_n)\Delta t^2. \tag{13.141}$$

This can be expressed as the difference of two consecutive velocities:

$$2(\mathbf{v}_{n+1} - \mathbf{v}_n) = (\mathbf{a}_{n+1} + \mathbf{a}_n)\Delta t. \tag{13.142}$$

Now we substitute

$$\mathbf{x}_{n-1} = \mathbf{x}_{n+1} - 2\mathbf{v}_n \Delta t \tag{13.143}$$

to get

$$\mathbf{x}_{n+1} = 2\mathbf{x}_n - \mathbf{x}_{n+1} + 2\mathbf{v_n}\Delta t + \mathbf{a}_n \Delta t^2 \tag{13.144}$$

which simplifies to

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{v}_n \Delta t + \frac{\mathbf{a}_n}{2}\Delta t^2. \tag{13.145}$$

Thus the equations of the velocity Verlet algorithm have been recovered. However, since the Verlet method is a 2-step method, the choice of initial values is important. The Stoermer-Verlet method starts from two coordinate sets x_0, x_1. The first step is

$$\mathbf{x}_2 = 2\mathbf{x_1} - \mathbf{x_0} + a_1 \Delta t^2 \tag{13.146}$$

$$\mathbf{v}_1 = \frac{\mathbf{x}_2 - \mathbf{x}_0}{2\Delta t} = \frac{\mathbf{x}_1 - \mathbf{x}_0}{\Delta t} + \frac{\mathbf{a}_1}{2}\Delta t^2. \tag{13.147}$$

The velocity Verlet method, on the other hand, starts from one set of coordinates and velocities $\mathbf{x}_1, \mathbf{v}_1$. Here the first step is

$$\mathbf{x}_2 = \mathbf{x}_1 + \mathbf{v}_1 \Delta t + \mathbf{a}_1 \frac{\Delta t^2}{2} \tag{13.148}$$

$$\mathbf{v}_2 = \mathbf{v}_1 + \frac{\mathbf{a}_1 + \mathbf{a}_2}{2}\Delta t. \tag{13.149}$$

The two methods give the same resulting trajectory if we choose

$$\mathbf{x}_0 = \mathbf{x}_1 - \mathbf{v}_1 \Delta t + \frac{\mathbf{a}_1}{2}\Delta t^2. \tag{13.150}$$

If, on the other hand, $\mathbf{x}_0$ is known with higher precision, the local error order of Stoermer's algorithm changes as can be seen from addition of the two Taylor series

$$\mathbf{x}(t_n + \Delta t) = \mathbf{x}_n + \mathbf{v}_n \Delta t + \frac{\mathbf{a}_n}{2}\Delta t^2 + \frac{\dot{\mathbf{a}}_n}{6}\Delta t^3 + \cdots \tag{13.151}$$

$$\mathbf{x}(t_n - \Delta t) = \mathbf{x}_n - \mathbf{v}_n \Delta t + \frac{\mathbf{a}_n}{2}\Delta t^2 - \frac{\dot{\mathbf{a}}_n}{6}\Delta t^3 + \cdots \tag{13.152}$$

which gives

$$\mathbf{x}(t_n + \Delta t) = 2\mathbf{x}(t_n) - \mathbf{x}(t_n - \Delta t) + \mathbf{a}_n \Delta t^2 + O(\Delta t^4) \tag{13.153}$$

$$\frac{\mathbf{x}(t_n + \Delta t) - \mathbf{x}(t_n - \Delta t)}{2\Delta t} = \mathbf{v}_n + O(\Delta t^2). \tag{13.154}$$

13.11.6 Error Accumulation for the Stoermer-Verlet Method

Equation (13.153) gives only the local error of one single step. Assume the start values $\mathbf{x}_0$ and $\mathbf{x}_1$ are exact. The next value $\mathbf{x}_2$ has an error with the leading term $\Delta x_2 = \alpha \Delta t^4$. If the trajectory is sufficiently smooth and the time step not too large the coefficient α will vary only slowly and the error of the next few iterations is given by

$$\begin{aligned}
\Delta x_3 &= 2\Delta x_2 - \Delta x_1 = 2\alpha \Delta t^4 \\
\Delta x_4 &= 2\Delta x_3 - \Delta x_2 = 3\alpha \Delta t^4 \\
&\vdots \\
\Delta x_{n+1} &= n\alpha \Delta t^4.
\end{aligned} \tag{13.155}$$

This shows that the effective error order of the Stoermer-Verlet method is only $O(\Delta t^3)$ similar to the velocity Verlet method.

13.11.7 Beeman's Method

Beeman and Schofield [167, 168] introduced a method which is very similar to the Stoermer-Verlet method but calculates the velocities with higher precision. This is important if, for instance, the kinetic energy has to be calculated. Starting from the Taylor series

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{v}_n \Delta t + \mathbf{a}_n \frac{\Delta t^2}{2} + \dot{\mathbf{a}}_n \frac{\Delta t^3}{6} + \ddot{\mathbf{a}}_n \frac{\Delta t^4}{24} + \cdots \tag{13.156}$$

the derivative of the acceleration is approximated by a backward difference

$$\begin{aligned}\mathbf{x}_{n+1} &= \mathbf{x}_n + \mathbf{v}_n \Delta t + \mathbf{a}_n \frac{\Delta t^2}{2} + \frac{\mathbf{a}_n - \mathbf{a}_{n-1}}{\Delta t} \frac{\Delta t^3}{6} + O(\Delta t^4) \\ &= \mathbf{x}_n + \mathbf{v}_n \Delta t + \frac{4\mathbf{a}_n - \mathbf{a}_{n-1}}{6} \Delta t^2 + O(\Delta t^4). \end{aligned} \tag{13.157}$$

This equation can be used as an explicit step to update the coordinates or as a predictor step in combination with the implicit corrector step

$$\begin{aligned}\mathbf{x}_{n+1} &= \mathbf{x}_n + \mathbf{v}_n \Delta t + \mathbf{a}_n \frac{\Delta t^2}{2} + \frac{\mathbf{a}_{n+1} - \mathbf{a}_n}{\Delta t} \frac{\Delta t^3}{6} + O(\Delta t^4) \\ &= \mathbf{x}_n + \mathbf{v}_n \Delta t + \frac{\mathbf{a}_{n+1} + 2\mathbf{a}_n}{6} \Delta t^2 + O(\Delta t^4) \end{aligned} \tag{13.158}$$

which can be applied repeatedly (usually two iterations are sufficient). Similarly, the Taylor series of the velocity is approximated by

$$\begin{aligned}\mathbf{v}_{n+1} &= \mathbf{v}_n + \mathbf{a}_n \Delta t + \dot{\mathbf{a}}_n \frac{\Delta t^2}{2} + \ddot{\mathbf{a}}_n \frac{\Delta t^3}{6} + \cdots \\ &= \mathbf{v}_n + \mathbf{a}_n \Delta t + \left(\frac{\mathbf{a}_{n+1} - \mathbf{a}_n}{\Delta t} + O(\Delta t) \right) \frac{\Delta t^2}{2} + \cdots \\ &= \mathbf{v}_n + \frac{\mathbf{a}_{n+1} + \mathbf{a}_n}{2} \Delta t + O(\Delta t^3). \end{aligned} \tag{13.159}$$

Inserting the velocity from (13.158) we obtain the corrector step for the velocity

$$\begin{aligned}\mathbf{v}_{n+1} &= \frac{\mathbf{x}_{n+1} - \mathbf{x}_n}{\Delta t} - \frac{\mathbf{a}_{n+1} + 2\mathbf{a}_n}{6} \Delta t + \frac{\mathbf{a}_{n+1} + \mathbf{a}_n}{2} \Delta t + O(\Delta t^3) \\ &= \frac{\mathbf{x}_{n+1} - \mathbf{x}_n}{\Delta t} + \frac{2\mathbf{a}_{n+1} + \mathbf{a}_n}{6} \Delta t + O(\Delta t^3). \end{aligned} \tag{13.160}$$

In combination with (13.157) this can be replaced by

$$\begin{aligned}\mathbf{v}_{n+1} &= \mathbf{v}_n + \frac{4\mathbf{a}_n - \mathbf{a}_{n-1}}{6} \Delta t + \frac{2\mathbf{a}_{n+1} + \mathbf{a}_n}{6} \Delta t + O(\Delta t^3) \\ &= \mathbf{v}_n + \frac{2\mathbf{a}_{n+1} + 5\mathbf{a}_n - \mathbf{a}_{n-1}}{6} \Delta t + O(\Delta t^3). \end{aligned} \tag{13.161}$$

Together, (13.157) and (13.161) provide an explicit method which is usually understood as Beeman's method. Inserting the velocity (13.160) from the previous step

$$\mathbf{v}_n = \frac{\mathbf{x}_n - \mathbf{x}_{n-1}}{\Delta t} + \frac{2\mathbf{a}_n + \mathbf{a}_{n-1}}{6} \Delta t + O(\Delta t^3) \tag{13.162}$$

into (13.157) gives

$$\mathbf{x}_{n+1} = 2\mathbf{x}_n - \mathbf{x}_{n-1} + \mathbf{a}_n \Delta t^2 + O(\Delta t^4) \tag{13.163}$$

which coincides with the Stoermer-Verlet method (13.138). We conclude that Beeman's method should produce the same trajectory as the Stoermer-Verlet method if numerical errors can be neglected and comparable initial values are used. In fact, the Stoermer-Verlet method may suffer from numerical extinction and Beeman's method provides a numerically more favorable alternative.

13.11.8 The Leapfrog Method

Closely related to the Verlet methods is the so called leapfrog method [165]. It uses the simple decomposition

$$e^{\mathcal{L}\Delta t} \approx e^{\mathcal{L}_A \Delta t} e^{\mathcal{L}_B \Delta t} \tag{13.164}$$

but introduces two different time grids for coordinates and velocities which are shifted by $\Delta t/2$ (Fig. 13.13).

The leapfrog algorithm is given by

$$\mathbf{v}_{n+1/2} = \mathbf{v}_{n-1/2} + \mathbf{a}_n \Delta t \tag{13.165}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{v}_{n+1/2} \Delta t. \tag{13.166}$$

Due to the shifted arguments the order of the method is increased as can be seen from the Taylor series:

$$\mathbf{x}(t_n) + \left(\mathbf{v}(t_n) + \frac{\Delta t}{2} \mathbf{a}(t_n) + \cdots \right) \Delta t = \mathbf{x}(t_n + \Delta t) + O(\Delta t^3) \tag{13.167}$$

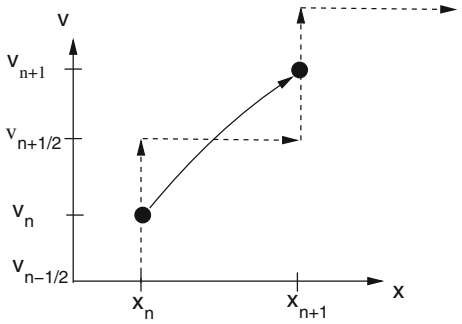


Fig. 13.13 (Leapfrog method) The exact integration path is approximated by one step with constant coordinates and one step with constant velocities. Two different grids are used for coordinates and velocities which are shifted by $\Delta t/2$

$$\mathbf{v}\left(t_n + \frac{\Delta t}{2}\right) - \mathbf{v}\left(t_n - \frac{\Delta t}{2}\right) = \mathbf{a}(t_n)\Delta t + O(\Delta t^3). \tag{13.168}$$

One disadvantage of the leapfrog method is that some additional effort is necessary if the velocities are needed. The simple expression

$$\mathbf{v}(t_n) = \frac{1}{2}\left(\mathbf{v}\left(t_n - \frac{\Delta t}{2}\right) + \mathbf{v}\left(t_n + \frac{\Delta t}{2}\right)\right) + O(\Delta t^2) \tag{13.169}$$

is of lower error order than (13.168).

Problems

Problem 13.1 Circular Orbits

In this computer experiment we consider a mass point moving in a central field. The equation of motion can be written as the following system of first order equations:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{v}_x \\ \dot{v}_y \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{1}{(x^2+y^2)^{3/2}} & 0 & 0 & 0 \\ 0 & -\frac{1}{(x^2+y^2)^{3/2}} & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix}. \tag{13.170}$$

For initial values

$$\begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{13.171}$$

the exact solution is given by

$$x = \cos t \quad y = \sin t. \tag{13.172}$$

The following methods are used to calculate the position $x(t)$, $y(t)$ and the energy

$$E_{tot} = E_{kin} + E_{pot} = \frac{1}{2}(v_x^2 + v_y^2) - \frac{1}{\sqrt{x^2+y^2}}. \tag{13.173}$$

- The explicit Euler method (13.3)

$$\begin{aligned} x(t_{n+1}) &= x(t_n) + v_x(t_n)\Delta t \\ y(t_{n+1}) &= y(t_n) + v_y(t_n)\Delta t \\ v_x(t_{n+1}) &= v_x(t_n) - \tfrac{x(t_n)}{R(t_n)^3}\Delta t \\ v_y(t_{n+1}) &= v_y(t_n) - \tfrac{y(t_n)}{R(t_n)^3}\Delta t. \end{aligned} \tag{13.174}$$

- The 2nd order Runge–Kutta method (13.7.1)

which consists of the predictor step

$$x(t_n + \Delta t/2) = x(t_n) + \frac{\Delta t}{2} v_x(t_n) \tag{13.175}$$

$$y(t_n + \Delta t/2) = y(t_n) + \frac{\Delta t}{2} v_y(t_n) \tag{13.176}$$

$$v_x(t_n + \Delta t/2) = v_x(t_n) - \frac{\Delta t}{2} \frac{x(t_n)}{R(t_n)^3} \tag{13.177}$$

$$v_y(t_n + \Delta t/2) = v_y(t_n) - \frac{\Delta t}{2} \frac{y(t_n)}{R(t_n)^3} \tag{13.178}$$

and the corrector step

$$x(t_{n+1}) = x(t_n) + \Delta t\ v_x(t_n + \Delta t/2) \tag{13.179}$$

$$y(t_{n+1}) = y(t_n) + \Delta t\ v_y(t_n + \Delta t/2) \tag{13.180}$$

$$v_x(t_{n+1}) = v_x(t_n) - \Delta t \frac{x(t_n + \Delta t/2)}{R^3(t_n + \Delta t/2)} \tag{13.181}$$

$$v_y(t_{n+1}) = v_y(t_n) - \Delta t \frac{y(t_n + \Delta t/2)}{R^3(t_n + \Delta t/2)}. \tag{13.182}$$

- The fourth order Runge–Kutta method (13.7.3)
- The Verlet method (13.11.5)

$$x(t_{n+1}) = x(t_n) + (x(t_n) - x(t_{n-1})) - \Delta t \frac{x(t_n)}{R^3(t_n)} \tag{13.183}$$

$$y(t_{n+1}) = y(t_n) + (y(t_n) - y(t_{n-1})) - \Delta t \frac{y(t_n)}{R^3(t_n)} \tag{13.184}$$

$$v_x(t_n) = \frac{x(t_{n+1}) - x(t_{n-1})}{2\Delta t} = \frac{x(t_n) - x(t_{n-1})}{\Delta t} - \frac{\Delta t}{2} \frac{x(t_n)}{R^3(t_n)} \tag{13.185}$$

$$v_y(t_n) = \frac{y(t_{n+1}) - y(t_{n-1})}{2\Delta t} = \frac{y(t_n) - y(t_{n-1})}{\Delta t} - \frac{\Delta t}{2} \frac{y(t_n)}{R^3(t_n)}. \tag{13.186}$$

To start the Verlet method we need additional coordinates at time $-\Delta t$ which can be chosen from the exact solution or from the approximation

$$x(t_{-1}) = x(t_0) - \Delta t\, v_x(t_0) - \frac{\Delta t^2}{2}\frac{x(t_0)}{R^3(t_0)} \tag{13.187}$$

$$y(t_{-1}) = y(t_0) - \Delta t\, v_y(t_0) - \frac{\Delta t^2}{2}\frac{y(t_0)}{R^3(t_0)}. \tag{13.188}$$

- The leapfrog method (13.11.8)

$$x(t_{n+1}) = x(t_n) + v_x(t_{n+1/2})\Delta t \tag{13.189}$$

$$y(t_{n+1}) = y(t_n) + v_y(t_{n+1/2})\Delta t \tag{13.190}$$

$$v_x(t_{n+1/2}) = v_x(t_{n-1/2}) - \frac{x(t_n)}{R(t_n)^3}\Delta t \tag{13.191}$$

$$v_y(t_{n+1/2}) = v_y(t_{n-1/2}) - \frac{y(t_n)}{R(t_n)^3}\Delta t \tag{13.192}$$

where the velocity at time t_n is calculated from

$$v_x(t_n) = v_x(t_{n+1/2}) - \frac{\Delta t}{2}\frac{x(t_{n+1})}{R^3(t_{n+1})} \tag{13.193}$$

$$v_y(t_n) = v_y(t_{n+1/2}) - \frac{\Delta t}{2}\frac{y(t_{n+1})}{R^3(t_{n+1})}. \tag{13.194}$$

To start the leapfrog method we need the velocity at time $t_{-1/2}$ which can be taken from the exact solution or from

$$v_x(t_{-1/2}) = v_x(t_0) - \frac{\Delta t}{2}\frac{x(t_0)}{R^3(t_0)} \tag{13.195}$$

$$v_y(t_{-1/2}) = v_y(t_0) - \frac{\Delta t}{2}\frac{y(t_0)}{R^3(t_0)}. \tag{13.196}$$

Compare the conservation of energy for the different methods as a function of the time step Δt. Study the influence of the initial values for leapfrog and Verlet methods.

Problem 13.2 N-body System

In this computer experiment we simulate the motion of three mass points under the influence of gravity. Initial coordinates and velocities as well as the masses can be varied. The equations of motion are solved with the 4th order Runge–Kutta method with quality control for different step sizes. The local integration error is estimated

using the step doubling method. Try to simulate a planet with a moon moving round a sun!

Problem 13.3 Adams-Bashforth Method

In this computer experiment we simulate a circular orbit with the Adams-Bashforth method of order 2 . . . 7. The absolute error at time T

$$\Delta(T) = |x(T) - \cos(T)| + |y(t) - \sin(T)| + |v_x(T) + \sin(T)| + |v_y(T) - \cos(T)| \tag{13.197}$$

is shown as a function of the time step Δt in a log-log plot. From the slope

$$s = \frac{d(\log_{10}(\Delta))}{d(\log_{10}(\Delta t))} \tag{13.198}$$

the leading error order s can be determined. For very small step sizes rounding errors become dominating which leads to an increase $\Delta \sim (\Delta t)^{-1}$.

Determine maximum precision and optimal step size for different orders of the method. Compare with the explicit Euler method.

Part II
Simulation of Classical and Quantum Systems

Chapter 14
Rotational Motion

An asymmetric top under the influence of time dependent external forces is a rather complicated subject in mechanics. Efficient methods to describe the rotational motion are important as well in astrophysics as in molecular physics. The orientation of a rigid body relative to the laboratory system can be described by a 3×3 *matrix. Instead of solving nine equations for all its components, the rotation matrix can be parametrized by the four real components of a quaternion. Euler angles use the minimum necessary number of three parameters but have numerical disadvantages. Care has to be taken to conserve the orthogonality of the rotation matrix. Omelyan's implicit quaternion method is very efficient and conserves orthogonality exactly. In computer experiments we compare different explicit and implicit methods for a free rotor, we simulate a rotor in an external field and the collision of two rotating molecules.*

14.1 Transformation to a Body Fixed Coordinate System

Let us define a rigid body as a set of mass points m_i with fixed relative orientation (described by distances and angles).

The position of m_i in the laboratory coordinate system *CS* will be denoted by $\mathbf{r_i}$. The position of the center of mass (COM) of the rigid body is

$$\mathbf{R} = \frac{1}{\sum_i m_i} \sum_i m_i \mathbf{r}_i \tag{14.1}$$

and the position of m_i within the COM coordinate system CS_c (Fig. 14.1) is $\boldsymbol{\rho}_i$:

$$\mathbf{r}_i = \mathbf{R} + \boldsymbol{\rho}_i. \tag{14.2}$$

Let us define a body fixed coordinate system CS_{cb}, where the position $\boldsymbol{\rho}_{\mathbf{ib}}$ of m_i is time independent $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\rho}_{\mathbf{ib}} = 0$. $\boldsymbol{\rho}_{\mathbf{i}}$ and $\boldsymbol{\rho}_{ib}$ are connected by a linear vector function

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_14

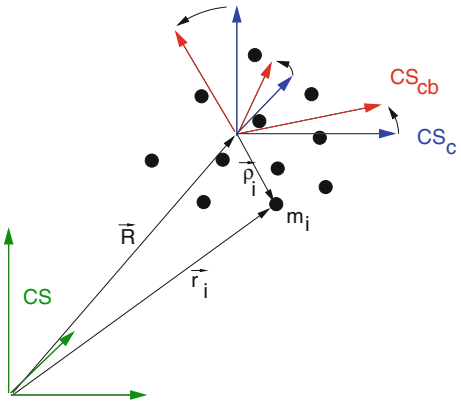


Fig. 14.1 (Coordinate systems) Three coordinate systems will be used: The laboratory system CS, the center of mass system CS_c and the body fixed system CS_{cb}

$$\boldsymbol{\rho_i} = A\boldsymbol{\rho}_{ib} \tag{14.3}$$

where A is a 3×3 matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}. \tag{14.4}$$

14.2 Properties of the Rotation Matrix

Rotation conserves the length of $\boldsymbol{\rho}$ [1]:

$$\boldsymbol{\rho}^T\boldsymbol{\rho} = (A\boldsymbol{\rho})^T(A\boldsymbol{\rho}) = \boldsymbol{\rho}^T A^T A\boldsymbol{\rho}. \tag{14.5}$$

Consider the matrix

$$M = A^T A - 1 \tag{14.6}$$

for which

$$\boldsymbol{\rho}^T M\boldsymbol{\rho} = 0 \tag{14.7}$$

holds for all vectors $\boldsymbol{\rho}$. Let us choose the unit vector in x-direction: $\boldsymbol{\rho} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$. Then we have

[1] $\boldsymbol{\rho}^T\boldsymbol{\rho}$ denotes the scalar product of two vectors whereas $\boldsymbol{\rho}\boldsymbol{\rho}^T$ is the outer or matrix product.

$$0 = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = M_{11}. \tag{14.8}$$

Similarly by choosing a unit vector in y or z direction we find $M_{22} = M_{33} = 0$.

Now choose $\boldsymbol{\rho} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$:

$$\begin{aligned} 0 &= \begin{pmatrix} 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} M_{11} + M_{12} \\ M_{21} + M_{22} \\ M_{31} + M_{32} \end{pmatrix} = M_{11} + M_{22} + M_{12} + M_{21}. \end{aligned} \tag{14.9}$$

Since the diagonal elements vanish we have $M_{12} = -M_{21}$. With $\boldsymbol{\rho} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$, $\boldsymbol{\rho} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$ we find $M_{13} = -M_{31}$ and $M_{23} = -M_{32}$, hence M is skew symmetric and has three independent components

$$M = -M^T = \begin{pmatrix} 0 & M_{12} & M_{13} \\ -M_{12} & 0 & M_{23} \\ -M_{13} & -M_{23} & 0 \end{pmatrix}. \tag{14.10}$$

Inserting (14.6) we have

$$(A^T A - 1) = -(A^T A - 1)^T = -(A^T A - 1) \tag{14.11}$$

which shows that $A^T A = 1$ or equivalently $A^T = A^{-1}$. Hence $(\det(A))^2 = 1$ and A is an orthogonal matrix. For a pure rotation without reflection only $\det(A) = +1$ is possible.

From

$$\mathbf{r_i} = \mathbf{R} + A\boldsymbol{\rho}_{\mathbf{ib}} \tag{14.12}$$

we calculate the velocity

$$\frac{\mathrm{d}\mathbf{r_i}}{\mathrm{d}t} = \frac{\mathrm{d}\mathbf{R}}{\mathrm{d}t} + \frac{\mathrm{d}A}{\mathrm{d}t}\boldsymbol{\rho}_{\mathbf{ib}} + A\frac{\mathrm{d}\boldsymbol{\rho}_{\mathbf{ib}}}{\mathrm{d}t} \tag{14.13}$$

but since $\boldsymbol{\rho}_{\mathbf{ib}}$ is constant by definition, the last summand vanishes

$$\dot{\mathbf{r}}_i = \dot{\mathbf{R}} + \dot{A}\boldsymbol{\rho}_{ib} = \dot{\mathbf{R}} + \dot{A}A^{-1}\boldsymbol{\rho}_i \tag{14.14}$$

and in the center of mass system we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\rho}_i = \dot{A}A^{-1}\boldsymbol{\rho}_i = W\boldsymbol{\rho}_i \tag{14.15}$$

with the matrix

$$W = \dot{A}A^{-1}. \tag{14.16}$$

14.3 Properties of W, Connection with the Vector of Angular Velocity

Since rotation does not change the length of $\boldsymbol{\rho}_i$, we have

$$0 = \frac{\mathrm{d}}{\mathrm{d}t}|\boldsymbol{\rho}_i|^2 \rightarrow 0 = \boldsymbol{\rho}_i \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\rho}_i = \boldsymbol{\rho}_i(W\boldsymbol{\rho}_i) \tag{14.17}$$

or in matrix notation

$$0 = \boldsymbol{\rho}_i^T W \boldsymbol{\rho}_i. \tag{14.18}$$

This holds for arbitrary $\boldsymbol{\rho}_i$. Hence W is skew symmetric and has three independent components

$$W = \begin{pmatrix} 0 & W_{12} & W_{13} \\ -W_{12} & 0 & W_{23} \\ -W_{13} & -W_{23} & 0 \end{pmatrix}. \tag{14.19}$$

Now consider an infinitesimal rotation by the angle $\mathrm{d}\varphi$ (Fig. 14.2).

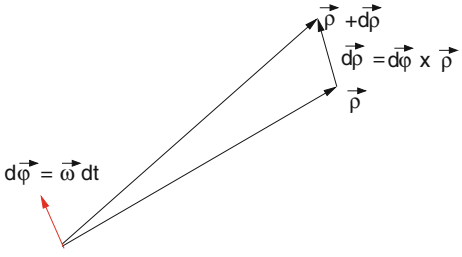


Fig. 14.2 Infinitesimal rotation

Then we have (the index i is suppressed)

$$d\boldsymbol{\rho} = \frac{\mathrm{d}\boldsymbol{\rho}}{\mathrm{d}t}dt = \begin{pmatrix} 0 & W_{12} & W_{13} \\ -W_{12} & 0 & W_{23} \\ -W_{13} & -W_{23} & 0 \end{pmatrix} \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{pmatrix} dt = \begin{pmatrix} W_{12}\rho_2 + W_{13}\rho_3 \\ -W_{12}\rho_1 + W_{23}\rho_3 \\ -W_{13}\rho_1 - W_{23}\rho_2 \end{pmatrix} dt \tag{14.20}$$

which can be written as a cross product:

$$d\boldsymbol{\rho} = d\boldsymbol{\varphi} \times \boldsymbol{\rho} \tag{14.21}$$

with

$$d\boldsymbol{\varphi} = \begin{pmatrix} -W_{23}dt \\ W_{13}dt \\ -W_{12}dt \end{pmatrix}. \tag{14.22}$$

But this can be expressed in terms of the angular velocity $\boldsymbol{\omega}$ as

$$d\boldsymbol{\varphi} = \boldsymbol{\omega}dt \tag{14.23}$$

and finally we have

$$d\boldsymbol{\varphi} = \boldsymbol{\omega}dt = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} dt \quad W = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \tag{14.24}$$

and the more common form of the equation of motion

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\rho} = W\boldsymbol{\rho} = \boldsymbol{\omega} \times \boldsymbol{\rho}. \tag{14.25}$$

Example:Rotation Around the z-axis

For constant angular velocity $\boldsymbol{\omega}$ the equation of motion

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\rho} = W\boldsymbol{\rho} \tag{14.26}$$

has the formal solution

$$\boldsymbol{\rho} = \mathrm{e}^{Wt}\,\boldsymbol{\rho}(0) = A(t)\,\boldsymbol{\rho}(0). \tag{14.27}$$

The angular velocity vector for rotation around the z-axis is

$$\boldsymbol{\omega} = \begin{pmatrix} 0 \\ 0 \\ \omega_3 \end{pmatrix} \tag{14.28}$$

and

$$W = \begin{pmatrix} 0 & -\omega_3 & 0 \\ \omega_3 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \tag{14.29}$$

Higher powers of W can be easily calculated since

$$W^2 = \begin{pmatrix} -\omega_3^2 & 0 & 0 \\ 0 & -\omega_3^2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{14.30}$$

$$W^3 = -\omega_3^2 \begin{pmatrix} 0 & -\omega_3 & 0 \\ \omega_3 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{14.31}$$

etc., and the rotation matrix is obtained from the Taylor series

$$\begin{aligned} A(t) &= \mathrm{e}^{Wt} = 1 + Wt + \frac{1}{2}W^2t^2 + \frac{1}{6}W^3t^3 + \cdots \\ &= 1 + \begin{pmatrix} \omega_3^2t^2 & 0 & 0 \\ 0 & \omega_3^2t^2 & 0 \\ 0 & 0 & 0 \end{pmatrix}\left(-\frac{1}{2} + \frac{\omega_3^2t^2}{24} + \cdots\right) + \begin{pmatrix} 0 & -\omega_3t & 0 \\ \omega_3t & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}\left(1 - \frac{\omega_3^2t^2}{6} + \cdots\right) \\ &= \begin{pmatrix} \cos(\omega_3t) & -\sin(\omega_3t) & \\ \sin(\omega_3t) & \cos(\omega_3t) & \\ & & 1 \end{pmatrix}. \end{aligned} \tag{14.32}$$

14.4 Transformation Properties of the Angular Velocity

Now imagine we are sitting on the rigid body and observe a mass point moving outside. Its position in the laboratory system is $\mathbf{r_1}$. In the body fixed system we observe it at

$$\boldsymbol{\rho}_{1b} = A^{-1}(\mathbf{r}_1 - \mathbf{R}) \tag{14.33}$$

and its velocity in the body fixed system is

$$\dot{\boldsymbol{\rho}}_{1b} = A^{-1}(\dot{\mathbf{r}}_1 - \dot{\mathbf{R}}) + \frac{\mathrm{d}A^{-1}}{\mathrm{d}t}(\mathbf{r}_1 - \mathbf{R}). \tag{14.34}$$

The time derivative of the inverse matrix follows from

$$0 = \frac{\mathrm{d}}{\mathrm{d}t}(A^{-1}A) = A^{-1}\dot{A} + \frac{\mathrm{d}A^{-1}}{\mathrm{d}t}A \tag{14.35}$$

$$\frac{\mathrm{d}A^{-1}}{\mathrm{d}t} = -A^{-1}\dot{A}A^{-1} = -A^{-1}W \tag{14.36}$$

and hence

$$\frac{\mathrm{d}A^{-1}}{\mathrm{d}t}(\mathbf{r}_1 - \mathbf{R}) = -A^{-1}W(\mathbf{r}_1 - \mathbf{R}). \tag{14.37}$$

Now we rewrite this using the angular velocity as it is observed in the body fixed system

$$-A^{-1}W(\mathbf{r}_1 - \mathbf{R}) = -W_bA^{-1}(\mathbf{r_1} - \mathbf{R}) = -W_b\boldsymbol{\rho}_{1b} = -\boldsymbol{\omega}_b \times \boldsymbol{\rho}_{1b} \tag{14.38}$$

where W transforms as like a rank-2 tensor

$$W_b = A^{-1}WA. \tag{14.39}$$

From this equation the transformation properties of $\boldsymbol{\omega}$ can be derived. We consider only rotation around one axis explicitly, since a general rotation matrix can always be written as a product of three rotations around different axes. For instance, rotation around the z-axis gives:

$$\begin{aligned}
W_b &= \begin{pmatrix} 0 & -\omega_{b3} & \omega_{b2} \\ \omega_{b3} & 0 & -\omega_{b1} \\ -\omega_{b2} & \omega_{b1} & 0 \end{pmatrix} = \\
&\begin{pmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \begin{pmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} = \\
&= \begin{pmatrix} 0 & -\omega_3 & \omega_2\cos\varphi - \omega_1\sin\varphi \\ \omega_3 & 0 & -(\omega_1\cos\varphi + \omega_2\sin\varphi) \\ -(\omega_2\cos\varphi - \omega_1\sin\varphi) & \omega_1\cos\varphi + \omega_2\sin\varphi & 0 \end{pmatrix}
\end{aligned} \tag{14.40}$$

which shows that

$$\begin{pmatrix}\omega_{1b}\\ \omega_{2b}\\ \omega_{3b}\end{pmatrix} = \begin{pmatrix}\cos\varphi & \sin\varphi & 0\\ -\sin\varphi & \cos\varphi & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}\omega_1\\ \omega_2\\ \omega_3\end{pmatrix} = A^{-1}\boldsymbol{\omega} \tag{14.41}$$

i.e. $\boldsymbol{\omega}$ transforms like a vector under rotations. However, there is a subtle difference considering general coordinate transformations involving reflections. For example, under reflection at the xy-plane W is transformed according to

$$W_b = \begin{pmatrix}1&0&0\\0&1&0\\0&0&-1\end{pmatrix}\begin{pmatrix}0&-\omega_3&\omega_2\\ \omega_3&0&-\omega_1\\ -\omega_2&\omega_1&0\end{pmatrix}\begin{pmatrix}1&0&0\\0&1&0\\0&0&-1\end{pmatrix}$$
$$= \begin{pmatrix}0&-\omega_3&-\omega_2\\ \omega_3&0&\omega_1\\ \omega_2&-\omega_1&0\end{pmatrix} \tag{14.42}$$

and the transformed angular velocity vector is

$$\begin{pmatrix}\omega_{1b}\\ \omega_{2b}\\ \omega_{3b}\end{pmatrix} = -\begin{pmatrix}1&0&0\\0&1&0\\0&0&-1\end{pmatrix}\begin{pmatrix}\omega_1\\ \omega_2\\ \omega_3\end{pmatrix}. \tag{14.43}$$

This is characteristic of a so called axial or pseudo-vector. Under a general coordinate transformation it transforms as

$$\boldsymbol{\omega}_b = \det(A)A\boldsymbol{\omega}. \tag{14.44}$$

14.5 Momentum and Angular Momentum

The total momentum is

$$\mathbf{P} = \sum_i m_i\dot{\mathbf{r}}_i = \sum_i m_i\dot{\mathbf{R}} = M\dot{\mathbf{R}} \tag{14.45}$$

since by definition we have $\sum_i m_i\boldsymbol{\rho}_i = 0$.

The total angular momentum can be decomposed into the contribution of the center of mass motion and the contribution relative to the center of mass

$$\mathbf{L} = \sum_i m_i\mathbf{r}_i \times \dot{\mathbf{r}}_i = M\mathbf{R}\times\dot{\mathbf{R}} + \sum_i m_i\boldsymbol{\rho}_i\times\dot{\boldsymbol{\rho}}_i = \mathbf{L}_{COM} + \mathbf{L}_{int}. \tag{14.46}$$

The second contribution is

$$\mathbf{L}_{int} = \sum_i m_i \boldsymbol{\rho}_i \times (\boldsymbol{\omega} \times \boldsymbol{\rho}_i) = \sum_i m_i \left(\boldsymbol{\omega} \rho_i^2 - \boldsymbol{\rho}_i (\boldsymbol{\rho}_i \boldsymbol{\omega}) \right). \tag{14.47}$$

This is a linear vector function of $\boldsymbol{\omega}$, which can be expressed simpler by introducing the tensor of inertia

$$I = \sum_i m_i \rho_i^2 \, 1 - m_i \boldsymbol{\rho}_i \boldsymbol{\rho}_i^T \tag{14.48}$$

or component-wise

$$I_{m,n} = \sum_i m_i \rho_i^2 \delta_{m,n} - m_i \rho_{i,m} \rho_{i,n} \tag{14.49}$$

as

$$\mathbf{L}_{int} = I\boldsymbol{\omega}. \tag{14.50}$$

14.6 Equations of Motion of a Rigid Body

Let $\mathbf{F}_i$ be an external force acting on m_i. Then the equation of motion for the center of mass is

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2} \sum_i m_i \mathbf{r}_i = M\ddot{\mathbf{R}} = \sum_i \mathbf{F}_i = \mathbf{F}_{ext}. \tag{14.51}$$

If there is no total external force $\mathbf{F}_{ext}$, the center of mass moves with constant velocity

$$\mathbf{R} = \mathbf{R}_0 + \mathbf{V}(t - t_0). \tag{14.52}$$

The time derivative of the angular momentum equals the total external torque

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{L} = \frac{\mathrm{d}}{\mathrm{d}t} \sum_i m_i \mathbf{r}_i \times \dot{\mathbf{r}}_i = \sum_i m_i \mathbf{r}_i \times \ddot{\mathbf{r}}_i = \sum_i \mathbf{r}_i \times \mathbf{F}_i = \sum_i \mathbf{N}_i = \mathbf{N}_{ext} \tag{14.53}$$

which can be decomposed into

$$\mathbf{N}_{ext} = \mathbf{R} \times \mathbf{F}_{ext} + \sum_i \boldsymbol{\rho}_i \times \mathbf{F}_i. \tag{14.54}$$

With the decomposition of the angular momentum

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{L} = \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{L}_{COM} + \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{L}_{int} \tag{14.55}$$

we have two separate equations for the two contributions:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{L}_{COM} = \frac{\mathrm{d}}{\mathrm{d}t}M\mathbf{R}\times\dot{\mathbf{R}} = M\mathbf{R}\times\ddot{\mathbf{R}} = \mathbf{R}\times\mathbf{F}_{ext} \tag{14.56}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{L}_{int} = \sum_i \boldsymbol{\rho}_i\times\mathbf{F}_i = \mathbf{N}_{ext} - \mathbf{R}\times\mathbf{F}_{ext} = \mathbf{N}_{int} \tag{14.57}$$

14.7 Moments of Inertia

The angular momentum (14.50) is

$$\mathbf{L}_{Rot} = I\boldsymbol{\omega} = AA^{-1}IAA^{-1}\boldsymbol{\omega} = AI_b\boldsymbol{\omega}_b \tag{14.58}$$

where the tensor of inertia in the body fixed system is

$$\begin{aligned} I_b &= A^{-1}IA = A^{-1}\left(\sum_i m_i\boldsymbol{\rho}_i^T\rho_i - m_i\boldsymbol{\rho}_i\boldsymbol{\rho}_i^T\right)A \\ &= \sum_i m_iA^T\boldsymbol{\rho}_i^T\boldsymbol{\rho}_iA - m_iA^T\boldsymbol{\rho}_i\boldsymbol{\rho}_i^TA \\ &= \sum_i m_i\rho_{ib}^2 - m_i\boldsymbol{\rho}_{ib}\boldsymbol{\rho}_{ib}^T. \end{aligned} \tag{14.59}$$

Since I_b does not depend on time (by definition of the body fixed system) we will use the principal axes of I_b as the axes of the body fixed system. Then I_b takes the simple form

$$I_b = \begin{pmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{pmatrix} \tag{14.60}$$

with the principle moments of inertia $I_{1,2,3}$.

14.8 Equations of Motion for a Rotor

The following equations describe pure rotation of a rigid body:

$$\frac{\mathrm{d}}{\mathrm{d}t}A = WA = AW_b \tag{14.61}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{L}_{int} = \mathbf{N}_{int} \tag{14.62}$$

$$W = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \qquad W_{ij} = -\varepsilon_{ijk}\omega_k \tag{14.63}$$

$$\mathbf{L}_{int} = A\mathbf{L}_{int,b} = I\omega = AI_b\boldsymbol{\omega}_b \tag{14.64}$$

$$\boldsymbol{\omega}_b = I_b^{-1}\mathbf{L}_{int,b} = \begin{pmatrix} I_1^{-1} & 0 & 0 \\ 0 & I_2^{-1} & 0 \\ 0 & 0 & I_3^{-1} \end{pmatrix} \mathbf{L}_{int,b} \qquad \boldsymbol{\omega} = A\boldsymbol{\omega}_b \tag{14.65}$$

$$I_b = \text{const.} \tag{14.66}$$

14.9 Explicit Methods

Equation (14.61) for the rotation matrix and (14.62) for the angular momentum have to be solved by a suitable algorithm. The simplest integrator is the explicit Euler method (Fig. 14.3) [169]:

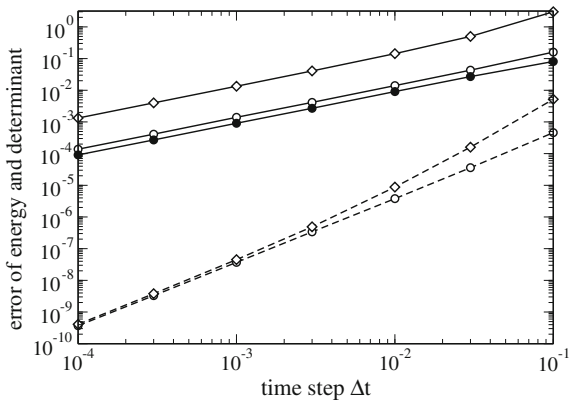


Fig. 14.3 (Global error of the explicit methods) The equations of a free rotor (14.8) are solved using the explicit first order (*full curves*) and second order (*dashed curves*) method. The deviations $|\det(A) - 1|$ (*diamonds*) and $|E_{kin} - E_{kin}(0)|$ (*circles*) at t=10 are shown as a function of the time step Δt. Full circles show the energy deviation of the first order method with reorthogonalization. The principal moments of inertia are $I_b = \mathrm{diag}(1, 2, 3)$ and the initial angular momentum is $\mathbf{L} = (1, 1, 1)$. See also Problem 14.1

$$A(t+\Delta t) = A(t) + A(t)W_b(t)\Delta t + O(\Delta t^2) \tag{14.67}$$

$$\mathbf{L}_{int}(t+\Delta t) = \mathbf{L}_{int}(t) + \mathbf{N}_{int}(t)\Delta t + O(\Delta t^2). \tag{14.68}$$

Expanding the Taylor series of $A(t)$ to second order we have the second order approximation (Fig. 14.3)

$$A(t+\Delta t) = A(t) + A(t)W_b(t)\Delta t + \frac{1}{2}\left(A(t)W_b^2(t) + A(t)\dot{W}_b(t)\right)\Delta t^2 + O(\Delta t^3). \tag{14.69}$$

A corresponding second order expression for the angular momentum involves the time derivative of the forces and is usually not practicable.

The time derivative of W can be expressed via the time derivative of the angular velocity which can be calculated as follows:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\omega}_b &= \frac{\mathrm{d}}{\mathrm{d}t}\left(I_b^{-1}A^{-1}\mathbf{L}_{int}\right) = I_b^{-1}\left(\frac{\mathrm{d}}{\mathrm{d}t}A^{-1}\right)\mathbf{L}_{int} + I_b^{-1}A^{-1}\mathbf{N}_{int} = \\ &= I_b^{-1}\left(-A^{-1}W\right)\mathbf{L}_{int} + I_b^{-1}A^{-1}\mathbf{N}_{int} = -I_b^{-1}W_b\mathbf{L}_{int,b} + I_b^{-1}\mathbf{N}_{int,b}.\end{aligned} \tag{14.70}$$

Alternatively, in the laboratory system

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\omega} &= \frac{\mathrm{d}}{\mathrm{d}t}(A\boldsymbol{\omega}_b) = WA\omega_b - AI_b^{-1}A^{-1}W\mathbf{L}_{int} + AI_b^{-1}A^{-1}\mathbf{N}_{int} \\ &= AI_b^{-1}A(\mathbf{N}_{int} - W\mathbf{L}_{int})\end{aligned} \tag{14.71}$$

where the first summand vanishes due to

$$WA\boldsymbol{\omega}_b = AW_b\boldsymbol{\omega}_b = A\boldsymbol{\omega}_b \times \boldsymbol{\omega}_b = 0. \tag{14.72}$$

Substituting the angular momentum we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\omega}_b = I_b^{-1}\mathbf{N}_{int,b} - I_b^{-1}W_bI_b\boldsymbol{\omega}_b \tag{14.73}$$

which reads in components:

$$\begin{aligned}\begin{pmatrix}\dot{\omega}_{b1}\\ \dot{\omega}_{b2}\\ \dot{\omega}_{b3}\end{pmatrix} &= \begin{pmatrix}I_{b1}^{-1}N_{b1}\\ I_{b2}^{-1}N_{b2}\\ I_{b3}^{-1}N_{b3}\end{pmatrix} \\ &- \begin{pmatrix}I_{b1}^{-1} & & \\ & I_{b2}^{-1} & \\ & & I_{b3}^{-1}\end{pmatrix}\begin{pmatrix}0 & -\omega_{b3} & \omega_{b2}\\ \omega_{b3} & 0 & -\omega_{b1}\\ -\omega_{b2} & \omega_{b1} & 0\end{pmatrix}\begin{pmatrix}I_{b1}\omega_{b1}\\ I_{b2}\omega_{b2}\\ I_{b3}\omega_{b3}\end{pmatrix}\end{aligned} \tag{14.74}$$

Evaluation of the product gives a set of equations which are well known as Euler's equations:

$$\begin{aligned}
\dot{\omega}_{b1} &= \tfrac{I_{b2}-I_{b3}}{I_{b1}}\omega_{b2}\omega_{b3} + \tfrac{N_{b1}}{I_{b1}} \\
\dot{\omega}_{b2} &= \tfrac{I_{b3}-I_{b1}}{I_{b2}}\omega_{b3}\omega_{b1} + \tfrac{N_{b2}}{I_{b2}} \\
\dot{\omega}_{b3} &= \tfrac{I_{b1}-I_{b2}}{I_{b3}}\omega_{b1}\omega_{b2} + \tfrac{N_{b3}}{I_{b3}}
\end{aligned} \tag{14.75}$$

14.10 Loss of Orthogonality

The simple methods above do not conserve the orthogonality of A. This is an effect of higher order but the error can accumulate quickly. Consider the determinant of A. For the simple explicit Euler scheme we have

$$\det(A + \Delta A) = \det(A + WA\Delta t) = \det A \ \det(1 + W\Delta t) = \det A \ (1 + \omega^2 \Delta t^2). \tag{14.76}$$

The error is of order Δt^2, but the determinant will continuously increase, i.e. the rigid body will explode. For the second order integrator we find

$$\begin{aligned}
&\det(A + \Delta A) = \det\left(A + WA\Delta t + \frac{\Delta t^2}{2}(W^2 A + \dot{W}A)\right) \\
&= \det A \ \det\left(1 + W\Delta t + \frac{\Delta t^2}{2}(W^2 + \dot{W})\right).
\end{aligned} \tag{14.77}$$

This can be simplified to give

$$\det(A + \Delta A) = \det A \ (1 + \dot{\boldsymbol{\omega}}\boldsymbol{\omega}\Delta t^3 + \cdots). \tag{14.78}$$

The second order method behaves somewhat better since the product of angular velocity and acceleration can change in time. To assure that A remains a rotation matrix we must introduce constraints or reorthogonalize A at least after some steps (for instance every time when $|\det(A) - 1|$ gets larger than a certain threshold). The following method with a symmetric correction matrix is a very useful alternative [170]. The non-singular square matrix A can be decomposed into the product of an orthonormal matrix $\tilde{A}$ and a positive semi-definite matrix S

$$A = \widetilde{A}S \tag{14.79}$$

with the positive definite square root of the symmetric matrix $A^T A$

$$S = (A^T A)^{1/2} \tag{14.80}$$

and

$$\tilde{A} = AS^{-1} = A(A^T A)^{-1/2} \tag{14.81}$$

which is orthonormal as can be seen from

$$\widetilde{A}^T\widetilde{A} = (S^{-1})^T A^T A S^{-1} = S^{-1}S^2S^{-1} = 1. \tag{14.82}$$

Since the deviation of A from orthogonality is small, we make the approximations

$$S = 1 + s \tag{14.83}$$

$$A^T A = S^2 \approx 1 + 2s \tag{14.84}$$

$$s \approx \frac{A^T A - 1}{2} \tag{14.85}$$

$$S^{-1} \approx 1 - s \approx 1 + \frac{1 - A^T A}{2} + \cdots \tag{14.86}$$

which can be easily evaluated.

14.11 Implicit Method

The quality of the method can be significantly improved by taking the time derivative at midstep (Fig. 14.4) (13.5):

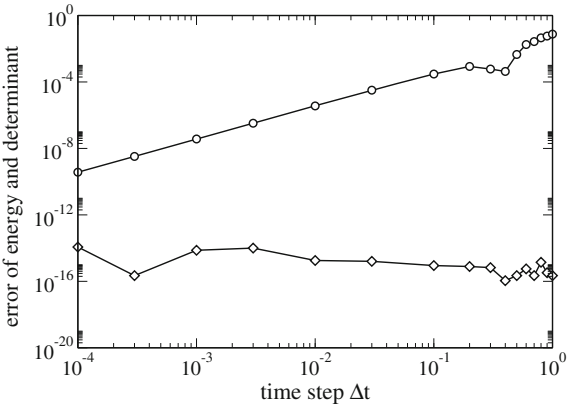


Fig. 14.4 (Global error of the implicit method) The equations of a free rotor (14.8) are solved using the implicit method. The deviations $|\det(A) - 1|$ (*diamonds*) and $|E_{kin} - E_{kin}(0)|$ (*circles*) at $t = 10$ are shown as a function of the time step Δt. Initial conditions as in Fig. 14.3. See also Problem 14.1

$$A(t+\Delta t) = A(t) + A\left(t+\frac{\Delta t}{2}\right) W\left(t+\frac{\Delta t}{2}\right)\Delta t + \cdots \tag{14.87}$$

$$\mathbf{L}_{int}(t+\Delta t) = \mathbf{L}_{int}(t) + \mathbf{N}_{int}\left(t+\frac{\Delta t}{2}\right)\Delta t + \cdots . \tag{14.88}$$

Taylor series expansion gives

$$\begin{aligned} &A\left(t+\frac{\Delta t}{2}\right) W\left(t+\frac{\Delta t}{2}\right)\Delta t \\ &= A(t)W(t)\Delta t + \dot{A}(t)W(t)\frac{\Delta t^2}{2} + A(t)\dot{W}(t)\frac{\Delta t^2}{2} + O(\Delta t^3) \qquad (14.89)\\ &= A(t)W(t)\Delta t + (A(t)W^2(t) + A(t)\dot{W}(t))\frac{\Delta t^2}{2} + O(\Delta t^3) \qquad (14.90)\end{aligned}$$

which has the same error order as the explicit second order method. The matrix $A(t+\frac{\Delta t}{2})$ at mid-time can be approximated by

$$\begin{aligned} &\frac{1}{2}(A(t) + A(t+\Delta t)) \\ &= A\left(t+\frac{\Delta t}{2}\right) + \frac{\Delta t^2}{4}\ddot{A}\left(t+\frac{\Delta t}{2}\right) + \cdots = A\left(t+\frac{\Delta t}{2}\right) + O(\Delta t^2) \end{aligned} \tag{14.91}$$

which does not change the error order of the implicit integrator which now becomes

$$A(t+\Delta t) = A(t) + \frac{1}{2}\left(A(t) + A(t+\Delta t)\right) W\left(t+\frac{\Delta t}{2}\right)\Delta t + O(\Delta t^3). \tag{14.92}$$

This equation can be formally solved by

$$A(t+\Delta t) = A(t)\left(1+\frac{\Delta t}{2}W\left(t+\frac{\Delta t}{2}\right)\right)\left(1-\frac{\Delta t}{2}W\left(t+\frac{\Delta t}{2}\right)\right)^{-1} = A(t)T_b\left(\frac{\Delta t}{2}\right). \tag{14.93}$$

Alternatively, using angular velocities in the laboratory system we have the similar expression

$$A(t+\Delta t) = \left[1-\frac{\Delta t}{2}W\left(t+\frac{\Delta t}{2}\right)\right]^{-1}\left[1+\frac{\Delta t}{2}W\left(t+\frac{\Delta t}{2}\right)\right]A(t) = T\left(\frac{\Delta t}{2}\right)A(t). \tag{14.94}$$

The angular velocities at midtime can be calculated with sufficient accuracy from

$$W\left(t+\frac{\Delta t}{2}\right) = W(t) + \frac{\Delta t}{2}\dot{W}(t) + O(\Delta t^2). \tag{14.95}$$

With the help of an algebra program we easily prove that

$$\det\left(1+\frac{\Delta t}{2}W\right)=\det\left(1-\frac{\Delta t}{2}W\right)=1+\frac{\omega^2\Delta t^2}{4} \tag{14.96}$$

and therefore the determinant of the rotation matrix is conserved. The necessary matrix inversion can be easily done:

$$\begin{aligned}&\left[1-\frac{\Delta t}{2}W\right]^{-1}\\&=\begin{pmatrix}1+\frac{\omega_1^2\Delta t^2}{4} & -\omega_3\frac{\Delta t}{2}+\omega_1\omega_2\frac{\Delta t^2}{4} & \omega_2\frac{\Delta t}{2}+\omega_1\omega_3\frac{\Delta t^2}{4}\\ \omega_3\frac{\Delta t}{2}+\omega_1\omega_2\frac{\Delta t^2}{4} & 1+\frac{\omega_2^2\Delta t^2}{4} & -\omega_1\frac{\Delta t}{2}+\omega_2\omega_3\frac{\Delta t^2}{4}\\ -\omega_2\frac{\Delta t}{2}+\omega_1\omega_3\frac{\Delta t^2}{4} & \omega_1\frac{\Delta t}{2}+\omega_2\omega_3\frac{\Delta t^2}{4} & 1+\frac{\omega_3^2\Delta t^2}{4}\end{pmatrix}\frac{1}{1+\omega^2\frac{\Delta t^2}{4}}.\end{aligned} \tag{14.97}$$

The matrix product is explicitly

$$\begin{aligned}T_b&=\left[1+\frac{\Delta t}{2}W_b\right]\left[1-\frac{\Delta t}{2}W_b\right]^{-1}\\&=\begin{pmatrix}1+\frac{\omega_{b1}^2-\omega_{b2}^2-\omega_{b3}^2}{4}\Delta t^2 & -\omega_{b3}\Delta t+\omega_{b1}\omega_{b2}\frac{\Delta t^2}{2} & \omega_{b2}\Delta t+\omega_{b1}\omega_{b3}\frac{\Delta t^2}{2}\\ \omega_{b3}\Delta t+\omega_{b1}\omega_{b2}\frac{\Delta t^2}{2} & 1+\frac{-\omega_{b1}^2+\omega_{b2}^2-\omega_{b3}^2}{4}\Delta t^2 & -\omega_{b1}\Delta t+\omega_{b2}\omega_{b3}\frac{\Delta t^2}{2}\\ -\omega_{b2}\Delta t+\omega_{b1}\omega_{b3}\frac{\Delta t^2}{2} & \omega_{b1}\Delta t+\omega_{b2}\omega_{b3}\frac{\Delta t^2}{2} & 1+\frac{-\omega_{b1}^2-\omega_{b2}^2+\omega_{b3}^2}{4}\Delta t^2\end{pmatrix}\\&\times\frac{1}{1+\omega_b^2\frac{\Delta t^2}{4}}.\end{aligned} \tag{14.98}$$

With the help of an algebra program it can be proved that this matrix is even orthogonal

$$T_b^T T_b=1 \tag{14.99}$$

and hence the orthonormality of A is conserved. The approximation for the angular momentum

$$\begin{aligned}&\mathbf{L}_{int}(t)+\mathbf{N}_{int}\left(t+\frac{\Delta t}{2}\right)\Delta t\\&=\mathbf{L}_{int}(t)+\mathbf{N}_{int}(t)\Delta t+\dot{\mathbf{N}}_{int}(t)\frac{\Delta t^2}{2}+\cdots=\mathbf{L}_{int}(t+\Delta t)+O(\Delta t^3)\end{aligned} \tag{14.100}$$

can be used in an implicit way

$$\mathbf{L}_{int}(t+\Delta t) = \mathbf{L}_{int}(t) + \frac{\mathbf{N}_{int}(t+\Delta t) + \mathbf{N}_{int}(t)}{2}\Delta t + O(\Delta t^3). \tag{14.101}$$

Alternatively Euler's equations can be used in the form [171, 172]

$$\omega_{b1}\left(t+\frac{\Delta t}{2}\right) = \omega_{b1}\left(t-\frac{\Delta t}{2}\right) + \frac{I_{b2}-I_{b3}}{I_{b1}}\omega_{b2}(t)\omega_{b3}(t)\Delta t + \frac{N_{b1}}{I_{b1}}\Delta t \text{ etc.} \tag{14.102}$$

where the product $\omega_{b2}(t)\omega_{b3}(t)$ is approximated by

$$\omega_{b2}(t)\omega_{b3}(t) = \frac{1}{2}\left[\omega_{b2}\left(t-\frac{\Delta t}{2}\right)\omega_{b3}\left(t-\frac{\Delta t}{2}\right) + \omega_{b2}\left(t+\frac{\Delta t}{2}\right)\omega_{b3}\left(t+\frac{\Delta t}{2}\right)\right]. \tag{14.103}$$

$\omega_{b1}(t+\frac{\Delta t}{2})$ is determined by iterative solution of the last two equations. Starting with $\omega_{b1}(t-\frac{\Delta t}{2})$ convergence is achieved after few iterations.

14.12 Example: Free Symmetric Rotor

For the special case of a free symmetric rotor ($I_{b2} = I_{b3}$, $\mathbf{N}_{int} = 0$) Euler's equations simplify to:

$$\dot{\omega}_{b1} = 0 \tag{14.104}$$

$$\dot{\omega}_{b2} = \frac{I_{b2(3)} - I_{b1}}{I_{b2(3)}}\omega_{b1}\omega_{b3} = \lambda\omega_{b3} \tag{14.105}$$

$$\dot{\omega}_{b3} = \frac{I_{b1} - I_{b2(3)}}{I_{b2(3)}}\omega_{b1}\omega_{b2} = -\lambda\omega_{b2} \tag{14.106}$$

$$\lambda = \frac{I_{b2(3)} - I_{b1}}{I_{b2(3)}}\omega_{b1}. \tag{14.107}$$

Coupled equations of this type appear often in physics. The solution can be found using a complex quantity

$$\Omega = \omega_{b2} + i\omega_{b3} \tag{14.108}$$

which obeys the simple differential equation

$$\dot{\Omega} = \dot{\omega}_{b2} + i\dot{\omega}_{b3} = -i(i\lambda\omega_{b3} + \lambda\omega_{b2}) = -i\lambda\Omega \tag{14.109}$$

with the solution

$$\Omega = \Omega_0 e^{-i\lambda t}. \tag{14.110}$$

Finally

$$\boldsymbol{\omega}_b = \begin{pmatrix} \omega_{b1}(0) \\ \Re(\Omega_0 e^{-i\lambda t}) \\ \Im(\Omega_0 e^{-i\lambda t}) \end{pmatrix} = \begin{pmatrix} \omega_{b1}(0) \\ \omega_{b2}(0)\cos(\lambda t) + \omega_{b3}(0)\sin(\lambda t) \\ \omega_{b3}(0)\cos(\lambda t) - \omega_{b2}(0)\sin(\lambda t) \end{pmatrix} \tag{14.111}$$

i.e. $\boldsymbol{\omega}_b$ rotates around the 1-axis with frequency λ.

14.13 Kinetic Energy of a Rotor

The kinetic energy of the rotor is

$$\begin{aligned} E_{kin} &= \sum_i \frac{m_i}{2}\dot{r}_i^2 = \sum_i \frac{m_i}{2}(\dot{\mathbf{R}} + \dot{A}\boldsymbol{\rho}_{ib})^2 \\ &= \sum_i \frac{m_i}{2}(\dot{R}^T + \boldsymbol{\rho}_{ib}^T\dot{A}^T)(\dot{R} + \dot{A}\boldsymbol{\rho}_{ib}) = \frac{M}{2}\dot{R}^2 + \sum_i \frac{m_i}{2}\boldsymbol{\rho}_{ib}^T\dot{A}^T\dot{A}\boldsymbol{\rho}_{ib}. \end{aligned} \tag{14.112}$$

The second part is the contribution of the rotational motion. It can be written as

$$E_{rot} = \sum_i \frac{m_i}{2}\boldsymbol{\rho}_{ib}^T W_b^T A^T A W_b \boldsymbol{\rho}_{ib} = -\sum_i \frac{m_i}{2}\boldsymbol{\rho}_{ib}^T W_b^2 \boldsymbol{\rho}_{ib} = \frac{1}{2}\omega_b^T I_b \boldsymbol{\omega}_b \tag{14.113}$$

since

$$-W_b^2 = \begin{pmatrix} \omega_{b3}^2 + \omega_{b2}^2 & -\omega_{b1}\omega_{b2} & -\omega_{b1}\omega_{b3} \\ -\omega_{b1}\omega_{b2} & \omega_{b1}^2 + \omega_{b3}^2 & -\omega_{b2}\omega_{b3} \\ -\omega_{b1}\omega_{b3} & -\omega_{b2}\omega_{b3} & \omega_{b1}^2 + \omega_{b2}^2 \end{pmatrix} = \omega_b^2 - \boldsymbol{\omega}_b\boldsymbol{\omega}_b^T. \tag{14.114}$$

14.14 Parametrization by Euler Angles

So far we had to solve equations for all 9 components of the rotation matrix. But there are six constraints since the column vectors of the matrix have to be orthonormalized. Therefore the matrix can be parametrized with less than 9 variables. In fact it is sufficient to use only three variables. This can be achieved by splitting the full rotation

into three rotations around different axis. Most common are Euler angles defined by the orthogonal matrix [173]

$$\begin{pmatrix} \cos\psi\cos\phi - \cos\theta\sin\phi\sin\psi & -\sin\psi\cos\phi - \cos\theta\sin\phi\cos\psi & \sin\theta\sin\phi \\ \cos\psi\sin\phi + \cos\theta\cos\phi\sin\psi & -\sin\psi\sin\phi + \cos\theta\cos\phi\cos\psi & -\sin\theta\cos\phi \\ \sin\theta\sin\psi & \sin\theta\cos\psi & \cos\theta \end{pmatrix} \tag{14.115}$$

obeying the equations

$$\dot{\phi} = \omega_x \frac{\sin\phi\cos\theta}{\sin\theta} + \omega_y \frac{\cos\phi\cos\theta}{\sin\theta} + \omega_z \tag{14.116}$$

$$\dot{\theta} = \omega_x \cos\phi + \omega_y \sin\phi \tag{14.117}$$

$$\dot{\psi} = \omega_x \frac{\sin\phi}{\sin\theta} - \omega_y \frac{\cos\phi}{\sin\theta}. \tag{14.118}$$

Different versions of Euler angles can be found in the literature, together with the closely related cardanic angles. For all of them a $\sin\theta$ appears in the denominator which causes numerical instabilities at the poles. One possible solution to this problem is to switch between two different coordinate systems.

14.15 Cayley–Klein-Parameters, Quaternions, Euler Parameters

There exists another parametrization of the rotation matrix which is very suitable for numerical calculations. It is connected with the algebra of the so called quaternions. The vector space of the complex 2×2 matrices can be spanned using Pauli matrices by

$$1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{14.119}$$

Any complex 2×2 matrix can be written as a linear combination

$$c_0 1 + \mathbf{c}\boldsymbol{\sigma}. \tag{14.120}$$

Accordingly any vector $\mathbf{x} \in R^3$ can be mapped onto a complex 2×2 matrix:

$$\mathbf{x} \;\rightarrow\; P = \begin{pmatrix} z & x - iy \\ x + iy & -z \end{pmatrix}. \tag{14.121}$$

Rotation of the coordinate system leads to the transformation

$$P' = QPQ^{\dagger} \tag{14.122}$$

where

$$Q = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \tag{14.123}$$

is a complex 2×2 rotation matrix. Invariance of the length ($|\mathbf{x}| = \sqrt{-\det(P)}$) under rotation implies that Q must be unitary, i.e. $Q^\dagger = Q^{-1}$ and its determinant must be 1. Explicitly

$$Q^\dagger = \begin{pmatrix} \alpha^* & \gamma^* \\ \beta^* & \delta^* \end{pmatrix} = Q^{-1} = \frac{1}{\alpha\delta - \beta\gamma} \begin{pmatrix} \delta & -\beta \\ -\gamma & \alpha \end{pmatrix} \tag{14.124}$$

and Q has the form

$$Q = \begin{pmatrix} \alpha & \beta \\ -\beta^* & \alpha^* \end{pmatrix} \quad \text{with} \quad |\alpha|^2 + |\beta|^2 = 1. \tag{14.125}$$

Setting $x_\pm = x \pm iy$, the transformed matrix has the same form as P:

$$\begin{aligned} &QPQ^\dagger \\ &= \begin{pmatrix} \alpha^*\beta x_+ + \beta^*\alpha x_- + (|\alpha|^2 - |\beta|^2)z & -\beta^2 x_+ + \alpha^2 x_- - 2\alpha\beta z \\ \alpha^{*2} x_+ - \beta^{*2} x_- - 2\alpha^*\beta^* z & -\alpha^*\beta x_+ - \alpha\beta^* x_- - (|\alpha|^2 - |\beta|^2)z \end{pmatrix} \\ &= \begin{pmatrix} z' & x'_- \\ x'_+ & -z' \end{pmatrix}. \end{aligned} \tag{14.126}$$

From comparison we find the transformed vector components:

$$\begin{aligned} x' &= \frac{1}{2}(x'_+ + x'_-) = \frac{1}{2}(\alpha^{*2} - \beta^2)x_+ + \frac{1}{2}(\alpha^2 - \beta^{*2})x_- - (\alpha\beta + \alpha^*\beta^*)z \\ &= \frac{\alpha^{*2} + \alpha^2 - \beta^{*2} - \beta^2}{2} x + \frac{i(\alpha^{*2} - \alpha^2 + \beta^{*2} - \beta^2)}{2} y - (\alpha\beta + \alpha^*\beta^*)z \end{aligned} \tag{14.127}$$

$$\begin{aligned} y' &= \frac{1}{2i}(x'_+ - x'_-) = \frac{1}{2i}(\alpha^{*2} + \beta^2)x_+ + \frac{1}{2i}(-\beta^{*2} - \alpha^2)x_- + \frac{1}{i}(-\alpha^*\beta^* + \alpha\beta)z \\ &= \frac{\alpha^{*2} - \alpha^2 - \beta^{*2} + \beta^2}{2i} x + \frac{\alpha^{*2} + \alpha^2 + \beta^{*2} + \beta^2}{2} y + i(\alpha^*\beta^* - \alpha\beta)z \end{aligned} \tag{14.128}$$

$$z' = (\alpha^*\beta + \alpha\beta^*)x + i(\alpha^*\beta - \alpha\beta^*)y + (|\alpha|^2 - |\beta|^2)z. \tag{14.129}$$

This gives us the rotation matrix in terms of the Cayley–Klein parameters α and β:

$$A = \begin{pmatrix} \frac{\alpha^{*2} + \alpha^2 - \beta^{*2} - \beta^2}{2} & \frac{i(\alpha^{*2} - \alpha^2 + \beta^{*2} - \beta^2)}{2} & -(\alpha\beta + \alpha^*\beta^*) \\ \frac{\alpha^{*2} - \alpha^2 - \beta^{*2} + \beta^2}{2i} & \frac{\alpha^{*2} + \alpha^2 + \beta^{*2} + \beta^2}{2} & \frac{1}{i}(-\alpha^*\beta^* + \alpha\beta) \\ (\alpha^*\beta + \alpha\beta^*) & i(\alpha^*\beta - \alpha\beta^*) & (|\alpha|^2 - |\beta|^2) \end{pmatrix}. \tag{14.130}$$

For practical calculations one often prefers to have four real parameters instead of two complex ones. The so called Euler parameters q_0, q_1, q_2, q_3 are defined by

$$\alpha = q_0 + iq_3 \qquad \beta = q_2 + iq_1. \tag{14.131}$$

Now the matrix Q

$$Q = \begin{pmatrix} q_0 + iq_3 & q_2 + iq_1 \\ -q_2 + iq_1 & q_0 - iq_3 \end{pmatrix} = q_0 1 + iq_1\sigma_x + iq_2\sigma_y + iq_3\sigma_z \tag{14.132}$$

becomes a so-called quaternion which is a linear combination of the four matrices

$$U = 1 \qquad I = i\sigma_z \qquad J = i\sigma_y \qquad K = i\sigma_x \tag{14.133}$$

which obey the following multiplication rules:

$$\begin{aligned} I^2 = J^2 = K^2 &= -U \\ IJ = -JI &= K \\ JK = -KJ &= I \\ KI = -IK &= J. \end{aligned} \tag{14.134}$$

In terms of Euler parameters the rotation matrix reads

$$A = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \tag{14.135}$$

and from the equation $\dot{A} = WA$ we derive the equation of motion for the quaternion

$$\begin{pmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & \omega_1 & \omega_2 & \omega_3 \\ -\omega_1 & 0 & -\omega_3 & \omega_2 \\ -\omega_2 & \omega_3 & 0 & -\omega_1 \\ -\omega_3 & -\omega_2 & \omega_1 & 0 \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} \tag{14.136}$$

or from $\dot{A} = AW_b$ the alternative equation

$$\begin{pmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & \omega_{1b} & \omega_{2b} & \omega_{3b} \\ -\omega_{1b} & 0 & \omega_{3b} & -\omega_{2b} \\ -\omega_{2b} & -\omega_{3b} & 0 & \omega_{1b} \\ -\omega_{3b} & \omega_{2b} & -\omega_{1b} & 0 \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}. \tag{14.137}$$

Both of these equations can be written briefly in the form

$$\dot{\mathbf{q}} = \widetilde{W}\mathbf{q}. \tag{14.138}$$

Example: Rotation Around the z-axis

Rotation around the z-axis corresponds to the quaternion with Euler parameters

$$\mathbf{q} = \begin{pmatrix} \cos\frac{\omega t}{2} \\ 0 \\ 0 \\ -\sin\frac{\omega t}{2} \end{pmatrix} \tag{14.139}$$

as can be seen from the rotation matrix

$$\begin{aligned} A &= \begin{pmatrix} \left(\cos\frac{\omega t}{2}\right)^2 - \left(\sin\frac{\omega t}{2}\right)^2 & -2\cos\frac{\omega t}{2}\sin\frac{\omega t}{2} & 0 \\ 2\cos\frac{\omega t}{2}\sin\frac{\omega t}{2} & \left(\cos\frac{\omega t}{2}\right)^2 - \left(\sin\frac{\omega t}{2}\right)^2 & 0 \\ 0 & 0 & \left(\cos\frac{\omega t}{2}\right)^2 + \left(\sin\frac{\omega t}{2}\right)^2 \end{pmatrix} \\ &= \begin{pmatrix} \cos\omega t & -\sin\omega t & 0 \\ \sin\omega t & \cos\omega t & 0 \\ 0 & 0 & 1 \end{pmatrix}. \end{aligned} \tag{14.140}$$

The time derivative of $\mathbf{q}$ obeys the equation

$$\dot{\mathbf{q}} = \frac{1}{2}\begin{pmatrix} 0 & 0 & 0 & \omega \\ 0 & 0 & -\omega & 0 \\ 0 & \omega & 0 & 0 \\ -\omega & 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} \cos\frac{\omega t}{2} \\ 0 \\ 0 \\ -\sin\frac{\omega t}{2} \end{pmatrix} = \begin{pmatrix} -\frac{\omega}{2}\sin\omega t \\ 0 \\ 0 \\ -\frac{\omega}{2}\cos\omega t \end{pmatrix}. \tag{14.141}$$

After a rotation by 2π the quaternion changes its sign, i.e. $\mathbf{q}$ and $-\mathbf{q}$ parametrize the same rotation matrix!

14.16 Solving the Equations of Motion with Quaternions

As with the matrix method we can obtain a simple first or second order algorithm from the Taylor series expansion

$$\mathbf{q}(t+\Delta t) = \mathbf{q}(t) + \widetilde{W}(t)\mathbf{q}(t)\Delta t + (\dot{\widetilde{W}}(t) + \widetilde{W}^2(t))\mathbf{q}(t)\frac{\Delta t^2}{2} + \cdots. \tag{14.142}$$

Now only one constraint remains, which is the conservation of the norm of the quaternion. This can be taken into account by rescaling the quaternion whenever its norm deviates too much from unity.

It is also possible to use Omelyan's [174] method:

$$\mathbf{q}(t+\Delta t) = \mathbf{q}(t) + \widetilde{W}\left(t + \frac{\Delta t}{2}\right)\frac{1}{2}(\mathbf{q}(t) + \mathbf{q}(t+\Delta t)) \tag{14.143}$$

gives

$$\mathbf{q}(t+\Delta t) = \left(1 - \frac{\Delta t}{2}\widetilde{W}\right)^{-1} \left(1 + \frac{\Delta t}{2}\widetilde{W}\right) \mathbf{q}(t) \tag{14.144}$$

where the inverse matrix is

$$\left(1 - \frac{\Delta t}{2}\widetilde{W}\right)^{-1} = \frac{1}{1 + \omega^2 \frac{\Delta t^2}{16}} \left(1 + \frac{\Delta t}{2}\widetilde{W}\right) \tag{14.145}$$

and the matrix product

$$\left(1 - \frac{\Delta t}{2}\widetilde{W}\right)^{-1} \left(1 + \frac{\Delta t}{2}\widetilde{W}\right) = \frac{1 - \omega^2 \frac{\Delta t^2}{16}}{1 + \omega^2 \frac{\Delta t^2}{16}} + \frac{\Delta t}{1 + \omega^2 \frac{\Delta t^2}{16}} \widetilde{W}. \tag{14.146}$$

This method conserves the norm of the quaternion and works quite well.

Problems

Problem 14.1 Free Rotor

In this computer experiment we compare different methods for a free rotor (Sect. 14.8, Fig. 14.5):

- explicit first order method (14.67)

$$A(t+\Delta t) = A(t) + A(t)W_b(t)\Delta t + O(\Delta t^2) \tag{14.147}$$

- explicit second order method (14.69)

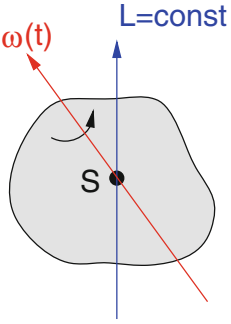


Fig. 14.5 Free asymmetric rotor

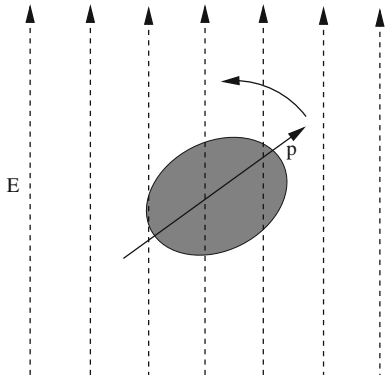


Fig. 14.6 Rotor in an electric field

$$A(t+\Delta t) = A(t) + A(t)W_b(t)\Delta t + \frac{1}{2}\left(A(t)W_b^2(t) + A(t)\dot{W}_b(t)\right)\Delta t^2 + O(\Delta t^3) \tag{14.148}$$

- implicit second order method (14.93)

$$A(t+\Delta t) = A(t)\left(1+\frac{\Delta t}{2}W\left(t+\frac{\Delta t}{2}\right)\right)\left(1-\frac{\Delta t}{2}W\left(t+\frac{\Delta t}{2}\right)\right)^{-1} + O(\Delta t^3). \tag{14.149}$$

The explicit methods can be combined with reorthogonalization according to (14.79) or with the Gram-Schmidt method. Reorthogonalization threshold and time step can be varied and the error of kinetic energy and determinant are plotted as a function of the total simulation time.

Problem 14.2 Rotor in a Field

In this computer experiment we simulate a molecule with a permanent dipole moment in a homogeneous electric field **E** (Fig. 14.6). We neglect vibrations and describe the molecule as a rigid body consisting of nuclei with masses m_i and partial charges Q_i. The total charge is $\sum_i Q_i = 0$. The dipole moment is

$$\mathbf{p} = \sum_i Q_i \mathbf{r}_i \tag{14.150}$$

and external force and torque are

$$\mathbf{F}_{ext} = \sum_i Q_i \mathbf{E} = 0 \tag{14.151}$$

$$\mathbf{N}_{ext} = \sum_i Q_i \mathbf{r}_i \times \mathbf{E} = \mathbf{p} \times \mathbf{E}. \tag{14.152}$$

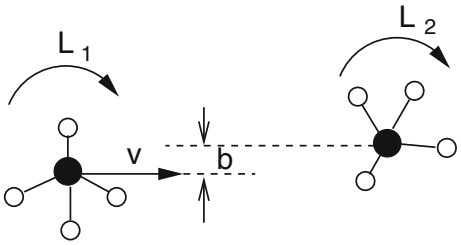


Fig. 14.7 Molecular collision

The angular momentum changes according to

$$\frac{d}{\Delta t}\mathbf{L}_{int} = \mathbf{p} \times \mathbf{E} \tag{14.153}$$

where the dipole moment is constant in the body fixed system. We use the implicit integrator for the rotation matrix (14.93) and the equation

$$\dot{\boldsymbol{\omega}}_b(t) = -I_b^{-1} W_b(t) \mathbf{L}_{int,b}(t) + I_b^{-1} A^{-1}(t)(\mathbf{p}(t) \times \mathbf{E}) \tag{14.154}$$

to solve the equations of motion numerically.

Obviously the component of the angular momentum parallel to the field is constant. The potential energy is

$$U = -\sum_i Q_i \mathbf{E}\mathbf{r}_i = -\mathbf{p}\mathbf{E}. \tag{14.155}$$

Problem 14.3 Molecular Collision

This computer experiment simulates the collision of two rigid methane molecules (Fig. 14.7). The equations of motion are solved with the implicit quaternion method (14.143) and the velocity Verlet method (13.11.4). The two molecules interact by a standard 6–12 Lennard-Jones potential (15.24) [163]. For comparison the attractive r^{-6} part can be switched off. The initial angular momenta as well as the initial velocity v and collision parameter b can be varied. Total energy and momentum are monitored and the decomposition of the total energy into translational, rotational and potential energy are plotted as a function of time.

Study the exchange of momentum and angular momentum and the transfer of energy between translational and rotational degrees of freedom.

Chapter 15
Molecular Mechanics

Classical molecular mechanics simulations have become a very valuable tool for the investigation of atomic and molecular systems [175–179], *mainly in the area of materials science and molecular biophysics. Based on the Born–Oppenheimer separation which assumes that the electrons move much faster than the nuclei, nuclear motion is described quantum mechanically by the Hamiltonian*

$$H = \left[T^{Nuc} + U\left(\mathbf{r}_j^{Nuc}\right)\right]. \tag{15.1}$$

Molecular mechanics uses the corresponding classical energy function

$$T^{Nuc} + U\left(\mathbf{r}_j^{Nuc}\right) = \sum_j \frac{(p_j^{Nuc})^2}{2m_j} + U\left(\mathbf{r}_j^{Nuc}\right) \tag{15.2}$$

which treats the atoms as mass points interacting by classical forces

$$\mathbf{F}_i = -\mathrm{grad}_{\mathbf{r}_i} U\left(\mathbf{r}_j^{Nuc}\right). \tag{15.3}$$

Stable structures, i.e. local minima of the potential energy can be found by the methods discussed in Chap. 6. Small amplitude motions around an equilibrium geometry are described by a harmonic normal mode analysis. Molecular dynamics (MD) simulations solve the classical equations of motion

$$m_i \frac{\mathrm{d}^2\mathbf{r}_i}{\mathrm{d}t^2} = \mathbf{F}_i = -\mathrm{grad}_{\mathbf{r}_i} U \tag{15.4}$$

numerically.

The potential energy function $U\left(\mathbf{r}_j^{Nuc}\right)$ *can be calculated with simplified quantum methods for not too large systems* [180, 181]. *Classical MD simulations for larger molecules use empirical force fields, which approximate the potential energy surface of the electronic ground state. They are able to describe structural and conformational changes but not chemical reactions which usually involve more than one*

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_15

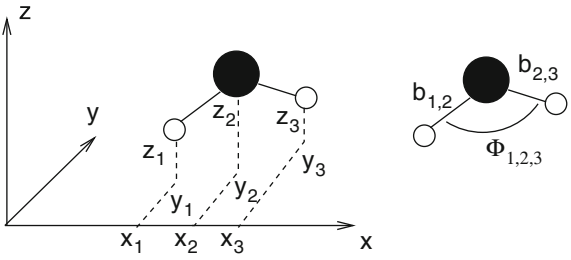


Fig. 15.1 (Molecular coordinates) Cartesian coordinates (***Left***) are used to solve the equations of motion whereas the potential energy is more conveniently formulated in internal coordinates (***Right***)

electronic state. Among the most popular classical force fields are AMBER [182], CHARMM [183] *and GROMOS* [184, 185].

In this chapter we discuss the most important interaction terms, which are conveniently expressed in internal coordinates, i.e. bond lengths, bond angles and dihedral angles. We derive expressions for the gradients of the force field with respect to Cartesian coordinates. In a computer experiment we simulate a glycine dipeptide and demonstrate the principles of energy minimization, normal mode analysis and dynamics simulation.

15.1 Atomic Coordinates

The most natural coordinates for the simulation of molecules are the Cartesian coordinates (Fig. 15.1) of the atoms,

$$\mathbf{r}_i = (x_i, y_i, z_i) \tag{15.5}$$

which can be collected into a $3N$-dimensional vector

$$(\xi_1, \xi_2 \cdots \xi_{3N}) = (x_1, y_1, z_1, x_2 \cdots x_N, y_N, z_N). \tag{15.6}$$

The second derivatives of the Cartesian coordinates appear directly in the equations of motion (15.4)

$$m_r \ddot{\xi}_r = F_r \quad r = 1 \cdots 3N. \tag{15.7}$$

Cartesian coordinates have no direct relation to the structural properties of molecules. For instance a protein is a long chain of atoms (the so called backbone) with additional side groups (Fig. 15.2).

The protein structure can be described more intuitively with the help of atomic distances and angles. Internal coordinates are (Fig. 15.3) distances between two bonded atoms (bond lengths)

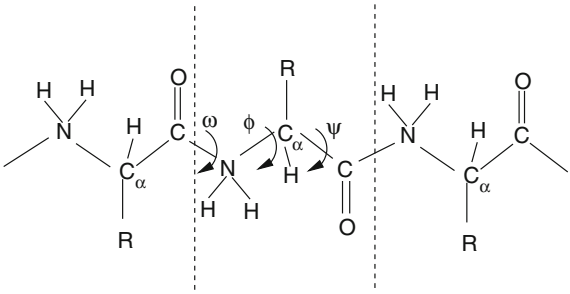


Fig. 15.2 (Conformation of a protein) The relative orientation of two successive protein residues can be described by three angles (Ψ, Φ, ω)

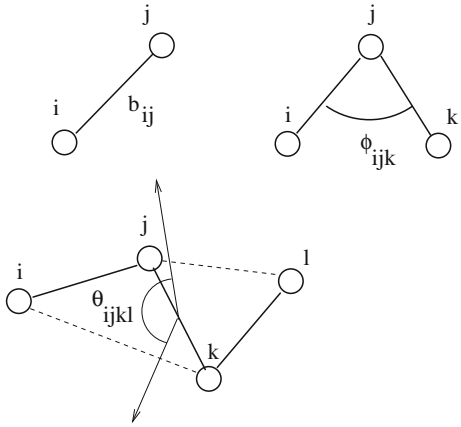


Fig. 15.3 (Internal coordinates) The structure of a molecule can be described by bond lengths, bond angles and dihedral angles

$$b_{ij} = |\mathbf{r}_{ij}| = |\mathbf{r}_i - \mathbf{r}_j|, \tag{15.8}$$

angles between two bonds (bond angles)

$$\phi_{ijk} = \arccos\left(\frac{\mathbf{r}_{ij}\mathbf{r}_{kj}}{|\mathbf{r}_{ij}|\,|\mathbf{r}_{kj}|}\right) \tag{15.9}$$

and dihedral angles which describe the planarity and torsions of the molecule. A dihedral angle (Fig. 15.4) is the angle between two planes which are defined by three bonds

$$\theta_{ijkl} = \text{sign}(\theta_{ijkl})\ \arccos(\mathbf{n}_{ijk}\mathbf{n}_{jkl}) \tag{15.10}$$

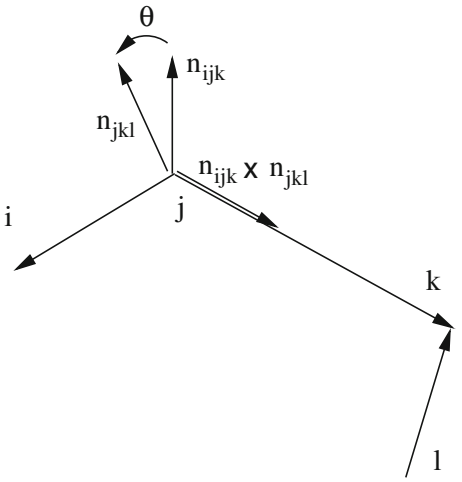


Fig. 15.4 Dihedral angle

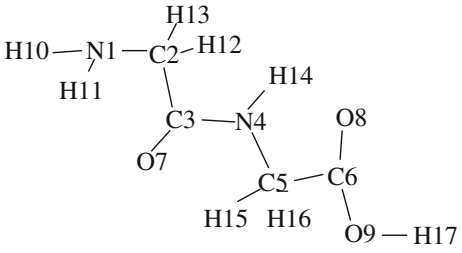


Fig. 15.5 (Glycine dipeptide model) The glycine dipeptide is the simplest model for a peptide. It is simulated in Problem 15.1. Optimized internal coordinates are shown in Table 15.1

$$\mathbf{n}_{ijk} = \frac{\mathbf{r}_{ij} \times \mathbf{r}_{kj}}{\left|\mathbf{r}_{ij} \times \mathbf{r}_{kj}\right|} \quad \mathbf{n}_{jkl} = \frac{\mathbf{r}_{kj} \times \mathbf{r}_{kl}}{\left|\mathbf{r}_{kj} \times \mathbf{r}_{kl}\right|} \tag{15.11}$$

where the conventional sign of the dihedral angle [186] is determined by

$$\text{sign}\theta_{ijkl} = \text{sign}\left(\mathbf{r}_{kj}(\mathbf{n}_{ijk} \times \mathbf{n}_{jkl})\right). \tag{15.12}$$

Internal coordinates are very convenient for the formulation of a force field. On the other hand, the kinetic energy (15.2) becomes complicated if expressed in internal coordinates. Therefore both kinds of coordinates are used in molecular dynamics calculations. The internal coordinates are usually arranged in Z-matrix form. Each line corresponds to one atom i and shows its position relative to three atoms j, k, l in terms of the bond length b_{ij}, the bond angle ϕ_{ijk} and the dihedral angle θ_{ijkl} (Fig. 15.5 and Table 15.1).

Table 15.1 (Z-matrix) The optimized values of the internal coordinates from Problem 15.1 are shown in Z-matrix form. Except for the first three atoms the position of atom i is given by its distance b_{ij} to atom j, the bond angle ϕ_{ijk} and the dihedral angle θ_{ijkl}

Number i	Label	j	k	l	Bond length b_{ij}(Å)	Bond angle ϕ_{ijk}	Dihedral θ_{ijkl}
1	N1						
2	C2	1			1.45		
3	C3	2	1		1.53	108.6	
4	N4	3	2	1	1.35	115.0	160.7
5	C5	4	3	2	1.44	122.3	−152.3
6	C6	5	4	3	1.51	108.7	−153.1
7	O7	3	2	1	1.23	121.4	−26.3
8	O8	6	5	4	1.21	124.4	123.7
9	O9	6	5	4	1.34	111.5	−56.5
10	H10	1	2	3	1.02	108.7	−67.6
11	H11	1	2	3	1.02	108.7	49.3
12	H12	2	3	4	1.10	109.4	−76.8
13	H13	2	3	4	1.10	109.4	38.3
14	H14	4	3	2	1.02	123.1	27.5
15	H15	5	4	3	1.10	111.2	−32.5
16	H16	5	4	3	1.10	111.1	86.3
17	H17	9	6	5	0.97	106.9	−147.4

15.2 Force Fields

Classical force fields are usually constructed as an additive combination of many interaction terms. Generally these can be divided into intramolecular contributions U_{bonded} which determine the configuration and motion of a single molecule and intermolecular contributions $U_{non-bonded}$ describing interactions between different atoms or molecules

$$U = U_{bonded} + U_{non-bonded}. \tag{15.13}$$

15.2.1 Intramolecular Forces

The most important intramolecular forces depend on the deviation of bond lengths, bond angles and dihedral angles from their equilibrium values. For simplicity a sum of independent terms is used as for the CHARMM force field [183, 187, 188]

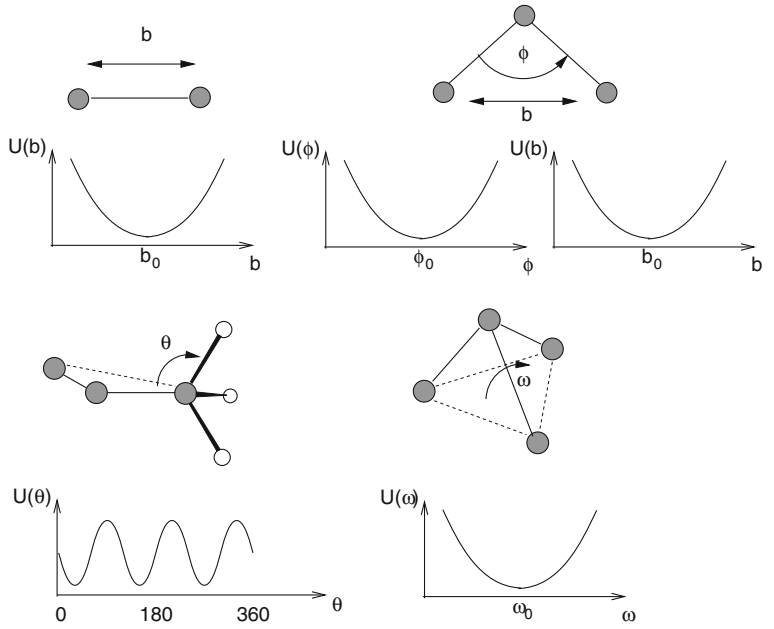


Fig. 15.6 Intramolecular forces

$$U_{intra} = \sum U_{ij}^{bond} + \sum U_{ijk}^{angle} + \sum U_{ijk}^{UB} + \sum U_{ijkl}^{dihedral} + \sum U_{ijkl}^{improper}. \tag{15.14}$$

The forces are derived from potential functions which are in the simplest case approximated by harmonic oscillator parabolas (Fig. 15.6), like the bond stretching energy

$$U_{ij}^{bond} = \frac{1}{2} k_{ij} (b_{ij} - b_{ij}^0)^2 \tag{15.15}$$

angle bending terms

$$U_{ijk}^{angle} = \frac{1}{2} k_{ijk} (\phi_{ijk} - \phi_{ijk}^0)^2 \tag{15.16}$$

together with the Urey-Bradly correction

$$U_{ijk}^{UB} = \frac{1}{2} k_{ijk} (b_{ik} - b_{ik}^0)^2 \tag{15.17}$$

and "improper dihedral" terms which are used to keep planarity

$$U_{ijkl}^{improper} = \frac{1}{2} k_{ijkl} (\theta_{ijkl} - \theta_{ijkl}^0)^2. \tag{15.18}$$

Torsional energy contributions are often described by a cosine function[1]

$$U_{ijkl}^{dihedral} = k_{ijkl} \left(1 - \cos(m\theta_{ijkl} - \theta_{ijkl}^0)\right) \tag{15.19}$$

where $m = 1, 2, 3, 4, 6$ describes the symmetry. For instance $m = 3$ for the three equivalent hydrogen atoms of a methyl group. In most cases the phase shift $\theta_{ijkl}^0 = 0$ or $\theta_{ijkl}^0 = \pi$. Then the dihedral potential can be expanded as a polynomial of $\cos\theta$, for instance

$$\text{m=1:} \quad U_{ijkl}^{dihedral} = k(1 \pm \cos\theta_{ijkl}) \tag{15.20}$$

$$\text{m=2:} \quad U_{ijkl}^{dihedral} = k \pm k(1 - 2(\cos\theta_{ijkl})^2) \tag{15.21}$$

$$\text{m=3:} \quad U_{ijkl}^{dihedral} = k(1 \pm 3\cos\theta_{ijkl} \mp 4(\cos\theta_{ijkl})^3). \tag{15.22}$$

For more general θ_{ijkl}^0 the torsional potential can be written as a polynomial of $\cos\theta_{ijkl}$ and $\sin\theta_{ijkl}$.

The atoms are classified by element and bonding environment. Atoms of the same atom type are considered equivalent and the parameters transferable (for an example see Tables 15.2, 15.3, 15.4).

15.2.2 Intermolecular Interactions

Interactions between non-bonded atoms

$$U_{non-bonded} = U^{Coul} + U^{vdW} \tag{15.23}$$

include the Coulomb interaction and the weak attractive van der Waals forces which are usually combined with a repulsive force at short distances to account for the Pauli principle. Very often a sum of pairwise Lennard-Jones potentials is used (Fig. 15.7) [163]

[1] Some force-fields like Desmond [189] or UFF [190] use a more general sum $k \sum_{m=0}^{M} c_m \cos(m\theta - \theta^0)$.

$$U^{vdw} = \sum_{A \neq B} \sum_{i \in A, j \in B} U_{i,j}^{vdw} = \sum_{A \neq B} \sum_{\text{ij}} 4\varepsilon_{ij} \left(\frac{\sigma_{ij}^{12}}{r_{ij}^{12}} - \frac{\sigma_{ij}^{6}}{r_{ij}^{6}} \right). \tag{15.24}$$

The charge distribution of a molecular system can be described by a set of multipoles at the position of the nuclei, the bond centers and further positions (lone pairs for example). Such distributed multipoles can be calculated quantum chemically for not too large molecules. In the simplest models only partial charges are taken into account giving the Coulomb energy as a sum of atom-atom interactions

$$U^{Coul} = \sum_{A \neq B} \sum_{i \in A, j \in B} \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}}. \tag{15.25}$$

More sophisticated force fields include higher charge multipoles and polarization effects.

15.3 Gradients

The equations of motion are usually solved in Cartesian coordinates and the gradients of the potential are needed in Cartesian coordinates. Since the potential depends only on relative position vectors $\mathbf{r}_{ij}$, the gradient with respect to a certain atom position $\mathbf{r}_k$ can be calculated from

Table 15.2 (Atom types of the glycine dipeptide) Atom types for glycine oligopeptides according to Bautista and Seminario [191]. The atoms are classified by element and bonding environment. Atoms of the same atom type are considered equivalent

Atom type	Atoms
C	C3
C_1	C2, C5
C_2	C6
N	N4
N_2	N1
O	O7
O_1	O9
O_2	O8
H	H14
H_1	H12, H13, H15, H16
H_2	H17
H_3	H10,H11

Table 15.3 (Bond stretching parameters) Equilibrium bond lengths (Å) and force constants (kcal $\text{mol}^{-1}\text{Å}^{-2}$) for the glycine dipeptide from [191]

Bond type	b^0	k	Bonds
$r_{C,N}$	1.346	1296.3	C3-N4
$r_{C1,N}$	1.438	935.5	N4-C5
$r_{C1,N2}$	1.452	887.7	N1-C2
$r_{C2,C1}$	1.510	818.9	C5-C6
$r_{C,C1}$	1.528	767.9	C2-C3
$r_{C2,O2}$	1.211	2154.5	C6-O8
$r_{C,O}$	1.229	1945.7	C3-O7
$r_{C2,O1}$	1.339	1162.1	C6-O9
$r_{N,H}$	1.016	1132.4	N4-H14
$r_{N2,H3}$	1.020	1104.5	N1-H10, N1-H11
$r_{C1,H1}$	1.098	900.0	C2-H12, C2-H13, C5-H15,C5-H16
$r_{O1,H2}$	0.974	1214.6	O9-H17

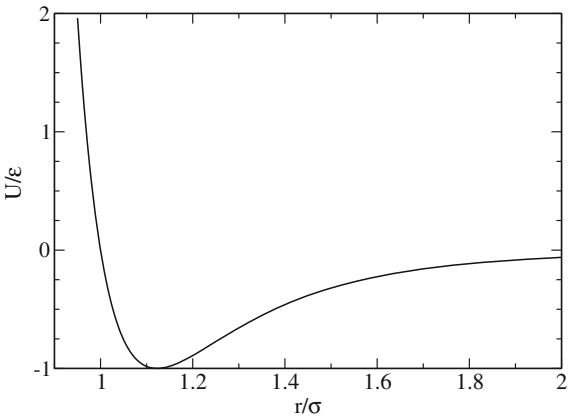


Fig. 15.7 (Lennard-Jones potential) The 6–12 potential (15.24) has its minimum at $r_{\min} = \sqrt[6]{2}\sigma \approx 1.12\sigma$ with $U_{\min} = -\epsilon$

$$\text{grad}_{\mathbf{r}_k} = \sum_{i<j} (\delta_{ik} - \delta_{jk}) \text{grad}_{\mathbf{r}_{ij}}. \tag{15.26}$$

Therefore it is sufficient to calculate gradients with respect to the difference vectors. Numerically efficient methods to calculate first and second derivatives of many force field terms are given in [192–194]. The simplest potential terms depend only on the distance of two atoms. For instance bond stretching terms, Lennard-Jones and Coulomb energies have the form

$$U_{ij} = U(r_{ij}) = U(|\mathbf{r}_{ij}|) \tag{15.27}$$

Table 15.4 (Bond angle parameters) Equilibrium bond angles (deg) and force constants (kcal $\text{mol}^{-1}\text{rad}^{-2}$) for the glycine dipeptide from [191]

Angle type	ϕ^0	k	Angles
$\phi_{N,C,C1}$	115.0	160.0	C2-C3-N4
$\phi_{C1,N,C}$	122.3	160.1	C3-N4-C5
$\phi_{C1,C2,O1}$	111.5	156.0	C5-C6-O9
$\phi_{C1,C2,O2}$	124.4	123.8	C5-C6-O8
$\phi_{C1,C,O}$	121.4	127.5	C2-C3-O7
$\phi_{O2,C2,O1}$	124.1	146.5	O8-C6-O9
$\phi_{N,C,O}$	123.2	132.7	N4-C3-O7
$\phi_{C,C1,H1}$	110.1	74.6	H12-C2-C3, H13-C2-C3
$\phi_{C2,C1,H1}$	109.4	69.6	H16-C5-C6, H15-C5-C6
$\phi_{C,N,H}$	123.1	72.0	C3-N4-H14
$\phi_{C1,N,H}$	114.6	68.3	C5-N4-H14
$\phi_{C1,N2,H3}$	108.7	71.7	H10-N1-C2, H11-N1-C2
$\phi_{H1,C1,H1}$	106.6	48.3	H13-C2-H12,H15-C5-H16
$\phi_{H3,N2,H3}$	107.7	45.2	H10-N1-H11
$\phi_{C,C1,N2}$	109.0	139.8	N1-C2-C3
$\phi_{C2,C1,N}$	108.6	129.0	N4-C5-C6
$\phi_{C2,O1,H2}$	106.9	72.0	H17-O9-C6
$\phi_{N,C1,H1}$	111.1	73.3	H15-C5-N4, H16-C5-N4
$\phi_{N2,C1,H1}$	112.6	80.1	H13-C2-N1, H12-C2-N1

where the gradient is

$$\text{grad}_{\mathbf{r}_{ij}} U_{ij} = \frac{\mathrm{d}U}{\mathrm{d}r} \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}|}. \tag{15.28}$$

The most important gradients of this kind are

$$\text{grad}_{\mathbf{r}_{ij}} U_{i,j}^{bond} = k(r_{ij} - b^0) \frac{\mathbf{r}_{ij}}{r_{ij}} = k \left(1 - \frac{b^0}{r_{ij}}\right) \mathbf{r}_{ij} \tag{15.29}$$

$$\text{grad}_{\mathbf{r}_{ij}} U_{i,j}^{vdw} = 24\varepsilon_{ij} \left(-2\frac{\sigma_{ij}^{12}}{r_{ij}^{14}} + \frac{\sigma_{ij}^{6}}{r_{ij}^{8}}\right) \mathbf{r}_{ij} \tag{15.30}$$

$$\text{grad}_{\mathbf{r}_{ij}} U_{ij}^{Coul} = -\frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}^3}\mathbf{r}_{ij}. \tag{15.31}$$

The gradient of the harmonic bond angle potential is

$$\text{grad}_{\mathbf{r}} U_{i,j,k}^{angle} = k(\phi_{ijk} - \phi^0)\text{grad}_{\mathbf{r}}\phi_{ijk} \tag{15.32}$$

where the gradient of the angle can be calculated from the gradient of its cosine

$$\begin{aligned}\text{grad}_{\mathbf{r}_{ij}}\phi_{ijk} &= -\frac{1}{\sin\phi_{ijk}}\text{grad}_{\mathbf{r}_{ij}}\cos\phi_{ijk} = -\frac{1}{\sin\phi_{ijk}}\left(\frac{\mathbf{r}_{kj}}{|\mathbf{r}_{ij}||\mathbf{r}_{kj}|} - \frac{\mathbf{r}_{ij}\mathbf{r}_{kj}}{|\mathbf{r}_{ij}|^3|\mathbf{r}_{kj}|}\mathbf{r}_{ij}\right)\\ &= -\frac{1}{\sin\phi_{ijk}}\left(\frac{\mathbf{r}_{kj}}{|\mathbf{r}_{ij}||\mathbf{r}_{kj}|} - \frac{\cos\phi_{ijk}}{|\mathbf{r}_{ij}|^2}\mathbf{r}_{ij}\right)\end{aligned} \tag{15.33}$$

$$\text{grad}_{\mathbf{r}_{kj}}\phi_{ijk} = -\frac{1}{\sin\phi_{ijk}}\left(\frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij}||\mathbf{r}_{kj}|} - \frac{\cos\phi_{ijk}}{|\mathbf{r}_{kj}|^2}\mathbf{r}_{kj}\right). \tag{15.34}$$

In principle, the sine function in the denominator could lead to numerical problems which can be avoided by treating angles close to 0 or π separately or using a function of $\cos\phi_{ijk}$ like the trigonometric potential

$$U_{ijk}^{angle} = \frac{1}{2}k_{ijk}(\cos\phi_{ijk} - \cos\phi_{ijk}^0)^2 \tag{15.35}$$

instead [190, 195, 196]. Alternatively, the gradient of ϕ can be brought to a form which is free of singularities by expressing the sine in the denominator by a cosine [197]

$$\begin{aligned}\text{grad}_{\mathbf{r}_{ij}}\phi_{ijk} &= -\frac{1}{\sqrt{r_{ij}^2 r_{kj}^2(1-\cos^2\phi_{ijk})}}\left(\mathbf{r}_{kj} - \frac{\mathbf{r}_{ij}\mathbf{r}_{kj}}{r_{ij}^2}\mathbf{r}_{ij}\right)\\ &= -\frac{r_{ij}^2\mathbf{r}_{kj} - (\mathbf{r}_{ij}\mathbf{r}_{kj})\mathbf{r}_{ij}}{r_{ij}\sqrt{(r_{ij}^2\mathbf{r}_{kj} - (\mathbf{r}_{ij}\mathbf{r}_{kj})\mathbf{r}_{ij})^2}} = -\frac{1}{r_{ij}}\frac{\mathbf{r}_{ij}\times(\mathbf{r}_{kj}\times\mathbf{r}_{ij})}{|\mathbf{r}_{ij}\times(\mathbf{r}_{kj}\times\mathbf{r}_{ij})|}\end{aligned} \tag{15.36}$$

and similarly

$$\text{grad}_{\mathbf{r}_{kj}}\phi_{ijk} = -\frac{1}{r_{kj}}\frac{\mathbf{r}_{kj}\times(\mathbf{r}_{ij}\times\mathbf{r}_{kj})}{|\mathbf{r}_{kj}\times(\mathbf{r}_{ij}\times\mathbf{r}_{kj})|}. \tag{15.37}$$

Gradients of the dihedral potential are most easily calculated for $\theta_{ijkl}^0 = 0$ or π. In that case, the dihedral potential is a polynomial of $\cos\theta_{ijkl}$ only (15.20)–(15.22) and

$$\text{grad}_{\mathbf{r}} U_{ijkl}^{dihedral} = \frac{\mathrm{d}U_{ijkl}^{dihedral}}{\mathrm{d}\cos\theta_{ijkl}} \text{grad}_{\mathbf{r}} \cos\theta_{ijkl} \tag{15.38}$$

whereas in the general case $0 < \theta_{ijkl} < \pi$ application of the chain rule gives

$$\text{grad} U_{ijkl}^{dihedral} = mk_{ijkl} \sin(m(\theta_{ijkl} - \theta^0)) \text{ grad } \theta_{ijkl}. \tag{15.39}$$

If this is evaluated with the help of

$$\text{grad}\theta_{ijkl} = -\frac{1}{\sin\theta_{ijkl}} \text{grad} \cos\theta_{ijkl} \tag{15.40}$$

singularities appear for $\theta = 0$ and π. The same is the case for the gradients of the harmonic improper potential

$$\text{grad} U_{ijkl}^{improper} = k(\theta_{ijkl} - \theta_{ijkl}^0) \text{ grad}\theta_{ijkl}. \tag{15.41}$$

Again, one possibility which has been often used, is to treat angles close to 0 or π separately [188]. However, the gradient of the angle θ_{ijkl} can be calculated directly, which is much more efficient [198].

The gradient of the cosine follows from application of the product rule

$$\text{grad}\cos\theta = \text{grad}\left(\frac{\mathbf{r}_{ij} \times \mathbf{r}_{kj}}{|\mathbf{r}_{ij} \times \mathbf{r}_{kj}|} \frac{\mathbf{r}_{kj} \times \mathbf{r}_{kl}}{|\mathbf{r}_{kj} \times \mathbf{r}_{kl}|}\right). \tag{15.42}$$

First we derive the differentiation rule

$$\begin{aligned} &\text{grad}_{\mathbf{a}}[(\mathbf{a} \times \mathbf{b})(\mathbf{c} \times \mathbf{d})] = \text{grad}_{\mathbf{a}}[(\mathbf{ac})(\mathbf{bd}) - (\mathbf{ad})(\mathbf{bc})] \\ &= \mathbf{c}(\mathbf{bd}) - \mathbf{d}(\mathbf{bc}) = \mathbf{b} \times (\mathbf{c} \times \mathbf{d}) \end{aligned} \tag{15.43}$$

which helps us to find

$$\text{grad}_{r_{ij}}(\mathbf{r}_{ij} \times \mathbf{r}_{kj})(\mathbf{r}_{kj} \times \mathbf{r}_{kl}) = \mathbf{r}_{kj} \times (\mathbf{r}_{kj} \times \mathbf{r}_{kl}) \tag{15.44}$$

$$\text{grad}_{r_{kl}}(\mathbf{r}_{ij} \times \mathbf{r}_{kj})(\mathbf{r}_{kj} \times \mathbf{r}_{kl}) = \mathbf{r}_{kj} \times (\mathbf{r}_{kj} \times \mathbf{r}_{ij}) \tag{15.45}$$

$$\text{grad}_{r_{kj}}(\mathbf{r}_{ij}\times\mathbf{r}_{kj})(\mathbf{r}_{kj}\times\mathbf{r}_{kl})=\mathbf{r}_{kl}\times(\mathbf{r}_{ij}\times\mathbf{r}_{kj})+\mathbf{r}_{ij}\times(\mathbf{r}_{kl}\times\mathbf{r}_{kj}). \tag{15.46}$$

and

$$\text{grad}_{\,r_{ij}}\frac{1}{\left|\mathbf{r}_{ij}\times\mathbf{r}_{kj}\right|}=-\frac{\mathbf{r}_{kj}\times(\mathbf{r}_{ij}\times\mathbf{r}_{kj})}{\left|\mathbf{r}_{ij}\times\mathbf{r}_{kj}\right|^3} \tag{15.47}$$

$$\text{grad}_{\,r_{kj}}\frac{1}{\left|\mathbf{r}_{ij}\times\mathbf{r}_{kj}\right|}=-\frac{\mathbf{r}_{ij}\times(\mathbf{r}_{kj}\times\mathbf{r}_{ij})}{\left|\mathbf{r}_{ij}\times\mathbf{r}_{kj}\right|^3} \tag{15.48}$$

$$\text{grad}_{\,r_{kj}}\frac{1}{\left|\mathbf{r}_{kj}\times\mathbf{r}_{kl}\right|}=-\frac{\mathbf{r}_{kl}\times(\mathbf{r}_{kj}\times\mathbf{r}_{kl})}{\left|\mathbf{r}_{kj}\times\mathbf{r}_{kl}\right|^3} \tag{15.49}$$

$$\text{grad}_{\,r_{kl}}\frac{1}{\left|\mathbf{r}_{kj}\times\mathbf{r}_{kl}\right|}=-\frac{\mathbf{r}_{kj}\times(\mathbf{r}_{kl}\times\mathbf{r}_{kj})}{\left|\mathbf{r}_{kj}\times\mathbf{r}_{kl}\right|^3}. \tag{15.50}$$

Finally we collect terms to obtain the gradients of the cosine [197]

$$\begin{aligned}\text{grad}_{\mathbf{r}_{ij}}\cos\theta_{ijkl}&=\frac{\mathbf{r}_{kj}\times(\mathbf{r}_{kj}\times\mathbf{r}_{kl})}{\left|\mathbf{r}_{ij}\times\mathbf{r}_{kj}\right|\left|\mathbf{r}_{kj}\times\mathbf{r}_{kl}\right|}-\frac{\mathbf{r}_{kj}\times(\mathbf{r}_{ij}\times\mathbf{r}_{kj})}{\left|\mathbf{r}_{ij}\times\mathbf{r}_{kj}\right|^2}\cos\theta_{ijkl} \\ &=\frac{\mathbf{r}_{kj}}{\left|\mathbf{r}_{ij}\times\mathbf{r}_{kj}\right|}\times\left(\mathbf{n}_{jkl}-\mathbf{n}_{ijk}\cos\theta\right)=\frac{\mathbf{r}_{kj}}{\left|\mathbf{r}_{ij}\times\mathbf{r}_{kj}\right|}\times\left(\mathbf{n}_{jkl}-\mathbf{n}_{ijk}(\mathbf{n}_{jkl}\mathbf{n}_{ijk})\right) \\ &=\frac{\mathbf{r}_{kj}}{\left|\mathbf{r}_{ij}\times\mathbf{r}_{kj}\right|}\times\left(\mathbf{n}_{ijk}\times(\mathbf{n}_{jkl}\times\mathbf{n}_{ijk})\right) \\ &=\frac{\mathbf{r}_{kj}}{\left|\mathbf{r}_{ij}\times\mathbf{r}_{kj}\right|}\times\left(\mathbf{n}_{ijk}\times\frac{1}{r_{kj}}(-\mathbf{r}_{kj})\sin\theta\right)=\frac{\sin\theta}{r_{kj}}\frac{\mathbf{r}_{kj}}{\left|\mathbf{r}_{ij}\times\mathbf{r}_{kj}\right|}\times\left(\mathbf{n}_{ijk}\times(-\mathbf{r}_{kj})\right)= \\ &\frac{\sin\theta}{r_{kj}}\frac{1}{\left|\mathbf{r}_{ij}\times\mathbf{r}_{kj}\right|}\left(-\mathbf{n}_{ijk}r_{kj}^2\right)=-r_{kj}\sin\theta\frac{\mathbf{n}_{ijk}}{\left|\mathbf{r}_{ij}\times\mathbf{r}_{kj}\right|}\end{aligned} \tag{15.51}$$

$$\begin{aligned}\text{grad}_{\mathbf{r}_{kl}}\cos\theta_{ijkl}&=\frac{\mathbf{r}_{kj}\times(\mathbf{r}_{kj}\times\mathbf{r}_{ij})}{\left|\mathbf{r}_{ij}\times\mathbf{r}_{kj}\right|\left|\mathbf{r}_{kj}\times\mathbf{r}_{kl}\right|}-\frac{\mathbf{r}_{kj}\times(\mathbf{r}_{kl}\times\mathbf{r}_{kj})}{\left|\mathbf{r}_{kj}\times\mathbf{r}_{kl}\right|^2}\cos\theta_{ijkl} \\ &=\frac{\mathbf{r}_{kj}\times}{\left|\mathbf{r}_{kj}\times\mathbf{r}_{kl}\right|}\left(-\mathbf{n}_{ijk}+\mathbf{n}_{jkl}\cos\theta\right)=\frac{\mathbf{r}_{kj}\times}{\left|\mathbf{r}_{kj}\times\mathbf{r}_{kl}\right|}\left(-\mathbf{n}_{ijk}+\mathbf{n}_{jkl}(\mathbf{n}_{ijk}\mathbf{n}_{jkl})\right) \\ &=-\frac{\mathbf{r}_{kj}}{\left|\mathbf{r}_{kj}\times\mathbf{r}_{kl}\right|}\times\left(\mathbf{n}_{jkl}\times(\mathbf{n}_{ijk}\times\mathbf{n}_{jkl})\right)\end{aligned} \tag{15.52}$$

$$= -\frac{\mathbf{r}_{kj}}{|\mathbf{r}_{kj} \times \mathbf{r}_{kl}|} \times \left(\mathbf{n}_{jkl} \times \left(\frac{\mathbf{r}_{kj}}{r_{kj}} \sin\theta\right)\right) = -\frac{\sin\theta}{r_{kj}\,|\mathbf{r}_{kj} \times \mathbf{r}_{kl}|}\mathbf{r}_{kj} \times \left(\mathbf{n}_{jkl} \times \mathbf{r}_{kj}\right)$$
$$= -\frac{r_{kj} \sin\theta}{|\mathbf{r}_{kj} \times \mathbf{r}_{kl}|}\mathbf{n}_{jkl}$$

$$\begin{aligned}
\operatorname{grad}_{\mathbf{r}_{kj}} \cos\theta_{ijkl} &= \frac{\mathbf{r}_{kl} \times (\mathbf{r}_{ij} \times \mathbf{r}_{kj}) + \mathbf{r}_{ij} \times (\mathbf{r}_{kl} \times \mathbf{r}_{kj})}{|\mathbf{r}_{ij} \times \mathbf{r}_{kj}|\,|\mathbf{r}_{kj} \times \mathbf{r}_{kl}|} \\
&- \frac{\mathbf{r}_{ij} \times (\mathbf{r}_{kj} \times \mathbf{r}_{ij})}{|\mathbf{r}_{ij} \times \mathbf{r}_{kj}|^2} \cos\theta - \frac{\mathbf{r}_{kl} \times (\mathbf{r}_{kj} \times \mathbf{r}_{kl})}{|\mathbf{r}_{kj} \times \mathbf{r}_{kl}|^2} \cos\theta \qquad (15.53) \\
&= \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij} \times \mathbf{r}_{kj}|} \times \left(-\mathbf{n}_{jkl} + \mathbf{n}_{ijk} \cos\theta\right) + \frac{\mathbf{r}_{kl}}{|\mathbf{r}_{kj} \times \mathbf{r}_{kl}|} \times \left(\mathbf{n}_{ijk} - \mathbf{n}_{jkl} \cos\theta\right) \\
&= -\frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij} \times \mathbf{r}_{kj}|} \times \left(\mathbf{n}_{ijk} \times (\mathbf{n}_{jkl} \times \mathbf{n}_{ijk})\right) + \frac{\mathbf{r}_{kl}}{|\mathbf{r}_{kj} \times \mathbf{r}_{kl}|} \left(\mathbf{n}_{jkl} \times (\mathbf{n}_{ijk} \times \mathbf{n}_{jkl})\right) \\
&= \frac{\mathbf{r}_{ij}}{|\mathbf{r}_{ij} \times \mathbf{r}_{kj}|} \times \left(\mathbf{n}_{ijk} \times \left(\frac{\mathbf{r}_{kj}}{r_{kj}} \sin\theta\right)\right) + \frac{\mathbf{r}_{kl}}{|\mathbf{r}_{kj} \times \mathbf{r}_{kl}|} \left(\mathbf{n}_{jkl} \times \left(\frac{\mathbf{r}_{kj}}{r_{kj}} \sin\theta\right)\right) \\
&= \frac{\sin\theta}{r_{kj}} \frac{1}{|\mathbf{r}_{ij} \times \mathbf{r}_{kj}|}\mathbf{r}_{ij} \times \left(\mathbf{n}_{ijk} \times \mathbf{r}_{kj}\right) + \frac{\sin\theta}{r_{kj}} \frac{1}{|\mathbf{r}_{kj} \times \mathbf{r}_{kl}|}\mathbf{r}_{kl} \times \left(\mathbf{n}_{jkl} \times \mathbf{r}_{kj}\right) \\
&= \frac{\sin\theta}{r_{kj}} \frac{\mathbf{n}_{ijk}(\mathbf{r}_{ij}\mathbf{r}_{kj})}{|\mathbf{r}_{ij} \times \mathbf{r}_{kj}|} + \frac{\sin\theta}{r_{kj}} \frac{\mathbf{n}_{jkl}(\mathbf{r}_{kl}\mathbf{r}_{kj})}{|\mathbf{r}_{kj} \times \mathbf{r}_{kl}|} \\
&= -\frac{\mathbf{r}_{ij}\mathbf{r}_{kj}}{r_{kj}^2} \operatorname{grad}_{ij} \cos\theta - \frac{\mathbf{r}_{kl}\mathbf{r}_{kj}}{r_{kj}^2} \operatorname{grad}_{kl} \cos\theta.
\end{aligned}$$

15.4 Normal Mode Analysis

The nuclear motion around an equilibrium configuration can be approximately described as the combination of independent harmonic normal modes. Equilibrium configurations can be found with the methods discussed in Sect. 6.2. The convergence is usually rather slow (Fig. 15.8) except for the full Newton-Raphson method, which needs the calculation and inversion of the Hessian matrix.

15.4.1 Harmonic Approximation

At an equilibrium configuration

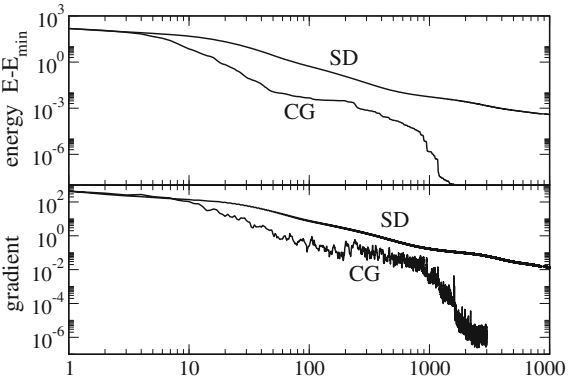


Fig. 15.8 (Convergence of energy and gradient) The energy of the glycine dipeptide is minimized with the methods of steepest descent and conjugate gradients

$$\xi_i = \xi_i^{eq} \tag{15.54}$$

the gradient of the potential energy vanishes. For small deviations from the equilibrium

$$\zeta_i = \xi_i - \xi_i^{eq} \tag{15.55}$$

approximation by a truncated Taylor series gives

$$U(\zeta_1 \cdots \zeta_{3N}) = U_0 + \frac{1}{2}\sum_{i,j}\frac{\partial^2 U}{\partial\zeta_i \partial\zeta_j}\zeta_i\zeta_j + \cdots \approx U_0 + \frac{1}{2}\sum_{i,j} H_{i,j}\zeta_i\zeta_j \tag{15.56}$$

and the equations of motion are approximately

$$m_i\ddot{\zeta}_i = -\frac{\partial}{\partial\zeta_i}U = -\sum_j H_{i,j}\zeta_j. \tag{15.57}$$

Assuming periodic oscillations

$$\zeta_i = \zeta_i^0 \mathrm{e}^{\mathrm{i}\omega t} \tag{15.58}$$

we have

$$m_i\omega^2\zeta_i^0 = \sum_j H_{ij}\zeta_j^0. \tag{15.59}$$

If mass weighted coordinates are used, defined as

$$\tau_i = \sqrt{m_i}\zeta_i \tag{15.60}$$

this becomes an ordinary eigenvalue problem

$$\omega^2 \tau_i^0 = \sum_j \frac{H_{ij}}{\sqrt{m_i m_j}} \tau_j^0. \tag{15.61}$$

The eigenvectors $\mathbf{u}_r$ of the symmetric matrix

$$\tilde{H}_{ij} = \frac{H_{ij}}{\sqrt{m_i m_j}} \tag{15.62}$$

are the solutions of

$$\sum_j \tilde{H}_{ij} u_{jr} = \lambda_r u_{ir} \tag{15.63}$$

and satisfy (15.61)

$$\omega^2 u_{ir} = \sum \tilde{H}_{ij} u_{jr} = \lambda_r u_{ir} \tag{15.64}$$

with normal mode frequencies

$$\omega_r = \sqrt{\lambda_r}. \tag{15.65}$$

Finally, the Cartesian coordinates are linear combinations of all normal modes

$$\zeta_i = \sum_r C_r \frac{u_{ir}}{\sqrt{m_i}} \mathrm{e}^{\mathrm{i}\omega_r t}. \tag{15.66}$$

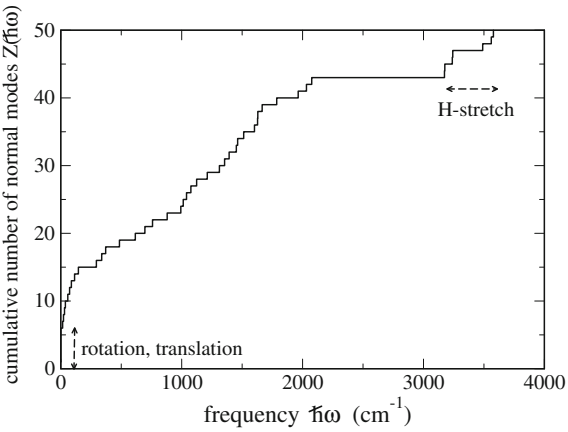


Fig. 15.9 (Normal mode distribution for the dipeptide model) The cumulative distribution (Sect. 9.1.2) of normal mode frequencies is shown for the glycine dipeptide. Translations and rotations of the molecule correspond to the lowest 6 frequencies which are close to zero. The highest frequencies between 3100 and 3600 cm^{-1} correspond to the stretching modes of the 8 hydrogen atoms

In a true local energy minimum the Hessian matrix H_{ij} is positive definite and all frequencies are real valued. The six lowest frequencies are close to zero and correspond to translations and rotations of the whole system (Fig. 15.9).

Problems

Problem 15.1 Simulation of a Glycine Dipeptide

In this computer experiment a glycine dipeptide (Fig. 15.5) is simulated. Parameters for bond stretching (Table 15.3) and bond angle (Table 15.4) terms have been derived from quantum calculations by Bautista and Seminario [191].

- Torsional potential terms (Table 15.5) can be added to make the structure more rigid. This is especially important for the $O9-H17$, $N4-H14$ and $N1-H10$ bonds, which rotate almost freely without torsional potentials.
- The energy can be minimized with the methods of steepest descent or conjugate gradients
- A normal mode analysis can be performed (the Hessian matrix is calculated by numerical differentiation). The rth normal mode can be visualized by modulating the coordinates periodically according to

$$\xi_i = \xi_i^{eq} + C_r \frac{u_{ir}}{\sqrt{m_i}} \cos \omega_r t. \tag{15.67}$$

- The motion of the atoms can be simulated with the Verlet method. You can stretch the $O9-H17$ or $N4-H14$ bond and observe, how the excitation spreads over the molecule.

Table 15.5 (Torsional potential terms) Torsional potential terms $V_{ijkl} = k_{ijkl}(1-\cos(\theta_{ijkl}-\theta^0_{ijkl})$, which can be added to the force field. Minimum angles are from the optimized structure without torsional terms (15.1). The barrier height of $2k_{ijkl} = 2\,\mathrm{kcal/mol}$ is only a guessed value

i	j	k	l	θ^0_{ijkl}	k_{ijkl}	Backbone
10	1	2	3	−67.6	1.0	
14	4	3	2	27.5	1.0	
17	9	6	5	−147.4	1.0	
4	3	2	1	160.7	1.0	Ψ
5	4	3	2	−152.3	1.0	ω
6	5	4	3	−153.1	1.0	Φ
8	6	5	4	123.7	1.0	
9	6	5	4	−56.5	1.0	
15	5	4	3	−32.5	1.0	
16	5	4	3	86.3	1.0	
7	3	2	1	−26.3	1.0	

Chapter 16
Thermodynamic Systems

An important application for computer simulations is the calculation of thermodynamic averages in an equilibrium system. We discuss two different examples:

In the first case the classical equations of motion are solved for a system of particles interacting pairwise by Lennard–Jones forces (Lennard–Jones fluid). The thermodynamic average is taken along the trajectory, i.e. over the calculated coordinates at different times $\mathbf{r}_i(t_n)$*. We evaluate the pair distance distribution function*

$$g(R) = \frac{1}{N^2 - N} < \sum_{i \neq j} \delta(r_{ij} - R) >, \tag{16.1}$$

the velocity auto-correlation function

$$C(t) = < \mathbf{v}(t_0)\mathbf{v}(t) > \tag{16.2}$$

and the mean square displacement

$$\Delta x^2 = < (\mathbf{x}(t) - \mathbf{x}(t_0))^2 > . \tag{16.3}$$

In the second case the Metropolis method is applied to a one- or two-dimensional system of interacting spins (Ising model). The thermodynamic average is taken over a set of random configurations $\mathbf{q}^{(n)}$*. We study the average magnetization*

$$< M > = \mu < S > \tag{16.4}$$

in a magnetic field and the phase transition to the ferromagnetic state.

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_16

16.1 Simulation of a Lennard–Jones Fluid

The Lennard–Jones fluid is a simple model of a realistic atomic fluid. It has been studied by computer simulations since Verlet's early work [164, 199] and serves as a test case for the theoretical description of liquids [200, 201] and the liquid-gas [202] and liquid-solid phase transitions [203, 204].

In the following we describe a simple computer model of 125 interacting particles[1] without internal degrees of freedom (see problems section). The force on atom i is given by the gradient of the pairwise Lennard–Jones potential (15.24)

$$\mathbf{F}_i = \sum_{j \neq i} \mathbf{F}_{ij} = -4\varepsilon \sum_{j \neq i} \nabla_i \left(\frac{\sigma^{12}}{r_{ij}^{12}} - \frac{\sigma^6}{r_{ij}^6} \right) = 4\varepsilon \sum_{j \neq i} \left(\frac{12\sigma^{12}}{r_{ij}^{14}} - \frac{6\sigma^6}{r_{ij}^8} \right) (\mathbf{r}_i - \mathbf{r}_j). \tag{16.5}$$

We use argon parameters $m = 6.69 \times 10^{-26}\,\mathrm{kg}$, $\varepsilon = 1.654 \times 10^{-21}\,\mathrm{J}$, $\sigma = 3.405 \times 10^{-10}\,\mathrm{m}$ [163]. After introduction of reduced units for length $\mathbf{r}^* = \frac{1}{\sigma}\mathbf{r}$, energy $E^* = \frac{1}{\varepsilon}E$ and time $t^* = \sqrt{\varepsilon/m\sigma^2}\quad t$, the potential energy

$$U^* = \sum_{\mathrm{ij}} 4 \left(\frac{1}{r_{ij}^{*12}} - \frac{1}{r_{ij}^{*6}} \right) \tag{16.6}$$

and the equation of motion

$$\frac{\mathrm{d}^2}{\mathrm{d}t^{*2}} \mathbf{r}_i^* = 4 \sum_{j \neq i} \left(\frac{12}{r_{ij}^{*14}} - \frac{6}{r_{ij}^{*8}} \right) (\mathbf{r}_i^* - \mathbf{r}_j^*) \tag{16.7}$$

become universal expressions, i.e. there exists only one universal Lennard–Jones system. To reduce computer time, usually the 6–12 potential is modified at larger distances which can influence the simulation results [205]. In our model a simple cutoff of potential and forces at $r_{\max} = 10\text{Å}$ is used.

16.1.1 Integration of the Equations of Motion

The equations of motion are integrated with the Verlet algorithm (Sect. 13.11.5)

$$\Delta\mathbf{r}_i = \mathbf{r}_i(t) - \mathbf{r}_i(t - \Delta t) + \frac{\mathbf{F}_i(t)}{m} \Delta t^2 \tag{16.8}$$

[1] This small number of particles allows a graphical representation of the system during the simulation.

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \Delta\mathbf{r}_i + O(\Delta t^4). \tag{16.9}$$

We use a higher order expression for the velocities to improve the accuracy of the calculated kinetic energy

$$\mathbf{v}_{i+1} = \frac{\Delta\mathbf{r}_i}{\Delta t} + \frac{5\mathbf{F}_i(t) - 2\mathbf{F}_i(t - \Delta t)}{6m}\Delta t + O(\Delta t^3). \tag{16.10}$$

16.1.2 Boundary Conditions and Average Pressure

Molecular dynamics simulations often involve periodic boundary conditions to reduce finite size effects. Here we employ an alternative method which simulates a box with elastic walls. This allows us to calculate explicitly the pressure on the walls of the box.

The atoms are kept in the cube by reflecting walls, i.e. whenever an atom passes a face of the cube, the normal component of the velocity vector is changed in sign (Fig. 16.1). Thus the kinetic energy is conserved but a momentum of $m\Delta v = 2mv_\perp$ is transferred to the wall. The average momentum change per time can be interpreted as a force acting upon the wall

$$F_\perp = < \frac{\sum_{refl.} 2mv_\perp}{dt} > . \tag{16.11}$$

The pressure p is given by

$$p = \frac{1}{6L^2} < \frac{\sum_{walls}\sum_{refl.} 2mv_\perp}{dt} > . \tag{16.12}$$

With the Verlet algorithm the reflection can be realized by exchanging the values of the corresponding coordinate at times t_n and t_{n-1}.

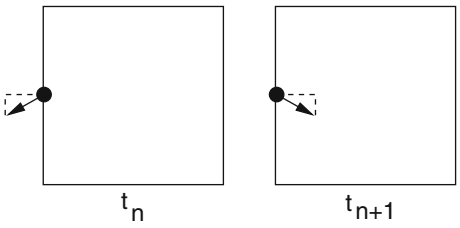


Fig. 16.1 Reflecting walls

16.1.3 Initial Conditions and Average Temperature

At the very beginning the $N = 125$ atoms are distributed over equally spaced lattice points within the cube. Velocities are randomly distributed according to a Gaussian distribution for each Cartesian component v_μ

$$f(v_\mu) = \sqrt{\frac{m}{2\pi k_B T}} \mathrm{e}^{-m v_\mu^2 / 2 k_B T} \tag{16.13}$$

corresponding to a Maxwell speed distribution

$$f(|v|) = \left(\frac{m}{2\pi k_B T}\right)^{3/2} 4\pi v^2 \mathrm{e}^{-m v^2 / 2 k_B T}. \tag{16.14}$$

Assuming thermal equilibrium, the effective temperature is calculated from the kinetic energy

$$k_B T = \frac{2}{3N} E_{kin}. \tag{16.15}$$

The desired temperature T_o is established by the rescaling procedure

$$\mathbf{v}_i \to \mathbf{v}_i \sqrt{\frac{k_B T_o}{k_B T_{actual}}} \tag{16.16}$$

which is applied repeatedly during an equilibration run. The velocity distribution $f(|v|)$ can be monitored. It approaches quickly a stationary Maxwell distribution (Fig. 16.2).

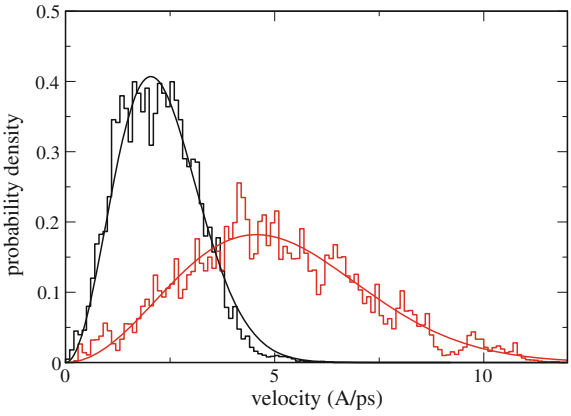


Fig. 16.2 (Velocity distribution) The velocity distribution is shown for $T = 100\,\mathrm{K}$ and $T = 500\,\mathrm{K}$ (*histograms*) and compared to the Maxwell speed distribution (*solid curves*)

A smoother method to control temperature is the Berendsen thermostat algorithm [206]

$$\mathbf{v}_i \rightarrow \mathbf{v}_i \sqrt{1 + \frac{\Delta t}{\tau_{therm}} \frac{kT_o - kT_{actual}}{kT_{actual}}} \tag{16.17}$$

where τ_{therm} is a suitable relaxation time (for instance $\tau_{therm} = 20\Delta t$). This method can be used also during the simulation. However, it does not generate the trajectory of a true canonical ensemble. If this is necessary, more complicated methods have to be used [207]

16.1.4 Analysis of the Results

After an initial equilibration phase the system is simulated at constant energy (NVE simulation) or at constant temperature (NVT) with the Berendsen thermostat method. Several static and dynamic properties can be determined.

16.1.4.1 Deviation from the Ideal Gas Behavior

A dilute gas is approximately ideal with

$$pV = Nk_BT. \tag{16.18}$$

For a real gas the interaction between the particles has to be taken into account. From the equipartition theorem it can be found that[2]

$$pV = Nk_BT + W \tag{16.19}$$

with the inner virial (Fig. 16.3)

$$W = < \frac{1}{3} \sum_i \mathbf{r}_i \mathbf{F}_i > \tag{16.20}$$

which can be expanded as a power series of the number density $n = N/V$ [208] to give

$$pV = Nk_BT(1 + b(T)\frac{N}{V} + c(T)\left(\frac{N}{V}\right)^2 + \cdots). \tag{16.21}$$

[2]MD simulations with periodic boundary conditions use this equation to calculate the pressure.

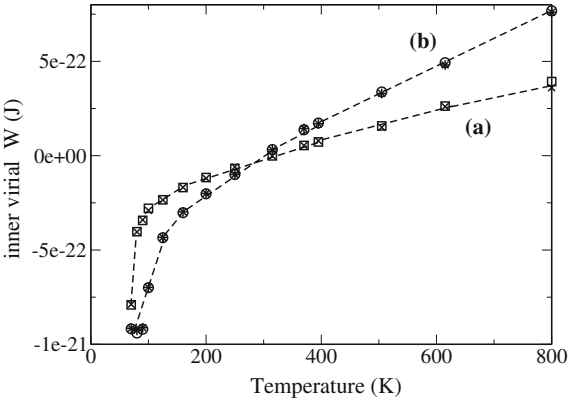


Fig. 16.3 (Inner virial) The inner virial W (16.20, *crosses* and *stars*) is compared to $pV - k_BT$ (*squares* and *circles*) for two values of the particle density $N/V = 10^{-3}\text{Å}^{-1}$ (**a**) and $1.95 \times 10^{-3}\text{Å}^{-1}$ (**b**), corresponding to reduced densities $n^* = \sigma^3 N/V$ of 0.040 and 0.077

The virial coefficient $b(T)$ can be calculated exactly for the Lennard–Jones gas [208]:

$$b(T) = -\frac{2\pi}{3}\sigma^3 \sum_{j=0}^{\infty} \frac{2^{j-3/2}}{j!}\Gamma\left(\frac{2j-1}{4}\right)\left(\frac{\varepsilon}{k_BT}\right)^{(j/2+1/4)}. \tag{16.22}$$

For comparison we calculate the quantity

$$\frac{V}{N}\left(\frac{pV}{Nk_BT} - 1\right) \tag{16.23}$$

which for small values of the particle density $n = N/V$ correlates well (Fig. 16.4) with expression (16.22).

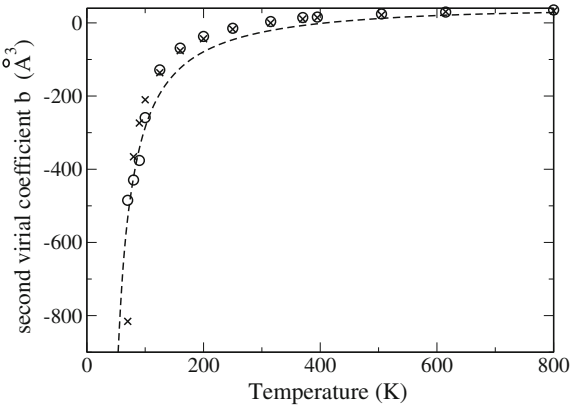


Fig. 16.4 (Second virial coefficient) The value of $\frac{V}{N}\left(\frac{pV}{k_BT} - 1\right)$ is shown for two values of the particle density $N/V = 10^{-3}\,\text{AA}$ (*crosses*) and $1.95 \times 10^{-3}\,\text{AA}^{-1}$ (*circles*) and compared to the exact second virial coefficient b (*dashed curve*) (16.22)

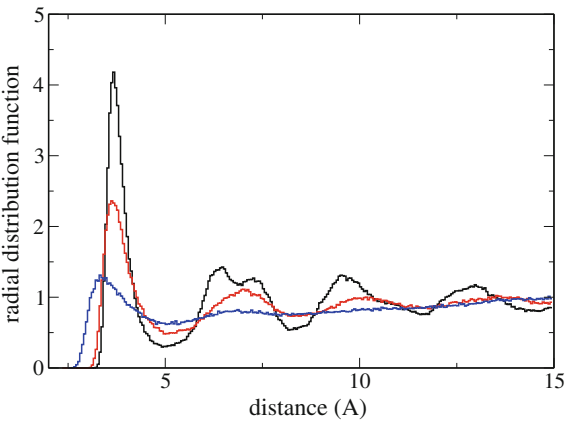


Fig. 16.5 (Radial pair distribution) The normalized radial distribution function $g(R)/g_{ideal}(R)$ is evaluated for $kT = 35\,\mathrm{K}$, 100 K, 1000 K and a density of $n = 0.025\,A^{-3}$ corresponding to a reduced density $n^* = \sigma^3 N/V$ of 1.0. At this density the Lennard–Jones system shows a liquid-solid transition at a temperature of ca. 180 K [204]

16.1.4.2 Structural Order

A convenient measure for structural order [209] is the radial pair distribution function (Fig. 16.5)

$$g(R) = < \frac{1}{N(N-1)} \sum_{i \neq j} \delta(r_{ij} - R) > = \frac{P(R < r_{ij} < R + dR)}{dR} \tag{16.24}$$

which is usually normalized with respect to an ideal gas, for which

$$g_{ideal}(R) = 4\pi n\, R^2 dR. \tag{16.25}$$

For small distances $g(R)/g_{ideal}(R)$ vanishes due to the strong repulsive force. It peaks at the distance of nearest neighbors and approaches unity at very large distances. In the condensed phase additional maxima appear showing the degree of short (liquid) and long range (solid) order.

Equation (16.25) is not valid for our small model system without periodic boundary conditions. Therefore g_{ideal} was calculated numerically to normalize the results shown in Fig. 16.5.

16.1.4.3 Ballistic and Diffusive Motion

The velocity auto-correlation function (Fig. 16.6)

$$C(t) = < \mathbf{v}(t)\mathbf{v}(t_0) > \tag{16.26}$$

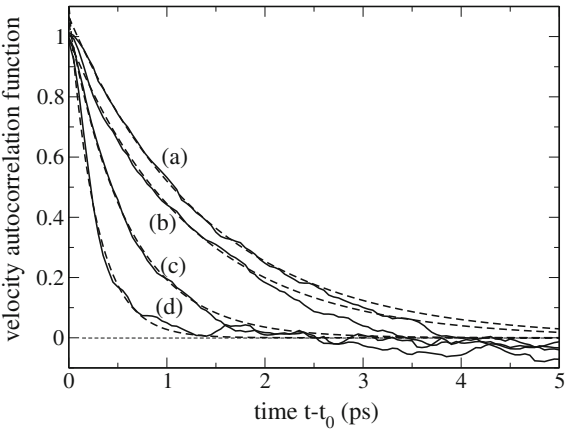


Fig. 16.6 (Velocity auto-correlation function) The Lennard–Jones system is simulated for $k_B T = 200\,\mathrm{K}$ and different values of the density $n^* = 0.12$ (a), 0.18 (b), 0.32 (c), 0.62 (d). The velocity auto-correlation function (*full curves*) is averaged over 20 trajectories and fitted by an exponential function (*dashed curves*)

decays as a function of the delay time $t - t_0$ due to collisions of the particles. In a stationary state it does not depend on the initial time t_0. Integration leads to the mean square displacement (Fig. 16.6)

$$\Delta x^2(t) = < (\mathbf{x}(t) - \mathbf{x}(t_0))^2 > . \tag{16.27}$$

In the absence of collisions the mean square displacement grows with $(t - t_0)^2$, representing a ballistic type of motion. Collisions lead to a diffusive kind of motion where the mean square displacement grows only linearly with time. The transition between this two types of motion can be analyzed within the model of Brownian motion [210] where the collisions are replaced by a fluctuating random force $\Gamma(t)$ and a damping constant γ.
The equation of motion in one dimension is

$$\dot{v} + \gamma v = \Gamma(t) \tag{16.28}$$

with

$$< \Gamma(t) > = 0 \tag{16.29}$$

$$< \Gamma(t)\Gamma(t') > = \frac{2\gamma k_B T}{m}\delta(t - t'). \tag{16.30}$$

The velocity correlation decays exponentially

$$< v(t)v(t_0) > = \frac{k_B T}{m} e^{-\gamma|t-t_0|} \tag{16.31}$$

with the average velocity square given by

$$< v^2 > = C(t_0) = \frac{k_B T}{m} = \frac{< E_{kin} >}{\frac{m}{2}} \tag{16.32}$$

and the integral of the correlation function equals

$$\int_{t_0}^{\infty} C(t)dt = \frac{k_B T}{\gamma m}. \tag{16.33}$$

The average of Δx^2 is

$$< (x(t) - x(t_0))^2 > = \frac{2k_B T}{m\gamma}(t - t_0) - \frac{2k_B T}{m\gamma^2}(1 - e^{-\gamma(t-t_0)}). \tag{16.34}$$

For small time differences $t - t_0$ the motion is ballistic with the thermal velocity

$$< (x(t) - x(t_0))^2 > \approx \frac{k_B T}{m}(t - t_0)^2 = < v^2 > (t - t_0)^2. \tag{16.35}$$

For large time differences diffusive motion emerges with

$$< (x(t) - x(t_0)^2 > \approx \frac{2k_B T}{m\gamma}(t - t_0) = 2D(t - t_0) \tag{16.36}$$

with the diffusion constant given by the Einstein relation

$$D = \frac{k_B T}{m\gamma}. \tag{16.37}$$

For a three-dimensional simulation the Cartesian components of the position or velocity vector add up independently. The diffusion coefficient can be determined from

$$D = \frac{1}{6} \lim_{t \to \infty} \frac{< (x(t) - x(t_0))^2 >}{t - t_0} \tag{16.38}$$

or, alternatively from (16.33) [163]

$$D = \frac{1}{3} \int_{t_0}^{\infty} < \mathbf{v}(t)\mathbf{v}(t_0) > dt. \tag{16.39}$$

This equation is more generally valid also outside the Brownian limit (Green–Kubo formula). The Brownian model represents the simulation data quite well at low particle densities (Figs. 16.6 and 16.7). For higher densities the velocity auto-correlation function shows a very rapid decay followed by a more or less structured tail. [163, 211, 212]

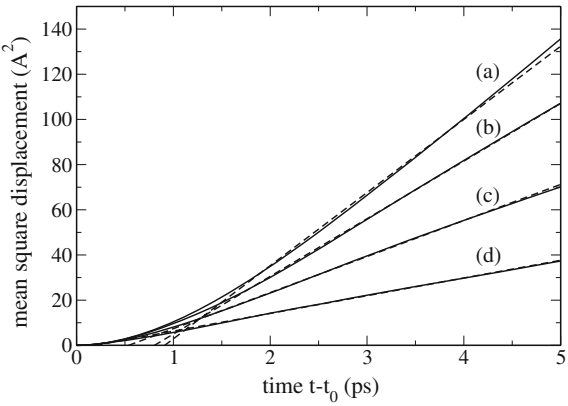


Fig. 16.7 (Mean square displacement) The Lennard–Jones system is simulated for $k_B T = 200K$ and different values of the density $n^* = 0.12$ (a), 0.18 (b), 0.32 (c), 0.62 (d). The mean square displacement (*full curves*) is averaged over 20 trajectories and fitted by a linear function (*dashed lines*) for $t - t_0 > 1.5$ps

16.2 Monte-Carlo Simulation

The basic principles of Monte Carlo simulations are discussed in Chap. 9. Here we will apply the Metropolis algorithm to simulate the Ising model in one or two dimensions. The Ising model [213, 214] is primarily a model for the phase transition of a ferromagnetic system. However, it has further applications for instance for a polymer under the influence of an external force or protonation equilibria in proteins.

16.2.1 One-Dimensional Ising Model

We consider a chain consisting of N spins which can be either up ($S_i = 1$) or down ($S_i = -1$). The total energy in a magnetic field is (Fig. 16.8)

$$H = -MB = -B \sum_{i=1}^{N} \mu S_i \tag{16.40}$$

and the average magnetic moment of one spin is

$$< M > = \mu \frac{e^{\mu B/kT} - e^{-\mu B/kT}}{e^{\mu B/kT} + e^{-\mu B/kT}} = \mu \tanh(\frac{\mu B}{kT}). \tag{16.41}$$

If interaction between neighboring spins is included, the energy of a configuration $(S_1 \cdots S_N)$ becomes

$$H = -\mu B \sum_{i=1}^{N} S_i - J \sum_{i=1}^{N-1} S_i S_{i+1}. \tag{16.42}$$

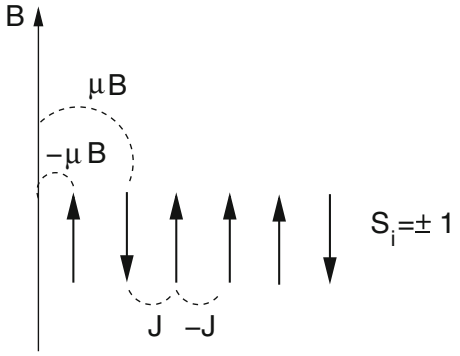


Fig. 16.8 (Ising model) N spins can be *up* or *down*. The interaction with the magnetic field is $-\mu B S_i$, the interaction between nearest neighbors is $-J S_i S_j$

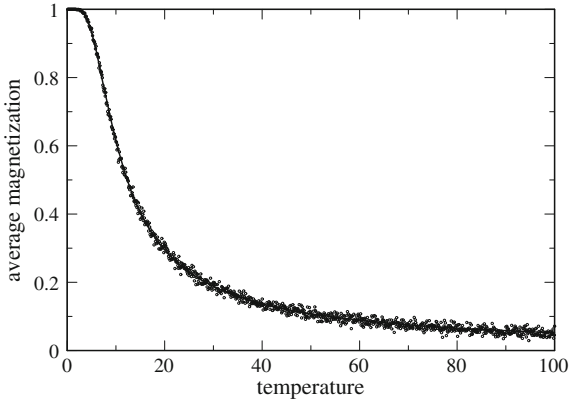


Fig. 16.9 (Numerical simulation of the 1-dimensional Ising model) The average magnetization per spin is calculated from a MC simulation (*circles*) and compared to the exact solution (16.43). Parameters are $\mu B = -5$ and $J = -2$

The 1-dimensional model can be solved analytically [208]. In the limit $N \to \infty$ the magnetization is

$$< M >= \mu \frac{\sinh(\frac{\mu B}{kT})}{\sqrt{\sinh^2(\frac{\mu B}{kT}) + e^{4J/kT}}}. \tag{16.43}$$

The numerical simulation (Fig. 16.9) starts either with the ordered state $S_i = 1$ or with a random configuration. New configurations are generated with the Metropolis method as follows:

Table 16.1 Transition probabilities for a 3-spin system ($p = 1/3$)

	+++	++−	+−+	+−−	−++	−+−	−−+	−−−
+++	0	p	p	0	p	0	0	0
++−	p	0	0	p	0	p	0	0
+−+	p	0	0	p	0	0	p	0
+−−	0	p	p	0	0	0	0	p
−++	p	0	0	0	0	p	p	0
−+−	0	p	0	0	p	0	0	p
−−+	0	0	p	0	p	0	0	p
−−−	0	0	0	p	0	p	p	0

- flip one randomly chosen spin S_i[3] and calculate the energy change due to the change $\Delta S_i = (-S_i) - S_i = -2S_i$

$$\Delta E = -\mu B \Delta S_i - J \Delta S_i (S_{i+1} + S_{i-1}) = 2\mu B S_i + 2J S_i \, (S_{i+1} + S_{i-1}) \, . \tag{16.44}$$

- if $\Delta E < 0$ then accept the flip, otherwise accept it with a probability of $P = e^{-\Delta E/kT}$

As a simple example consider N=3 spins which have 8 possible configurations. The probabilities of the trial step $T_{i \to j}$ are shown in Table 16.1. The table is symmetric and all configurations are connected.

16.2.2 Two-Dimensional Ising Model

For dimension $d > 1$ the Ising model behaves qualitatively different as a phase transition appears. For $B = 0$ (Fig. 16.10) the 2-dimensional Ising-model with 4 nearest neighbors can be solved analytically [215, 216]. The magnetization disappears above the critical temperature T_c, which is given by

$$\frac{J}{kT_c} = -\frac{1}{2} \ln(\sqrt{2} - 1) \approx \frac{1}{2.27}. \tag{16.45}$$

Below T_c the average magnetization is given by

$$< M > = \left(1 - \frac{1}{\sinh^4(\frac{2J}{kT})} \right)^{\frac{1}{8}} . \tag{16.46}$$

[3] Or try one spin after the other.

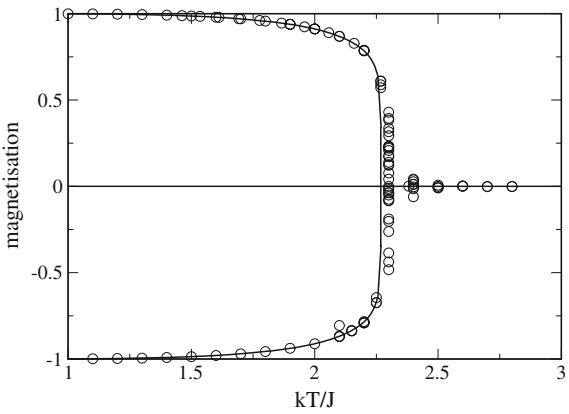


Fig. 16.10 (Numerical simulation of the 2-dimensional Ising model) The average magnetization per spin is calculated for $B = 0$ from a MC simulation (*circles*) and compared to (16.46)

Problems

Problem 16.1 Lennard–Jones Fluid

In this computer experiment a Lennard–Jones fluid is simulated. The pressure is calculated from the average transfer of momentum (16.12) and compared with expression (16.19).

- Equilibrate the system and observe how the distribution of squared velocities approaches a Maxwell distribution.
- Equilibrate the system for different values of temperature and volume and investigate the relation between pV/N and kT.
- observe the radial distribution function for different values of temperature and densities. Try to locate phase transitions.
- determine the decay time of the velocity correlation function and compare with the behavior of the mean square displacement which shows a transition from ballistic to diffusive motion.

Problem 16.2 One-Dimensional Ising Model

In this computer experiment we simulate a linear chain of $N = 500$ spins with periodic boundaries and interaction between nearest neighbors only. We go along the chain and try to flip one spin after the other according to the Metropolis method.

After trying to flip the last spin S_N the total magnetization

$$M = \sum_{i=1}^{N} S_i \tag{16.47}$$

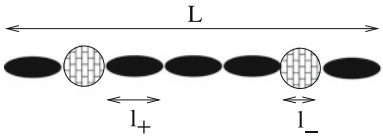


Fig. 16.11 Two state model

is calculated. It is averaged over 500 such cycles and then compared graphically with the analytical solution for the infinite chain (16.43). Temperature and magnetic field can be varied.

Problem 16.3 Two-State Model for a Polymer

Consider a polymer (Fig. 16.11) consisting of N units which can be in two states $S_i = +1$ or $S_i = -1$ with corresponding lengths l_+ and l_-. The interaction between neighboring units takes one of the values w_{++}, w_{+-}, w_{--}. Under the influence of an external force κ the energy of the polymer is

$$E = -\kappa \sum_i l(S_i) + \sum_i w(S_i, S_{i+1}). \tag{16.48}$$

This model is isomorphic to the one-dimensional Ising model.

$$E = -\kappa N \frac{l_- + l_+}{2} - \kappa \frac{l_+ - l_-}{2} \sum S_i \tag{16.49}$$

$$+ \sum \left(w_{+-} + \frac{w_{++} - w_{+-}}{2} S_i + \frac{w_{+-} - w_{--}}{2} S_{i+1} \right.$$

$$\left. + \frac{w_{++} + w_{--} - 2w_{+-}}{2} S_i S_{i+1} \right) \tag{16.50}$$

$$= \kappa N \frac{l_- + l_+}{2} + N w_{+-}$$

$$- \kappa \frac{l_+ - l_-}{2} M + \frac{w_{++} - w_{--}}{2} M$$

$$+ \frac{w_{++} + w_{--} - 2w_{+-}}{2} \sum S_i S_{i+1}. \tag{16.51}$$

Comparison with (16.42) shows the correspondence

$$-J = \frac{w_{++} + w_{--} - 2w_{+-}}{2} \tag{16.52}$$

$$-\mu B = -\kappa \frac{l_+ - l_-}{2} + \frac{w_{++} - w_{--}}{2} \tag{16.53}$$

$$L = \sum l(S_i) = N\frac{l_+ + l_-}{2} + \frac{l_+ - l_-}{2}M. \tag{16.54}$$

In this computer experiment we simulate a linear chain of $N = 20$ units with periodic boundaries and nearest neighbor interaction as in the previous problem.

The fluctuations of the chain conformation are shown graphically and the magnetization of the isomorphic Ising model is compared with the analytical expression for the infinite system (16.43). Temperature and magnetic field can be varied as well as the coupling J. For negative J the anti-ferromagnetic state becomes stable at low magnetic field strengths.

Problem 16.4 Two-Dimensional Ising Model

In this computer experiment a 200×200 square lattice with periodic boundaries and interaction with the 4 nearest neighbors is simulated. The fluctuations of the spins can be observed. At low temperatures ordered domains with parallel spin appear. The average magnetization is compared with the analytical expression for the infinite system (16.46).

Chapter 17
Random Walk and Brownian Motion

Random walk processes are an important class of stochastic processes. They have many applications in physics, computer science, ecology, economics and other fields. A random walk [217] *is a sequence of successive random steps. In this chapter we study Markovian* [218, 219][1] *discrete time*[2] *models. In one dimension the position of the walker after n steps approaches a Gaussian distribution, which does not depend on the distribution of the single steps. This follows from the central limit theorem and can be checked in a computer experiment. A 3-dimensional random walk provides a simple statistical model for the configuration of a biopolymer, the so called freely jointed chain model. In a computer experiment we generate random structures and calculate the gyration tensor, an experimentally observable quantity, which gives information on the shape of a polymer. Simulation of the dynamics is simplified if the fixed length segments of the freely jointed chain are replaced by Hookean springs. This is utilized in a computer experiment to study the dependence of the polymer extension on an applied external force (this effect is known as entropic elasticity). The random motion of a heavy particle in a bath of light particles, known as Brownian motion, can be described by Langevin dynamics, which replace the collisions with the light particles by an average friction force proportional to the velocity and a randomly fluctuating force with zero mean and infinitely short correlation time. In a computer experiment we study Brownian motion in a harmonic potential.*

17.1 Markovian Discrete Time Models

The time evolution of a system is described in terms of an N-dimensional vector $\mathbf{r}(t)$, which can be for instance the position of a molecule in a liquid, or the price of a fluctuating stock. At discrete times $t_n = n\Delta t$ the position changes suddenly (Fig. 17.1)

$$\mathbf{r}(t_{n+1}) = \mathbf{r}(t_n) + \Delta\mathbf{r}_n \tag{17.1}$$

[1] Different steps are independent.

[2] A special case of the more general continuous time random walk with a waiting time distribution of $P(\tau) = \delta(\tau - \Delta t)$.

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_17

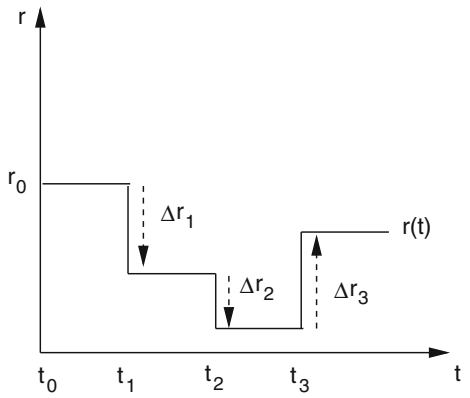


Fig. 17.1 Discrete time random walk

where the steps are distributed according to the probability distribution[3]

$$P(\Delta \mathbf{r}_n = \mathbf{b}) = f(\mathbf{b}). \tag{17.2}$$

The probability of reaching the position $\mathbf{R}$ after $n+1$ steps obeys the equation

$$\begin{aligned} P_{n+1}(\mathbf{R}) &= P\left(\mathbf{r}(t_{n+1}) = \mathbf{R}\right) \\ &= \int d^N\mathbf{b}\, P_n(\mathbf{R}-\mathbf{b}) f(\mathbf{b}). \end{aligned} \tag{17.3}$$

17.2 Random Walk in One Dimension

Consider a random walk in one dimension. We apply the central limit theorem to calculate the probability distribution of the position r_n after n steps. The first two moments and the standard deviation of the step distribution are

$$\overline{b} = \int db\, b f(b) \quad \overline{b^2} = \int db\, b^2 f(b) \quad \sigma_b = \sqrt{\overline{b^2} - \overline{b}^2}. \tag{17.4}$$

Hence the normalized quantity

$$\xi_i = \frac{\Delta x_i - \overline{b}}{\sigma_b} \tag{17.5}$$

[3]General random walk processes are characterized by a distribution function $P(\mathbf{R}, \mathbf{R}')$. Here we consider only correlated processes for which $P(\mathbf{R}, \mathbf{R}') = P(\mathbf{R}' - \mathbf{R})$.

is a random variable with zero average and unit standard deviation. The distribution function of the new random variable

$$\eta_n = \frac{\xi_1 + \xi_2 + \cdots + \xi_n}{\sqrt{n}} = \frac{r_n - n\bar{b}}{\sigma_b \sqrt{n}} \tag{17.6}$$

approaches a normal distribution for large n

$$f(\eta_n) \to \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\eta_n^2/2} \tag{17.7}$$

and finally from

$$f(r_n)dr_n = f(\eta_n)d\eta_n = f(\eta_n)\frac{dr_n}{\sigma_b\sqrt{n}}$$

we have

$$f(r_n) = \frac{1}{\sqrt{2\pi n}\sigma_b} \exp\left\{-\frac{(r_n - n\bar{b})^2}{2n\sigma_b^2}\right\}. \tag{17.8}$$

The position of the walker after n steps obeys approximately a Gaussian distribution centered at $\bar{r}_n = n\bar{b}$ with a standard deviation of

$$\sigma_{r_n} = \sqrt{n}\sigma_b. \tag{17.9}$$

17.2.1 Random Walk with Constant Step Size

In the following we consider the classical example of a 1-dimensional random walk process with constant step size. At time t_n the walker takes a step of length Δx to the left with probability p or to the right with probability $q = 1 - p$ (Figs. 17.2, 17.3).

The corresponding step size distribution function is

$$f(b) = p\delta(b + \Delta x) + q\delta(b - \Delta x) \tag{17.10}$$

with the first two moments

$$\bar{b} = (q - p)\Delta x \quad \overline{b^2} = \Delta x^2. \tag{17.11}$$

Let the walker start at $r(t_0) = 0$. The probability $P_n(m)$ of reaching position $m\Delta x$ after n steps obeys the recursion

$$P_{n+1}(m) = pP_n(m + 1) + qP_n(m - 1) \tag{17.12}$$

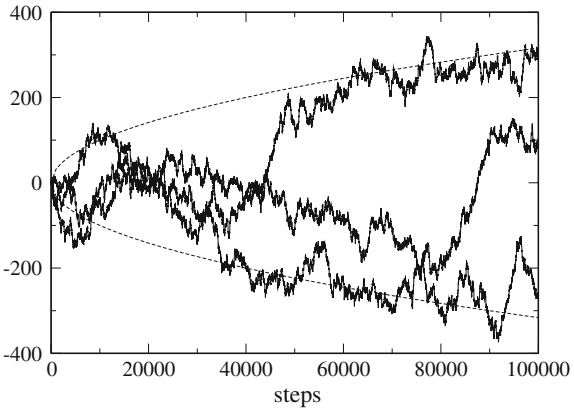
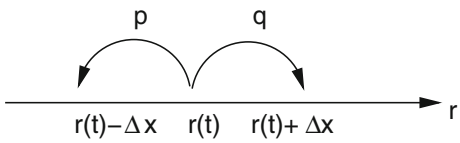


Fig. 17.2 (Random walk with constant step size) The figure shows the position r_n for three different 1-dimensional random walks with step size $\Delta x = \pm 1$. The *dashed curves* show the width $\pm\sigma = \pm\sqrt{n}$ of the Gaussian approximation (17.8)

Fig. 17.3 Random walk with constant step size



which obviously leads to a binomial distribution. From the expansion of

$$(p+q)^n = \sum \binom{n}{m} p^m q^{n-m} \tag{17.13}$$

we see that

$$P_n(n-2m) = \binom{n}{m} p^m q^{n-m} \tag{17.14}$$

or after substitution $m' = n - 2m = -n, -n+2, \ldots n-2, n$:

$$P_n(m') = \binom{n}{(n-m')/2} p^{(n-m')/2} q^{(n+m')/2}. \tag{17.15}$$

Since the steps are uncorrelated we easily find the first two moments

$$\overline{r}_n = \sum_{i=1}^{n} \overline{\Delta x_i} = n\overline{b} = n\Delta x(q-p) \tag{17.16}$$

and

$$\overline{r_n^2} = \overline{\left(\sum_{i=1}^{n} \Delta x_i\right)^2} = \overline{\sum_{i,j=1}^{n} \Delta x_i \Delta x_j} = \sum_{i=1}^{n} \overline{(\Delta x_i)^2} = n\overline{b^2} = n\Delta x^2. \tag{17.17}$$

17.3 The Freely Jointed Chain

We consider a simple statistical model for the conformation of a biopolymer like DNA or a protein.

The polymer is modeled by a 3-dimensional chain consisting of M units with constant bond length and arbitrary relative orientation (Fig. 17.4). The configuration can be described by a point in a $3(M+1)$-dimensional space which is reached after M steps $\Delta \mathbf{r}_i = \mathbf{b}_i$ of a 3-dimensional random walk with constant step size

$$\mathbf{r}_M = \mathbf{r}_0 + \sum_{i=1}^{M} \mathbf{b}_i. \tag{17.18}$$

17.3.1 Basic Statistic Properties

The M bond vectors

$$\mathbf{b}_i = \mathbf{r}_i - \mathbf{r}_{i-1} \tag{17.19}$$

have a fixed length $|\mathbf{b}_i| = b$ and are oriented randomly. The first two moments are

$$\overline{\mathbf{b}_i} = 0 \quad \overline{\mathbf{b}_i^2} = b^2. \tag{17.20}$$

Since different units are independent

$$\overline{\mathbf{b}_i \mathbf{b}_j} = \delta_{i,j} b^2. \tag{17.21}$$

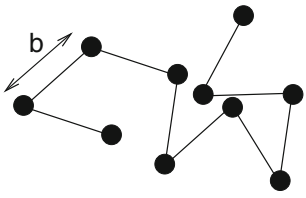


Fig. 17.4 Freely jointed chain with constant bond length b

Obviously the relative position of segment j

$$\mathbf{R}_j = \mathbf{r}_j - \mathbf{r}_0 = \sum_{i=1}^{j} \mathbf{b}_i$$

has zero mean

$$\overline{\mathbf{R}_j} = \sum_{i=1}^{j} \overline{\mathbf{b}_i} = 0 \tag{17.22}$$

and its second moment is

$$\overline{R_j^2} = \overline{\left(\sum_{i=1}^{j} \mathbf{b}_i \sum_{k=1}^{j} \mathbf{b}_k \right)} = \sum_{i,k=1}^{j} \overline{\mathbf{b}_i \mathbf{b}_k} = jb^2. \tag{17.23}$$

For the end to end distance (Fig. 17.5)

$$\mathbf{R}_M = \mathbf{r}_M - \mathbf{r}_0 = \sum_{i=1}^{M} \mathbf{b}_i \tag{17.24}$$

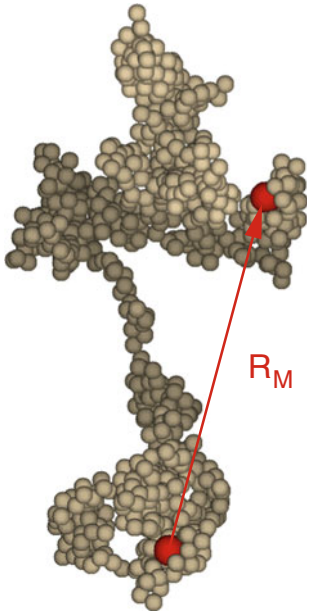


Fig. 17.5 (Freely jointed chain) The figure shows a random 3-dimensional structure with 1000 segments visualized as balls (Molden graphics [220])

this gives

$$\overline{\mathbf{R}}_M = 0, \quad \overline{R_M^2} = Mb^2. \tag{17.25}$$

Let us apply the central limit theorem for large M. For the x coordinate of the end to end vector we have

$$X = \sum_{i=1}^{M} \mathbf{b}_i \mathbf{e}_x = b \sum_i \cos\theta_i. \tag{17.26}$$

With the help of the averages[4]

$$\overline{\cos\theta_i} = \frac{1}{4\pi} \int_0^{2\pi} d\phi \int_0^{\pi} \cos\theta \, \sin\theta d\theta = 0 \tag{17.27}$$

$$\overline{(\cos\theta_i)^2} = \frac{1}{4\pi} \int_0^{2\pi} d\phi \int_0^{\pi} \cos^2\theta \, \sin\theta d\theta = \frac{1}{3} \tag{17.28}$$

we find that the scaled difference

$$\xi_i = \sqrt{3} \cos\theta_i \tag{17.29}$$

has zero mean and unit variance and therefore the sum

$$\widetilde{X} = \frac{\sqrt{3}}{b\sqrt{M}} X = \sqrt{\frac{3}{M}} \sum_{i=1}^{M} \cos\theta_i \tag{17.30}$$

converges to a normal distribution:

$$P(\widetilde{X}) = \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{\widetilde{X}^2}{2} \right\}. \tag{17.31}$$

Hence

$$P(X) = \frac{1}{\sqrt{2\pi}} \frac{\sqrt{3}}{b\sqrt{M}} \exp\left\{ -\frac{3}{2Mb^2} X^2 \right\} \tag{17.32}$$

[4]For a 1-dimensional polymer $\overline{\cos\theta_i} = 0$ and $\overline{(\cos\theta_i)^2} = 1$. In two dimensions $\overline{\cos\theta_i} = \frac{1}{\pi}\int_0^{\pi} \cos\theta \, d\theta = 0$ and $\overline{(\cos\theta_i)^2} = \frac{1}{\pi}\int_0^{\pi} \cos^2\theta \, d\theta = \frac{1}{2}$. To include these cases the factor 3 in the exponent of (17.33) should be replaced by the dimension d.

and finally in 3 dimensions

$$P(\mathbf{R}_M) = P(X)\,P(Y)\,P(Z) = \frac{\sqrt{27}}{b^3\sqrt{(2\pi M)^3}} \exp\left\{-\frac{3}{2Mb^2}\mathbf{R}_M^2\right\}. \tag{17.33}$$

17.3.2 Gyration Tensor

For the center of mass

$$\mathbf{R}_c = \frac{1}{M}\sum_{i=1}^{M}\mathbf{R}_i \tag{17.34}$$

we find

$$\overline{\mathbf{R}_c} = 0 \quad \overline{R_c^2} = \frac{1}{M^2}\sum_{i,j}\overline{\mathbf{R}_i\mathbf{R}_j} \tag{17.35}$$

and since

$$\overline{\mathbf{R}_i\mathbf{R}_j} = \min(i,j)\,b^2 \tag{17.36}$$

we have

$$\overline{R_c^2} = \frac{b^2}{M^2}\left(2\sum_{i=1}^{M} i(M-i+1) - \sum_{i=1}^{M} i\right) = \frac{b^2}{M^2}\left(\frac{M^3}{3} + \frac{M^2}{2} + \frac{M}{6}\right) \approx \frac{Mb^2}{3}. \tag{17.37}$$

The gyration radius [221] is generally defined by

$$R_g^2 = \frac{1}{M}\sum_{i=1}^{M}\overline{(\mathbf{R}_i - \mathbf{R}_c)^2} \tag{17.38}$$

$$= \frac{1}{M}\sum_{i=1}^{M}\left(\overline{R_i^2} + \overline{R_c^2} - 2\frac{1}{M}\sum_{j=1}^{M}\overline{\mathbf{R}_i\mathbf{R}_j}\right) = \frac{1}{M}\sum_{i}\left(\overline{R_i^2}\right) - \overline{R_c^2} \tag{17.39}$$

$$= b^2\frac{M+1}{2} - \frac{b^2}{M^2}\left(\frac{M^3}{3} + \frac{M^2}{2} + \frac{M}{6}\right) = b^2\left(\frac{M}{6} - \frac{1}{6M}\right) \approx \frac{Mb^2}{6}. \tag{17.40}$$

R_g can be also written as

$$R_g^2 = \left(\frac{1}{M} \sum_i \overline{R_i^2} - \frac{1}{M^2} \sum_{ij} \overline{\mathbf{R}_i \mathbf{R}_j} \right) = \frac{1}{2M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \overline{(\mathbf{R}_i - \mathbf{R}_j)^2} \tag{17.41}$$

and can be experimentally measured with the help of scattering phenomena. It is related to the gyration tensor which is defined as

$$\Omega_g = \frac{1}{M} \sum_i \overline{(\mathbf{R}_i - \mathbf{R}_c)(\mathbf{R}_i - \mathbf{R}_c)^T}. \tag{17.42}$$

Its trace is

$$\mathrm{tr}(\Omega_g) = R_g^2 \tag{17.43}$$

and its eigenvalues give us information about the shape of the polymer (Fig. 17.6).

17.3.3 Hookean Spring Model

Simulation of the dynamics of the freely jointed chain is complicated by the constraints which are implied by the constant chain length. Much simpler is the

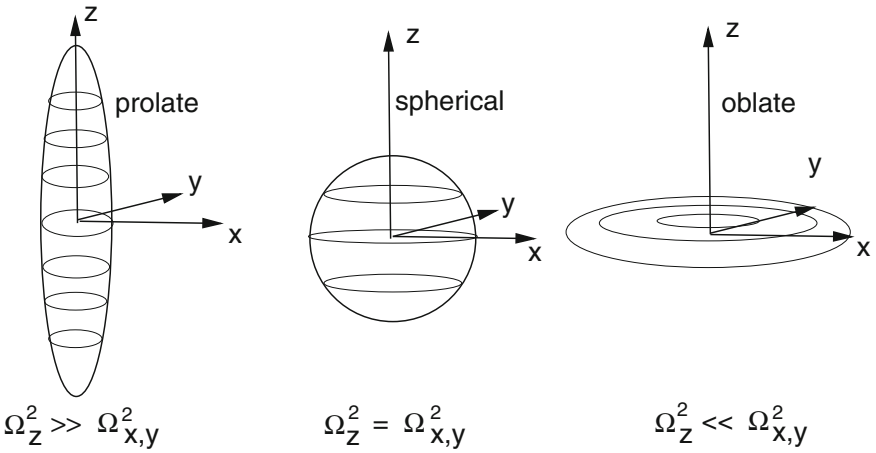


Fig. 17.6 (Gyration tensor) The eigenvalues of the gyration tensor give information on the shape of the polymer. If the extension is larger (smaller) along one direction than in the perpendicular plane, one eigenvalue is larger (smaller) than the two other

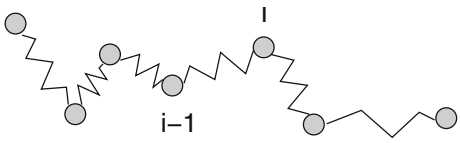


Fig. 17.7 Polymer model with Hookean springs

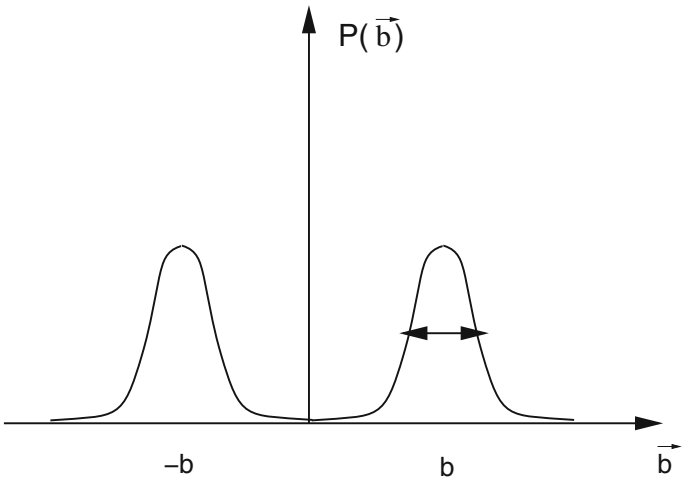


Fig. 17.8 (Distribution of bond vectors) The bond vector distribution for a 1-dimensional chain of springs has maxima at $\pm b$. For large force constants the width of the two peaks becomes small and the chain of springs resembles a freely jointed chain with constant bond length

simulation of a model which treats the segments as Hookean springs (Fig. 17.7). In the limit of a large force constant the two models give equivalent results.

We assume that the segments are independent (self crossing is not avoided). Then for one segment the energy contribution is

$$E_i = \frac{f}{2} \left(|\mathbf{b}_i| - b\right)^2 . \tag{17.44}$$

If the fluctuations are small

$$\overline{||\mathbf{b}_i| - b|} \ll b \tag{17.45}$$

then (Fig. 17.8)

$$\overline{|\mathbf{b}_i|} \approx b \quad \overline{\mathbf{b}_i^2} \approx b^2 \tag{17.46}$$

and the freely jointed chain model (17.33) gives the entropy as a function of the end to end vector

$$S = -k_B \ln\left(P(\mathbf{R}_M)\right) = -k_B \ln\left(\frac{\sqrt{27}}{b^3\sqrt{(2\pi M)^3}}\right) + \frac{3k_B}{2Mb^2}{\mathbf{R}_M}^2. \tag{17.47}$$

If one end of the polymer is fixed at $\mathbf{r}_0 = 0$ and a force $\boldsymbol{\kappa}$ is applied to the other end, the free energy is given by

$$F = TS - \boldsymbol{\kappa}\mathbf{R}_M = \frac{3k_BT}{2Mb^2}\mathbf{R}_M^2 - \boldsymbol{\kappa}\mathbf{R}_M + \text{const.} \tag{17.48}$$

In thermodynamic equilibrium the free energy is minimal, hence the average extension is

$$\overline{\mathbf{R}_M} = \frac{Mb^2}{3k_BT}\boldsymbol{\kappa}. \tag{17.49}$$

This linear behavior is similar to a Hookean spring with an effective force constant

$$f_{eff} = \frac{Mb^2}{3k_BT} \tag{17.50}$$

and is only valid for small forces. For large forces the freely jointed chain asymptotically reaches its maximum length of $R_{M,\max} = Mb$, whereas for the chain of springs $R_M \to M(b + \kappa/f)$.

17.4 Langevin Dynamics

A heavy particle moving in a bath of much smaller and lighter particles (for instance atoms and molecules of the air) shows what is known as Brownian motion [222–224]. Due to collisions with the thermally moving bath particles it experiences a fluctuating force which drives the particle into a random motion. The French physicist Paul Langevin developed a model to describe this motion without including the light particles explicitly. The fluctuating force is divided into a macroscopic friction force proportional to the velocity

$$\mathbf{F}_{fr} = -\gamma\mathbf{v} \tag{17.51}$$

and a randomly fluctuating force with zero mean and infinitely short correlation time

$$\overline{\mathbf{F}_{rand}(t)} = 0 \quad \overline{\mathbf{F}_{rand}(t)\mathbf{F}_{rand}(t')} = \overline{\mathbf{F}_{rand}^2}\delta(t - t'). \tag{17.52}$$

The equations of motion for the heavy particle are

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} &= \mathbf{v}\\ \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{v} &= -\gamma\mathbf{v} + \frac{1}{m}\mathbf{F}_{fr}(t) - \frac{1}{m}\nabla U(\mathbf{x})\end{aligned} \tag{17.53}$$

with the macroscopic friction coefficient γ and the potential $U(\mathbf{x})$.

The behavior of the random force can be better understood if we introduce a time grid $t_{n+1} - t_n = \Delta t$ and take the limit $\Delta t \to 0$. We assume that the random force has a constant value during each interval

$$\mathbf{F}_{rand}(t) = \mathbf{F}_n \quad t_n \le t < t_{n+1} \tag{17.54}$$

and that the values at different intervals are uncorrelated

$$\overline{\mathbf{F}_n \mathbf{F}_m} = \delta_{m,n}\overline{\mathbf{F}_n^2}. \tag{17.55}$$

The auto-correlation function then is given by

$$\overline{\mathbf{F}_{rand}(t)\mathbf{F}_{rand}(t')} = \begin{cases} 0 & \text{different intervals} \\ \overline{\mathbf{F}_n^2} & \text{same interval.} \end{cases} \tag{17.56}$$

Division by Δt gives a sequence of functions which converges to a delta function in the limit $\Delta t \to 0$

$$\frac{1}{\Delta t}\overline{\mathbf{F}_{rand}(t)\mathbf{F}_{rand}(t')} \to \overline{\mathbf{F}_n^2}\,\delta(t - t'). \tag{17.57}$$

Hence we find

$$\overline{\mathbf{F}_n^2} = \frac{1}{\Delta t}\overline{\mathbf{F}_{rand}^2}. \tag{17.58}$$

Within a short time interval $\Delta t \to 0$ the velocity changes by

$$\mathbf{v}(t_n + \Delta t) = \mathbf{v} - \gamma\mathbf{v}\Delta t - \frac{1}{m}\nabla U(\mathbf{x})\Delta t + \frac{1}{m}\mathbf{F}_n \Delta t + \cdots \tag{17.59}$$

and taking the square gives

$$\mathbf{v}^2(t_n + \Delta t) = \mathbf{v}^2 - 2\gamma\mathbf{v}^2\Delta t - \frac{2}{m}\mathbf{v}\nabla U(\mathbf{x})\Delta t + \frac{2}{m}\mathbf{v}\mathbf{F}_n\Delta t + \frac{\mathbf{F}_n^2}{m^2}(\Delta t)^2 + \cdots. \tag{17.60}$$

Hence for the total energy

$$E(t_n + \Delta t) = \frac{m}{2}\mathbf{v}^2(t_n + \Delta t) + U(\mathbf{x}(t_n + \Delta t))$$
$$= \frac{m}{2}\mathbf{v}^2(t_n + \Delta t) + U(\mathbf{x}) + \mathbf{v}\nabla U(\mathbf{x})\Delta t + \cdots \tag{17.61}$$

we have

$$E(t_n + \Delta t) = E(t_n) - m\gamma\mathbf{v}^2\Delta t + \mathbf{v}\mathbf{F}_n\Delta t + \frac{\mathbf{F}_n^2}{2m}(\Delta t)^2 + \cdots . \tag{17.62}$$

On the average the total energy $\overline{E}$ should be constant and furthermore in d dimensions

$$\frac{m}{2}\overline{\mathbf{v}^2} = \frac{d}{2}k_B T. \tag{17.63}$$

Therefore we conclude

$$m\gamma\overline{\mathbf{v}^2} = \frac{\Delta t}{2m}\overline{\mathbf{F}_n^2} = \frac{1}{2m}\overline{\mathbf{F}_{rand}^2} \tag{17.64}$$

from which we obtain finally

$$\overline{\mathbf{F}_n^2} = \frac{2m\gamma d}{\Delta t}\, k_B T. \tag{17.65}$$

Problems

Problem 17.1 Random Walk in One Dimension

This program generates random walks with (a) fixed step length $\Delta x = \pm 1$ or (b) step length equally distributed over the interval $-\sqrt{3} < \Delta x < \sqrt{3}$. It also shows the variance, which for large number of walks approaches $\sigma = \sqrt{n}$. See also Fig. 17.2

Problem 17.2 Gyration Tensor

The program calculates random walks with M steps of length b. The bond vectors are generated from M random points $\mathbf{e}_i$ on the unit sphere as $\mathbf{b}_i = b\mathbf{e}_i$. End to end distance, center of gravity and gyration radius are calculated and can be averaged over numerous random structures. The gyration tensor (Sect. 17.3.2) is diagonalized and the ordered eigenvalues are averaged.

Problem 17.3 Brownian Motion in a Harmonic Potential

The program simulates a particle in a 1-dimensional harmonic potential

$$U(\mathbf{x}) = \frac{f}{2}x^2 - \kappa x \tag{17.66}$$

where κ is an external force. We use the improved Euler method (13.36). First the coordinate and the velocity at mid time are estimated

$$\mathbf{x}\left(t_n + \frac{\Delta t}{2}\right) = \mathbf{x}(t_n) + \mathbf{v}(t_n)\frac{\Delta t}{2} \tag{17.67}$$

$$\mathbf{v}\left(t_n + \frac{\Delta t}{2}\right) = \mathbf{v}(t_n) - \gamma\mathbf{v}(t_n)\frac{\Delta t}{2} + \frac{\mathbf{F}_n}{m}\frac{\Delta t}{2} - \frac{f}{m}\mathbf{x}(t_n)\frac{\Delta t}{2} \tag{17.68}$$

where $\mathbf{F}_n$ is a random number obeying (17.65). Then the values at t_{n+1} are calculated as

$$\mathbf{x}(t_n + \Delta t) = \mathbf{x}(t_n) + \mathbf{v}\left(t_n + \frac{\Delta t}{2}\right)\Delta t \tag{17.69}$$

$$\mathbf{v}(t_n + \Delta t) = \mathbf{v}(t_n) - \gamma\mathbf{v}\left(t_n + \frac{\Delta t}{2}\right)\Delta t + \frac{\mathbf{F}_n}{m}\Delta t - \frac{f}{m}\mathbf{x}\left(t_n + \frac{\Delta t}{2}\right)\Delta t. \tag{17.70}$$

Problem 17.4 Force Extension Relation

The program simulates a chain of springs Sect. 17.3.3 with potential energy

$$U = \frac{f}{2}\sum(|\mathbf{b}_i| - b)^2 - \boldsymbol{\kappa}\mathbf{R}_M. \tag{17.71}$$

The force can be varied and the extension along the force direction is averaged over numerous time steps.

Chapter 18
Electrostatics

The electrostatic potential $\Phi(\mathbf{r})$ of a charge distribution $\rho(\mathbf{r})$ is a solution[1] *of Poisson's equation*

$$\triangle \Phi(\mathbf{r}) = -\rho(\mathbf{r}) \tag{18.1}$$

which, for spatially varying dielectric constant $\varepsilon(\mathbf{r})$ becomes

$$div(\varepsilon(\mathbf{r})\ \ grad\ \Phi(\mathbf{r})) = -\rho(\mathbf{r}) \tag{18.2}$$

and, if mobile charges are taken into account, like for an electrolyte or semiconductor, turns into the Poisson–Boltzmann equation

$$div(\varepsilon(\mathbf{r})\ \ grad\ \Phi(\mathbf{r})) = -\rho_{fix}(\mathbf{r}) - \sum_i n_i^0 Z_i e\, \mathrm{e}^{-Z_i e\Phi(\mathbf{r})/k_B T}. \tag{18.3}$$

In this chapter we discretize the Poisson and the linearized Poisson–Boltzmann equation by finite volume methods which are applicable even in case of discontinuous ε. We solve the discretized equations iteratively with the method of successive over-relaxation. The solvation energy of a charged sphere in a dielectric medium is calculated to compare the accuracy of several methods. This can be studied also in a computer experiment.

Since the Green's function is analytically available for the Poisson and Poisson–Boltzmann equations, alternatively the method of boundary elements can be applied, which can reduce the computer time, for instance for solvation models. A computer experiment simulates a point charge within a spherical cavity and calculates the solvation energy with the boundary element method.

[1] *The solution depends on the boundary conditions, which in the simplest case are given by* $\lim_{|\mathbf{r}|\to\infty} \Phi(\mathbf{r}) = 0$.

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_18

18.1 Poisson Equation

From a combination of the basic equations of electrostatics

$$\operatorname{div} D(\mathbf{r}) = \rho(\mathbf{r}) \tag{18.4}$$

$$D(\mathbf{r}) = \varepsilon(\mathbf{r}) E(\mathbf{r}) \tag{18.5}$$

$$E(\mathbf{r}) = -\operatorname{grad} \Phi(\mathbf{r}) \tag{18.6}$$

the generalized Poisson equation is obtained

$$\operatorname{div}(\varepsilon(\mathbf{r}) \operatorname{grad}\Phi(\mathbf{r})) = -\rho(\mathbf{r}) \tag{18.7}$$

which can be written in integral form with the help of Gauss' theorem

$$\oint_{\partial V} d\mathbf{A} \operatorname{div}(\varepsilon(\mathbf{r}) \operatorname{grad}\Phi(\mathbf{r})) = \int_V dV\, \varepsilon(\mathbf{r}) \operatorname{grad}\Phi(\mathbf{r})) = -\int_V dV\, \rho(\mathbf{r}). \tag{18.8}$$

If $\varepsilon(\mathbf{r})$ is continuously differentiable, the product rule for differentiation gives

$$\varepsilon(\mathbf{r})\, \triangle\Phi(\mathbf{r}) + (\operatorname{grad} \varepsilon(\mathbf{r}))\, (\operatorname{grad}\, \Phi(\mathbf{r})) = -\rho(\mathbf{r}) \tag{18.9}$$

which for constant ε simplifies to the Poisson equation

$$\Delta\Phi(\mathbf{r}) = -\frac{\rho(\mathbf{r})}{\varepsilon}. \tag{18.10}$$

18.1.1 Homogeneous Dielectric Medium

We begin with the simplest case of a dielectric medium with constant ε and solve (18.10) numerically. We use a finite volume method (Sect. 12.3) which corresponds to a finite element method with piecewise constant test functions. The integration volume is divided into small cubes V_{ijk} which are centered at the grid points (Fig. 18.1)

$$r_{ijk} = (hi, hj, hk). \tag{18.11}$$

Integration of (18.10) over the control volume V_{ijk} around $\mathbf{r}_{ijk}$ gives

$$\int_V dV \operatorname{div} \operatorname{grad}\, \Phi = \oint_{\partial V} \operatorname{grad}\, \Phi \mathbf{dA} = -\frac{1}{\varepsilon} \int_V dV\, \rho(\mathbf{r}) = -\frac{Q_{ijk}}{\varepsilon}. \tag{18.12}$$

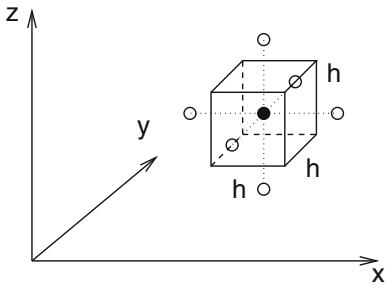


Fig. 18.1 (Finite volume for the Poisson equation) The control volume is a small cube centered at a grid point (*full circle*)

Q_{ijk} is the total charge in the control volume. The flux integral is approximated by (12.85)

$$\oint_{\partial V} \text{grad}\, \Phi \, \mathbf{dA} = -h^2 \left(\frac{\partial \Phi}{\partial x}(x_{i+1/2}, y_j, z_k) - \frac{\partial \Phi}{\partial x}(x_{i-1/2}, y_j, z_k) \right.$$
$$\left. + \frac{\partial \Phi}{\partial y}(x_i y_{j+1/2}, z_k) - \frac{\partial \Phi}{\partial y}(x_i, y_{j-1/2}, z_k) + \frac{\partial \Phi}{\partial z}(x_i, y_j, z_{k+1/2}) - \frac{\partial \Phi}{\partial z}(x_i, y_j, z_{k-1/2}) \right). \tag{18.13}$$

The derivatives are approximated by symmetric differences

$$\begin{aligned}
\oint_{\partial V} \text{grad}\, \Phi \, \mathbf{dA} &= -h \big\{ \big(\Phi(x_{i+1}, y_j, z_k) - \Phi(x_i, y_j, z_k) \big) \\
&- \big(\Phi(x_i, y_j, z_k) - \Phi(x_{i-1}, y_j, z_k) \big) \\
&+ \big(\Phi(x_i, y_{j+1}, z_k) - \Phi(x_i, y_j, z_k) \big) \\
&- \big(\Phi(x_i, y_j, z_k) - \Phi(x_i, y_{j-1}, z_k) \big) \\
&+ \big(\Phi(x_i, y_j, z_{k+1}) - \Phi(x_i, y_j, z_k) \big) \\
&- \big(\Phi(x_i, y_j, z_k) - \Phi(x_i, y_j, z_{k-1}) \big) \big\} \\
&= -h \big(\Phi(x_{i-1}, y_j, z_k) + \Phi(x_{i+1}, y_j, z_k) + \Phi(x_i, y_{j-1}, z_k) + \Phi(x_i, y_{j+1}, z_k) \\
&+ \Phi(x_i, y_j, z_{k-1}) + \Phi(x_i, y_j, z_{k+1}) - 6\Phi(x_i, y_j, z_k) \big)
\end{aligned} \tag{18.14}$$

which coincides with the simplest discretization of the second derivatives (3.40). Finally we obtain the discretized Poisson equation in the more compact form

$$\sum_{s=1}^{6} (\Phi(r_{ijk} + d\mathbf{r}_s) - \Phi(r_{ijk})) = -\frac{Q_{ijk}}{\varepsilon h} \tag{18.15}$$

which involves an average over the 6 neighboring cells

$$d\mathbf{r}_1 = (-h, 0, 0) \ \ldots d\mathbf{r}_6 = (0, 0, h). \tag{18.16}$$

18.1.2 Numerical Methods for the Poisson Equation

Equation (18.15) is a system of linear equations with very large dimension (for a grid with $100 \times 100 \times 100$ points the dimension of the matrix is $10^6 \times 10^6$!). Our computer experiments use the iterative method (Sect. 5.5)

$$\Phi^{new}(r_{ijk}) = \frac{1}{6}\left(\sum_s \Phi^{old}(\mathbf{r}_{ijk} + d\mathbf{r}_s) + \frac{Q_{ijk}}{\varepsilon h}\right). \tag{18.17}$$

Jacobi's method (5.121 on p. 80) makes all the changes in one step whereas the Gauss–Seidel method (5.124 on p. 80) makes one change after the other. The chessboard (or black red method) divides the grid into two subgrids (with $i+j+k$ even or odd) which are treated subsequently. The vector $d\mathbf{r}_s$ connects points of different subgrids. Therefore it is not necessary to store intermediate values like for the Gauss–Seidel method.

Convergence can be improved with the method of successive over-relaxation (SOR, 5.128 on p. 81) using a mixture of old and new values

$$\Phi^{new}(r_{ijk}) = (1-\omega)\Phi^{old}(\mathbf{r}_{ijk}) + \omega\frac{1}{6}\left(\sum \Phi^{old}(\mathbf{r}_{ijk} + d\mathbf{r}_s) + \frac{Q_{ijk}}{\varepsilon h}\right) \tag{18.18}$$

with the relaxation parameter ω. For $1 < \omega < 2$ convergence is faster than for $\omega = 1$. The optimum choice of ω for the Poisson problem in any dimension is discussed in [225].

Convergence can be further improved by multigrid methods [226, 227]. Error components with short wavelengths are strongly damped during a few iterations whereas it takes a very large number of iterations to remove the long wavelength components. But here a coarser grid is sufficient and reduces computing time. After a few iterations a first approximation Φ_1 is obtained with the finite residual

$$r_1 = \Delta\Phi_1 + \frac{1}{\varepsilon}\rho. \tag{18.19}$$

Then more iterations on a coarser grid are made to find an approximate solution Φ_2 of the equation

$$\Delta\Phi = -r_1 = -\frac{1}{\varepsilon}\rho - \Delta\Phi_1. \tag{18.20}$$

The new residual is

$$r_2 = \Delta\Phi_2 + r_1. \tag{18.21}$$

Function values of Φ_2 on the finer grid are obtained by interpolation and finally the sum $\Phi_1 + \Phi_2$ provides an improved approximation to the solution since

$$\Delta(\Phi_1 + \Phi_2) = -\frac{1}{\varepsilon}\rho + r_1 + (r_2 - r_1) = -\frac{1}{\varepsilon}\rho + r_2. \tag{18.22}$$

This method can be extended to a hierarchy of many grids.

Alternatively, the Poisson equation can be solved non-iteratively with pseudospectral methods [228, 229]. For instance, if the boundary is the surface of a cube, eigenfunctions of the Laplacian are for homogeneous boundary conditions ($\Phi = 0$) given by

$$N_{\mathbf{k}}(\mathbf{r}) = \sin(k_x x)\sin(k_y y)\sin(k_z z) \tag{18.23}$$

and for no-flow boundary conditions ($\frac{\partial}{\partial n}\Phi = 0$) by

$$N_{\mathbf{k}}(\mathbf{r}) = \cos(k_x x)\cos(k_y y)\cos(k_z z) \tag{18.24}$$

which can be used as expansion functions for the potential

$$\Phi(\mathbf{r}) = \sum_{k_x,k_y,k_z} \Phi_{\mathbf{k}} N_{\mathbf{k}}(\mathbf{r}). \tag{18.25}$$

Introducing collocation points $\mathbf{r}_j$ the condition on the residual becomes

$$0 = \triangle\Phi(\mathbf{r}_j) + \frac{1}{\varepsilon}\rho(\mathbf{r}_j) = \sum_{k_x,k_y,k_z} k^2 \Phi_{\mathbf{k}} N_{\mathbf{k}}(\mathbf{r}_j) + \frac{1}{\varepsilon}\rho(\mathbf{r}_j) \tag{18.26}$$

which can be inverted with an inverse discrete sine transformation, (respectively an inverse discrete cosine transformation for no-flux boundary conditions) to obtain the Fourier components of the potential. Another discrete sine (or cosine) transformation gives the potential in real space.

18.1.3 Charged Sphere

As a simple example we consider a sphere of radius R with a homogeneous charge density of

$$\rho_0 = e \cdot \frac{3}{4\pi R^3}. \tag{18.27}$$

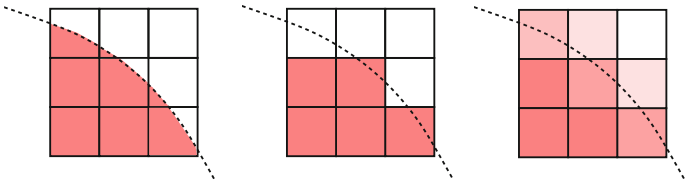


Fig. 18.2 (Discretization of the discontinuous charge density) ***Left*** the most precise method divides the control volumes at the boundary into two irregularly shaped parts. ***Middle*** assigning either the value ρ_0 or zero retains the discontinuity but changes the shape of the boundary. ***Right*** averaging over a control volume smears out the discontinuous transition

The exact potential is given by

$$\Phi(r) = \frac{e}{4\pi\varepsilon_0 R} + \frac{e}{8\pi\varepsilon_0 R}\left(1 - \frac{r^2}{R^2}\right) \quad \text{for } r < R$$
$$\Phi(r) = \frac{e}{4\pi\epsilon_0 r} \quad \text{for } r > R. \tag{18.28}$$

The charge density (18.27) is discontinuous at the surface of the sphere. Integration over a control volume smears out this discontinuity which affects the potential values around the boundary (Fig. 18.2). Alternatively we could assign the value $\rho(\mathbf{r}_{ijk})$ which is either ρ_0 (18.27) or zero to each control volume which retains a sharp transition but changes the shape of the boundary surface and does not conserve the total charge. This approach was discussed in the first edition of this book in connection with a finite differences method. The most precise but also complicated method divides the control volumes at the boundary into two irregularly shaped parts [230, 231].

Initial guess as well as boundary values are taken from

$$\Phi_0(r) = \frac{e}{4\pi\varepsilon_0 \max(r, h)} \tag{18.29}$$

which provides proper boundary values but is far from the final solution inside the sphere. The interaction energy is given by (Sect. 18.5)

$$E_{int} = \frac{1}{2}\int_V \rho(\mathbf{r})\Phi(\mathbf{r})dV = \frac{3}{20}\frac{e^2}{\pi\varepsilon_0 R}\,. \tag{18.30}$$

Calculated potential (Fig. 18.3) and interaction energy (Figs. 18.4, 18.5) converge rapidly. The optimum relaxation parameter is around $\omega \approx 1.9$.

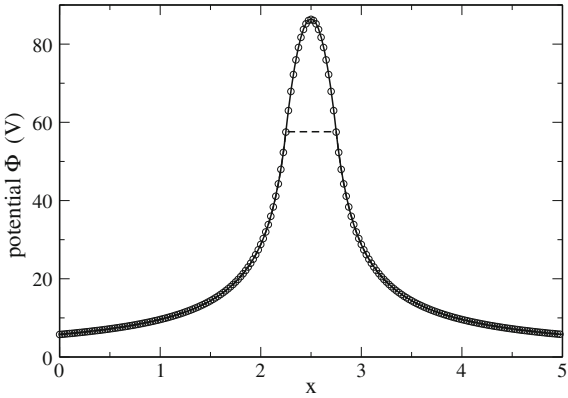


Fig. 18.3 (Electrostatic potential of a charged sphere) A charged sphere is simulated with radius $R = 0.25$ and a homogeneous charge density $\rho = e \cdot 3/4\pi R^3$. The grid consists of 200^3 points with a spacing of $h = 0.025$. The calculated potential (*circles*) is compared to the exact solution (18.28, *solid curve*), the initial guess is shown by the *dashed line*

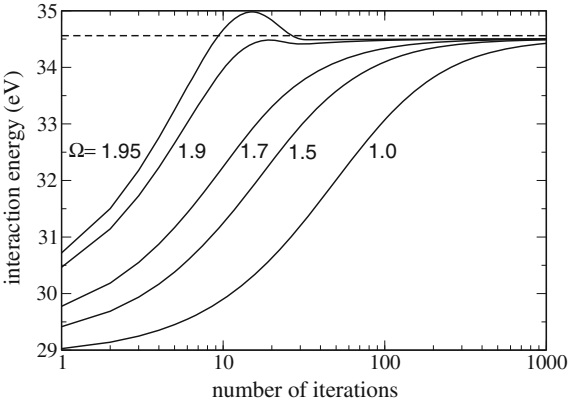


Fig. 18.4 (Influence of the relaxation parameter) The convergence of the interaction energy (18.30, which has a value of 34.56 eV for this example) is studied as a function of the relaxation parameter ω. The optimum value is around $\omega \approx 1.9$. For $\omega > 2$ there is no convergence. The *dashed line* shows the exact value

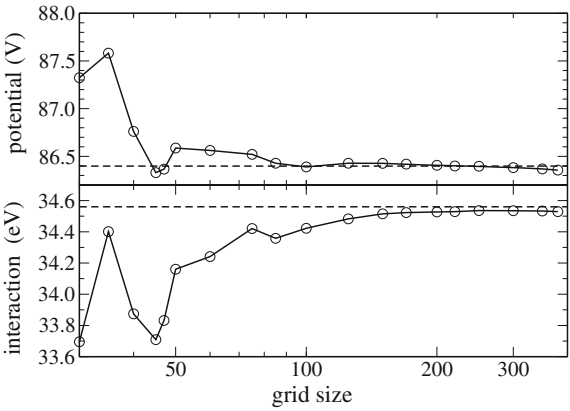


Fig. 18.5 (Influence of grid size) The convergence of the interaction energy (18.30) and the central potential value are studied as a function of grid size. The *dashed lines* show the exact values

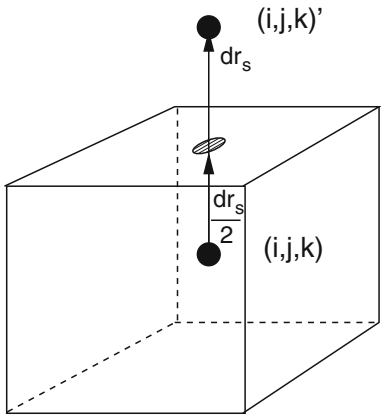


Fig. 18.6 Face center of the control volume

18.1.4 Variable ε

In the framework of the finite volume method we take the average over a control volume to discretize ε[2] and Φ

$$\varepsilon_{ijk} = \overline{\varepsilon}(\mathbf{r}_{ijk}) = \frac{1}{h^3}\int_{V_{ijk}} dV\, \varepsilon(\mathbf{r}) \tag{18.31}$$

$$\Phi_{ijk} = \overline{\Phi}(\mathbf{r}_{ijk}) = \frac{1}{h^3}\int_{V_{ijk}} dV\, \Phi(\mathbf{r}). \tag{18.32}$$

Integration of (18.7) gives

$$\int_V dV\ \mathrm{div}\ (\varepsilon(\mathbf{r})\ \mathrm{grad}\ \Phi(\mathbf{r})) = \oint_{\partial V} \varepsilon(\mathbf{r})\ \mathrm{grad}\ \Phi \mathbf{dA} = -\int_V dV\, \rho(\mathbf{r}) = -Q_{ijk}. \tag{18.33}$$

The surface integral is

$$\oint_{\partial V} \mathbf{dA}\, \varepsilon\, \mathrm{grad}\Phi = \sum_{s\in faces} \int_{A_s} dA\, \varepsilon(\mathbf{r})\, \frac{\partial}{\partial n}\Phi. \tag{18.34}$$

Applying the midpoint rule (12.77) we find (Fig. 18.6)

$$\oint_{\partial V} \mathbf{dA}\, \varepsilon\, \mathrm{grad}\Phi \approx h^2 \sum_{r=1}^{6} \varepsilon\left(\mathbf{r}_{ijk} + \frac{1}{2}d\mathbf{r}_s\right) \frac{\partial}{\partial n}\Phi\left(\mathbf{r}_{ijk} + \frac{1}{2}d\mathbf{r}_s\right). \tag{18.35}$$

[2] But see Sect. 18.1.5 for the case of discontinuous ε.

The potential Φ as well as the product $\varepsilon(\mathbf{r})\frac{\partial\Phi}{\partial n}$ are continuous, therefore we make the approximation [230]

$$\varepsilon\left(\mathbf{r}_{ijk}+\frac{1}{2}d\mathbf{r}_s\right)\frac{\partial\Phi}{\partial n}\left(\mathbf{r}_{ijk}+\frac{1}{2}d\mathbf{r}_s\right)=\overline{\varepsilon}(\mathbf{r}_{ijk})\frac{\overline{\Phi}\left(\mathbf{r}_{ijk}+\frac{1}{2}d\mathbf{r}_s\right)-\overline{\Phi}(\mathbf{r}_{ijk})}{\frac{h}{2}}$$
$$=\overline{\varepsilon}(\mathbf{r}_{ijk}+d\mathbf{r}_s)\frac{\overline{\Phi}(\mathbf{r}_{ijk}+d\mathbf{r}_s)-\overline{\Phi}\left(\mathbf{r}_{ijk}+\frac{1}{2}d\mathbf{r}_s\right)}{\frac{h}{2}}. \tag{18.36}$$

From this equation the unknown potential value on the face of the control volume $\overline{\Phi}(\mathbf{r}_{ijk}+\frac{1}{2}d\mathbf{r}_s)$ (Fig. 18.6) can be calculated

$$\overline{\Phi}\left(\mathbf{r}_{ijk}+\frac{1}{2}d\mathbf{r}_s\right)=\frac{\overline{\varepsilon}(\mathbf{r}_{ijk})\overline{\Phi}(\mathbf{r}_{ijk})+\overline{\varepsilon}(\mathbf{r}_{ijk}+d\mathbf{r}_s)\overline{\Phi}(\mathbf{r}_{ijk}+d\mathbf{r}_s)}{\overline{\varepsilon}(\mathbf{r}_{ijk})+\overline{\varepsilon}(\mathbf{r}_{ijk}+d\mathbf{r}_s)} \tag{18.37}$$

which gives

$$\varepsilon\left(\mathbf{r}_{ijk}+\frac{1}{2}d\mathbf{r}_s\right)\frac{\partial}{\partial n}\Phi\left(\mathbf{r}_{ijk}+\frac{1}{2}d\mathbf{r}_s\right)=\frac{2\overline{\varepsilon}(\mathbf{r}_{ijk})\overline{\varepsilon}(\mathbf{r}_{ijk}+d\mathbf{r}_s)}{\overline{\varepsilon}(\mathbf{r}_{ijk})+\overline{\varepsilon}(\mathbf{r}_{ijk}+d\mathbf{r}_s)}\ \frac{\overline{\Phi}(r_{ijk}+dr_s)-\overline{\Phi}(r_{ijk})}{h}. \tag{18.38}$$

Finally we obtain the discretized equation

$$-Q_{ijk}=h\sum_{s=1}^{6}\frac{2\overline{\varepsilon}(\mathbf{r}_{ijk}+d\mathbf{r}_s)\overline{\varepsilon}(\mathbf{r}_{ijk})}{\overline{\varepsilon}(\mathbf{r}_{ijk}+d\mathbf{r}_s)+\overline{\varepsilon}(\mathbf{r}_{ijk})}(\overline{\Phi}(\mathbf{r}_{ijk}+d\mathbf{r}_s)-\overline{\Phi}(\mathbf{r}_{ijk})) \tag{18.39}$$

which can be solved iteratively according to

$$\Phi^{new}(\mathbf{r}_{ijk})=\frac{\sum\frac{2\varepsilon(\mathbf{r}_{ijk}+d\mathbf{r}_s)\varepsilon(\mathbf{r}_{ijk})}{\varepsilon(\mathbf{r}_{ijk}+d\mathbf{r}_s)+\varepsilon(\mathbf{r}_{ijk})}\Phi^{old}(\mathbf{r}_{ijk}+d\mathbf{r}_s)+\frac{Q_{ijk}}{h}}{\sum\frac{2\varepsilon(\mathbf{r}_{ijk}+d\mathbf{r}_s)\varepsilon(\mathbf{r}_{ijk})}{\varepsilon(\mathbf{r}_{ijk}+d\mathbf{r}_s)+\varepsilon(\mathbf{r}_{ijk})}}. \tag{18.40}$$

18.1.5 Discontinuous ε

For practical applications models are often used with piecewise constant ε. A simple example is the solvation of a charged molecule in a dielectric medium (Fig. 18.9). Here $\varepsilon=\varepsilon_0$ within the molecule and $\varepsilon=\varepsilon_0\varepsilon_1$ within the medium. At the boundary ε is discontinuous. In (18.40) the discontinuity is replaced by a smooth transition between the two values of ε (Fig. 18.7).

If the discontinuity of ε is inside a control volume V_{ijk} then (18.31) takes the arithmetic average

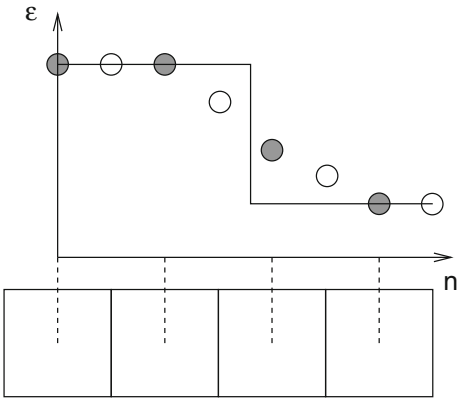


Fig. 18.7 (Transition of ε) The discontinuous $\varepsilon(r)$ (*black line*) is averaged over the control volumes to obtain the discretized values ε_{ijk} (*full circles*). Equation (18.40) takes the harmonic average over two neighbor cells (*open circles*) and replaces the discontinuity by a smooth transition over a distance of about h

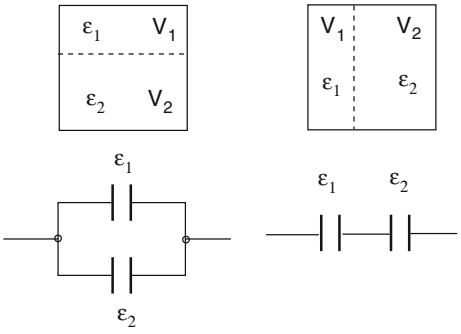


Fig. 18.8 Average of ε over a control volume

$$\overline{\varepsilon}_{ijk} = V_{ijk}^{(1)}\varepsilon_1 + V_{ijk}^{(2)}\varepsilon_2 \tag{18.41}$$

which corresponds to the parallel connection of two capacities (Fig. 18.8). Depending on geometry, a serial connection may be more appropriate which corresponds to the weighted harmonic average

$$\overline{\varepsilon}_{ijk} = \frac{1}{V_{ijk}^{(1)}\varepsilon_1^{-1} + V_{ijk}^{(2)}\varepsilon_2^{-1}}. \tag{18.42}$$

18.1.6 Solvation Energy of a Charged Sphere

We consider again a charged sphere, which is now embedded in a dielectric medium (Fig. 18.9) with relative dielectric constant ε_1.

For a spherically symmetrical problem (18.7) can be solved by application of Gauss's theorem

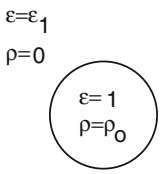


Fig. 18.9 (Solvation of a charged sphere in a dielectric medium) Charge density and dielectric constant are discontinuous at the surface of the sphere

$$4\pi r^2 \varepsilon(r) \frac{\mathrm{d}\Phi}{\mathrm{d}r} = -4\pi \int_0^r \rho(r') r'^2 dr' = -q(r) \tag{18.43}$$

$$\Phi(r) = -\int_0^r \frac{q(r)}{4\pi r^2 \varepsilon(r)} + \Phi(0). \tag{18.44}$$

For the charged sphere we find

$$q(r) = \begin{cases} Qr^3/R^3 \text{ for } r < R \\ Q \text{ for } r > R \end{cases} \tag{18.45}$$

$$\Phi(r) = -\frac{Q}{4\pi\varepsilon_0 R^3} \frac{r^2}{2} + \Phi(0) \quad \text{for } r < R \tag{18.46}$$

$$\Phi(r) = -\frac{Q}{8\pi\varepsilon_0 R} + \Phi(0) + \frac{Q}{4\pi\varepsilon_0\varepsilon_1} \left(\frac{1}{r} - \frac{1}{R} \right) \quad \text{for } r > R. \tag{18.47}$$

The constant $\Phi(0)$ is chosen to give vanishing potential at infinity

$$\Phi(0) = \frac{Q}{4\pi\varepsilon_0\varepsilon_1 R} + \frac{Q}{8\pi\varepsilon_0 R}. \tag{18.48}$$

The interaction energy is

$$E_{int} = \frac{1}{2} \int_0^R 4\pi r^2 dr \, \rho\Phi(r) = \frac{Q^2(5+\varepsilon_1)}{40\pi\varepsilon_0\varepsilon_1 R}. \tag{18.49}$$

Numerical results for $\varepsilon_1 = 4$ are shown in Fig. 18.10.

18.1.7 The Shifted Grid Method

An alternative approach uses a different grid for ε which is shifted by $h/2$ in all directions (Fig. 18.11) [232] or, more generally, a dual grid (12.74).

$$\varepsilon_{ijk} = \overline{\varepsilon}(\mathbf{r}_{i+1/2,,j+1/2,k+1/2}). \tag{18.50}$$

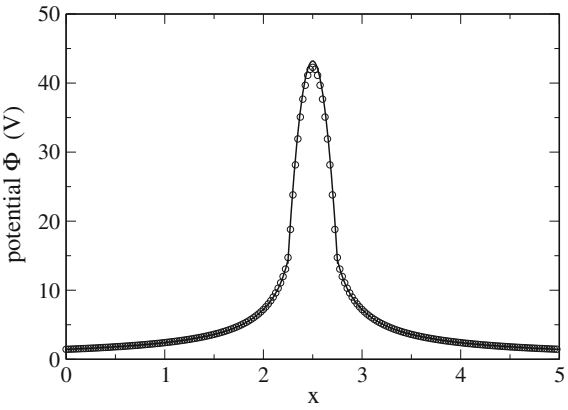


Fig. 18.10 (Charged sphere in a dielectric medium) Numerical results for $\varepsilon_1 = 4$ outside the sphere and 200^3 grid points (*circles*) are compared to the exact solution (18.46,18.47, *solid curves*)

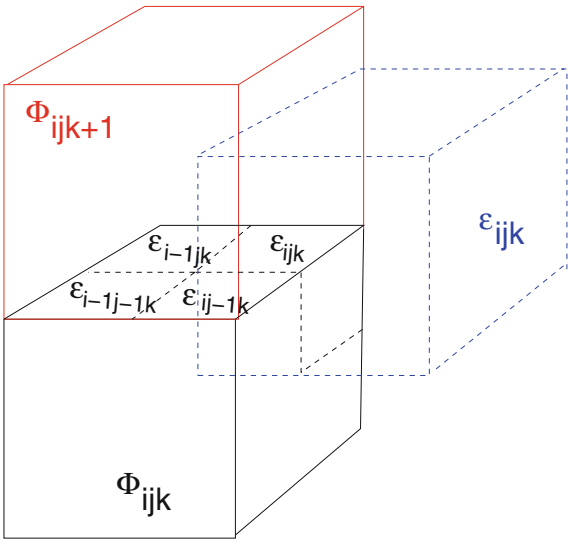


Fig. 18.11 (Shifted grid method) A different grid is used for the discretization of ε which is shifted by $h/2$ in all directions

The value of ε has to be averaged over four neighboring cells to obtain the discretized equation

$$
\begin{aligned}
-\frac{Q_{ijk}}{h^2} &= \sum_s \varepsilon(\mathbf{r}_{ijk} + d\mathbf{r}_s)\frac{\partial \Phi}{\partial n}(\mathbf{r}_{ijk} + d\mathbf{r}_s) \\
&= \frac{\Phi_{i,j,k+1} - \Phi_{i,j,k}}{h}\,\frac{\varepsilon_{ijk} + \varepsilon_{i,j-1,k} + \varepsilon_{i-1,j,k} + \varepsilon_{i-1,j-1,k}}{4} \\
&+ \frac{\Phi_{i,j,k-1} - \Phi_{i,j,k}}{h}\,\frac{\varepsilon_{ijk-1} + \varepsilon_{i,j-1,k-1} + \varepsilon_{i-1,j,k-1} + \varepsilon_{i-1,j-1,k-1}}{4} \\
&+ \frac{\Phi_{i+1,j,k} - \Phi_{i,j,k}}{h}\,\frac{\varepsilon_{ijk} + \varepsilon_{i,j-1,k} + \varepsilon_{i,j,k-1} + \varepsilon_{i,j-1,k-1}}{4}
\end{aligned}
$$

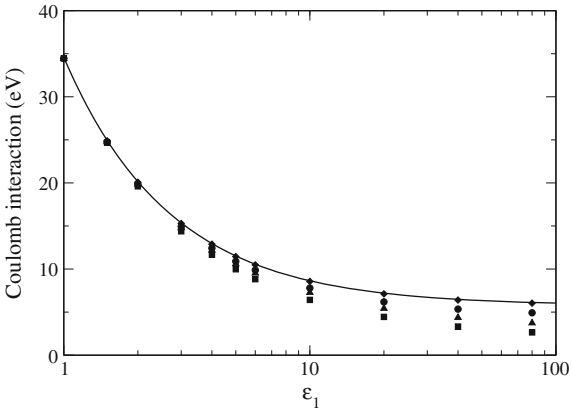


Fig. 18.12 (Comparison of numerical errors) The Coulomb interaction of a charged sphere is calculated with several methods for 100^3 grid points. *circles* (18.40, ε averaged) *diamonds* (18.40, ε^{-1} averaged) *squares* (18.51, ε averaged), *triangles* (18.51, ε^{-1} averaged), *solid curve* analytical solution (18.49)

$$
\begin{aligned}
&+ \frac{\Phi_{i-1,j,k} - \Phi_{i,j,k}}{h} \frac{\varepsilon_{i-1jk} + \varepsilon_{i-1,j-1,k} + \varepsilon_{i-1,j,k-1} + \varepsilon_{i-1,j-1,k-1}}{4} \\
&+ \frac{\Phi_{i,j+1,k} - \Phi_{i,j,k}}{h} \frac{\varepsilon_{ijk} + \varepsilon_{i-1,j,k} + \varepsilon_{i,j,k-1} + \varepsilon_{i-1,j,k-1}}{4} \\
&+ \frac{\Phi_{i,j-1,k} - \Phi_{i,j,k}}{h} \frac{\varepsilon_{ij-1k} + \varepsilon_{i-1,j-1,k} + \varepsilon_{i,j-1,k-1} + \varepsilon_{i-1,j-1,k-1}}{4}. \qquad (18.51)
\end{aligned}
$$

The shifted-grid method is especially useful if ε changes at planar interfaces. Numerical results of several methods are compared in Fig. 18.12.

18.2 Poisson–Boltzmann Equation

Electrostatic interactions are very important in molecular physics. Bio-molecules are usually embedded in an environment which is polarizable and contains mobile charges ($Na^+, K^+, Mg^{++}, Cl^- \cdots$).

We divide the charge density formally into a fixed and a mobile part

$$\rho(\mathbf{r}) = \rho_{fix}(\mathbf{r}) + \rho_{mobile}(\mathbf{r}). \qquad (18.52)$$

The fixed part represents, for instance, the charge distribution of a protein molecule which, neglecting polarization effects, is a given quantity and provides the inhomogeneity of the equation. The mobile part, on the other hand, represents the sum of all mobile charges (e is the elementary charge and Z_i the charge number of ion species i)

$$\rho_{mobile}(\mathbf{r}) = \sum_i Z_i e\, n_i(\mathbf{r}) \qquad (18.53)$$

which move around until an equilibrium is reached which is determined by the mutual interaction of the ions. The famous Debye–Huckel [233] and Gouy–Chapman models [234, 235] assume that the electrostatic interaction

$$U(\mathbf{r}) = Z_i e \Phi(\mathbf{r}) \tag{18.54}$$

is dominant and the density of the ions n_i is given by a Boltzmann-distribution

$$n_i(\mathbf{r}) = n_i^{(0)} \mathrm{e}^{-Z_i e \Phi(\mathbf{r})/k_B T}. \tag{18.55}$$

The potential $\Phi(\mathbf{r})$ has to be calculated in a self consistent way together with the density of mobile charges. The charge density of the free ions is

$$\rho_{mobile}(\mathbf{r}) = \sum_i n_i^{(0)} e Z_i e^{-Z_i e \Phi/k_B T} \tag{18.56}$$

and the Poisson equation (18.7) turns into the Poisson–Boltzmann equation [236]

$$\mathrm{div}(\varepsilon(\mathbf{r})\, \mathrm{grad}\Phi(\mathbf{r})) + \sum_i n_i^{(0)} e Z_i e^{-Z_i e \Phi/k_B T} = -\rho_{fix}(\mathbf{r}). \tag{18.57}$$

18.2.1 Linearization of the Poisson–Boltzmann Equation

For small ion concentrations the exponential can be expanded

$$e^{-Z_i e \Phi/kT} \approx 1 - \frac{Z_i e \Phi}{k_B T} + \frac{1}{2}\left(\frac{Z_i e \Phi}{k_B T}\right)^2 + \cdots. \tag{18.58}$$

For a neutral system

$$\sum_i n_i^{(0)} Z_i e = 0 \tag{18.59}$$

and the linearized Poisson–Boltzmann-equation is obtained:

$$\mathrm{div}(\varepsilon(\mathbf{r})\, \mathrm{grad}\, \Phi(\mathbf{r})) - \sum_i n_i^{(0)} \frac{Z_i^2 e^2}{k_B T} \Phi(\mathbf{r}) = -\rho_{fix}. \tag{18.60}$$

With

$$\varepsilon(\mathbf{r}) = \varepsilon_0 \varepsilon_r(\mathbf{r}) \tag{18.61}$$

and the definition

$$\kappa(\mathbf{r})^2 = \frac{e^2}{\varepsilon_0 \varepsilon_r(\mathbf{r}) k_B T} \sum n_i^{(0)} Z_i^2 \tag{18.62}$$

we have finally

$$\operatorname{div}(\varepsilon_r(\mathbf{r}) \operatorname{grad} \Phi(\mathbf{r})) - \varepsilon_r \kappa^2 \Phi = -\frac{1}{\varepsilon_0}\rho. \tag{18.63}$$

For a charged sphere with radius a embedded in a homogeneous medium the solution of (18.63) is given by

$$\Phi = \frac{A}{r} e^{-\kappa r} \quad A = \frac{e}{4\pi\varepsilon_0\varepsilon_r} \frac{e^{\kappa a}}{1+\kappa a}. \tag{18.64}$$

The potential is shielded by the ions. Its range is of the order $\lambda_{Debye} = 1/\kappa$ (the so-called Debye length).

18.2.2 Discretization of the Linearized Poisson Boltzmann Equation

To solve (18.63) the discrete equation (18.39) is generalized to [237]

$$\sum \frac{2\varepsilon_r(\mathbf{r}_{ijk} + d\mathbf{r}_s)\varepsilon_r(\mathbf{r}_{ijk})}{\varepsilon_r(\mathbf{r}_{ijk} + d\mathbf{r}_s) + \varepsilon_r(\mathbf{r}_{ijk})} \left(\Phi(\mathbf{r}_{ijk} + d\mathbf{r}_s) - \Phi(\mathbf{r}_{ijk})\right)$$
$$- \varepsilon_r(\mathbf{r}_{ijk})\kappa^2(\mathbf{r}_{ijk})h^2\Phi(\mathbf{r}_{ijk}) = -\frac{Q_{ijk}}{h\varepsilon_0}. \tag{18.65}$$

If ε is constant then we iterate

$$\Phi^{new}(\mathbf{r}_{ijk}) = \frac{\frac{Q_{ijk}}{h\varepsilon_0\varepsilon_r} + \sum \Phi^{old}(\mathbf{r}_{ijk} + d\mathbf{r}_s)}{6 + h^2\kappa^2(\mathbf{r}_{ijk})}. \tag{18.66}$$

18.3 Boundary Element Method for the Poisson Equation

Often continuum models are used to describe the solvation of a subsystem which is treated with a high accuracy method. The polarization of the surrounding solvent or protein is described by its dielectric constant ε and the subsystem is placed inside a cavity with $\varepsilon = \varepsilon_0$ (Fig. 18.13). Instead of solving the Poisson equation for a large solvent volume another kind of method is often used which replaces the polarization of the medium by a distribution of charges over the boundary surface.

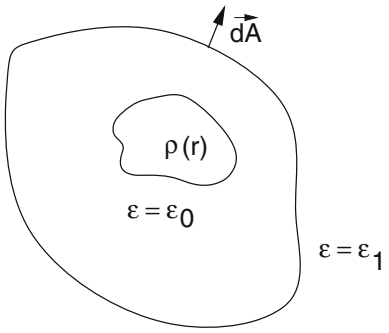


Fig. 18.13 Cavity in a dielectric medium

In the following we consider model systems which are composed of two spatial regions:

- the outer region is filled with a dielectric medium (ε_1) and contains no free charges
- the inner region ("Cavity") contains a charge distribution $\rho(r)$ and its dielectric constant is $\varepsilon = \varepsilon_0$.

18.3.1 Integral Equations for the Potential

Starting from the Poisson equation

$$\mathrm{div}(\varepsilon(\mathbf{r})\mathrm{grad}\Phi(\mathbf{r})) = -\rho(\mathbf{r}) \tag{18.67}$$

we will derive some useful integral equations in the following. First we apply Gauss's theorem to the expression [150]

$$\begin{aligned}&\mathrm{div}\left[G(\mathbf{r}-\mathbf{r}')\varepsilon(\mathbf{r})\mathrm{grad}(\Phi(\mathbf{r})) - \Phi(\mathbf{r})\varepsilon(\mathbf{r})\mathrm{grad}(G(\mathbf{r}-\mathbf{r}'))\right]\\ &= -\rho(\mathbf{r})G(\mathbf{r}-\mathbf{r}') - \Phi(\mathbf{r})\varepsilon(\mathbf{r})\mathrm{divgrad}(G(\mathbf{r}-\mathbf{r}')) - \Phi(\mathbf{r})\mathrm{grad}\varepsilon(\mathbf{r})\mathrm{grad}(G(\mathbf{r}-\mathbf{r}'))\end{aligned} \tag{18.68}$$

with the yet undetermined function $G(\mathbf{r}-\mathbf{r}')$. Integration over a volume V gives

$$\begin{aligned}&-\int_V dV\ \left(\rho(\mathbf{r})G(\mathbf{r}-\mathbf{r}') + \Phi(\mathbf{r})\varepsilon(\mathbf{r})\mathrm{divgrad}(G(\mathbf{r}-\mathbf{r}'))\right.\\ &\left.+\Phi(\mathbf{r})\mathrm{grad}\varepsilon(\mathbf{r})\mathrm{grad}(G(\mathbf{r}-\mathbf{r}'))\right)\\ &= \oint_{\partial V} dA\left(G(\mathbf{r}-\mathbf{r}')\varepsilon(\mathbf{r})\frac{\partial}{\partial n}(\Phi(\mathbf{r})) - \Phi(\mathbf{r})\varepsilon(\mathbf{r})\frac{\partial}{\partial n}(G(\mathbf{r}-\mathbf{r}'))\right).\end{aligned} \tag{18.69}$$

Now choose G as the fundamental solution of the Poisson equation

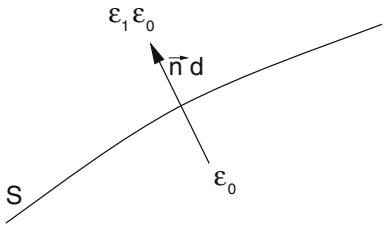


Fig. 18.14 Discontinuity at the cavity boundary

$$G_0(\mathbf{r}-\mathbf{r}') = -\frac{1}{4\pi|\mathbf{r}-\mathbf{r}'|} \tag{18.70}$$

which obeys

$$\text{div grad}G_0 = \delta(\mathbf{r}-\mathbf{r}') \tag{18.71}$$

to obtain the following integral equation for the potential:

$$\Phi(\mathbf{r}')\varepsilon(\mathbf{r}) = \int_V dV \frac{\rho(\mathbf{r})}{4\pi|\mathbf{r}-\mathbf{r}'|} + \frac{1}{4\pi}\int_V dV\,\Phi(\mathbf{r})\text{grad}\varepsilon(\mathbf{r})\text{grad}\left(\frac{1}{|\mathbf{r}-\mathbf{r}'|}\right)$$
$$-\frac{1}{4\pi}\oint_{\partial V} dA\left(\frac{1}{|\mathbf{r}-\mathbf{r}'|}\varepsilon(\mathbf{r})\frac{\partial}{\partial n}(\Phi(\mathbf{r})) + \Phi(\mathbf{r})\varepsilon(\mathbf{r})\frac{\partial}{\partial n}\left(\frac{1}{|\mathbf{r}-\mathbf{r}'|}\right)\right). \tag{18.72}$$

First consider as the integration volume a sphere with increasing radius. Then the surface integral vanishes for infinite radius ($\Phi \to 0$ at large distances) [150].

The gradient of $\varepsilon(\mathbf{r})$ is nonzero only on the boundary surface (Fig. 18.14) of the cavity and with the limiting procedure ($d \to 0$)

$$\text{grad}\varepsilon(\mathbf{r})dV = \mathbf{n}\frac{\epsilon_1 - 1}{d}\epsilon_0 dV = dA\,\mathbf{n}(\epsilon_1 - 1)\epsilon_0$$

we obtain

$$\Phi(\mathbf{r}') = \frac{1}{\varepsilon(\mathbf{r}')}\int_{cav} dV\frac{\rho(\mathbf{r})}{4\pi|\mathbf{r}-\mathbf{r}'|} + \frac{(\epsilon_1-1)\epsilon_0}{4\pi\varepsilon(\mathbf{r}')}\oint_S dA\,\Phi(\mathbf{r})\frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}'|}. \tag{18.73}$$

This equation allows to calculate the potential inside and outside the cavity from the given charge density and the potential at the boundary.

Next we apply (18.72) to the cavity volume (where $\varepsilon = \epsilon_0$) and obtain

$$\Phi_{in}(\mathbf{r}') = \int_V dV\,\frac{\rho(\mathbf{r})}{4\pi|\mathbf{r}-\mathbf{r}'|\epsilon_0}$$
$$-\frac{1}{4\pi}\oint_S dA\left(\Phi_{in}(\mathbf{r})\frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}'|} - \frac{1}{|\mathbf{r}-\mathbf{r}'|}\frac{\partial}{\partial n}\Phi_{in}(\mathbf{r})\right). \tag{18.74}$$

From comparison with (18.73) we have

$$\oint_S dA \frac{1}{|\mathbf{r}-\mathbf{r}'|}\frac{\partial}{\partial n}\Phi_{in}(\mathbf{r}) = \epsilon_1 \oint_S dA \Phi_{in}(\mathbf{r})\frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}'|}$$

and the potential can be alternatively calculated from the values of its normal gradient at the boundary

$$\Phi(\mathbf{r}') = \frac{1}{\varepsilon(\mathbf{r}')}\int_{cav} dV \frac{\rho(\mathbf{r})}{4\pi|\mathbf{r}-\mathbf{r}'|} + \frac{\left(1-\frac{1}{\epsilon_1}\right)\epsilon_0}{4\pi\varepsilon(\mathbf{r}')}\oint_S dA \frac{1}{|\mathbf{r}-\mathbf{r}'|}\frac{\partial}{\partial n}\Phi_{in}(\mathbf{r}). \tag{18.75}$$

This equation can be interpreted as the potential generated by the charge density ρ plus an additional surface charge density

$$\sigma(\mathbf{r}) = \left(1-\frac{1}{\epsilon_1}\right)\epsilon_0 \frac{\partial}{\partial n}\Phi_{in}(\mathbf{r}). \tag{18.76}$$

Integration over the volume outside the cavity (where $\varepsilon = \epsilon_1\epsilon_0$) gives the following expression for the potential:

$$\Phi_{out}(\mathbf{r}') = \frac{1}{4\pi}\oint_S dA \left(\Phi_{out}(\mathbf{r})\frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}'|} - \frac{1}{|\mathbf{r}-\mathbf{r}'|}\frac{\partial}{\partial n}\Phi_{out}(\mathbf{r})\right). \tag{18.77}$$

At the boundary the potential is continuous

$$\Phi_{out}(\mathbf{r}) = \Phi_{in}(\mathbf{r}) \quad \mathbf{r} \in A \tag{18.78}$$

whereas the normal derivative (hence the normal component of the electric field) has a discontinuity

$$\epsilon_1 \frac{\partial \Phi_{out}}{\partial n} = \frac{\partial \Phi_{in}}{\partial n}. \tag{18.79}$$

18.3.2 Calculation of the Boundary Potential

For a numerical treatment the boundary surface is approximated by a finite set of small surface elements $S_i, i = 1 \cdots N$ centered at $\mathbf{r}_i$ with an area A_i and normal vector $\mathbf{n}_i$ (Fig. 18.15). (We assume planar elements in the following, the curvature leads to higher order corrections).

The corresponding values of the potential and its normal derivative are denoted as $\Phi_i = \Phi(\mathbf{r}_i)$ and $\frac{\partial \Phi_i}{\partial n} = \mathbf{n}_i \,\mathrm{grad}\Phi(\mathbf{r}_i)$. At a point $\mathbf{r}_j^{\pm}$ close to the element S_j we obtain the following approximate equations:

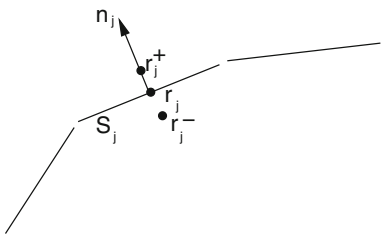


Fig. 18.15 Representation of the boundary by surface elements

$$\Phi_{in}(\mathbf{r}_j^-) = \int_V dV\ \frac{\rho(\mathbf{r})}{4\pi|\mathbf{r}-\mathbf{r}_j^-|\epsilon_0}$$
$$-\frac{1}{4\pi}\sum_i \Phi_i \oint_{S_i} dA\ \frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}_j^-|} + \frac{1}{4\pi}\sum_i \frac{\partial \Phi_{i,in}}{\partial n}\oint_{S_i} dA\ \frac{1}{|\mathbf{r}-\mathbf{r}_j^-|} \tag{18.80}$$

$$\Phi_{out}(\mathbf{r}_j^+) = \frac{1}{4\pi}\sum_i \Phi_i \oint_{S_i} dA\ \frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}_j^+|} - \frac{1}{4\pi}\sum_i \frac{\partial \Phi_{i,out}}{\partial n}\oint_{S_i} dA\ \frac{1}{|\mathbf{r}-\mathbf{r}_j^+|}. \tag{18.81}$$

These two equations can be combined to obtain a system of equations for the potential values only. To that end we approach the boundary symmetrically with $\mathbf{r}_i^{\pm} = \mathbf{r}_i \pm d\mathbf{n}_i$. Under this circumstance

$$\oint_{S_i} dA\ \frac{1}{|\mathbf{r}-\mathbf{r}_j^+|} = \oint_{S_i} dA\ \frac{1}{|\mathbf{r}-\mathbf{r}_j^-|}$$
$$\oint_{S_i} dA\ \frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}_i^+|} = -\oint_{S_i} dA\ \frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}_i^-|}$$
$$\oint_{S_i} dA\ \frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}_j^+|} = \oint_{S_i} dA\ \frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}_j^-|} \quad j \neq i \tag{18.82}$$

and we find

$$(1+\epsilon_1)\Phi_j = \int_V dV\ \frac{\rho(\mathbf{r})}{4\pi\epsilon_0|\mathbf{r}-\mathbf{r}_j|}$$
$$-\frac{1}{4\pi}\sum_{i\neq j}(1-\epsilon_1)\Phi_i \oint_{S_i} dA\ \frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}_j^-|} - \frac{1}{4\pi}(1+\epsilon_1)\Phi_j \oint_{S_j} dA\ \frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}_j^-|}. \tag{18.83}$$

The integrals for $i \neq j$ can be approximated by

$$\oint_{S_i} dA\ \frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}_j^-|} = A_i\,\mathbf{n}_i \mathrm{grad}_i \frac{1}{|\mathbf{r}_i-\mathbf{r}_j|}. \tag{18.84}$$

The second integral has a simple geometrical interpretation (Fig. 18.16).

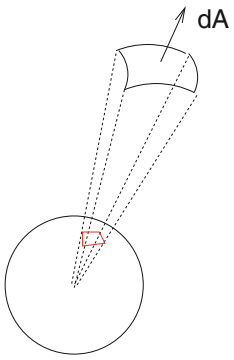


Fig. 18.16 Projection of the surface element

Since $\text{grad}\frac{1}{|r-r'|} = -\frac{1}{|r-r'|^2}\frac{r-r'}{|r-r'|}$ the area element $d\mathbf{A}$ is projected onto a sphere with unit radius. The integral $\oint_{S_j} d\mathbf{A}\ \text{grad}_{r-}\frac{1}{|\mathbf{r_j}-\mathbf{r}_j^-|}$ is given by the solid angle of $\mathbf{S}_j$ with respect to r'. For $r' \to r_j$ from inside this is just minus half of the full space angle of 4π. Thus we have

$$(1+\epsilon_1)\Phi_j = \int_V dV\ \frac{\rho(\mathbf{r})}{4\pi|\mathbf{r}-\mathbf{r}_j|\epsilon_0} - \frac{1}{4\pi}\sum_{i\neq j}(1-\epsilon_1)\Phi_i A_i \frac{\partial}{\partial n_i}\frac{1}{|\mathbf{r}_i-\mathbf{r}_j|} + \frac{1}{2}(1+\epsilon_1)\Phi_j \tag{18.85}$$

or

$$\Phi_j = \frac{2}{1+\epsilon_1}\int_V dV\ \frac{\rho(\mathbf{r})}{4\pi\epsilon_0|\mathbf{r}-\mathbf{r}_j|} + \frac{1}{2\pi}\sum_{i\neq j}\frac{\epsilon_1-1}{\epsilon_1+1}\Phi_i A_i \frac{\partial}{\partial n_i}\frac{1}{|\mathbf{r}_i-\mathbf{r}_j|}. \tag{18.86}$$

This system of equations can be used to calculate the potential on the boundary. The potential inside the cavity is then given by (18.73). Numerical stability is improved by a related method which considers the potential gradient along the boundary. Taking the normal derivative

$$\frac{\partial}{\partial n_j} = \mathbf{n}_j \text{grad}_{r_j\pm} \tag{18.87}$$

of (18.80, 18.81) gives

$$\frac{\partial}{\partial n_j}\Phi_{in}(\mathbf{r}_j^-) = \frac{\partial}{\partial n_j}\int_V dV\ \frac{\rho(\mathbf{r})}{4\pi|\mathbf{r}-\mathbf{r}_j^-|\epsilon_0} - \frac{1}{4\pi}\sum_i \Phi_i \oint_{S_i} dA\ \frac{\partial^2}{\partial n\partial n_j}\frac{1}{|\mathbf{r}-\mathbf{r}_j^-|} + \frac{1}{4\pi}\sum_i \frac{\partial\Phi_{i,in}}{\partial n}\oint_{S_i} dA\ \frac{\partial}{\partial n_j}\frac{1}{|\mathbf{r}-\mathbf{r}_j^-|} \tag{18.88}$$

$$\frac{\partial}{\partial n_j}\Phi_{out}(\mathbf{r}_j^+) = \frac{1}{4\pi}\sum_i \Phi_i \oint_{S_i} dA\, \frac{\partial^2}{\partial n \partial n_j}\frac{1}{|\mathbf{r}-\mathbf{r}_j^+|}$$
$$-\frac{1}{4\pi}\sum_i \frac{\partial \Phi_{i,out}}{\partial n}\oint_{S_i} dA\, \frac{\partial}{\partial n_j}\frac{1}{|\mathbf{r}-\mathbf{r}_j^+|}. \tag{18.89}$$

In addition to (18.82) we have now

$$\oint_{S_i} dA\, \frac{\partial^2}{\partial n \partial n_j}\frac{1}{|\mathbf{r}-\mathbf{r}_j^-|} = \oint_{S_i} dA\, \frac{\partial^2}{\partial n \partial n_j}\frac{1}{|\mathbf{r}-\mathbf{r}_j^+|} \tag{18.90}$$

and the sum of the two equations gives

$$\left(1+\frac{1}{\epsilon_1}\right)\frac{\partial}{\partial n_j}\Phi_{in,j}$$
$$= \frac{\partial}{\partial n_j}\left(\int_V dV\, \frac{\rho(\mathbf{r})}{4\pi\epsilon_0|\mathbf{r}-\mathbf{r}_j|} + \frac{1-\frac{1}{\epsilon_1}}{4\pi}\sum_{i\neq j} A_i \frac{\partial \Phi_{i,in}}{\partial n}\frac{1}{|\mathbf{r}_i-\mathbf{r}_j|}\right)$$
$$+\frac{1+\frac{1}{\epsilon_1}}{2\pi}\frac{\partial \Phi_{j,in}}{\partial n} \tag{18.91}$$

or finally

$$\frac{\partial}{\partial n_j}\Phi_{in,j} = \frac{2\epsilon_1}{\epsilon_1+1}\frac{\partial}{\partial n_j}\int_V dV\, \frac{\rho(\mathbf{r})}{4\pi\epsilon_0|\mathbf{r}-\mathbf{r}_j|}$$
$$+2\frac{\epsilon_1-1}{\epsilon_1+1}\sum_{i\neq j} A_i \frac{\partial \Phi_{i,in}}{\partial n}\frac{\partial}{\partial n_j}\frac{1}{|\mathbf{r}_i-\mathbf{r}_j|}. \tag{18.92}$$

In terms of the surface charge density this reads:

$$\sigma_j' = 2\varepsilon_0\frac{(1-\varepsilon_1)}{(1+\varepsilon_1)}\left(-\mathbf{n}_j \mathrm{grad}\int dV \frac{\rho(\mathbf{r})}{4\pi\epsilon_0|\mathbf{r}-\mathbf{r}'|} + \frac{1}{4\pi\varepsilon_0}\sum_{i\neq j}\sigma_i' A_i \frac{\mathbf{n}_j(\mathbf{r}_j-\mathbf{r}_i)}{|\mathbf{r}_i-\mathbf{r}_j|^3}\right). \tag{18.93}$$

This system of linear equations can be solved directly or iteratively (a simple damping scheme $\sigma_m' \to \omega\sigma_m' + (1-\omega)\sigma_{m,old}'$ with $\omega \approx 0.6$ helps to get rid of oscillations). From the surface charges $\sigma_i A_i$ the potential is obtained with the help of (18.75).

18.4 Boundary Element Method for the Linearized Poisson–Boltzmann Equation

We consider now a cavity within an electrolyte. The fundamental solution of the linear Poisson–Boltzmann equation (18.63)

$$G_\kappa(\mathbf{r}-\mathbf{r}') = -\frac{e^{-\kappa|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} \tag{18.94}$$

obeys

$$\text{div grad}G_\kappa(\mathbf{r}-\mathbf{r}') - \kappa^2 G_\kappa(\mathbf{r}-\mathbf{r}') = \delta(\mathbf{r}-\mathbf{r}'). \tag{18.95}$$

Inserting into Green's theorem (18.69) we obtain the potential outside the cavity

$$\Phi_{out}(\mathbf{r}') = -\oint_S dA \left(\Phi_{out}(\mathbf{r}) \frac{\partial}{\partial n} G_\kappa(\mathbf{r}-\mathbf{r}') - G_\kappa(\mathbf{r}-\mathbf{r}') \frac{\partial}{\partial n} \Phi_{out}(\mathbf{r}) \right) \tag{18.96}$$

which can be combined with (18.74, 18.79) to give the following equations [238]

$$\begin{aligned}(1+\epsilon_1)\Phi(\mathbf{r}') &= \oint_S dA \left[\Phi(\mathbf{r}) \frac{\partial}{\partial n}(G_0 - \epsilon_1 G_\kappa) - (G_0 - G_\kappa) \frac{\partial}{\partial n} \Phi_{in}(\mathbf{r}) \right] \\ &+ \int_{cav} \frac{\rho(\mathbf{r})}{4\pi\epsilon_0|\mathbf{r}-\mathbf{r}'|} dV \end{aligned} \tag{18.97}$$

$$\begin{aligned}(1+\epsilon_1)\frac{\partial}{\partial n'}\Phi_{in}(\mathbf{r}') &= \oint_S dA \Phi(\mathbf{r}) \frac{\partial^2}{\partial n \partial n'}(G_0 - G_\kappa) \\ &- \oint_S dA \frac{\partial}{\partial n}\Phi_{in}(\mathbf{r}) \frac{\partial}{\partial n'} \left(G_0 - \frac{1}{\epsilon_1} G_k \right) + \frac{\partial}{\partial n'} \int_{cav} \frac{\rho(\mathbf{r})}{4\pi\epsilon|\mathbf{r}-\mathbf{r}'|} dV. \end{aligned} \tag{18.98}$$

For a set of discrete boundary elements the following equations determine the values of the potential and its normal derivative at the boundary:

$$\begin{aligned}\frac{1+\epsilon_1}{2}\Phi_j &= \sum_{i \neq j} \Phi_i \oint dA \frac{\partial}{\partial n}(G_0 - \epsilon_1 G_\kappa) - \sum_{i \neq j} \frac{\partial}{\partial n}\Phi_{i,in} \oint dA (G_0 - G_\kappa) \\ &+ \int \frac{\rho(\mathbf{r})}{4\pi\epsilon_0|\mathbf{r}-\mathbf{r}_i|} dV \end{aligned} \tag{18.99}$$

$$\begin{aligned}\frac{1+\epsilon_1}{2}\frac{\partial}{\partial n'}\Phi_{i,in} &= \sum_{i \neq j} \Phi_i \oint dA \frac{\partial^2}{\partial n \partial n'}(G_0 - G_\kappa) \\ &- \sum_{i \neq j} \frac{\partial}{\partial n}\Phi_{i,in} \oint dA \frac{\partial}{\partial n'} \left(G_0 - \frac{1}{\epsilon_1} G_k \right) + \frac{\partial}{\partial n'} \int \frac{\rho(\mathbf{r})}{4\pi\epsilon|\mathbf{r}-\mathbf{r}_i|} dV. \end{aligned} \tag{18.100}$$

The situation is much more involved than for the simpler Poisson equation (with $\kappa = 0$) since the calculation of many integrals including such with singularities is necessary [238, 239].

18.5 Electrostatic Interaction Energy (Onsager Model)

A very important quantity in molecular physics is the electrostatic interaction of a molecule and the surrounding solvent [240, 241]. We calculate it by taking a small part of the charge distribution from infinite distance ($\Phi(r \to \infty) = 0$) into the cavity. The charge distribution thereby changes from $\lambda\rho(r)$ to $(\lambda + d\lambda)\rho(r)$ with $0 \leq \lambda \leq 1$. The corresponding energy change is

$$dE = \int d\lambda \cdot \rho(r)\, \Phi_\lambda(r) dV$$
$$= \int d\lambda \cdot \rho(r) \left(\sum_n \frac{\sigma_n(\lambda) A_n}{4\pi\varepsilon_0 |r - r_n|} + \int \frac{\lambda\rho(r')}{4\pi\varepsilon_0 |r - r'|} dV' \right) dV. \tag{18.101}$$

Multiplication of the equations (18.93) by a factor of λ shows that the surface charges $\lambda\sigma_n$ are the solution corresponding to the charge density $\lambda\rho(r)$. It follows that $\sigma_n(\lambda) = \lambda\sigma_n$ and hence

$$dE = \lambda d\lambda \int \rho(r) \left(\sum_n \frac{\sigma_n A_n}{4\pi\varepsilon_0 |r - r_n|} + \frac{\rho(r')}{4\pi\varepsilon_0 |r - r'|} dV' \right). \tag{18.102}$$

The second summand is the self energy of the charge distribution which does not depend on the medium. The first summand vanishes without a polarizable medium and gives the interaction energy. Hence we have the final expression

$$E_{int} = \int dE = \int_0^1 \lambda d\lambda \int \rho(r) \sum_n \frac{\sigma_n A_n}{4\pi\varepsilon_0 |r - r_n|} dV$$
$$= \sum_n \sigma_n A_n \int \frac{\rho(r)}{8\pi\varepsilon_0 |r - r_n|} dV. \tag{18.103}$$

For the special case of a spherical cavity with radius a an analytical solution by a multipole expansion is available [242]

$$E_{int} = -\frac{1}{8\pi\varepsilon_0} \sum_l \sum_{m=-l}^{l} \frac{(l+1)(\varepsilon_1 - 1)}{[l + \varepsilon_1(l+1)]\, a^{2l+1}} M_l^m M_l^m \tag{18.104}$$

with the multipole moments

$$M_l^m = \int \rho(r,\theta,\varphi)\sqrt{\frac{4\pi}{2l+1}} r^l Y_l^m(\theta,\varphi) dV. \tag{18.105}$$

The first two terms of this series are:

$$E_{int}^{(0)} = -\frac{1}{8\pi\varepsilon_0}\frac{\varepsilon_1 - 1}{\varepsilon_1 a} M_0^0 M_0^0 = -\frac{1}{8\pi\varepsilon_0}\left(1 - \frac{1}{\varepsilon_1}\right)\frac{Q^2}{a} \tag{18.106}$$

$$E_{int}^{(1)} = -\frac{1}{8\pi\varepsilon_0}\frac{2(\varepsilon_1 - 1)}{(1+2\varepsilon_1)a^3}(M_1^{-1}M_1^{-1} + M_1^0 M_1^0 + M_1^1 M_1^1)$$

$$= -\frac{1}{8\pi\varepsilon_0}\frac{2(\varepsilon_1 - 1)}{1+2\varepsilon_1}\frac{\mu^2}{a^3}. \tag{18.107}$$

18.5.1 Example: Point Charge in a Spherical Cavity

Consider a point charge Q in the center of a spherical cavity of radius R (Fig. 18.17). The dielectric constant is given by

$$\varepsilon = \begin{cases} \varepsilon_0 & r < R \\ \varepsilon_1\varepsilon_0 & r > R \end{cases}. \tag{18.108}$$

Electric field and potential are inside the cavity

$$E = \frac{Q}{4\pi\varepsilon_0 r^2} \quad \Phi = \frac{Q}{4\pi\varepsilon_0 r} + \frac{Q}{4\pi\varepsilon_0 R}\left(\frac{1}{\varepsilon_1} - 1\right) \tag{18.109}$$

and outside

$$E = \frac{Q}{4\pi\varepsilon_1\varepsilon_0 r^2} \quad \Phi = \frac{Q}{4\pi\varepsilon_1\varepsilon_0 r} \quad r > R \tag{18.110}$$

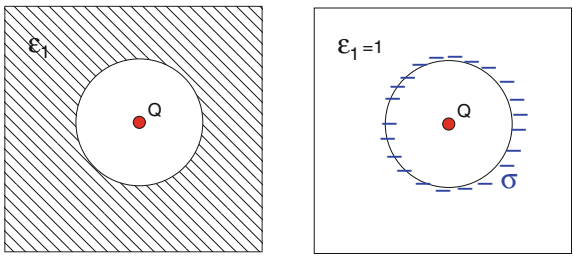


Fig. 18.17 Surface charges

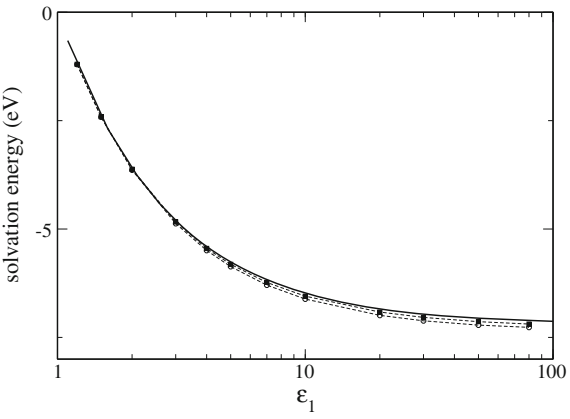


Fig. 18.18 (Solvation energy with the boundary element method) A spherical cavity is simulated with radius a = 1Å which contains a point charge in its center. The solvation energy is calculated with 25 × 25 (*circles*) and 50 × 50 (*squares*) surface elements of equal size. The exact expression (18.106) is shown by the *solid curve*

which in terms of the surface charge density σ is

$$E = \frac{Q + 4\pi R^2 \sigma}{4\pi\varepsilon_0 r^2} \quad r > R \tag{18.111}$$

with the total surface charge

$$4\pi R^2 \sigma = Q\left(\frac{1}{\varepsilon_1} - 1\right). \tag{18.112}$$

The solvation energy (18.103) is given by

$$E_{int} = \frac{Q^2}{8\pi\varepsilon_0}\left(\frac{1}{\varepsilon_1} - 1\right) \tag{18.113}$$

which is the first term (18.106) of the multipole expansion. Figure 18.18 shows numerical results.

Problems

Problem 18.1 Linearized Poisson–Boltzmann Equation

This computer experiment simulates a homogeneously charged sphere in a dielectric medium (Fig. 18.19). The electrostatic potential is calculated from the linearized Poisson Boltzmann equation (18.65) on a cubic grid of up to 100^3 points. The potential $\Phi(x)$ is shown along a line through the center together with a log-log plot of the maximum change per iteration

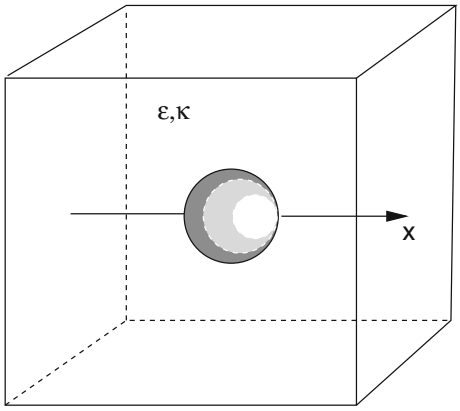


Fig. 18.19 Charged sphere in a dielectric medium

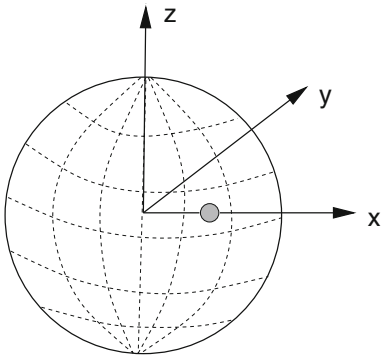


Fig. 18.20 Point charge inside a spherical cavity

$$|\Phi^{(n+1)}(\mathbf{r}) - \Phi^{(n)}(\mathbf{r})| \tag{18.114}$$

as a measure of convergence.

Explore the dependence of convergence on

- the initial values which can be chosen either $\Phi(\mathbf{r}) = 0$ or from the analytical solution

$$\Phi(\mathbf{r}) = \begin{cases} \frac{Q}{8\pi\epsilon\epsilon_0 a}\frac{2+\epsilon(1+\kappa a)}{1+\kappa a} - \frac{Q}{8\pi\epsilon_0 a^3}r^2 & \text{for } r < a \\ \frac{Qe^{-\kappa(r-a)}}{4\pi\epsilon_0\epsilon(\kappa a+1)r} & \text{for } r > a. \end{cases} \tag{18.115}$$

- the relaxation parameter ω for different combinations of ϵ and κ
- the resolution of the grid

Problem 18.2 Boundary Element Method

In this computer experiment the solvation energy of a point charge within a spherical cavity (Fig. 18.20) is calculated with the boundary element method (18.93).

The calculated solvation energy is compared to the analytical value from (18.104)

$$E_{solv} = \frac{Q^2}{8\pi\epsilon_0 R} \sum_{n=1}^{\infty} \frac{s^{2n}}{R^{2n}} \frac{(\epsilon_1 - \epsilon_2)(n+1)}{n\epsilon_1 + (n+1)\epsilon_2} \tag{18.116}$$

where R is the cavity radius and s is the distance of the charge from the center of the cavity.
Explore the dependence of accuracy and convergence on

- the damping parameter ω
- the number of surface elements ($6 \times 6 \cdots 42 \times 42$) which can be chosen either as $d\phi d\theta$ or $d\phi d\ \cos\theta$ (equal areas)
- the position of the charge

Chapter 19
Advection

Transport processes are very important in physics and engineering sciences. Transport of a conserved quantity like energy or concentration of a certain substance (e.g. salt) in a moving fluid is due to the effects of diffusion (Chap. 21) *and advection (which denotes transport by the bulk motion). The combination of these two transport mechanisms is usually called convection.*

In this chapter we investigate the advection equation in one spatial dimension

$$\frac{\partial}{\partial t} f(x,t) = -c \frac{\partial}{\partial x} f(x,t). \tag{19.1}$$

Numerical solutions are obtained with simple and more elaborate methods using finite differences, finite volumes and finite elements. Accuracy and stability of different methods are compared. The linear advection equation is an ideal test case but the methods are also useful for general nonlinear advection equations including the famous system of Navier–Stokes equations.

19.1 The Advection Equation

Consider a fluid moving with velocity $\mathbf{u}(\mathbf{r})$ and let $f(\mathbf{r}, t)$ denote the concentration of the substance. Its time dependence obeys the conservation law

$$\frac{\partial}{\partial t} f = \text{ div } (D \text{ grad } f - \mathbf{u} f) + S(\mathbf{r}, t) = - \text{ div } (\mathbf{J}_{diff} + \mathbf{J}_{adv}) + S(\mathbf{r}, t) \tag{19.2}$$

or in integral form

$$\frac{\partial}{\partial t} \int_V dV \; f(\mathbf{r}, t) + \oint_{\partial V} \mathbf{J}(\mathbf{r}, t) d\mathbf{A} = \int_V dV \; S(\mathbf{r}, t). \tag{19.3}$$

Without diffusion and sources or sinks, the flux of the substance is given by

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_19

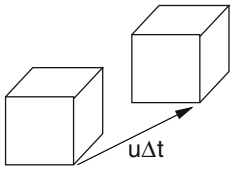


Fig. 19.1 Advection in an incompressible fluid

$$\mathbf{J}(\mathbf{r},t) = \mathbf{u}(\mathbf{r},t)\,f(\mathbf{r},t) \tag{19.4}$$

and the continuity equation for the substance concentration reads

$$\frac{\partial}{\partial t} f + \text{div}\,(f\,\mathbf{u}) = 0. \tag{19.5}$$

Introducing the substantial derivative we obtain

$$0 = \frac{\partial}{\partial t} f + \text{div}\,(f\,\mathbf{u}) = \frac{\partial}{\partial t} f + (\mathbf{u}\ \text{grad})\ f + f\ \text{div}\mathbf{u} \tag{19.6}$$

$$= \frac{\mathrm{d}f}{\mathrm{d}t} + f\ \text{div}\mathbf{u}. \tag{19.7}$$

For the common case of an incompressible fluid div $\mathbf{u} = 0$ and the advection equation simplifies to

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\partial}{\partial t} f(\mathbf{r},t) + (\mathbf{u}(\mathbf{r},t)\ \text{grad})\ f(\mathbf{r},t) = 0 \tag{19.8}$$

which has a very simple interpretation. Consider a small element of the fluid (Fig. 19.1), which during a time interval Δt moves from the position $\mathbf{r}$ to $\mathbf{r} + \Delta\mathbf{r} = \mathbf{r} + \mathbf{u}\Delta t$. The amount of substance does not change and we find

$$f(\mathbf{r},t) = f(\mathbf{r} + \mathbf{u}\Delta t, t + \Delta t) = f(\mathbf{r},t) + \frac{\partial f}{\partial t}\Delta t + \mathbf{u}\Delta t\ \text{grad}\ f + \cdots \tag{19.9}$$

which in the limit of small Δt becomes (19.8).

19.2 Advection in One Dimension

In one dimension div $\mathbf{u} = \frac{\partial u_x}{\partial x}$=0 implies constant velocity $u_x = c$. The differential equation

$$\frac{\partial f(x,t)}{\partial t} + c\frac{\partial f(x,t)}{\partial x} = 0 \tag{19.10}$$

can be solved exactly with d' Alembert's method. After substitution

$$x' = x - ct \quad t' = t \tag{19.11}$$

$$f(x,t) = f(x' + ct', t') = \phi(x', t')$$

$$\frac{\partial}{\partial x} = \frac{\partial}{\partial x'} \quad \frac{\partial}{\partial t} = \frac{\partial}{\partial t'} - c\frac{\partial}{\partial x'} \tag{19.12}$$

it becomes

$$0 = \left(\frac{\partial}{\partial t'} - c\frac{\partial}{\partial x'} + c\frac{\partial}{\partial x'}\right)\phi = \frac{\partial}{\partial t'}\phi \tag{19.13}$$

hence ϕ does not depend on time and the solution has the

$$f(x,t) = \phi(x') = \phi(x - ct) \tag{19.14}$$

where the constant envelope is determined by the initial values

$$\phi(x') = f(x, t = 0). \tag{19.15}$$

After spatial Fourier transformation

$$\hat{f}(k,t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \mathrm{e}^{\mathrm{i}kx} f(x,t) dx \tag{19.16}$$

the advection equation becomes an ordinary differential equation

$$\frac{\mathrm{d}\hat{f}(t,k)}{\mathrm{d}t} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \mathrm{e}^{\mathrm{i}kx} \mathrm{i}ck\, f(x,t) dx = \mathrm{i}ck\, \hat{f}(t,k) \tag{19.17}$$

quite similar to the example of a simple rotation (p. 13). Therefore we have to expect similar problems with the simple Euler integration methods (p. 293).

For a Fourier component of f in space and time (i.e. a plane wave moving in x-direction)

$$g_{\omega k} = \mathrm{e}^{\mathrm{i}(\omega t - kx)} \tag{19.18}$$

we find a linear dispersion relation, i.e. all Fourier components move with the same velocity

$$\omega = ck. \tag{19.19}$$

19.2.1 Spatial Discretization with Finite Differences

The simplest discretization (p. 259) is obtained by introducing a regular grid

$$x_m = m\Delta x \quad m = 1, 2 \ldots M \tag{19.20}$$

$$f_m(t) = f(x_m, t) \tag{19.21}$$

and approximating the gradient by a finite difference quotient.

In the following we use periodic boundary conditions by setting $f_0 \equiv f_M$, $f_{M+1} \equiv f_1$ which are simplest to discuss and allow us to simulate longer times on a finite domain.

19.2.1.1 First Order Forward and Backward Differences (Upwind Scheme)

First we use a first order backward difference in space

$$\frac{\mathrm{d}f_m(t)}{\mathrm{d}t} = c\frac{f_m(t) - f_{m-1}(t)}{\Delta x}. \tag{19.22}$$

From a Taylor series expansion

$$f(x - \Delta x) = f(x) - \frac{\partial f}{\partial x}\Delta x + \frac{(\Delta x)^2}{2}\frac{\partial^2 f}{\partial x^2} \cdots = \exp\left\{-\Delta x \frac{\partial}{\partial x}\right\} f(x) \tag{19.23}$$

we see that the leading error of the finite difference approximation

$$\frac{\partial f}{\partial t} - c\frac{f(x) - f(x - \Delta x)}{\Delta x} = \frac{\partial f}{\partial t} - c\frac{\partial f}{\partial x} + c\frac{\Delta x}{2}\frac{\partial^2 f}{\partial x^2} + \cdots \tag{19.24}$$

looks like a diffusion term for positive velocity c and is therefore called "numerical diffusion". Negative velocities, instead lead to an unphysical sharpening of the function f.

For $c < 0$ we have to reverse the space direction and use a forward difference

$$\frac{\mathrm{d}f_m(t)}{\mathrm{d}t} = c\frac{f_{m+1}(t) - f_m(t)}{\Delta x} \tag{19.25}$$

for which the sign of the second derivative changes

$$\frac{\partial f}{\partial t} - c\frac{f(x+\Delta x) - f(x)}{\Delta x} = \frac{\partial f}{\partial t} - c\frac{\partial f}{\partial x} - c\frac{\Delta x}{2}\frac{\partial^2 f}{\partial x^2} + \cdots . \tag{19.26}$$

Using the backward difference we obtain a system of ordinary differential equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{M-1} \\ f_M \end{pmatrix} = -\frac{c}{\Delta x}\begin{pmatrix} 1 & & & & -1 \\ -1 & 1 & & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 1 & \\ & & & -1 & 1 \end{pmatrix}\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{M-1} \\ f_M \end{pmatrix} \tag{19.27}$$

or shorter

$$\frac{\mathrm{d}\mathbf{f}}{\mathrm{d}t} = -\frac{c}{\Delta x}M\mathbf{f} \tag{19.28}$$

with the formal solution

$$\mathbf{f}(t) = \exp\left\{-\frac{c}{\Delta x}M\,t\right\}\mathbf{f}(t=0). \tag{19.29}$$

The eigenpairs of M are easily found (see p. 221). Inserting the Ansatz

$$\mathbf{f_k} = \begin{pmatrix} \mathrm{e}^{-\mathrm{i}k\Delta x} \\ \vdots \\ \mathrm{e}^{-M\mathrm{i}k\Delta x} \end{pmatrix} \tag{19.30}$$

corresponding to a Fourier component (19.18) into the eigenvalue equation we obtain

$$M\mathbf{f_k} = \begin{pmatrix} \mathrm{e}^{-\mathrm{i}k\Delta x} - \mathrm{e}^{-M\mathrm{i}k\Delta x} \\ \mathrm{e}^{-2\mathrm{i}k\Delta x} - \mathrm{e}^{-1\mathrm{i}k\Delta x} \\ \vdots \\ \mathrm{e}^{-(M-1)\mathrm{i}k\Delta x} - \mathrm{e}^{-(M-2)\mathrm{i}k\Delta x} \\ \mathrm{e}^{-M\mathrm{i}k\Delta x} - \mathrm{e}^{-(M-1)\mathrm{i}k\Delta x} \end{pmatrix}. \tag{19.31}$$

Solutions are found for values of k given by

$$\mathrm{e}^{-M\mathrm{i}k\Delta x} = 1,\ k = 0, \frac{2\pi}{M\Delta x}, \ldots (M-1)\frac{2\pi}{M\Delta x} \tag{19.32}$$

or, reducing k-values to the first Brillouin zone (p. 132)

$$k = -\left(\frac{M}{2} - 1\right)\frac{2\pi}{M\Delta x} - \frac{2\pi}{M\Delta x}, 0, \frac{2\pi}{M\Delta x}, \ldots \frac{M}{2}\frac{2\pi}{M\Delta x} \quad M \text{ even} \tag{19.33}$$

$$k = -\frac{M}{2}\frac{2\pi}{M\Delta x} - \frac{2\pi}{M\Delta x}, 0, \frac{2\pi}{M\Delta x}, \ldots \frac{M}{2}\frac{2\pi}{M\Delta x} \quad M \text{ odd} \tag{19.34}$$

for which

$$M\mathbf{f_k} = \lambda_k \mathbf{f}_k = (1 - \mathrm{e}^{\mathrm{i}k\Delta x})\,\mathbf{f}_k. \tag{19.35}$$

The eigenvalues of $-\frac{c}{\Delta x}M$ are complex valued

$$\sigma_k = -\frac{c}{\Delta x}(1 - \mathrm{e}^{\mathrm{i}k\Delta x}) = \frac{c}{\Delta x}(\cos k\Delta x - 1) + \mathrm{i}\frac{c}{\Delta x}\sin k\Delta x \tag{19.36}$$

and so is the dispersion

$$\omega_k = -\mathrm{i}\sigma_k = \frac{c}{\Delta x}\sin k\Delta x - \mathrm{i}\frac{c}{\Delta x}(\cos k\Delta x\ - 1). \tag{19.37}$$

If we take instead the forward difference we find similarly

$$\sigma_k = -\frac{c}{\Delta x}(\mathrm{e}^{-\mathrm{i}k\Delta x} - 1) = -\frac{c}{\Delta x}(\cos k\Delta x - 1) + \mathrm{i}\frac{c}{\Delta x}\sin k\Delta x \tag{19.38}$$

$$\omega_k = -\mathrm{i}\lambda_k = \frac{c}{\Delta x}\sin k\Delta x + \mathrm{i}\frac{c}{\Delta x}(\cos k\Delta x\ - 1). \tag{19.39}$$

19.2.1.2 Second Order Symmetric Difference

A symmetric difference quotient has higher error order and no diffusion term

$$\frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} = \frac{\sinh\left(\Delta x\frac{\partial}{\partial x}\right)}{\Delta x} f = \frac{\partial f}{\partial x} + \frac{(\Delta x)^2}{6}\frac{\partial^3 f}{\partial x^3} + \cdots \tag{19.40}$$

It provides the system of ordinary differential equations.

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{M-1} \\ f_M \end{pmatrix} = -\frac{c}{\Delta x}\begin{pmatrix} 0 & 1/2 & & & -1/2 \\ -1/2 & 0 & 1/2 & & \\ & \ddots & \ddots & \ddots & \\ & & -1/2 & 0 & 1/2 \\ 1/2 & & & -1/2 & 0 \end{pmatrix}\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{M-1} \\ f_M \end{pmatrix}. \tag{19.41}$$

The eigenpairs of M are easily found (see p. 221) from

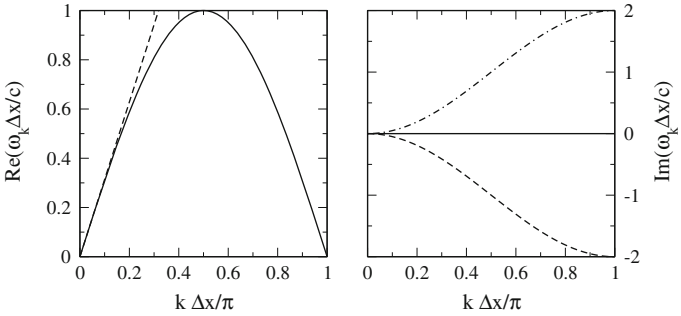


Fig. 19.2 (Dispersion of finite difference quotients) The dispersion of first order (19.37, 19.39) and second order (19.45) difference quotients is shown. ***Left*** the real part of ω_k (*full curve*) which is the same in all three cases is compared to the linear dispersion of the exact solution (*broken line*). ***Right*** imaginary part of ω_k (*dashed curve* = forward difference, *dash-dotted curve* = backward difference, *full line* = second order symmetric difference)

$$M\mathbf{f_k} = \frac{1}{2}\begin{pmatrix} \mathrm{e}^{-2\mathrm{i}k\Delta x} - \mathrm{e}^{-M\mathrm{i}k\Delta x} \\ \mathrm{e}^{-3\mathrm{i}k\Delta x} - \mathrm{e}^{-\mathrm{i}k\Delta x} \\ \vdots \\ \mathrm{e}^{-M\mathrm{i}k\Delta x} - \mathrm{e}^{-(M-2)\mathrm{i}k\Delta x} \\ \mathrm{e}^{-\mathrm{i}k\Delta x} - \mathrm{e}^{-(M-1)\mathrm{i}k\Delta x} \end{pmatrix} \tag{19.42}$$

$$= \frac{1}{2}(\mathrm{e}^{-\mathrm{i}k\Delta x} - \mathrm{e}^{\mathrm{i}k\Delta x})\mathbf{f}_k = -\mathrm{i}\sin k\Delta x\,\mathbf{f}_k \tag{19.43}$$

hence the eigenvalues of $-\frac{c}{\Delta x}M$ are purely imaginary and there is no damping (Fig. 19.2)

$$\sigma_k = \mathrm{i}\frac{c}{\Delta x}\sin k\Delta x \tag{19.44}$$

$$\omega_k = -\mathrm{i}\sigma_k = \frac{c}{\Delta x}\sin k\Delta x. \tag{19.45}$$

19.2.2 Explicit Methods

Time integration with an explicit forward Euler step proceeds according to (p. 293)

$$\mathbf{f}(t+\Delta t) = \mathbf{f}(t) + \frac{\partial f}{\partial t}\Delta t = \mathbf{f}(t) - c\frac{\partial f}{\partial x}\Delta t \tag{19.46}$$

and can be formulated in matrix notation as

$$\mathbf{f}(t + \Delta t) = A\,\mathbf{f}(t) = (1 - \alpha M)\mathbf{f}(t) \tag{19.47}$$

where the matrix M depends on the discretization method.

19.2.2.1 Forward in Time, Backward in Space

Combination with the backward difference quotient gives the FTBS (forward in time backward in space) method

$$f(x, t + \Delta t) = f(x, t) - \alpha\,(f(x, t) - f(x - \Delta x, t)) \tag{19.48}$$

with the so called Courant number [243][1]

$$\alpha = c\frac{\Delta t}{\Delta x}. \tag{19.49}$$

The eigenvalues of $1 - \alpha M$ are

$$\begin{aligned}\sigma_k &= 1 - \alpha(1 - \mathrm{e}^{\mathrm{i}k\Delta x}) \\ &= 1 - \alpha(1 - \cos k\Delta x) + \mathrm{i}\alpha \sin k\Delta x\end{aligned} \tag{19.50}$$

with absolute square (Fig. 19.3)

$$|\sigma_k|^2 = 1 + 2(\alpha^2 - \alpha)(1 - \cos k\Delta x). \tag{19.51}$$

Stability requires that $|\sigma_k| \leq 1$, i.e.

$$2(\alpha^2 - \alpha)(1 - \cos k\Delta x) \leq 0 \tag{19.52}$$

and, since $(1 - \cos k\Delta x) \geq 0$

$$(\alpha - 1)\alpha \leq 0 \tag{19.53}$$

with the solution[2]

$$0 \leq \alpha \leq 1. \tag{19.54}$$

[1] Also known as CFL (after the names of Courant, Friedrichs, Lewy).

[2] The so called Courant– Friedrichs– Lewy condition (or CFL condition).

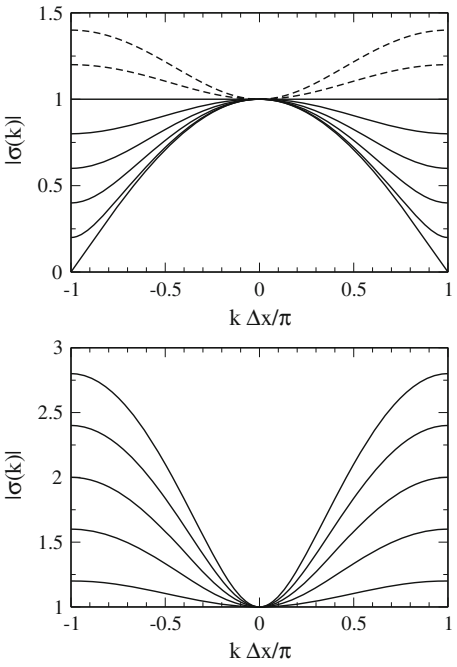


Fig. 19.3 (Stability of the FTBS method) ***Top*** the magnitude of the eigenvalue $|\sigma_k|$ is shown as a function of k for positive values of the Courant number (from *Bottom* to *Top*) $\alpha = 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1$. The method is stable for $\alpha \leq 1$ (*full curves*) and unstable for $\alpha > 1$ (*dashed curves*). ***Bottom*** the magnitude of the eigenvalue $|\sigma_k|$ is shown as a function of k for negative values of the Courant number (from *Bottom* to *Top*) $\alpha = -0.1, -0.3, -0.5, -0.7, -0.9$. The method is unstable for all $\alpha < 0$

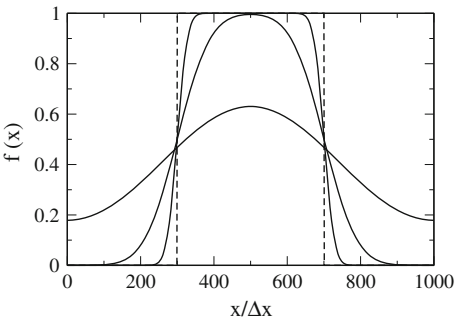


Fig. 19.4 (Performance of the FTBS method) An initially rectangular pulse (*dashed curve*) is propagated with the FTBS method ($\Delta x = 0.01$, $\Delta t = 0.005$, $\alpha = 0.5$). Due to numerical diffusion the shape is rapidly smoothened. Results are shown after 1,10 and 100 round trips (2000 time steps each)

The FTBS method works, but shows strong damping due to numerical diffusion (Fig. 19.4). Its dispersion relation is

$$\omega_k \Delta t = -\mathrm{i} \ln(\sigma_k) = -\mathrm{i} \ln\left([1 - \alpha(1 - \cos k\Delta x) + \mathrm{i}\alpha \sin k\Delta x]\right). \tag{19.55}$$

19.2.2.2 Forward in Time, Forward in Space

For a forward difference we obtain similarly

$$f(x, t + \Delta t) = f(x, t) - \alpha\left(f(x + \Delta x, t) - f(x, t)\right) \tag{19.56}$$

$$\sigma_k = 1 - \alpha(\mathrm{e}^{-\mathrm{i}k\Delta x} - 1) \tag{19.57}$$

$$|\sigma_k|^2 = 1 + 2(\alpha^2 + \alpha)(1 - \cos k\Delta x) \tag{19.58}$$

which is the same result as for the backward difference with α replaced by $-\alpha$.

19.2.2.3 Forward in Time, Centered in Space

For a symmetric difference quotient, the eigenvalues of M are purely imaginary, all eigenvalues of $(1 + \alpha M)$

$$\sigma_k = 1 + \mathrm{i}\alpha \sin k\Delta x \tag{19.59}$$

$$|\sigma_k|^2 = 1 + \alpha^2 \sin^2 k\Delta x \tag{19.60}$$

have absolute values $|\sigma_k| > 1$ and this method (FTCS, forward in time centered in space) is unstable (Fig. 19.5).

19.2.2.4 Lax-Friedrichs-Scheme

Stability can be achieved by a modification which is known as Lax-Friedrichs-scheme. The value of $f(x,t)$ is averaged over neighboring grid points

$$\begin{aligned} f(x, t+\Delta t) &= \frac{f(x+\Delta x) + f(x-\Delta x)}{2} - \frac{\alpha}{2}\left(f(x+\Delta x) - f(x-\Delta x)\right) \\ &= \left[\frac{1-\alpha}{2}\exp\left(\Delta x\frac{\partial}{\partial x}\right) + \frac{1+\alpha}{2}\exp\left(-\Delta x\frac{\partial}{\partial x}\right)\right] f(x,t) \\ &= 1 - \alpha\frac{\partial f}{\partial x} + \frac{(\Delta x)^2}{2}\frac{\partial^2 f}{\partial x^2} + \cdots . \end{aligned} \tag{19.61}$$

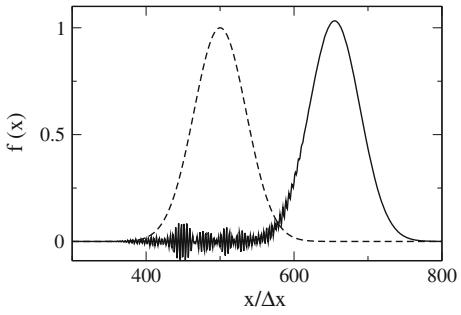


Fig. 19.5 (Instability of the FTCS method) An initially Gaussian pulse (*dashed curve*) is propagated with the FTCS method ($\Delta x = 0.01$, $\Delta t = 0.005$, $\alpha = 0.5$). Numerical instabilities already show up after 310 time steps and blow up rapidly afterwards

The error order is $O(\Delta x^2)$ as for the FTCS method but the leading error has now diffusive character. We obtain the system of equations

$$\mathbf{f}(t+\Delta t) = \frac{1}{2}\begin{pmatrix} & 1-\alpha & & & 1+\alpha \\ 1+\alpha & & 1-\alpha & & \\ & \ddots & \ddots & \ddots & \\ & & 1+\alpha & & 1-\alpha \\ 1-\alpha & & & 1+\alpha & \end{pmatrix} \mathbf{f}(t). \tag{19.62}$$

The eigenvalues follow from

$$(1-\alpha)\mathrm{e}^{-\mathrm{i}(n+1)k\Delta x} + (1+\alpha)\mathrm{e}^{-\mathrm{i}(n-1)k\Delta x} = \mathrm{e}^{-\mathrm{i}nk\Delta x}\left[(1-\alpha)\mathrm{e}^{-\mathrm{i})k\Delta x} + (1+\alpha)\mathrm{e}^{\mathrm{i}k\Delta x}\right] \tag{19.63}$$

and are given by

$$\begin{aligned} \sigma_k &= \frac{1}{2}\left[(1-\alpha)\mathrm{e}^{-\mathrm{i}k\Delta x} + (1+\alpha)\mathrm{e}^{\mathrm{i}k\Delta x}\right] \\ &= \cos k\Delta x + \mathrm{i}\alpha \sin k\Delta x. \end{aligned} \tag{19.64}$$

The absolute square is

$$\begin{aligned} |\sigma_k|^2 &= \frac{1}{4}\left[(1-\alpha)\mathrm{e}^{-\mathrm{i})k\Delta x} + (1+\alpha)\mathrm{e}^{\mathrm{i}k\Delta x}\right]\left[(1-\alpha)\mathrm{e}^{\mathrm{i})k\Delta x} + (1+\alpha)\mathrm{e}^{-\mathrm{i}k\Delta x}\right] \\ &= \frac{1}{4}\left[(1-\alpha)^2 + (1+\alpha)^2 + (1-\alpha^2)(\mathrm{e}^{-2\mathrm{i}k\Delta x} + \mathrm{e}^{2\mathrm{i}k\Delta x})\right] \\ &= \frac{1}{2}\left[1+\alpha^2 + (1-\alpha^2)\cos 2k\Delta x\right] \\ &= 1 - (1-\alpha^2)(\sin k\Delta x)^2 \end{aligned} \tag{19.65}$$

and the method is stable for

$$(1-\alpha^2)(\sin k\Delta x)^2 \geq 0 \tag{19.66}$$

which is the case if the Courant condition holds

$$|\alpha| \leq 1. \tag{19.67}$$

19.2.2.5 Lax-Wendroff Method

The Lax-Friedrichs method can be further improved to reduce numerical diffusion and obtain a method which is higher order in time. It is often used for hyperbolic partial differential equations. From the time derivative of the advection equation

$$\frac{\partial}{\partial t}\left(\frac{\partial f}{\partial t}\right) = -c\frac{\partial}{\partial x}\left(\frac{\partial f}{\partial t}\right) = c^2\frac{\partial^2 f}{\partial x^2} \tag{19.68}$$

we obtain the Taylor expansion

$$f(t+\Delta t) = f(t) - \Delta t\, c\frac{\partial f}{\partial x} + \frac{(\Delta t)^2}{2}c^2\frac{\partial^2 f}{\partial x^2} + \cdots \tag{19.69}$$

which we discretize to obtain the Lax-Wendroff scheme

$$\begin{aligned} f(x,t+\Delta t) &= f(x,t) - \Delta t\, c\frac{f(x+\Delta x,t) - f(x-\Delta x,t)}{2\Delta x} \\ &+ \frac{(\Delta t)^2}{2}c^2\frac{f(x+\Delta x,t) + f(x-\Delta x,t) - 2f(x,t)}{(\Delta x)^2}. \end{aligned} \tag{19.70}$$

This algorithm can also be formulated as a predictor-corrector method. First we calculate intermediate values at $t + \Delta t/2, x \pm \Delta x/2$ with the Lax method

$$\begin{aligned} f(x+\frac{\Delta x}{2}, t+\frac{\Delta t}{2}) &= \frac{f(x+\Delta x,t)+f(x,t)}{2} - c\Delta t\frac{f(x+\Delta x,t)-f(x,t)}{2\Delta x} \\ f(x-\frac{\Delta x}{2}, t+\frac{\Delta t}{2}) &= \frac{f(x,t)+f(x-\Delta x,t)}{2} - c\Delta t\frac{f(x,t)-f(x-\Delta x,t)}{2\Delta x} \end{aligned} \tag{19.71}$$

which are then combined in a corrector step

$$f(x,t+\Delta t) = f(x,t) - c\Delta t\frac{f(x+\frac{\Delta x}{2}, t+\frac{\Delta t}{2}) - f(x-\frac{\Delta t}{2}, t+\frac{\Delta t}{2})}{\Delta x}. \tag{19.72}$$

Insertion of the predictor step (19.71) shows the equivalence with (19.70).

$$\begin{aligned} f(x,t) - \frac{c\Delta t}{\Delta x}\Bigg[&\frac{f(x+\Delta x,t)+f(x,t)}{2} - c\Delta t\frac{f(x+\Delta x,t)-f(x,t)}{2\Delta x} \\ &- \frac{f(x,t)+f(x-\Delta x,t)}{2} + c\Delta t\frac{f(x,t)-f(x-\Delta x,t)}{2\Delta x}\Bigg] \end{aligned}$$

$$= f(x,t) - c\frac{\Delta t}{2\Delta x}\left[\frac{f(x+\Delta x,t) - f(x-\Delta x,t)}{2}\right]$$
$$+ \frac{c^2(\Delta t)^2}{2(\Delta x)^2}\left[f(x+\Delta x,t) - 2f(x,t) + f(x-\Delta x,t)\right]. \tag{19.73}$$

In matrix notation the Lax-Wendroff method reads

$$\mathbf{f}(t+\Delta t) = \begin{pmatrix} \ddots & & \\ \frac{\alpha+\alpha^2}{2} & 1-\alpha^2 & \frac{\alpha^2-\alpha}{2} \\ & & \ddots \end{pmatrix} \mathbf{f}(t) \tag{19.74}$$

with eigenvalues

$$\sigma_k = 1 - \alpha^2 + \frac{\alpha^2+\alpha}{2}\mathrm{e}^{\mathrm{i}k\Delta x} + \frac{\alpha^2-\alpha}{2}\mathrm{e}^{-\mathrm{i}k\Delta x}$$
$$= 1 - \alpha^2 + \alpha^2\cos k\Delta x + \mathrm{i}\alpha\sin k\Delta x \tag{19.75}$$

and

$$|\sigma_k|^2 = (1 + \alpha^2(\cos(k\Delta x) - 1))^2 + \alpha^2\sin^2 k\Delta x$$

$$= 1 - \alpha^2(1-\alpha^2)(1-\cos k\Delta x)^2 \tag{19.76}$$

which is ≤ 1 for

$$\alpha^2(1-\alpha^2)(1-\cos k\Delta x)^2 \geq 0 \tag{19.77}$$

which reduces to the CFL condition

$$|\alpha| \leq 1. \tag{19.78}$$

19.2.2.6 Leapfrog Method

The Leapfrog method uses symmetric differences for both derivatives

$$\frac{f(x,t+\Delta t) - f(x,t-\Delta t)}{2\Delta t} = -c\frac{f(x+\Delta x,t) - f(x-\Delta x,t)}{2\Delta x} \tag{19.79}$$

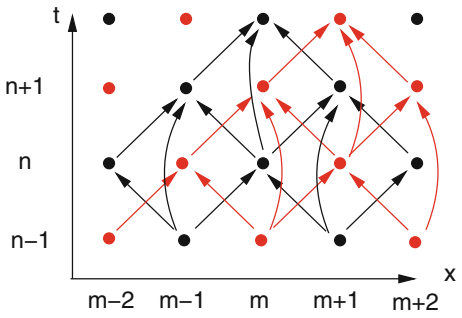


Fig. 19.6 Leapfrog method for advection

to obtain a second order two step method

$$f(x, t + \Delta t) = f(x, t - \Delta t) - \alpha\,[f(x + \Delta x, t) - f(x - \Delta x, t)] \tag{19.80}$$

on a grid which is equally spaced in space and time.

The calculated data form two independent subgrids (Fig. 19.6). For long integration times this can lead to problems if the results on the subgrids become different due to numerical errors. Introduction of a diffusive coupling term can help to avoid such difficulties.

To analyze stability, we write the two step method as a one step method, treating the values at even and odd time steps as independent variables

$$g_m^n = f(m\Delta x, 2n\Delta t) \quad h_m^n = f(m\Delta x, (2n+1)\Delta t) \tag{19.81}$$

for which the Leapfrog scheme becomes

$$h_m^n = h_m^{n-1} - \alpha(g_{m+1}^n - g_{m-1}^n) \tag{19.82}$$

$$g_m^{n+1} = g_m^n - \alpha(h_{m+1}^n - h_{m-1}^n). \tag{19.83}$$

Combining this two equations we obtain the one step iteration

$$g_m^n = f_m^{2n} \quad h_m^n = f_m^{2n+1}$$

$$\begin{pmatrix} h^n \\ g^{n+1} \end{pmatrix} = \begin{pmatrix} 1 & \\ -\alpha M & 1 \end{pmatrix} \begin{pmatrix} 1 & -\alpha M \\ & 1 \end{pmatrix} \begin{pmatrix} h^{n-1} \\ g^n \end{pmatrix} = \begin{pmatrix} 1 & -\alpha M \\ -\alpha M & 1 + \alpha^2 M^2 \end{pmatrix} \begin{pmatrix} h^{n-1} \\ g^n \end{pmatrix}. \tag{19.84}$$

The eigenvalues of the matrix M are $\lambda_k = -2\mathrm{i} \sin k\Delta x$, hence the eigenvalues of the Leapfrog scheme

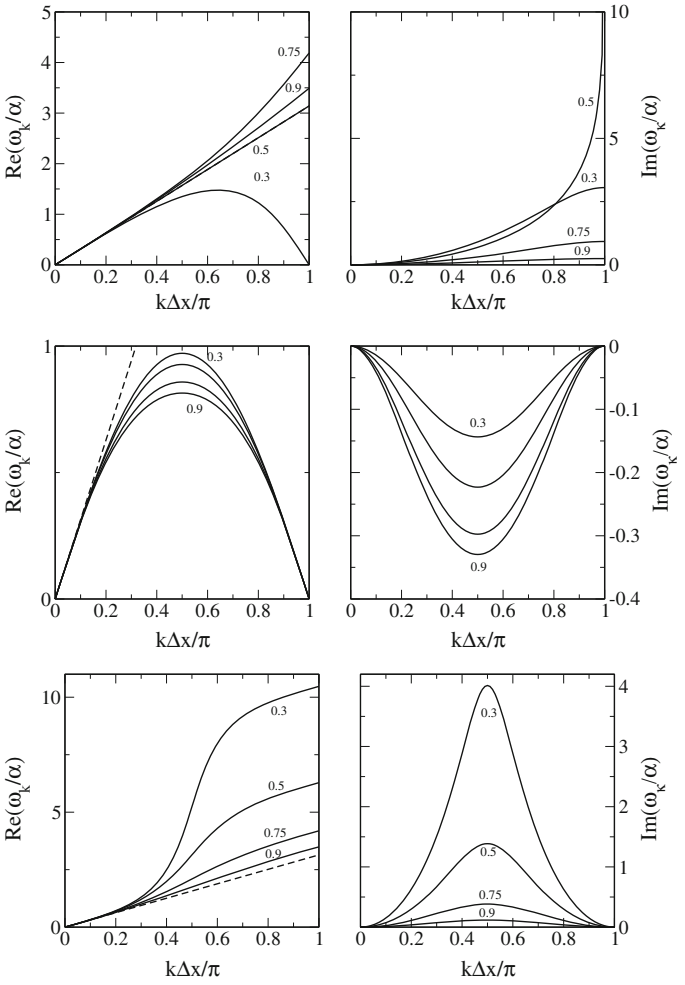


Fig. 19.7 (Dispersion of first order explicit methods) Real (***Left***) and imaginary (***Right***) part of ω_k are shown for the first order explicit FTBS (***Top***), FTCS (***Middle***) and Lax-Friedrich (***Bottom***) methods for values of $\alpha = 0.3, 0.5, 0.75, 0.9$

$$\sigma_k = 1 + \frac{\alpha^2\lambda^2}{2} \pm \sqrt{\alpha^2\lambda^2 + \frac{\alpha^4\lambda^4}{4}}$$
$$= 1 - 2\alpha^2 \sin^2 k\Delta x \pm 2\sqrt{\alpha^2 \sin^2 k\Delta x(\alpha^2 \sin^2 k\Delta x - 1)}. \tag{19.85}$$

For $|\alpha| \leq 1$ the squareroot is purely imaginary and

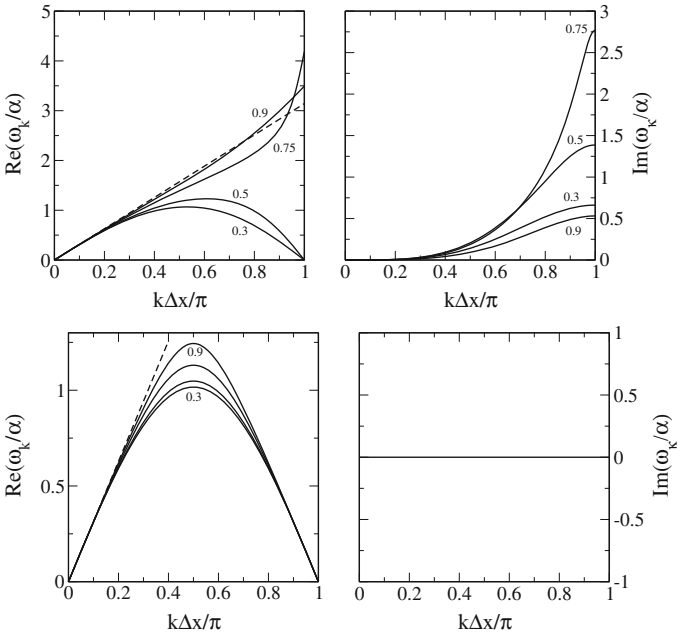


Fig. 19.8 (Dispersion of second order explicit methods) Real (***Left***) and imaginary (***Right***) part of ω_k are shown for the second order explicit Lax-Wendroff (***Top***) and leapfrog (***Bottom***) methods for values of $\alpha = 0.3, 0.5, 0.75, 0.9$

$$|\sigma_k|^2 = 1$$

i.e. the method is unconditionally stable and diffusionless. The dispersion

$$2\omega\Delta t = -\mathrm{i}\ln\sigma_k = \arctan\left(\pm\frac{2\sqrt{\alpha^2\sin^2 k\Delta x - \alpha^4\sin^4 k\Delta x}}{1-2\alpha^2\sin^2 k\Delta x}\right) \tag{19.86}$$

has two branches. Expanding for small $k\Delta x$ we find

$$\omega \approx \pm ck + \cdots . \tag{19.87}$$

Only the plus sign corresponds to a physical mode. The negative sign corresponds to the so called computational mode which can lead to artificial rapid oscillations. These can be removed by special filter algorithms [244, 245].

Figures. 19.7, 19.8 and 19.9 show a comparison of different explicit methods.

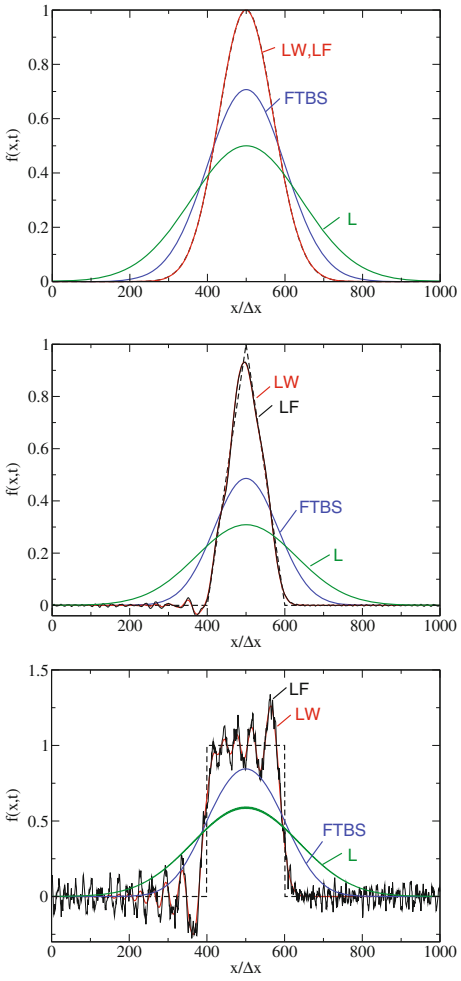


Fig. 19.9 (Comparison of explicit methods) The results from the FTBS, Lax-Friedrichs(L, *green*), Lax-Wendroff (LW, *black*) and leapfrog (LF, *red*) methods after 10 roundtrips are shown. Initial values (*black dashed curves*) are Gaussian (***Top***), triangular (***Middle***) and rectangular (***Bottom***). $\Delta x = 0.01$, $\Delta t = 0.005$, $\alpha = 0.5$

19.2.3 Implicit Methods

Time integration by implicit methods improves the stability but can be time consuming since inversion of a matrix is involved. A simple Euler backward step (13.4) takes the derivative at $t + \Delta t$

$$\mathbf{f}(t + \Delta t) = \mathbf{f}(t) - \alpha M \mathbf{f}(t + \Delta t) \tag{19.88}$$

which can be formally written as

$$\mathbf{f}(t + \Delta t) = (1 + \alpha M)^{-1} \mathbf{f}(t). \tag{19.89}$$

The Crank–Nicolson method (13.5) takes the average of implicit and explicit Euler step

$$\mathbf{f}(t+\Delta t) = \mathbf{f}(t) - \frac{\alpha}{2} M \left[\mathbf{f}(t+\Delta t) + \mathbf{f}(t)\right] \tag{19.90}$$

$$\mathbf{f}(t+\Delta t) = \left(1 + \frac{\alpha}{2} M\right)^{-1} \left(1 - \frac{\alpha}{2} M\right) \mathbf{f}(t). \tag{19.91}$$

Both methods require to solve a linear system of equations.

19.2.3.1 First Order Implicit Method

Combining the back steps in time and space we obtain the BTBS (backward in time, backward in space) method

$$\mathbf{f}(t+\Delta t) = \left(1 + \alpha \begin{pmatrix} 1 & & & -1 \\ -1 & 1 & & \\ & \ddots & \ddots & \ddots \\ & & -1 & 1 \\ & & & -1 \quad 1 \end{pmatrix}\right)^{-1} \mathbf{f}(t). \tag{19.92}$$

The tridiagonal structure of the matrix $1 + \alpha M$ simplifies the solution of the system. The eigenvalues of $(1 + \alpha M)^{-1}$ are

$$\sigma_k = \frac{1}{1 + \alpha(1 - \mathrm{e}^{\mathrm{i}k\Delta x})} \tag{19.93}$$

$$|\sigma_k|^2 = \frac{1}{(1+\alpha)^2 + \alpha^2 - 2\alpha(1+\alpha)\cos(k\Delta x)} \leq 1 \tag{19.94}$$

and the method is unconditionally stable.

19.2.3.2 Second Order Crank–Nicolson Method

The Crank–Nicolson method with the symmetric difference quotient gives a second order method

$$\left(1 + \frac{\alpha}{2} M\right) \mathbf{f}(t+\Delta t) = \left(1 - \frac{\alpha}{2} M\right) \mathbf{f}(t). \tag{19.95}$$

The eigenvalues of $(1+\frac{\alpha}{2}M)^{-1}(1-\frac{\alpha}{2}M)$ are

$$\sigma_k = \frac{1+\frac{\alpha}{2}\mathrm{i}\sin k\Delta x}{1-\frac{\alpha}{2}\mathrm{i}\sin k\Delta x} \tag{19.96}$$

$$= \frac{1-\frac{\alpha^2}{4}\sin^2 k\Delta x + \mathrm{i}\alpha\sin k\Delta x}{1+\frac{\alpha^2}{4}\sin^2 k\Delta x} \tag{19.97}$$

with

$$|\sigma_k|^2 = 1. \tag{19.98}$$

There is no damping but strong dispersion at larger values of α, slowing down partial waves with higher $k-$values (Fig. 19.11)

This method is unconditionally stable (Fig. 19.10). It may, however, show oscillations if the time step is chosen too large (Fig. 19.11). It can be turned into an explicit method by an iterative approach (iterated Crank–Nicolson method, see p. 475), which avoids solution of a linear system but is only stable for $\alpha \leq 1$.

19.2.4 Finite Volume Methods

Finite volume methods [246] are very popular for equations in the form of a conservation law

$$\frac{\partial f(\mathbf{x},t)}{\partial t} = -\mathrm{div}\mathbf{J}(f(\mathbf{x},t)). \tag{19.99}$$

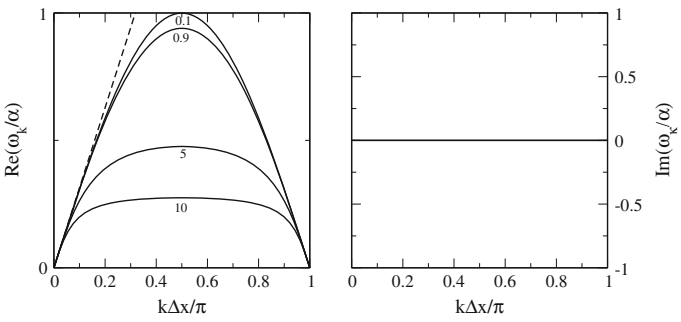


Fig. 19.10 (Dispersion of the Crank–Nicolson method) Real (***Left***) and imaginary part (***Right***) part of ω_k are shown for $\alpha = 0.1, 9, 5, 10$. This implicit method is stable for $\alpha > 1$ but dispersion becomes noticeable at higher values

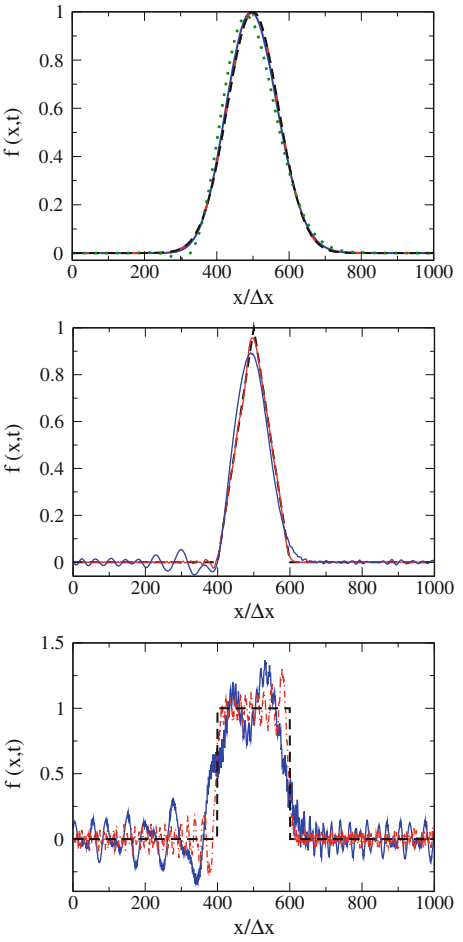


Fig. 19.11 (Performance of the Crank–Nicolson method) Results of the implicit Crank–Nicolson method after 10 roundtrips are shown. Initial values (*black dashed curves*) are Gaussian (***Top***), triangular (***Middle***) and rectangular (***Bottom***). $\Delta x = 0.01$, $\Delta t = 0.01$ ($\alpha = 1$, *red dash-dotted curve*) $\Delta t = 0.1 (\alpha = 10$, *blue full curve)* $\Delta t = 0.2 (\alpha = 20$, *green dotted curve*, only shown for Gaussian initial values)

In one dimension the control volumes are intervals, in the simplest case centered at equidistant grid points x_n

$$V_n = [x_n - \frac{\Delta x}{2}, x_n + \frac{\Delta x}{2}]. \tag{19.100}$$

Integration over one interval gives an equation for the cell average

$$\frac{\partial \overline{f}_n(t)}{\partial t} = \frac{1}{\Delta x} \frac{\partial}{\partial t} \int_{x_n - \Delta x/2}^{x_n + \Delta x/2} f(x,t) = -\frac{1}{\Delta x} \int_{x_n - \Delta x/2}^{x_n + \Delta x/2} \frac{\partial}{\partial x} J(x,t)$$

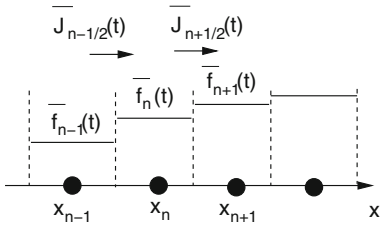


Fig. 19.12 (Finite volume method) The change of the cell average $\overline{f}_n$ is balanced by the fluxes through the cell interfaces $\overline{J}_{n\pm 1/2}$

$$= -\frac{1}{\Delta x}\left[J(x_n + \frac{\Delta x}{2}, t) - J(x_n - \frac{\Delta x}{2}, t)\right]. \tag{19.101}$$

Formally this can be integrated

$$\overline{f}_n(t+\Delta t) - \overline{f}(t) = -\frac{1}{\Delta x}\left[\int_t^{t+\Delta t} J\left(x_n + \frac{\Delta x}{2}, t'\right) dt - \int_t^{t+\Delta t} J\left(x_n - \frac{\Delta x}{2}, t'\right) dt\right] \tag{19.102}$$

and with the temporally averaged fluxes through the control volume boundaries

$$\overline{J}_{n\pm 1/2}(t) = \frac{1}{\Delta t}\int_t^{t+\Delta t} J\left(x_n \pm \frac{\Delta x}{2}, t'\right) dt \tag{19.103}$$

it takes the simple form (Fig. 19.12)

$$\overline{f}_n(t+\Delta t) = \overline{f}_n(t) - \frac{\Delta t}{\Delta x}\left[\overline{J}_{n+1/2}(t) - \overline{J}_{n-1/2}(t)\right]. \tag{19.104}$$

A numerically scheme for a conservation law is called conservative if it can be written in this form with some approximation $\overline{J}_{n\pm 1/2}(t)$ of the fluxes at the cell interfaces. Conservative schemes are known to converge to a weak solution of the conservation law under certain conditions (stability and consistency).

To obtain a practical scheme, we have to approximate the fluxes in terms of the cell averages. Godunov's famous method [247] uses a piecewise constant approximation of $f(x,t)$

$$f(x,t) \approx \overline{f}_n(t) \quad \text{for } x_{n-1/2} \le x \le x_{n+1/2}. \tag{19.105}$$

To construct the fluxes, we have to solve the Riemann problem[3]

$$\frac{\partial f}{\partial t} = -\frac{\partial J}{\partial x} \tag{19.106}$$

[3] A conservation law with discontinuous initial values.

with discontinuous initial values

$$f(x,t) = \begin{cases} \overline{f}_n(t) & \text{if } x \le x_{n+1/2} \\ \overline{f}_{n+1}(t) & \text{if } x \ge x_{n+1/2} \end{cases} \tag{19.107}$$

in the time interval

$$t \le t' \le t + \Delta t. \tag{19.108}$$

For the linear advection equation with $J(x,t) = cf(x,t)$ the solution is easily found as the discontinuity just moves with constant velocity. For $c\Delta t \le \Delta x$, d'Alembert's method gives

$$f(x_n + \frac{\Delta x}{2}, t') = \overline{f}_n(t) \tag{19.109}$$

and the averaged flux is

$$\overline{J}_{n+1/2}(t) = \frac{c}{\Delta t}\int_t^{t+\Delta t} f(x_n + \frac{\Delta x}{2}, t')dt = c\overline{f}_n(t). \tag{19.110}$$

Finally we end up with the FTBS upwind scheme (Sect. 19.2.2)

$$\overline{f}_n(t+\Delta t) = \overline{f}_n(t) - \frac{c\Delta t}{\Delta x}\left[\overline{f}_n(t) - \overline{f}_{n-1}(t)\right]. \tag{19.111}$$

For general conservation laws, approximate methods have to be used to solve the Riemann problem (so called Riemann solvers [248]).

Higher order methods can be obtained by using higher order piecewise interpolation functions. If we interpolate linearly

$$f(x,t) \approx \overline{f}_n(t) + (x - x_n)\sigma_n(t) \quad \text{for } x_{n-1/2} \le x \le x_{n+1/2}$$

the solution to the Riemann problem is

$$f(x_n + \frac{\Delta x}{2}, t') = f(x_n + \frac{\Delta x}{2} - c(t'-t), t) = \overline{f}_n(t) + \left[\frac{\Delta x}{2} - c(t'-t)\right]\sigma_n(t)$$

$$= \overline{f}_n(t) + \left[\frac{\Delta x}{2} - c(t'-t)\right]\frac{\overline{f}_{n+1} - \overline{f}_{n-1}}{2\Delta x}.$$

The time averaged fluxes are

$$\overline{J}_{n+1/2} = c\overline{f}_n(t) + c\left[\frac{\Delta x}{2} - c\frac{\Delta t}{2}\right]\sigma_n(t)$$

and we end up with

$$\overline{f}_n(t+\Delta t) = \overline{f}_n(t) - \frac{c\Delta t}{\Delta x}\left[\overline{f}_n(t) - \overline{f}_{n-1}(t) + \left(\frac{\Delta x}{2} - c\frac{\Delta t}{2}\right)(\sigma_n - \sigma_{n-1})\right]. \tag{19.112}$$

If we take the slopes from the forward differences

$$\sigma_n = \frac{\overline{f}_{n+1} - \overline{f}_n}{\Delta x} \tag{19.113}$$

we end up with

$$\begin{aligned}
\overline{f}_n(t+\Delta t) &= \overline{f}_n(t) - \frac{c\Delta t}{\Delta x}\left[\overline{f}_n(t) - \overline{f}_{n-1}(t)\right] - \frac{c\Delta t}{2\Delta x}(\Delta x - c\Delta t)\frac{\overline{f}_{n+1} - 2\overline{f}_n + \overline{f}_{n-1}}{\Delta x} \\
&= \overline{f}_n(t) - \frac{c\Delta t}{\Delta x}\left[\overline{f}_n(t) - \overline{f}_{n-1}(t) + \frac{\overline{f}_{n+1} - 2\overline{f}_n + \overline{f}_{n-1}}{2} + \frac{(c\Delta t)^2}{2\Delta x}\frac{\overline{f}_{n+1} - 2\overline{f}_n + \overline{f}_{n-1}}{\Delta x}\right. \\
&= \overline{f}_n(t) - \frac{c\Delta t}{\Delta x}\left[\frac{\overline{f}_{n+1} - \overline{f}_{n-1}}{2} + \frac{(c\Delta t)^2}{2\Delta x}\frac{\overline{f}_{n+1} - 2\overline{f}_n + \overline{f}_{n-1}}{\Delta x}\right.
\end{aligned} \tag{19.114}$$

i.e. we end up with the Lax-Wendroff scheme. Different approximations for the slopes are possible (backward difference, symmetric differences) leading to the schemes of Fromm and Beam-Warming.

19.2.5 Taylor–Galerkin Methods

The error order of finite difference methods can be improved by using a finite element discretization [249, 250]. We start from the Taylor series expansion in the time step

$$f(t+\Delta t) = f(t) + \Delta t\frac{\partial f}{\partial t} + \frac{(\Delta t)^2}{2}\frac{\partial^2 f}{\partial t^2} + \frac{(\Delta t)^3}{6}\frac{\partial^3 f}{\partial t^3} + \cdots \tag{19.115}$$

which is also the basis of the Lax-Wendroff method (19.70) and make use of the advection equation to substitute time derivatives by spatial derivatives

$$f(t+\Delta t) = f(t) - \Delta t\, c \frac{\partial f}{\partial x} + \frac{(\Delta t)^2}{2} c^2 \frac{\partial^2 f}{\partial x^2} + \frac{(\Delta t)^3}{6} c^2 \frac{\partial^3 f}{\partial t \partial x^2} + \cdots \tag{19.116}$$

where we use a mixed expression for the third derivative to allow the usage of linear finite elements. We approximate the third derivative as

$$\frac{\partial^3 f}{\partial t \partial x^2} = \frac{\partial^2}{\partial x^2} \frac{f(x, t+\Delta t) - f(x,t)}{\Delta t} + \cdots \tag{19.117}$$

and obtain an implicit expression which is a third order accurate extension of the Lax-Wendroff scheme

$$\left[1 - \frac{(\Delta t)^2}{6} c^2 \frac{\partial^2}{\partial x^2}\right] (f(x, t+\Delta t) - f(x,t)) = -\Delta t\, c \frac{\partial f}{\partial x} + \frac{(\Delta t)^2}{2} c^2 \frac{\partial^2 f}{\partial x^2}. \tag{19.118}$$

Application of piecewise linear elements on a regular grid (p. 282) produces the following Lax-Wendroff Taylor-Galerkin scheme

$$\left[1 + \frac{1}{6}\left(1 - \alpha^2\right) D_2\right] (\mathbf{f}(t+\Delta t) - \mathbf{f}(t)) = \left[-\alpha\, M_1 + \frac{\alpha^2}{2} M_2\right] \mathbf{f}(t). \tag{19.119}$$

The Taylor-Galerkin method can be also combined with other schemes like leapfrog or Crank–Nicolson [250]. It can be generalized to advection-diffusion problems and it can be turned into an explicit scheme [251] by series expansion of the inverse in

$$\mathbf{f}(t+\Delta t) = \mathbf{f}(t) + \left[1 + \frac{1}{6}\left(1 - \alpha^2\right) M_2\right]^{-1} \left[-\alpha\, M_1 + \frac{\alpha^2}{2} M_2\right] \mathbf{f}(t). \tag{19.120}$$

The eigenvalues are

$$\sigma_k = 1 + \frac{\alpha \mathrm{i} \sin k\Delta x - 2\alpha^2 \sin^2 \frac{k\Delta x}{2}}{1 - \frac{2}{3}(1-\alpha^2) \sin^2 \frac{k\Delta x}{2}}. \tag{19.121}$$

The method is stable for $|\alpha| \leq 1$. Due to its higher error order it shows less dispersion and damping than the Lax-Wendroff method (Fig. 19.13) and provides superior results (Fig. 19.14).

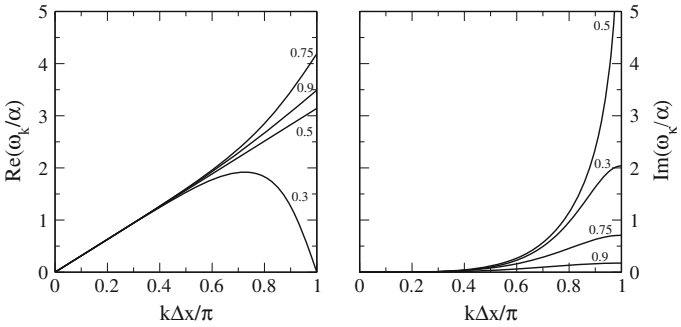


Fig. 19.13 (Dispersion of the Taylor-Galerkin Lax-Wendroff method) Real (***Left***) and imaginary part (***Right***) part of ω_k are shown for $\alpha = 0.3, 0.5, 0.75, 0.9$

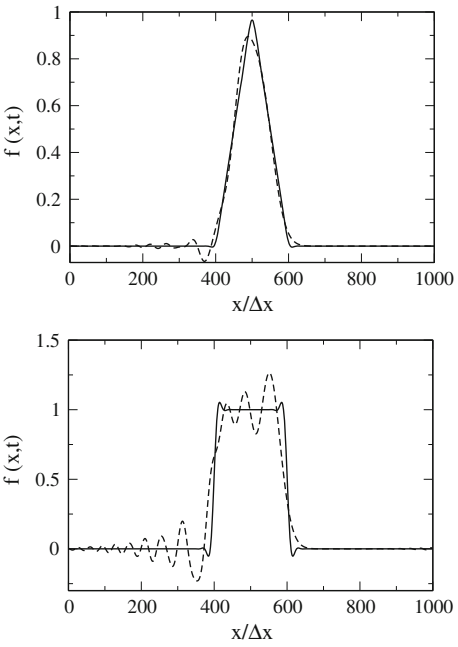


Fig. 19.14 (Performance of the Taylor-Galerkin Lax-Wendroff method) Results of the Lax-Wendroff (*dashed curves*) and Taylor-Galerkin Lax-Wendroff (*full curves*) methods are compared after 25 roundtrips (2000 steps each). $\Delta x = 0.01$, $\Delta t = 0.005$, $\alpha = 0.5$

19.3 Advection in More Dimensions

While in one dimension for an incompressible fluid $c = \text{const}$, this is not necessarily the case in more dimensions.

19.3.1 Lax–Wendroff Type Methods

In more dimensions we substitute

$$\frac{\partial f}{\partial t} = -\mathbf{u}\nabla f \tag{19.122}$$

$$\frac{\partial^2 f}{\partial t^2} = \frac{\partial}{\partial t}\left(\frac{\partial f}{\partial t}\right) = -\mathbf{u}\nabla\left(\frac{\partial f}{\partial t}\right) = (\mathbf{u}\nabla)(\mathbf{u}\nabla) f \tag{19.123}$$

in the series expansion

$$f(t+\Delta t) - f(t) = \Delta t \frac{\partial f}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2 f}{\partial t^2} + \cdots \tag{19.124}$$

to obtain a generalization of the Taylor expansion (19.69)

$$f(t+\Delta t) - f(t) = -\Delta t\, \mathbf{u}\nabla f + \frac{(\Delta t)^2}{2}(\mathbf{u}\nabla)(\mathbf{u}\nabla) f + \cdots \tag{19.125}$$

which then has to be discretized in space by the usual methods of finite differences or finite elements [250]. Other one-dimensional schemes like leapfrog also can be generalized to more dimensions.

19.3.2 Finite Volume Methods

In multidimensions we introduce a, not necessarily regular, mesh of control volumes V_i. The surface of V_i is divided into interfaces $A_{i,\alpha}$ to the neighboring cells. Application of the integral form of the continuity equation (19.3) gives

$$\frac{\partial}{\partial t}\int_{V_i} dV\, f(\mathbf{r},t) = -\oint_{\partial V_i} \mathbf{J}(\mathbf{r},t) d\mathbf{A} \tag{19.126}$$

and after time integration

$$\overline{f}_i(t+\Delta t) - \overline{f}(t) = -\Delta t \sum_\alpha \overline{J}_{i,\alpha}(t) \tag{19.127}$$

with the cell averages

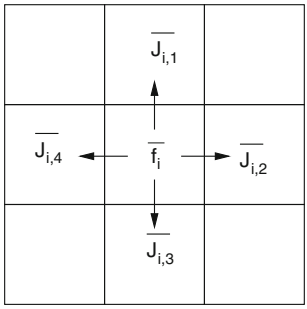


Fig. 19.15 Averaged fluxes in 2 dimensions

$$\overline{f}_i(t) = \frac{1}{V_i}\frac{\partial}{\partial t}\int_{V_i} dV\ f(\mathbf{r},t) \tag{19.128}$$

and the flux averages

$$\overline{J}_{i,\alpha}(t) = \frac{1}{\Delta t}\frac{1}{V_i}\sum_{\alpha}\int_t^{t+\Delta t} dt' \quad \oint_{A_{i,\alpha}} \mathbf{J}(\mathbf{r},t')d\mathbf{A}. \tag{19.129}$$

For a regular mesh with cubic control volumes the sum is over all cell faces

$$\begin{aligned}\overline{f}_{ijk}(t+\Delta t) = \overline{f}_{ijk}(t) - \Delta t\left[\overline{J}_{i+1/2,j,k}(t) + \overline{J}_{i,j+1/2,k}(t) + \overline{J}_{i,j,k+1/2,}(t)\right.\\ \left. -\overline{J}_{i-1/2,j,k}(t) - -\overline{J}_{i,j-1/2,k}(t) - \overline{J}_{i,j,k-1/2}(t)\right].\end{aligned} \tag{19.130}$$

The function values have to be reconstructed from the cell averages, e.g. piecewise constant

$$f(\mathbf{r},t) = \overline{f}_i(t) \quad \text{for } \mathbf{r} \in V_i \tag{19.131}$$

and the fluxes through the cell surface approximated in a suitable way, e.g. constant over a surface element (Fig. 19.15)

$$\mathbf{J}(\mathbf{r},t) = \mathbf{J}_{i,\alpha}(t) \quad \text{for } \mathbf{r} \in A_{i,\alpha}. \tag{19.132}$$

Then the Riemann problem has to be solved approximately to obtain the fluxes for times $t \ldots t + \Delta t$. This method is also known as *reconstruct evolve average* (REA) method. An overview of average flux methods is presented in [252].

19.3.3 Dimensional Splitting

Splitting methods are very useful to divide a complicated problem into simpler steps. The time evolution of the advection equation can be written as a sum of three contributions[4]

$$\frac{\partial f}{\partial t} = -\text{div}(\mathbf{u} f) = -\frac{\partial (u_x f)}{\partial x} - \frac{\partial (u_y f)}{\partial y} - \frac{\partial (u_z f)}{\partial z} \tag{19.133}$$

or, for an incompressible fluid

$$\frac{\partial f}{\partial t} = -\mathbf{u}\ \text{grad}\ f = -u_x \frac{\partial f}{\partial x} - u_y \frac{\partial f}{\partial y} - u_z \frac{\partial f}{\partial z} \tag{19.134}$$

which has the form

$$\frac{\partial f}{\partial t} = A f = (A_x + A_y + A_z) f. \tag{19.135}$$

The time evolution can be approximated by

$$f(t + \Delta t) = \mathrm{e}^{\Delta t A} f(t) \approx \mathrm{e}^{\Delta t\, A_x} \mathrm{e}^{\Delta t\, A_y} \mathrm{e}^{\Delta t\, A_z} f(t) \tag{19.136}$$

i.e. by a sequence of one-dimensional time evolutions. Accuracy can be improved by applying a symmetrical Strang-splitting

$$f(t + \Delta t) \approx \mathrm{e}^{\Delta t/2\, A_x} \mathrm{e}^{\Delta t/2\, A_y} \mathrm{e}^{\Delta t\, A_z} \mathrm{e}^{\Delta t/2\, A_y} \mathrm{e}^{\Delta t/2\, A_x} f(t). \tag{19.137}$$

Problems

Problem 19.1 Advection in one Dimension

In this computer experiment we simulate 1-dimensional advection with periodic boundary conditions. Different initial values (rectangular, triangular or Gaussian pulses of different widths) and methods (**F**orward in **T**ime **B**ackward in **S**pace, Lax-Friedrichs, leapfrog, Lax-Wendroff, implicit Crank–Nicolson, Taylor-Galerkin Lax-Wendroff) can be compared. See also Figs. 19.4, 19.11, 19.14 and 19.9.

[4] This is also the case if a diffusion term $D\left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}\right)$ is included.

Chapter 20
Waves

Waves are oscillations that move in space and time and are able to transport energy from one point to another. Quantum mechanical wavefunctions are discussed in Chap. 23. In this chapter we simulate classical waves which are, for instance, important in acoustics and electrodynamics. We use the method of finite differences to discretize the wave equation in one spatial dimension

$$\frac{\partial^2}{\partial t^2} f(t,x) = c^2 \frac{\partial^2}{\partial x^2} f(t,x). \tag{20.1}$$

Numerical solutions are obtained by an eigenvector expansion using trigonometric functions or by time integration. Accuracy and stability of different methods are compared. The wave function is second order in time and can be integrated directly with a two step method. Alternatively, it can be converted into a first order system of equations of double dimension. Here, the velocity appears explicitly and velocity dependent damping can be taken into account. Finally, the second order wave equation can be replaced by two coupled first order equations for two variables (like velocity and density in case of acoustic waves), which can be solved by quite general methods. We compare the leapfrog, Lax–Wendroff and Crank–Nicolson methods. Only the Crank–Nicolson method is stable for Courant numbers $\alpha > 1$. It is an implicit method and can be solved iteratively. In a series of computer experiments we simulate waves on a string. We study reflection at an open or fixed boundary and at the interface between two different media. We compare dispersion and damping for different methods.

20.1 Classical Waves

In classical physics there are two main types of waves:

Electromagnetic waves do not require a medium. They are oscillations of the electromagnetic field and propagate also in vacuum. As an example consider a plane wave which propagates in x-direction and is linearly polarized (Fig. 20.1). The electric and magnetic field have the form

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_20

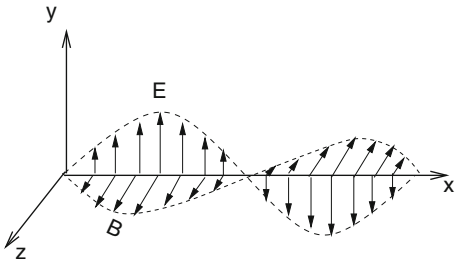


Fig. 20.1 Electromagnetic wave

$$\mathbf{E} = \begin{pmatrix} 0 \\ E_y(x,t) \\ 0 \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} 0 \\ 0 \\ B_z(x,t) \end{pmatrix}. \tag{20.2}$$

Maxwell's equations read in the absence of charges and currents

$$\text{div}\mathbf{E} = \text{div}\ \mathbf{B} = 0, \quad \text{rot}\ \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \quad \text{rot}\mathbf{B} = \mu_0 \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t}. \tag{20.3}$$

The fields (20.2) have zero divergence and satisfy the first two equations. Application of the third and fourth equation gives

$$\frac{\partial E_y}{\partial x} = -\frac{\partial B_z}{\partial t} \quad -\frac{\partial B_z}{\partial x} = \mu_0 \varepsilon_0 \frac{\partial E_y}{\partial t} \tag{20.4}$$

which can be combined to a one-dimensional wave-equation

$$\frac{\partial^2 E_y}{\partial t^2} = c^2 \frac{\partial^2 E_y}{\partial x^2} \tag{20.5}$$

with velocity $c = (\mu_0 \varepsilon_0)^{-1/2}$.

Mechanical waves propagate through an elastic medium like air, water or an elastic solid. The material is subject to external forces deforming it and elastic forces which try to restore the deformation. As a result the atoms or molecules move around their equilibrium positions. As an example consider one-dimensional acoustic waves in an organ pipe (Fig. 20.2):

A mass element

$$dm = \varrho dV = \varrho A dx \tag{20.6}$$

at position x experiences an external force due to the air pressure which, according to Newton's law changes the velocity v of the element as described by Euler's equation[1]

[1] we consider only small deviations from the equilibrium values ϱ_0, p_0, $v_0 = 0$.

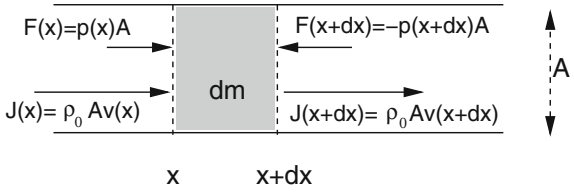


Fig. 20.2 (Acoustic waves in one dimension) A mass element $dm = \varrho A dx$ at position x experiences a total force $F = F(x) + F(x + dx) = -A\frac{\partial p}{\partial x}dx$. Due to the conservation of mass the change of the density $\frac{\partial \varrho}{\partial t}$ is given by the net flux $J = J(x) - J(x + dx) = -\varrho_0 A\frac{\partial v}{\partial x}dx$

$$\varrho_0 \frac{\partial}{\partial t} v = -\frac{\partial p}{\partial x}. \tag{20.7}$$

The pressure is a function of the density

$$\frac{p}{p_0} = \left(\frac{\varrho}{\varrho_0}\right)^n \quad \left(\frac{\mathrm{d}p}{\mathrm{d}\varrho}\right)_0 = n\frac{p_0}{\varrho_0} = c^2 \tag{20.8}$$

where $n = 1$ for an isothermal ideal gas and $n \approx 1.4$ for air under adiabatic conditions (no heat exchange), therefore

$$\varrho_0 \frac{\partial}{\partial t} v = -c^2 \frac{\partial \varrho}{\partial x}. \tag{20.9}$$

From the conservation of mass the continuity equation (12.10) follows

$$\frac{\partial}{\partial t}\varrho = -\varrho_0 \frac{\partial}{\partial x} v. \tag{20.10}$$

Combining the time derivative of (20.10) and the spatial derivative of (20.9) we obtain again the one-dimensional wave equation

$$\frac{\partial^2}{\partial t^2}\varrho = c^2 \frac{\partial^2}{\partial x^2}\varrho. \tag{20.11}$$

The wave-equation can be factorized as

$$\left(\frac{\partial}{\partial t} + c\frac{\partial}{\partial x}\right)\left(\frac{\partial}{\partial t} - c\frac{\partial}{\partial x}\right)\varrho = \left(\frac{\partial}{\partial t} - c\frac{\partial}{\partial x}\right)\left(\frac{\partial}{\partial t} + c\frac{\partial}{\partial x}\right)\varrho = 0 \tag{20.12}$$

which shows that solutions of the advection equation

$$\left(\frac{\partial}{\partial t} \pm c\frac{\partial}{\partial x}\right)\varrho = 0 \tag{20.13}$$

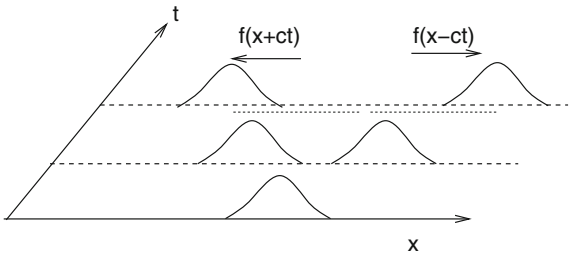


Fig. 20.3 d'Alembert solution to the wave equation

are also solutions of the wave equation, which have the form

$$\varrho = f(x \pm ct). \tag{20.14}$$

In fact a general solution of the wave equation is given according to d'Alembert as the sum of two waves running to the left and right side with velocity c and a constant envelope (Fig. 20.3)

$$\varrho = f_1(x + ct) + f_2(x - ct). \tag{20.15}$$

A special solution of this kind is the plane wave solution

$$f(x,t) = e^{i\omega t \pm ikx}$$

with the dispersion relation

$$\omega = ck. \tag{20.16}$$

20.2 Spatial Discretization in One Dimension

We use the simplest finite difference expression for the spatial derivative (Sects. 3.4 and 12.2)

$$\frac{\partial^2}{\partial x^2} f(x,t) = \frac{f(t, x + \Delta x) + f(t, x - \Delta x) - 2f(t,x)}{\Delta x^2} + O(\Delta x^2) \tag{20.17}$$

and a regular grid

$$x_m = m\Delta x \quad m = 1, 2 \ldots M \tag{20.18}$$

$$f_m = f(x_m). \tag{20.19}$$

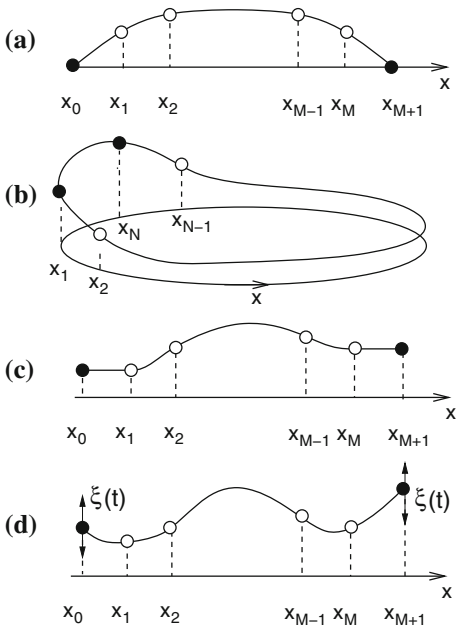


Fig. 20.4 (Boundary Conditions for 1-dimensional waves) Additional boundary points x_0, x_{M+1} are used to realize the boundary conditions. (**a**) **Fixed boundaries** $f(x_0) = 0$ $\frac{\partial^2}{\partial x^2} f(x_1) = \frac{1}{\Delta x^2}(f(x_2) - 2f(x_1))$ or $f(x_{M+1}) = 0, \frac{\partial^2}{\partial x^2} f(x_M) = \frac{1}{\Delta x^2}(f(x_{M-1}) - 2f(x_M))$.(**b**) **Periodic boundary conditions** $x_0 \equiv x_M$, $\frac{\partial^2}{\partial x^2} f(x_1) = \frac{1}{\Delta x^2}(f(x_2) + f(x_M) - 2f(x_1))\, x_{M+1} \equiv x_1, \frac{\partial^2}{\partial x^2} f(x_M) = \frac{1}{\Delta x^2}(f(x_{M-1}) + f(x_1) - 2f(x_{NM}))$.(**c**) **Open boundaries** $\frac{\partial}{\partial x} f(x_1) = \frac{f(x_2)-f(x_0)}{2\Delta x} = 0, \frac{\partial^2}{\partial x^2} f(x_1) = \frac{1}{\Delta x^2}(2f(x_2) - 2f(x_1))$ or $\frac{\partial}{\partial x} f(x_M) = \frac{f(x_{M+1})-f(x_{M-1})}{2\Delta x} = 0 \frac{\partial^2}{\partial x^2} f(x_M) = \frac{1}{\Delta x^2}(2f((x_{M-1}) - 2f(x_M))$.(**d**) **Moving boundaries** $f(x_0, t) = \xi_0(t) \frac{\partial^2}{\partial x^2} f(x_1) = \frac{1}{\Delta x^2}(f(x_2) - 2f(x_1) + \xi_0(t))$ or $f(x_{M+1}, t) = \xi_{M+1}(t)$, $\frac{\partial^2}{\partial x^2} f(x_M) = \frac{1}{\Delta x^2}(f(x_{M-1}) - 2f(x_M) + \xi_{N+1}(t))$

This turns the wave equation into the system of ordinary differential equations (Sect. 12.2.3)

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2} f_m = c^2 \frac{f_{m+1} + f_{m-1} - 2f_m}{\Delta x^2} \tag{20.20}$$

where f_0 and f_{M+1} have to be specified by suitable boundary conditions (Fig. 20.4). In matrix notation we have

$$\mathbf{f}(t) = \begin{pmatrix} f_1(t) \\ \vdots \\ f_M(t) \end{pmatrix} \tag{20.21}$$

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathbf{f}(t) = A\mathbf{f}(t) + \mathbf{S}(t) \tag{20.22}$$

where for

$$\text{fixed boundaries} \quad A = \begin{pmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{pmatrix} \frac{c^2}{\Delta x^2} \quad \mathbf{S}(t) = 0 \tag{20.23}$$

$$\text{periodic boundaries}^2 \quad A = \begin{pmatrix} -2 & 1 & & & & 1 \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ 1 & & & & 1 & -2 \end{pmatrix} \frac{c^2}{\Delta x^2} \quad \mathbf{S}(t) = 0 \tag{20.24}$$

$$\text{open boundaries} \quad A = \begin{pmatrix} -2 & 2 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 2 & -2 \end{pmatrix} \frac{c^2}{\Delta x^2} \quad \mathbf{S}(t) = 0 \tag{20.25}$$

$$\text{moving boundaries} \quad A = \begin{pmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{pmatrix} \frac{c^2}{\Delta x^2} \quad \mathbf{S}(t) = \begin{pmatrix} \xi_0(t) \\ 0 \\ \vdots \\ \\ \vdots \\ 0 \\ \xi_{M+1}(t) \end{pmatrix}. \tag{20.26}$$

A combination of different boundary conditions for both sides is possible.

Equation (20.20) corresponds to a series of mass points which are connected by harmonic springs (Fig. 20.5), a model, which is used in solid state physics to describe longitudinal acoustic waves [253].

[2] This corresponds to the boundary condition $f_0 = f_2$, $\frac{\partial}{\partial x} f(x_1) = 0$. Alternatively we could use $f_0 = f_1$, $\frac{\partial}{\partial x} f(x_{1/2}) = 0$ which replaces the 2 s in the first and last row by 1 s.

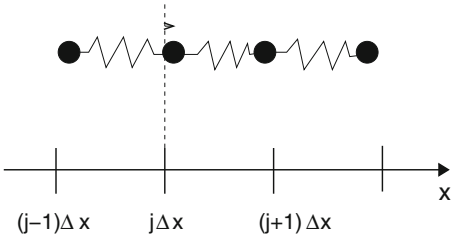


Fig. 20.5 (Atomistic model for longitudinal waves) A set of mass points m is connected by springs with stiffness K. The elongation of mass point number j from its equilibrium position $x_j = j\Delta x$ is ξ_j. The equations of motion $m\ddot{\xi}_j = -K(\xi_j - \xi_{j-1}) - K(\xi_j - \xi_{j+1})$ coincide with (20.20) with a velocity of $c = \Delta x\sqrt{\frac{k\Delta x}{m}}$

20.3 Solution by an Eigenvector Expansion

For fixed boundaries (20.20) reads in matrix form

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathbf{f}(t) = A\mathbf{f}(t) \tag{20.27}$$

with the vector of function values:

$$\mathbf{f}(t) = \begin{pmatrix} f_1(t) \\ \vdots \\ f_M(t) \end{pmatrix} \tag{20.28}$$

and the matrix

$$A = \begin{pmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{pmatrix} \frac{c^2}{\Delta x^2} \tag{20.29}$$

which can be diagonalized exactly (Sect. 10.3). The two boundary points $f(0) = 0$ and $f((M+1)\Delta x) = 0$ can be added without any changes. The eigenvalues are

$$\lambda = 2\frac{c^2}{\Delta x^2}(\cos(k\Delta x) - 1) = -\frac{4c^2}{\Delta x^2}\sin^2\left(\frac{k\Delta x}{2}\right) = (i\omega_k)^2 \quad k\Delta x = \frac{\pi l}{(M+1)}, l = 1 \ldots M \tag{20.30}$$

with the frequencies

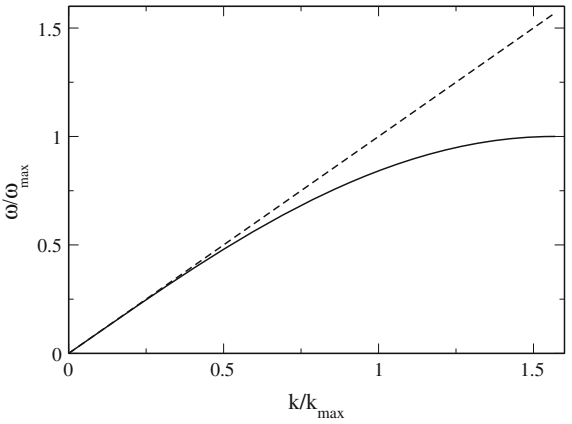


Fig. 20.6 (Dispersion of the discrete wave equation) The dispersion of the discrete wave equation approximates the linear dispersion of the continuous wave equation only at small values of k. At $k_{max} = \pi/\Delta x$ it saturates at $\omega_{max} = 2c/\Delta x = (2/\pi)\, ck_{max}$

$$\omega_k = \frac{2c}{\Delta x} \sin\left(\frac{k\Delta x}{2}\right). \tag{20.31}$$

This result deviates from the dispersion relation of the continuous wave equation (20.11) $\omega_k = ck$ and approximates it only for $k\Delta x \ll 1$ (Fig. 20.6).

The general solution has the form (Sect. 12.2.4)

$$f_n(t) = \sum_{l=1}^{M} \left(C_{l+} e^{i\omega_l t} + C_{l-} e^{-i\omega_l t}\right) \sin\left(m \frac{\pi l}{(M+1)}\right). \tag{20.32}$$

The initial amplitudes and velocities are

$$\begin{aligned} f_n(t=0) &= \sum_{l=1}^{M} (C_{l+} + C_{l-}) \sin\left(m \frac{\pi l}{(M+1)}\right) = F_m \\ \frac{\mathrm{d}}{\mathrm{d}t} f_m(t=0, x_m) &= \sum_{l=1}^{M} i\omega_l \ (C_{l+} - C_{l-}) \sin\left(m \frac{\pi l}{(M+1)}\right) = G_m \end{aligned} \tag{20.33}$$

with F_m and G_m given. Different eigenfunctions of a tridiagonal matrix are mutually orthogonal

$$\sum_{m=1}^{M} \sin\left(m \frac{\pi l}{M+1}\right) \sin\left(m \frac{\pi l'}{M+1}\right) = \frac{M}{2} \delta_{l,l'} \tag{20.34}$$

and the coefficients $C_{l\pm}$ follow from a discrete Fourier transformation:

$$\widetilde{F_l} = \frac{1}{M}\sum_{m=1}^{M} \sin\left(m\frac{\pi l}{N+1}\right) F_m$$
$$= \frac{1}{M}\sum_{m=1}^{M}\sum_{l'=1}^{M}(C_{l'+} + C_{l'-}) \sin\left(m\frac{\pi l'}{M+1}\right) \sin\left(m\frac{\pi l}{M+1}\right) = \frac{1}{2}(C_{l+} + C_{l-}) \tag{20.35}$$

$$\widetilde{G_l} = \frac{1}{M}\sum_{m=1}^{M} \sin\left(m\frac{\pi l}{N+1}\right) G_n$$
$$= \frac{1}{M}\sum_{m=1}^{M}\sum_{l'=1}^{NM} i\omega_l\,(C_{l+} - C_{l-}) \sin\left(m\frac{\pi l'}{M+1}\right) \sin\left(m\frac{\pi l}{M+1}\right) = \frac{1}{2} i\omega_l (C_{l+} - C_{l-}) \tag{20.36}$$

$$C_{l+} = \widetilde{F_l} + \frac{1}{i\omega_l}\widetilde{G_l}$$
$$C_{l-} = \widetilde{F_l} - \frac{1}{i\omega_l}\widetilde{G_l}. \tag{20.37}$$

Finally the explicit solution of the wave equation is

$$f_m(t) = \sum_{l=1}^{M} 2(\widetilde{F_l}\cos(\omega_l\, t) + \frac{\widetilde{G_l}}{\omega_l}\sin(\omega_l\, t)) \sin\left(m\frac{\pi l}{M+1}\right). \tag{20.38}$$

Periodic or open boundaries can be treated similarly as the matrices can be diagonalized exactly (Sect. 10.3). For moving boundaries the expansion coefficients are time dependent (Sect. 12.2.4).

20.4 Discretization of Space and Time

Using the finite difference expression also for the second time derivative the fully discretized wave equation is

$$\frac{f(t+\Delta t, x) + f(t-\Delta t, x) - 2f(t,x)}{\Delta t^2}$$
$$= c^2 \frac{f(t, x+\Delta x) + f(t, x-\Delta x) - 2f(t,x)}{\Delta x^2} + O(\Delta x^2, \Delta t^2). \tag{20.39}$$

For a plane wave

$$f = \mathrm{e}^{\mathrm{i}(\omega t - kx)} \tag{20.40}$$

we find

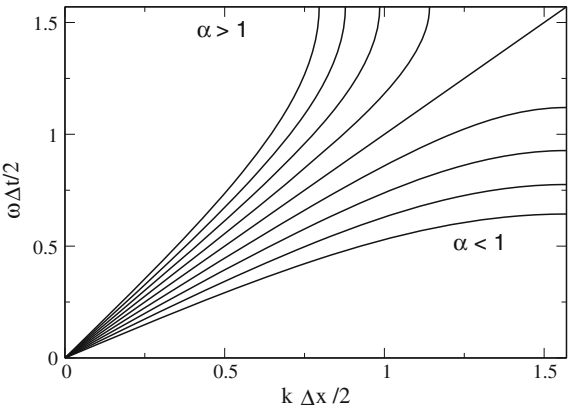


Fig. 20.7 (Dispersion of the discrete wave equation) Only for $\alpha = 1$ or for small values of $k\Delta x$ and $\omega\Delta t$ is the dispersion approximately linear. For $\alpha < 1$ only frequencies $\omega < \omega_{\max} = 2\arcsin(\alpha)/\Delta t$ are allowed whereas for $\alpha > 1$ the range of k-values is bounded by $k_{\max} = 2\arcsin(1/\alpha)/\Delta x$

$$e^{i\omega\Delta t} + e^{-i\omega\Delta t} - 2 = c^2 \frac{\Delta t^2}{\Delta x^2} \left(e^{ik\Delta x} + e^{-ik\Delta x} - 2\right) \tag{20.41}$$

which can be written as

$$\sin \frac{\omega\Delta t}{2} = \alpha \sin \frac{k\Delta x}{2} \tag{20.42}$$

with the so-called Courant-number [243]

$$\alpha = c\frac{\Delta t}{\Delta x}. \tag{20.43}$$

From (20.42) we see that the dispersion relation is linear only for $\alpha = 1$. For $\alpha \neq 1$ not all values of ω and k allowed (Fig. 20.7).

20.5 Numerical Integration with a Two-Step Method

We solve the discrete wave equation (20.39) with fixed or open boundaries for

$$\begin{aligned} f(t + \Delta t, x) = {} & 2f(t, x)(1 - \alpha^2) + \alpha^2(f(t, x + \Delta x) + f(t, x - \Delta x)) \\ & - f(t - \Delta t, x) + O(\Delta t^2, \Delta x^2) \end{aligned} \tag{20.44}$$

on the regular grids

$$x_m = m\Delta x \quad m = 1, 2 \ldots M \tag{20.45}$$

$$t_n = n\Delta t \quad n = 1, 2 \ldots N \tag{20.46}$$

$$\mathbf{f}_n = \begin{pmatrix} f_1^n \\ \vdots \\ f_M^n \end{pmatrix} = \begin{pmatrix} f(t_n, x_1) \\ \vdots \\ f(t_n, x_M) \end{pmatrix} \tag{20.47}$$

by applying the iteration

$$f_m^{n+1} = 2(1-\alpha^2) f_m^n + \alpha^2 f_{m+1}^n + \alpha^2 f_{m-1}^n - f_m^{n-1}. \tag{20.48}$$

This is a two-step method which can be rewritten as a one-step method of double dimension

$$\begin{pmatrix} \mathbf{f}_{n+1} \\ \mathbf{f}_n \end{pmatrix} = T \begin{pmatrix} \mathbf{f}_n \\ \mathbf{f}_{n-1} \end{pmatrix} = \begin{pmatrix} 2+\alpha^2 M & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{f}_n \\ \mathbf{f}_{n-1} \end{pmatrix} \tag{20.49}$$

with the tridiagonal matrix

$$M = \begin{pmatrix} -2 & a_1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & a_N & -2 \end{pmatrix} \tag{20.50}$$

where a_1 and a_N have the values 1 for a fixed or 2 for an open end.
The matrix M has eigenvalues (Sect. 10.3)

$$\lambda = 2\cos(k\Delta x) - 2 = -4\sin^2\left(\frac{k\Delta x}{2}\right). \tag{20.51}$$

To simulate excitation of waves by a moving boundary we add one grid point with given elongation $\xi_0(t)$ and change the first equation into

$$f(t_{n+1}, x_1) = 2(1-\alpha^2) f(t_n, x_1) + \alpha^2 f(t_n, x_2) + \alpha^2 \xi_0(t_n) - f(t_{n-1}, x_1). \tag{20.52}$$

Repeated iteration gives the series of function values

$$\begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_0 \end{pmatrix}, \begin{pmatrix} \mathbf{f}_2 \\ \mathbf{f}_1 \end{pmatrix} = T \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_0 \end{pmatrix}, \begin{pmatrix} \mathbf{f}_3 \\ \mathbf{f}_2 \end{pmatrix} = T^2 \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_0 \end{pmatrix}, \cdots \tag{20.53}$$

A necessary condition for stability is that all eigenvalues of T have absolute values smaller than one. Otherwise small perturbations would be amplified. The eigenvalue equation for T is

$$\begin{pmatrix} 2+\alpha^2 M - \sigma & -1 \\ 1 & -\sigma \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{20.54}$$

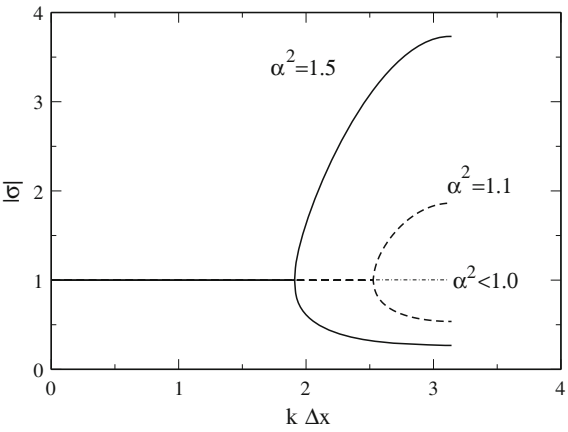


Fig. 20.8 (Stability regions of the two-step method) Instabilities appear for $|\alpha| > 1$. One of the two eigenvalues σ becomes unstable ($|\sigma| > 1$) for waves with large k-values

We substitute the solution of the second equation

$$u = \sigma v \tag{20.55}$$

into the first equation and use the eigenvectors of M (Sect. 10.3) to obtain the eigenvalue equation

$$(2 + \alpha^2\lambda - \sigma)\sigma v - v = 0. \tag{20.56}$$

Hence σ is one of the two roots of

$$\sigma^2 - \sigma(\alpha^2\lambda + 2) + 1 = 0 \tag{20.57}$$

which are given by (Fig. 20.8)

$$\sigma = 1 + \frac{\alpha^2\lambda}{2} \pm \sqrt{\left(\frac{\alpha^2\lambda}{2} + 1\right)^2 - 1}. \tag{20.58}$$

From

$$\lambda = -4\sin^2\left(\frac{k\Delta x}{2}\right)$$

we find

$$-4 < \lambda < 0 \tag{20.59}$$

$$1 - 2\alpha^2 < \frac{\alpha^2\lambda}{2} + 1 < 1 \tag{20.60}$$

and the square root in (20.58) is imaginary if

$$-1 < \frac{\alpha^2\lambda}{2} + 1 < 1 \tag{20.61}$$

which is the case for

$$\sin^2\left(\frac{k\Delta x}{2}\right)\alpha^2 < 1. \tag{20.62}$$

This holds for all k only if

$$|\alpha| < 1. \tag{20.63}$$

But then

$$|\sigma|^2 = \left(1 + \frac{\alpha^2\lambda}{2}\right)^2 + \left(1 - \left(\frac{\alpha^2\lambda}{2} + 1\right)^2\right) = 1 \tag{20.64}$$

and the algorithm is (conditionally) stable. If on the other hand $|\alpha| > 1$ then for some k-values the square root is real. Here we have

$$1 + \frac{\alpha^2\lambda}{2} < -1 \tag{20.65}$$

and finally

$$1 + \frac{\alpha^2\lambda}{2} - \sqrt{\left(1 + \frac{\alpha^2\lambda}{2}\right)^2 - 1} < -1 \tag{20.66}$$

which shows that instabilities are possible in this case.

20.6 Reduction to a First Order Differential Equation

A general method to reduce the order of an ordinary differential equation (or a system of such) introduces the time derivatives as additional variables (Chap. 13). The spatially discretized one-dimensional wave equation (20.22) can be transformed into a system of double dimension

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{f}(t) = \mathbf{v}(t) \tag{20.67}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{v}(t) = \frac{c^2}{\Delta x^2} M\mathbf{f}(t) + \mathbf{S}(t). \tag{20.68}$$

We use the improved Euler method (Sect. 13.5)

$$\mathbf{f}(t+\Delta t) = \mathbf{f}(t) + \mathbf{v}(t+\frac{\Delta t}{2})\Delta t + O(\Delta t^3) \tag{20.69}$$

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t) + \left[\frac{c^2}{\Delta x^2} M\mathbf{f}(t+\frac{\Delta t}{2}) + \mathbf{S}(t+\frac{\Delta t}{2})\right]\Delta t + O(\Delta t^3) \tag{20.70}$$

and two different time grids

$$\mathbf{f}_n = \mathbf{f}(t_n) \quad \mathbf{S}_n = \mathbf{S}(t_n) \quad n = 0, 1\ldots \tag{20.71}$$

$$\mathbf{f}(t_{n+1}) = \mathbf{f}(t_n) + \mathbf{v}(t_{n+1/2})\Delta t \tag{20.72}$$

$$\mathbf{v}_n = \mathbf{v}(t_{n-1/2}) \quad n = 0, 1\ldots \tag{20.73}$$

$$\mathbf{v}(t_{n+1/2}) = \mathbf{v}(t_{n-1/2}) + \left[\frac{c^2}{\Delta x^2} M\mathbf{f}(t_n) + \mathbf{S}(t_n)\right]\Delta t. \tag{20.74}$$

We obtain a leapfrog (Fig. 20.9) like algorithm (p. 398)

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \left[\frac{c^2}{\Delta x^2} M\mathbf{f}_n + \mathbf{S}_n\right]\Delta t \tag{20.75}$$

$$\mathbf{f}_{n+1} = \mathbf{f}_n + \mathbf{v}_{n+1}\Delta t \tag{20.76}$$

where the updated velocity (20.75) has to be inserted into (20.76). This can be combined into the iteration

$$\begin{pmatrix}\mathbf{f}_{n+1}\\ \mathbf{v}_{n+1}\end{pmatrix} = \begin{pmatrix}\mathbf{f}_n + \mathbf{v}_n\Delta t + \left[\frac{c^2}{\Delta x^2} M\mathbf{f}_n + \mathbf{S}_n\right]\Delta t^2\\ \mathbf{v}_n + \left[\frac{c^2}{\Delta x^2} M\mathbf{f}_n + \mathbf{S}_n\right]\Delta t\end{pmatrix} = \begin{pmatrix}1 + \frac{c^2\Delta t^2}{\Delta x^2} M & \Delta t\\ \frac{c^2\Delta t}{\Delta x^2} M & 1\end{pmatrix}\begin{pmatrix}\mathbf{f}_n\\ \mathbf{v}_n\end{pmatrix} + \begin{pmatrix}\mathbf{S}_n\Delta t^2\\ \mathbf{S}_n\Delta t\end{pmatrix}. \tag{20.77}$$

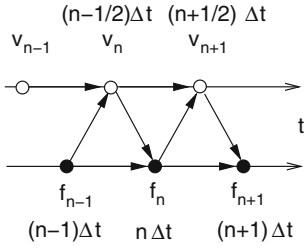


Fig. 20.9 Leapfrog method

Since the velocity appears explicitly we can easily add a velocity dependent damping like

$$-\gamma v(t_n, x_m) \tag{20.78}$$

which we approximate by

$$-\gamma v(t_n - \frac{\Delta t}{2}, x_m) \tag{20.79}$$

under the assumption of weak damping

$$\gamma \Delta t \ll 1. \tag{20.80}$$

To study the stability of this algorithm we consider the homogeneous problem with fixed boundaries. With the Courant number $\alpha = \frac{c\Delta t}{\Delta x}$ (20.77) becomes

$$\begin{pmatrix} \mathbf{f}_{n+1} \\ \mathbf{v}_{n+1} \end{pmatrix} = \begin{pmatrix} 1 + \alpha^2 M & \Delta t(1 - \gamma \Delta t) \\ \frac{\alpha^2}{\Delta t} M & 1 - \gamma \Delta t \end{pmatrix} \begin{pmatrix} \mathbf{f}_n \\ \mathbf{v}_n \end{pmatrix}. \tag{20.81}$$

Using the eigenvectors and eigenvalues of M (Sect. 10.3)

$$\lambda = -4 \sin^2 \left(\frac{k \Delta x}{2} \right) \tag{20.82}$$

we find the following equation for the eigenvalues σ:

$$\begin{aligned} (1 + \alpha^2 \lambda - \sigma) u + \Delta t (1 - \gamma \Delta t) v &= 0 \\ \alpha^2 \lambda u + \Delta t (1 - \gamma \Delta t - \sigma) v &= 0. \end{aligned} \tag{20.83}$$

Solving the second equation for u and substituting into the first equation we have

$$[(1 + \alpha^2 \lambda - \sigma) \frac{\Delta t}{-\alpha^2 \lambda} (1 - \gamma \Delta t - \sigma) + \Delta t (1 - \gamma \Delta t)] = 0 \tag{20.84}$$

hence

$$\begin{aligned} (1 + \alpha^2 \lambda - \sigma)(1 - \gamma \Delta t - \sigma) - \alpha^2 \lambda (1 - \gamma \Delta t) &= 0 \\ \sigma^2 - \sigma(2 - \gamma \Delta t + \alpha^2 \lambda) + (1 - \gamma \Delta t) &= 0 \\ \sigma = 1 - \frac{\gamma \Delta t}{2} + \frac{\alpha^2 \lambda}{2} \pm \sqrt{\left(1 - \frac{\gamma \Delta t}{2} + \frac{\alpha^2 \lambda}{2}\right)^2 - (1 - \gamma \Delta t)}. \end{aligned} \tag{20.85}$$

Instabilities are possible if the square root is real and $\sigma < -1$. ($\sigma > 1$ is not possible). This is the case for

$$-1 + \frac{\gamma \Delta t}{2} \approx -\sqrt{1 - \gamma \Delta t} < 1 - \frac{\gamma \Delta t}{2} + \frac{\alpha^2 \lambda}{2} < \sqrt{1 - \gamma \Delta t} \approx 1 - \frac{\gamma \Delta t}{2} \tag{20.86}$$

$$-2 + \gamma\Delta t < \frac{\alpha^2\lambda}{2} < 0. \tag{20.87}$$

The right inequality is satisfied, hence it remains

$$\alpha^2 \sin^2\left(\frac{k\Delta x}{2}\right) < 1 - \frac{\gamma\Delta t}{2}. \tag{20.88}$$

This holds for all k-values if it holds for the maximum of the sine-function

$$\alpha^2 < 1 - \frac{\gamma\Delta t}{2}. \tag{20.89}$$

This shows that inclusion of the damping term even favors instabilities.

20.7 Two Variable Method

For the 1-dimensional wave equation (20.11) there exists another possibility to reduce the order of the time derivative by splitting it up into two first order equations similar to (20.9, 20.10)

$$\frac{\partial}{\partial t} f(t,x) = c\frac{\partial}{\partial x} g(t,x) \tag{20.90}$$

$$\frac{\partial}{\partial t} g(t,x) = c\frac{\partial}{\partial x} f(t,x). \tag{20.91}$$

Several algorithms can be applied to solve these equations [254]. We discuss only methods which are second order in space and time and are rather general methods to solve partial differential equations. The boundary conditions need some special care. For closed boundaries with $f(x_0) = 0$ obviously $\frac{\partial f}{\partial t}(x_0) = 0$ whereas $\frac{\partial f}{\partial x}(x_0)$ is finite. Hence a closed boundary for $f(t,x)$ is connected with an open boundary for $g(t,x)$ with $\frac{\partial g}{\partial x}(x_0) = 0$ and vice versa. This is well known from acoustics (Fig. 20.10).

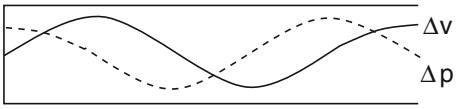


Fig. 20.10 (Standing waves in an organ pipe) At the closed (***Left***) end the amplitude of the longitudinal velocity is zero whereas the amplitudes of pressure and density changes are extremal. This is reversed at the open (***Right***) end

20.7.1 Leapfrog Scheme

We use symmetric differences (Sect. 3.2) for the first derivatives

$$\frac{f(t+\frac{\Delta t}{2},x)-f(t-\frac{\Delta t}{2},x)}{\Delta t}=c\frac{g(t,x+\frac{\Delta x}{2})-g(x-\frac{\Delta x}{2})}{\Delta x}+O(\Delta x^2,\Delta t^2) \tag{20.92}$$

$$\frac{g(t+\frac{\Delta t}{2},x)-g(t-\frac{\Delta t}{2},x)}{\Delta t}=c\frac{f(t,x+\frac{\Delta x}{2})-f(x-\frac{\Delta x}{2})}{\Delta x}+O(\Delta x^2,\Delta t^2) \tag{20.93}$$

to obtain the following scheme

$$g((t_{n+1/2},x_{m+1/2})=g(t_{n-1/2},x_{m+1/2})+\alpha\left(f(t_n,x_{m+1})\right)-f(t_n,x_{m-1}) \tag{20.94}$$

$$f(t_{n+1},x_m)=f(t_n,x_m)+\alpha\left(g(t_{n+1/2},x_{m+1/2})-g((t_{n+1/2},x_{m-1/2})\right). \tag{20.95}$$

Using different time grids for the two variables

$$\mathbf{f}_n=\begin{pmatrix}f_1^n\\ \vdots\\ f_M^n\end{pmatrix}=\begin{pmatrix}f(t_n,x_1)\\ \vdots\\ f(t_n,x_M)\end{pmatrix}\quad \mathbf{g}_n=\begin{pmatrix}g_1^n\\ \vdots\\ g_M^n\end{pmatrix}=\begin{pmatrix}g((t_{n-1/2},x_{1/2})\\ \vdots\\ g(t_{n-1/2},x_{M-1/2})\end{pmatrix} \tag{20.96}$$

this translates into the algorithm (Fig. 20.11)

$$g_m^{n+1}=g_m^n+\alpha\left(f_m^n-f_{m-1}^n\right) \tag{20.97}$$

$$f_m^{n+1}=f_m^n+\alpha\left(g_{m+1}^{n+1}-g_m^{n+1}\right)=f_m^n+\alpha\left(g_{m+1}^n-g_m^n\right)+\alpha^2(f_{m+1}^n-2f_m^n+f_{m-1}^n). \tag{20.98}$$

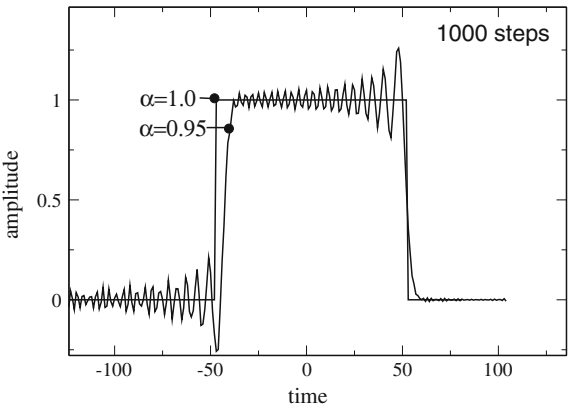


Fig. 20.11 (Simulation with the leapfrog method) A rectangular pulse is simulated with the two-variable leapfrog method. While for $\alpha = 1$ the pulse shape has not changed after 1000 steps, for smaller values the short wavelength components are lost due to dispersion

To analyze the stability we insert

$$f_m^n = u\mathrm{e}^{an}\mathrm{e}^{\mathrm{i}km\Delta x} \quad g_m^n = v\mathrm{e}^{an}\mathrm{e}^{\mathrm{i}km\Delta x} \tag{20.99}$$

and obtain the equations

$$Gv = v + \alpha u(1 - \mathrm{e}^{-\mathrm{i}k\Delta x}) \tag{20.100}$$

$$Gu = u + \alpha v(\mathrm{e}^{\mathrm{i}k\Delta x} - 1) + \alpha^2 u(2\cos k\Delta x - 2) \tag{20.101}$$

which in matrix form read

$$G\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 1 + \alpha^2(2\cos k\Delta x - 2) & \alpha(\mathrm{e}^{\mathrm{i}k\Delta x} - 1) \\ \alpha(1 - \mathrm{e}^{-\mathrm{i}k\Delta x}) & 1 \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix}. \tag{20.102}$$

The maximum amplification factor G is given by the largest eigenvalue, which is one of the roots of

$$(1 - 4\alpha^2 \sin^2\left(\frac{k\Delta x}{2}\right) - \sigma)(1 - \sigma) + 4\alpha^2 \sin^2\left(\frac{k\Delta x}{2}\right) = 0$$

$$(1 - \sigma + \alpha^2\lambda^2)(1 - \sigma) - \alpha^2\lambda^2 = 0 \tag{20.103}$$

$$\sigma = 1 - 2\alpha^2 \sin^2\left(\frac{k\Delta x}{2}\right) \pm \sqrt{\left(1 - 2\alpha^2 \sin^2\left(\frac{k\Delta x}{2}\right)\right)^2 - 1}. \tag{20.104}$$

The eigenvalues coincide with those of the two-step method (20.58).

20.7.2 Lax–Wendroff Scheme

The Lax–Wendroff scheme can be derived from the Taylor series expansion

$$f(t + \Delta t, x) = f(t, x) + \frac{\partial f(t, x)}{\partial t}\Delta t + \frac{1}{2}\Delta t^2 \frac{\partial^2 f(t, x)}{\partial t^2} + \dots$$
$$= f(t, x) + c\Delta t \frac{\partial g(t, x)}{\partial x} + \frac{c^2\Delta t^2}{2}\frac{\partial^2 f(t, x)}{\partial t^2} + \dots \tag{20.105}$$

$$g(t + \Delta t, x) = g(t, x) + \frac{\partial g(t, x)}{\partial t}\Delta t + \frac{1}{2}\Delta t^2 \frac{\partial^2 g(t, x)}{\partial t^2} + \dots$$
$$= g(t, x) + c\Delta t \frac{\partial f(t, x)}{\partial x} + \frac{c^2\Delta t^2}{2}\frac{\partial^2 g(t, x)}{\partial t^2} + \dots . \tag{20.106}$$

It uses symmetric differences on regular grids (20.45, 20.46) to obtain the iteration

$$f_m^{n+1} = f_m^n + c\Delta t \frac{g_{m+1}^n - g_{m-1}^n}{2\Delta x} + c^2\Delta t^2 \frac{f_{m+1}^n + f_{m-1}^n - 2f_m^n}{2\Delta x^2} \tag{20.107}$$

$$g_m^{n+1} = g_m^n + c\Delta t \frac{f_{m+1}^n - f_{m-1}^n}{2\Delta x} + c^2\Delta t^2 \frac{g_{m+1}^n + g_{m-1}^n - 2g_m^n}{2\Delta x^2} \tag{20.108}$$

$$\begin{pmatrix} \mathbf{f}^{n+1} \\ \mathbf{g}^{n+1} \end{pmatrix} = \begin{pmatrix} 1 + \frac{\alpha^2}{2} M & \frac{\alpha}{2} D \\ \frac{\alpha}{2} D & 1 + \frac{\alpha^2}{2} M \end{pmatrix} \begin{pmatrix} \mathbf{f}^n \\ \mathbf{g}^n \end{pmatrix} \tag{20.109}$$

with the tridiagonal matrix

$$D = \begin{pmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 0 & 1 \\ & & & -1 & 0 \end{pmatrix}. \tag{20.110}$$

To analyze the stability we insert

$$f_m^n = u\mathrm{e}^{an}\mathrm{e}^{\mathrm{i}km\Delta x} \quad g_m^n = v\mathrm{e}^{an}\mathrm{e}^{\mathrm{i}km\Delta x} \tag{20.111}$$

and calculate the eigenvalues (compare with 20.102) of

$$\begin{pmatrix} 1 + \alpha^2(\cos k\Delta x - 1) & \mathrm{i}\alpha \sin k\Delta x \\ \mathrm{i}\alpha \sin k\Delta x & 1 + \alpha^2(\cos k\Delta x - 1) \end{pmatrix} \tag{20.112}$$

which are given by

$$\sigma = 1 + \alpha^2(\cos k\Delta x - 1) \pm \sqrt{\alpha^2(\cos^2 k\Delta x - 1)}. \tag{20.113}$$

The root is always imaginary and

$$|\sigma|^2 = 1 + (\alpha^4 - \alpha^2)(\cos k\Delta x - 1)^2 \le 1 + 4(\alpha^4 - \alpha^2).$$

For $\alpha < 1$ we find $|\sigma| < 1$. The method is stable but there is wavelength dependent damping (Figs. 20.12 and 20.13).

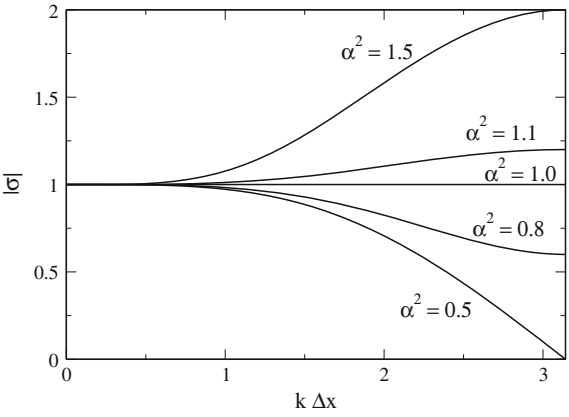


Fig. 20.12 (Stability region of the Lax–Wendroff method) Instabilities appear for $|\alpha| > 1$. In the opposite case short wavelength modes are damped

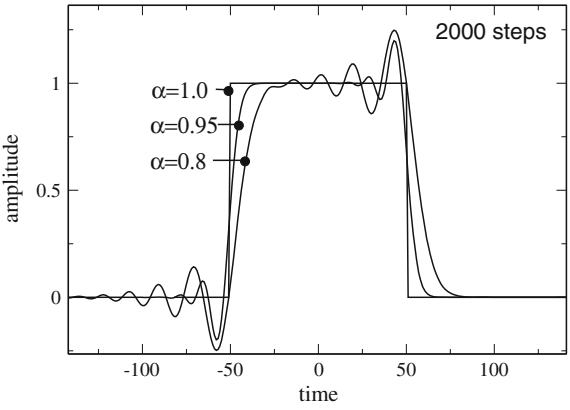


Fig. 20.13 (Simulation with the Lax–Wendroff method) A rectangular pulse is simulated with the two-variable Lax–Wendroff method. While for $\alpha = 1$ the pulse shape has not changed after 2000 steps, for smaller values the short wavelength components are lost due to dispersion and damping

20.7.3 Crank–Nicolson Scheme

This method takes the average of the explicit and implicit Euler methods

$$f(t+\Delta t) = f(t) + \frac{c}{2}\left(\frac{\partial g}{\partial x}(t,x) + \frac{\partial g}{\partial x}(t+\Delta t,x)\right)\Delta t \tag{20.114}$$

$$g(t+\Delta t) = g(t) + \frac{c}{2}\left(\frac{\partial f}{\partial x}(t,x) + \frac{\partial f}{\partial x}(t+\Delta t,x)\right)\Delta t \tag{20.115}$$

and uses symmetric differences on the regular grids (20.45, 20.46) to obtain

$$f_m^{n+1} = f_m^n + \frac{\alpha}{4}\left(g_{m+1}^n - g_{m-1}^n + g_{m+1}^{n+1} - g_{m-1}^{n+1}\right) \tag{20.116}$$

$$g_m^{n+1} = g_m^n + \frac{\alpha}{4}\left(f_{m+1}^n - f_{m-1}^n + f_{m+1}^{n+1} - f_{m-1}^{n+1}\right) \tag{20.117}$$

which reads in matrix notation

$$\begin{pmatrix} \mathbf{f}_{n+1} \\ \mathbf{g}_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & \frac{\alpha}{4}D \\ \frac{\alpha}{4}D & 1 \end{pmatrix} \begin{pmatrix} \mathbf{f}_n \\ \mathbf{g}_n \end{pmatrix} + \begin{pmatrix} & \frac{\alpha}{4}D \\ \frac{\alpha}{4}D & \end{pmatrix} \begin{pmatrix} \mathbf{f}_{n+1} \\ \mathbf{g}_{n+1} \end{pmatrix}. \tag{20.118}$$

This equation can be solved formally by collecting terms at time t_{n+1}

$$\begin{pmatrix} 1 & -\frac{\alpha}{4}D \\ -\frac{\alpha}{4}D & 1 \end{pmatrix} \begin{pmatrix} \mathbf{f}_{n+1} \\ \mathbf{g}_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & \frac{\alpha}{4}D \\ \frac{\alpha}{4}D & 1 \end{pmatrix} \begin{pmatrix} \mathbf{f}_n \\ \mathbf{g}_n \end{pmatrix} \tag{20.119}$$

and multiplying with the inverse matrix from left

$$\begin{pmatrix} \mathbf{f}_{n+1} \\ \mathbf{g}_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & -\frac{\alpha}{4}D \\ -\frac{\alpha}{4}D & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & \frac{\alpha}{4}D \\ \frac{\alpha}{4}D & 1 \end{pmatrix} \begin{pmatrix} \mathbf{f}_n \\ \mathbf{g}_n \end{pmatrix}. \tag{20.120}$$

Now, if $\mathbf{u}$ is an eigenvector of D with purely imaginary eigenvalue λ (Sect. 10.3)

$$\begin{pmatrix} 1 & \frac{\alpha}{4}D \\ \frac{\alpha}{4}D & 1 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \pm\mathbf{u} \end{pmatrix} = \begin{pmatrix} (1 \pm \frac{\alpha}{4}\lambda)\mathbf{u} \\ (\frac{\alpha}{4}\lambda \pm 1)\mathbf{u} \end{pmatrix} = (1 \pm \frac{\alpha}{4}\lambda) \begin{pmatrix} \mathbf{u} \\ \pm\mathbf{u} \end{pmatrix} \tag{20.121}$$

and furthermore

$$\begin{pmatrix} 1 & -\frac{\alpha}{4}D \\ -\frac{\alpha}{4}D & 1 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \pm\mathbf{u} \end{pmatrix} = \begin{pmatrix} (1 \mp \frac{\alpha}{4}\lambda)\mathbf{u} \\ (-\frac{\alpha}{4}\lambda \pm 1)\mathbf{u} \end{pmatrix} = (1 \mp \frac{\alpha}{4}\lambda) \begin{pmatrix} \mathbf{u} \\ \pm\mathbf{u} \end{pmatrix}. \tag{20.122}$$

But, since the eigenvalue of the inverse matrix is the reciprocal of the eigenvalue, the eigenvalues of

$$T = \begin{pmatrix} 1 & -\frac{\alpha}{4}D \\ -\frac{\alpha}{4}D & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 & \frac{\alpha}{4}D \\ \frac{\alpha}{4}D & 1 \end{pmatrix} \tag{20.123}$$

are given by

$$\sigma = \frac{1 \pm \frac{\alpha}{4}\lambda}{1 \mp \frac{\alpha}{4}\lambda}. \tag{20.124}$$

Since λ is imaginary, we find $|\sigma| = 1$. The Crank–Nicolson method is stable and does not show damping like the Lax–Wendroff method. However, there is considerable dispersion. Solution of the linear system (20.119) is complicated and can be replaced by an iterative predictor-corrector method. Starting from the initial guess

$$\begin{pmatrix} {}^{(0)}\mathbf{f}_{n+1} \\ {}^{(0)}\mathbf{g}_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & \frac{\alpha}{2}D \\ \frac{\alpha}{2}D & 1 \end{pmatrix} \begin{pmatrix} \mathbf{f}_n \\ \mathbf{g}_n \end{pmatrix} \tag{20.125}$$

we iterate

$$\begin{pmatrix} {}^{(0)}\mathbf{f}_{n+1/2} \\ {}^{(0)}\mathbf{g}_{n+1/2} \end{pmatrix} = \frac{1}{2}\begin{pmatrix} {}^{(0)}\mathbf{f}_{n+1} \\ {}^{(0)}\mathbf{g}_{n+1} \end{pmatrix} + \frac{1}{2}\begin{pmatrix} \mathbf{f}_n \\ {}^{)}\mathbf{g}_n \end{pmatrix} = \begin{pmatrix} 1 & \frac{\alpha}{4}D \\ \frac{\alpha}{4}D & 1 \end{pmatrix}\begin{pmatrix} \mathbf{f}_n \\ \mathbf{g}_n \end{pmatrix}$$

$$\begin{pmatrix} {}^{(1)}\mathbf{f}_{n+1} \\ {}^{(1)}\mathbf{g}_{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_n \\ \mathbf{g}_n \end{pmatrix} + \begin{pmatrix} & \frac{\alpha}{2}D \\ \frac{\alpha}{2}D & \end{pmatrix}\begin{pmatrix} {}^{(0)}\mathbf{f}_{n+1/2} \\ {}^{(0)}\mathbf{g}_{n+1/2} \end{pmatrix}$$

$$= \begin{pmatrix} 1 & \frac{\alpha}{4}D \\ \frac{\alpha}{4}D & 1 \end{pmatrix}\begin{pmatrix} \mathbf{f}_n \\ \mathbf{g}_n \end{pmatrix} + \begin{pmatrix} & \frac{\alpha}{4}D \\ \frac{\alpha}{4}D & \end{pmatrix}\begin{pmatrix} {}^{(0)}\mathbf{f}_{n+1} \\ {}^{(0)}\mathbf{g}_{n+1} \end{pmatrix} \tag{20.126}$$

$$\begin{pmatrix} {}^{(1)}\mathbf{f}_{n+1/2} \\ {}^{(1)}\mathbf{g}_{n+1/2} \end{pmatrix} = \frac{1}{2}\begin{pmatrix} \mathbf{f}_n \\ {}^{)}\mathbf{g}_n \end{pmatrix} + \frac{1}{2}\begin{pmatrix} {}^{(1)}\mathbf{f}_{n+1} \\ {}^{(1)}\mathbf{g}_{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_n \\ \mathbf{g}_n \end{pmatrix} + \begin{pmatrix} & \frac{\alpha}{4}D \\ \frac{\alpha}{4}D & \end{pmatrix}\begin{pmatrix} {}^{(0)}\mathbf{f}_{n+1/2} \\ {}^{(0)}\mathbf{g}_{n+1/2} \end{pmatrix} \tag{20.127}$$

$$\begin{pmatrix} {}^{(2)}\mathbf{f}_{n+1} \\ {}^{(2)}\mathbf{g}_{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_n \\ \mathbf{g}_n \end{pmatrix} + \begin{pmatrix} & \frac{\alpha}{2}D \\ \frac{\alpha}{2}D & \end{pmatrix}\begin{pmatrix} {}^{(1)}\mathbf{f}_{n+1/2} \\ {}^{(1)}\mathbf{g}_{n+1/2} \end{pmatrix}$$

$$= \begin{pmatrix} 1 & \frac{\alpha}{4}D \\ \frac{\alpha}{4}D & 1 \end{pmatrix}\begin{pmatrix} \mathbf{f}_n \\ \mathbf{g}_n \end{pmatrix} + \begin{pmatrix} & \frac{\alpha}{4}D \\ \frac{\alpha}{4}D & \end{pmatrix}\begin{pmatrix} {}^{(1)}\mathbf{f}_{n+1} \\ {}^{(1)}\mathbf{g}_{n+1} \end{pmatrix}. \tag{20.128}$$

In principle this iteration could be repeated more times, but as Teukolsky showed [255], two iterations are optimal for hyperbolic equations like the advection or wave equation. The region of stability is reduced (Figs. 20.14 and 20.15) compared to the implicit Crank–Nicolson method. The eigenvalues are

$${}^{(0)}\sigma = 1 \pm \mathrm{i}\alpha \sin k\Delta x \quad |{}^{(0)}\sigma| > 1 \tag{20.129}$$

$${}^{(1)}\sigma = 1 \pm \mathrm{i}\alpha \sin k\Delta x - \frac{\alpha^2}{2}\sin^2 k\Delta x \quad |{}^{(1)}\sigma| > 1 \tag{20.130}$$

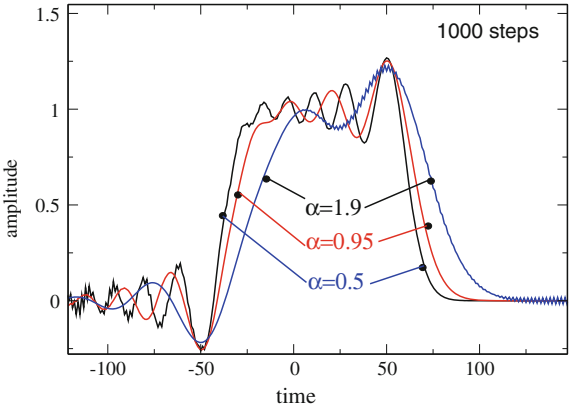


Fig. 20.14 (Simulation with the iterated Crank–Nicolson method) A rectangular pulse is simulated with the two-variable iterated Crank–Nicolson method. Only this method is stable for values $\alpha > 1$

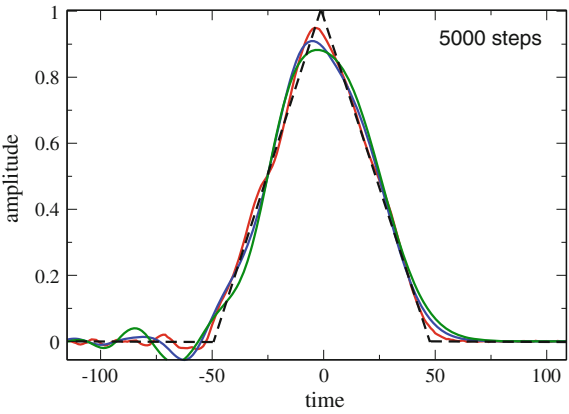


Fig. 20.15 (Simulation of a triangular pulse) A triangular pulse is simulated with different two-variable methods (*dashed curve* initial conditions, *red* leapfrog, *blue* Lax–Wendroff, *green* iterated Crank–Nicolson). This pulse contains less short wavelength components than the square pulse and shows much less deformation even after 5000 steps

$$
\begin{aligned}
{}^{(2)}\sigma &= 1 - \frac{\alpha^2}{2}\sin^2 k\Delta x \pm \mathrm{i}(\alpha \sin k\Delta x - \frac{\alpha^3 \sin^3 k\Delta x}{4}) \\
|{}^{(2)}\sigma|^2 &= 1 - \frac{\alpha^4 \sin^4 k\Delta x}{4} + \frac{\alpha^6 \sin^6 k\Delta x}{16} \leq 1 \text{ for } |\alpha| \leq 2. \qquad (20.131)
\end{aligned}
$$

Problems

Problem 20.1 Waves on a Damped String

In this computer experiment we simulate waves on a string with a moving boundary with the method from Sect. 20.6.

- Excite the left boundary with a continuous sine function and try to generate standing waves
- Increase the velocity until instabilities appear
- Compare reflection at open and fixed right boundary
- Observe the dispersion of pulses with different shape and duration
- The velocity can be changed by a factor n (refractive index) in the region $x > 0$. Observe reflection at the boundary $x = 0$

Problem 20.2 Waves with the Fourier Transform Method

In this computer experiment we use the method from Sect. 20.3 to simulate waves on a string with fixed boundaries.

- Different initial excitations of the string can be selected
- The dispersion can be switched off by using $\omega_k = ck$ instead of the proper eigenvalues (20.31)

Problem 20.3 Two Variable Methods

In this computer experiment we simulate waves with periodic boundary conditions. Different initial values (rectangular, triangular or Gaussian pulses of different widths) and methods (leapfrog, Lax–Wendroff, iterated Crank–Nicolson) can be compared.

Chapter 21
Diffusion

Diffusion is one of the simplest non-equilibrium processes. It describes the transport of heat [256, 257] *and the time evolution of differences in substance concentrations* [258]. *In this chapter, the one-dimensional diffusion equation*

$$\frac{\partial}{\partial t} f(t,x) = D \frac{\partial^2}{\partial x^2} f(t,x) + S(t,x) \tag{21.1}$$

is semi-discretized with finite differences. The time integration is performed with three different Euler methods. The explicit Euler method is conditionally stable only for small Courant number $\alpha = \frac{D\Delta t}{\Delta x^2} < 1/2$, *which makes very small time steps necessary. The fully implicit method is unconditionally stable but its dispersion deviates largely from the exact expression. The Crank–Nicolson method is also unconditionally stable. However, it is more accurate and its dispersion relation is closer to the exact one. Extension to more than one dimension is easily possible, but the numerical effort increases drastically as there is no formulation involving simple tridiagonal matrices like in one dimension. The split operator approximation uses the one-dimensional method independently for each dimension. It is very efficient with almost no loss in accuracy. In a computer experiment the different schemes are compared for diffusion in two dimensions.*

21.1 Particle Flux and Concentration Changes

Let $f(\mathbf{x},t)$ denote the concentration of a particle species and $\mathbf{J}$ the corresponding flux of particles. Consider a small cube with volume h^3 (Fig. 21.1). The change of the number of particles within this volume is given by the integral form of the conservation law (12.10)

$$\frac{\partial}{\partial t} \int_V dV\ f(\mathbf{r},t) + \oint_{\partial V} \mathbf{J}(\mathbf{r},t) d\mathbf{A} = \int_V dV\ S(\mathbf{r},t) \tag{21.2}$$

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_21

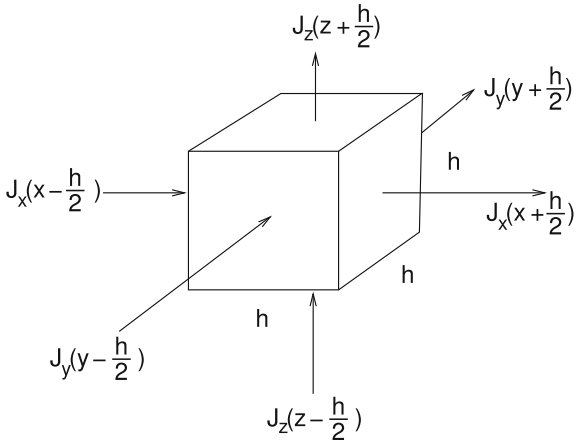


Fig. 21.1 Flux through a volume element

where the source term $S(\mathbf{r})$ accounts for creation or destruction of particles due to for instance chemical reactions. In Cartesian coordinates we have

$$\begin{aligned}
&\int_{x-h/2}^{x+h/2} dx' \int_{y-h/2}^{y+h/2} dy' \int_{z-h/2}^{z+h/2} dz' \left(\frac{\partial}{\partial t} f(x', y', z', t) - S(x', y', z', t) \right) \\
&+ \int_{x-h/2}^{x+h/2} dx' \int_{y-h/2}^{y+h/2} dy' \left(J_z(x', y', z + \frac{h}{2}) - J_z(x', y', z - \frac{h}{2}) \right) \\
&+ \int_{x-h/2}^{x+h/2} dx' \int_{z-h/2}^{z+h/2} dz' \left(J_y(x', y + \frac{h}{2}, z') - J_y(x', y - \frac{h}{2}, z') \right) \\
&+ \int_{z-h/2}^{z+h/2} dz' \int_{y-h/2}^{y+h/2} dy' \left(J_z(x + \frac{h}{2}, y', z') - J_z(x - \frac{h}{2}, y', z') \right) = 0. \quad (21.3)
\end{aligned}$$

In the limit of small h this turns into the differential form of the conservation law

$$h^3 \left(\frac{\partial}{\partial t} f(x, y, z, t) - S(x, y, z, t) \right) + h^2 \left(h \frac{\partial J_x}{\partial x} + h \frac{\partial J_y}{\partial y} + h \frac{\partial J_z}{\partial z} \right) = 0 \quad (21.4)$$

or after division by h^3

$$\frac{\partial}{\partial t} f(\mathbf{r}, t) = -\text{div}\mathbf{J}(\mathbf{r}, t) + S(\mathbf{r}, t). \quad (21.5)$$

Within the framework of linear response theory the flux is proportional to the gradient of f (Fig. 21.2),

$$\mathbf{J} = -D \text{ grad } f. \quad (21.6)$$

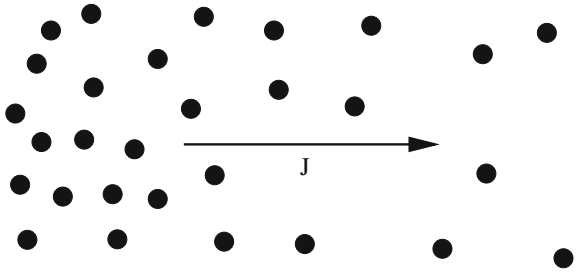


Fig. 21.2 Diffusion due to a concentration gradient

Together we obtain the diffusion equation

$$\frac{\partial f}{\partial t} = \text{div}(D\text{grad} f) + S \tag{21.7}$$

which in the special case of constant D simplifies to

$$\frac{\partial f}{\partial t} = D\Delta f + S. \tag{21.8}$$

21.2 Diffusion in One Dimension

We will use the finite differences method which works well if the diffusion constant D is constant in time and space. We begin with diffusion in one dimension and use regular grids $t_n = n\Delta t$, $x_m = m\Delta x$, $f_m^n = f(t_n, x_m)$ and the discretized second derivative

$$\frac{\partial^2 f}{\partial x^2} = \frac{f(x+\Delta x) + f(x-\Delta x) - 2f(x)}{\Delta x^2} + O(\Delta x^2) \tag{21.9}$$

to obtain the semi-discrete diffusion equation

$$\dot{f}(t, x_m) = \frac{D}{\Delta x^2}\left(f(t, x_{m+1}) + f(t, x_{m-1}) - 2f(t, x_m)\right) + S(t, x_m) \tag{21.10}$$

or in matrix notation

$$\dot{\mathbf{f}}(t) = \frac{D}{\Delta x^2} M\mathbf{f}(t) + \mathbf{S}(t) \tag{21.11}$$

with the tridiagonal matrix

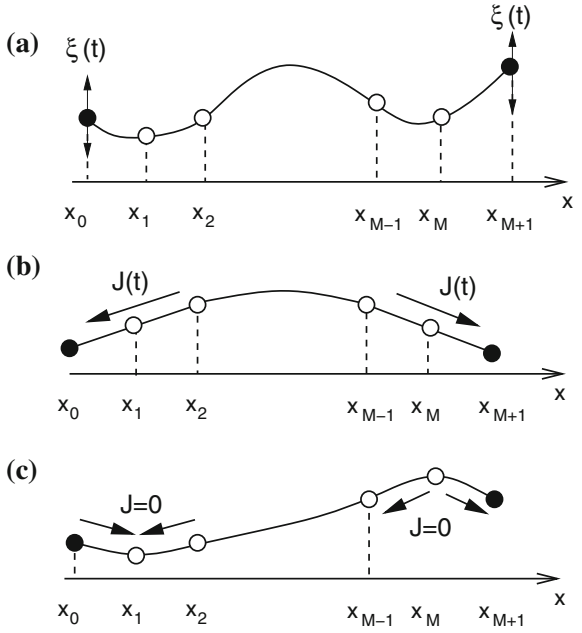


Fig. 21.3 (Boundary conditions for 1-dimensional diffusion) Additional boundary points x_0, x_{M+1} are used to realize the boundary conditions, (**a**) **Dirichlet boundary conditions** the function values at the boundary are given $f(t,x_0) = \xi_0(t)$ $\frac{\partial^2}{\partial x^2} f(x_1) = \frac{1}{\Delta x^2}\left(f(x_2) - 2f(x_1) + \xi_0(t)\right)$ or $f(t,x_{M+1}) = \xi_{M+1}(t)$, $\frac{\partial^2}{\partial x^2} f(x_M) = \frac{1}{\Delta x^2}\left(f(x_{M-1}) - 2f(x_M) + \xi_{M+1}(t)\right)$, (**b**) **Neumann boundary conditions** the flux through the boundary is given, hence the derivative $\frac{\partial f}{\partial x}$ at the boundary $f(t,x_0) = f(t,x_2) + 2\frac{\Delta x}{D} J_1(t)$ $\frac{\partial^2}{\partial x^2} f(x_1) = \frac{1}{\Delta x^2}\left(2f(x_2) - 2f(x_1) + 2\frac{\Delta x}{D} J_1(t)\right)$ or $f(t,x_{M+1}) = f(t,x_{M-1}) - \frac{2\Delta x}{D} J_M(t)$ $\frac{\partial^2}{\partial x^2} f(x_M) = \frac{1}{\Delta x^2}\left(2f(x_{M-1}) - 2f(x_M) - \frac{2\Delta x}{D} J_M(t)\right)$, (**c**) **No-flow boundary conditions** there is no flux through the boundary, hence the derivative $\frac{\partial f}{\partial x} = 0$ at the boundary $f(t,x_0) = f(t,x_2)$ $\frac{\partial^2}{\partial x^2} f(x_1) = \frac{1}{\Delta x^2}\left(2f(x_2) - 2f(x_1)\right)$ or $f(t,x_M) = f(t,x_{M-2})$ $\frac{\partial^2}{\partial x^2} f(x_M) = \frac{1}{\Delta x^2}\left(2f(x_{M-1}) - 2f(x_M)\right)$

$$M = \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix}. \tag{21.12}$$

Boundary conditions can be taken into account by introducing extra boundary points x_0, x_{M+1} (Fig. 21.3).

21.2.1 Explicit Euler (Forward Time Centered Space) Scheme

A simple Euler step (13.3) makes the approximation

$$f_m^{n+1} - f_m^n = \dot{f}(t_n, x_m)\Delta t = D\frac{\Delta t}{\Delta x^2}\left(f_{m+1}^n + f_{m-1}^n - 2f_m^n\right) + S_m^n \Delta t. \tag{21.13}$$

For homogeneous boundary conditions $f = 0$ this becomes in matrix form

$$\begin{pmatrix} f_1^{n+1} \\ \vdots \\ f_M^{n+1} \end{pmatrix} = A \begin{pmatrix} f_1^n \\ \vdots \\ f_M^n \end{pmatrix} + \begin{pmatrix} S_1^n \Delta t \\ \vdots \\ S_M^n \Delta t \end{pmatrix} \tag{21.14}$$

with the tridiagonal matrix

$$A = \begin{pmatrix} 1 - 2D\frac{\Delta t}{\Delta x^2} & D\frac{\Delta t}{\Delta x^2} & & & \\ D\frac{\Delta t}{\Delta x^2} & 1 - 2D\frac{\Delta t}{\Delta x^2} & & & \\ & \ddots & \ddots & \ddots & \\ & & D\frac{\Delta t}{\Delta x^2} & 1 - 2D\frac{\Delta t}{\Delta x^2} & D\frac{\Delta t}{\Delta x^2} \\ & & & D\frac{\Delta t}{\Delta x^2} & 1 - 2D\frac{\Delta t}{\Delta x^2} \end{pmatrix} = 1 + \alpha M \tag{21.15}$$

where α is the Courant number for diffusion

$$\alpha = D\frac{\Delta t}{\Delta x^2}. \tag{21.16}$$

The eigenvalues of M are (compare 20.30)

$$\lambda = -4\sin^2\left(\frac{k\Delta x}{2}\right) \text{ with } k\Delta x = \frac{\pi}{M+1}, \frac{2\pi}{M+1}, \cdots \frac{M\pi}{M+1} \tag{21.17}$$

and hence the eigenvalues of A are given by

$$1 + \alpha\lambda = 1 - 4\alpha\sin^2\frac{k\Delta x}{2}. \tag{21.18}$$

The algorithm is stable if

$$|1 + \alpha\lambda| < 1 \quad \text{for all } \lambda \tag{21.19}$$

which holds if

$$-1 < 1 - 4\alpha \sin^2 \frac{k\Delta x}{2} < 1. \tag{21.20}$$

The maximum of the sine function is $\sin(\frac{M\pi}{2(M+1)}) \approx 1$. Hence the right hand inequation is satisfied and from the left one we have

$$-1 < 1 - 4\alpha. \tag{21.21}$$

The algorithm is stable for

$$\alpha = D\frac{\Delta t}{\Delta x^2} < \frac{1}{2}. \tag{21.22}$$

The dispersion relation follows from inserting a plane wave ansatz

$$\mathrm{e}^{\mathrm{i}\omega\Delta t} = 1 - 4\alpha \sin^2\left(\frac{k\Delta x}{2}\right). \tag{21.23}$$

For $\alpha > 1/4$ the right hand side changes sign at

$$k_c \Delta x = 2\mathrm{arcsin}\sqrt{\frac{1}{4\alpha}}. \tag{21.24}$$

The imaginary part of ω has a singularity at k_c and the real part has a finite value of π for $k > k_c$ (Fig. 21.4). Deviations from the exact dispersion

$$\omega = \mathrm{i}k^2 \tag{21.25}$$

are large, except for very small k.

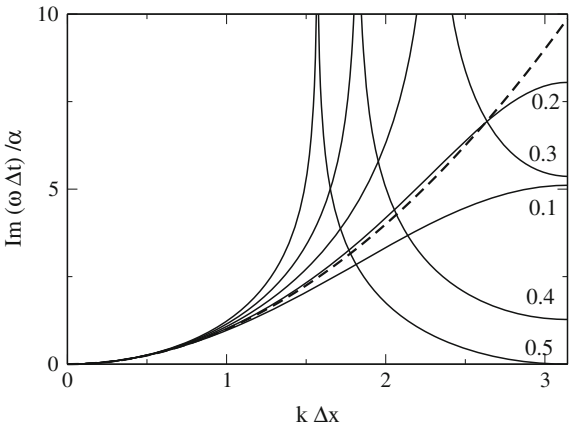


Fig. 21.4 (Dispersion of the explicit Euler method) The dispersion of the explicit method is shown for different values of the Courant number α and compared to the exact dispersion (*dashed curve*). The imaginary part of ω shows a singularity for $\alpha > 1/4$. Above the singularity ω is complex valued

21.2.2 Implicit Euler (Backward Time Centered Space) Scheme

Next we use the backward difference

$$f_m^{n+1} - f_m^n = \dot{f}(t_{n+1}, x_m)\Delta t$$

$$= D\frac{\partial^2 f}{\partial x^2}(t_{n+1}, x_m)\Delta t + S(t_{n+1}, x_m)\Delta t \tag{21.26}$$

to obtain the implicit method

$$f_m^{n+1} - \alpha\left(f_{m+1}^{n+1} + f_{m-1}^{n+1} - 2f_m^{n+1}\right) = f_m^n + S_m^{n+1}\Delta t \tag{21.27}$$

or in matrix notation

$$A\mathbf{f}_{n+1} = \mathbf{f}_n + \mathbf{S}_{n+1}\Delta t \quad \text{with } A = 1 - \alpha M \tag{21.28}$$

which can be solved formally by

$$\mathbf{f}_{n+1} = A^{-1}\mathbf{f}_n + A^{-1}\mathbf{S}_{n+1}\Delta t. \tag{21.29}$$

The eigenvalues of A are

$$\lambda(A) = 1 + 4\alpha \sin^2 \frac{k\Delta x}{2} > 1 \tag{21.30}$$

and the eigenvalues of A^{-1}

$$\lambda(A^{-1}) = \lambda(A)^{-1} = \frac{1}{1 + 4\alpha \sin^2 \frac{k\Delta x}{2}}. \tag{21.31}$$

The implicit method is unconditionally stable since

$$|\lambda(A^{-1})| < 1. \tag{21.32}$$

The dispersion relation of the implicit scheme follows from

$$\mathrm{e}^{\mathrm{i}\omega\Delta t} = \frac{1}{1 + 4\alpha \sin^2\left(\frac{k\Delta x}{2}\right)}. \tag{21.33}$$

There is no singularity and ω is purely imaginary. Still, deviations from the exact expression are large (Fig. 21.5).

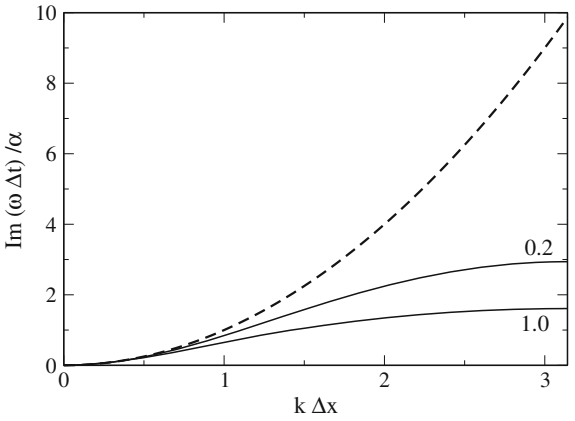


Fig. 21.5 (Dispersion of the implicit Euler method) The dispersion of the fully implicit method is shown for two different values of the Courant number α and compared to the exact dispersion (*dashed curve*)

Formally a matrix inversion is necessary. Numerically it is much more efficient to solve the tridiagonal system of equations (page 75).

$$(1 - \alpha M) f(t_{n+1}) = f(t_n) + S(t_{n+1}) \Delta t. \tag{21.34}$$

21.2.3 Crank–Nicolson Method

The Crank–Nicolson method [259] which is often used for diffusion problems, combines implicit and explicit methods. It uses the Heun method (Sect. 13.5) for the time integration

$$f_m^{n+1} - f_m^n = \frac{\Delta t}{2} \left(\frac{\partial f}{\partial t}(t_{n+1}, x_m) + \frac{\partial f}{\partial t}(t_n, x_m) \right) \tag{21.35}$$

$$= D \frac{\Delta t}{2} \left(\frac{\partial^2 f}{\partial x^2}(t_{n+1}, x_m) + \frac{\partial^2 f}{\partial x^2}(t_n, x_m) \right) + (S(t_n, x_m) + S(t_{n+1}, x_m)) \frac{\Delta t}{2} \tag{21.36}$$

$$= D \frac{\Delta t}{2} \left(\frac{f_{m+1}^n + f_{m-1}^n - 2 f_m^n}{\Delta x^2} + \frac{f_m^{n+1} + f_{m-1}^{n+1} - 2 f_m^{n+1}}{\Delta x^2} \right) + \frac{S_m^n + S_m^{n+1}}{2} \Delta t. \tag{21.37}$$

This approximation is second order both in time and space and becomes in matrix notation

$$\left(1-\frac{\alpha}{2}M\right)\mathbf{f}_{n+1}=\left(1+\frac{\alpha}{2}M\right)\mathbf{f}_n+\frac{\mathbf{S}_n+\mathbf{S}_{n+1}}{2}\Delta t \tag{21.38}$$

which can be solved by

$$\mathbf{f}_{n+1}=\left(1-\frac{\alpha}{2}M\right)^{-1}\left(1+\frac{\alpha}{2}M\right)\mathbf{f}_n+\left(1-\frac{\alpha}{2}M\right)^{-1}\frac{\mathbf{S}_n+\mathbf{S}_{n+1}}{2}\Delta t. \tag{21.39}$$

Again it is numerically much more efficient to solve the tridiagonal system of equations (21.38) than to calculate the inverse matrix.

The eigenvalues of this method are

$$\lambda=\frac{1+\frac{\alpha}{2}\mu}{1-\frac{\alpha}{2}\mu}\text{ with }\mu=-4\sin^2\frac{k\Delta x}{2}\in[-4,0]. \tag{21.40}$$

Since $\alpha\mu<0$ it follows

$$1+\frac{\alpha}{2}\mu<1-\frac{\alpha}{2}\mu \tag{21.41}$$

and hence

$$\lambda<1. \tag{21.42}$$

On the other hand we have

$$1>-1 \tag{21.43}$$

$$1+\frac{\alpha}{2}\mu>-1+\frac{\alpha}{2}\mu \tag{21.44}$$

$$\lambda>-1. \tag{21.45}$$

This shows that the Crank–Nicolson method is stable [260]. The dispersion follows from

$$\mathrm{e}^{\mathrm{i}\omega\Delta t}=\frac{1-2\alpha\sin^2\left(\frac{k\Delta x}{2}\right)}{1+2\alpha\sin^2\left(\frac{k\Delta x}{2}\right)}. \tag{21.46}$$

For $\alpha>1/2$ there is a sign change of the right hand side at

$$k_c\Delta x=2\arcsin\sqrt{\frac{1}{2\alpha}}. \tag{21.47}$$

The imaginary part of ω has a singularity at k_c and ω is complex valued for $k>k_c$ (Fig. 21.6).

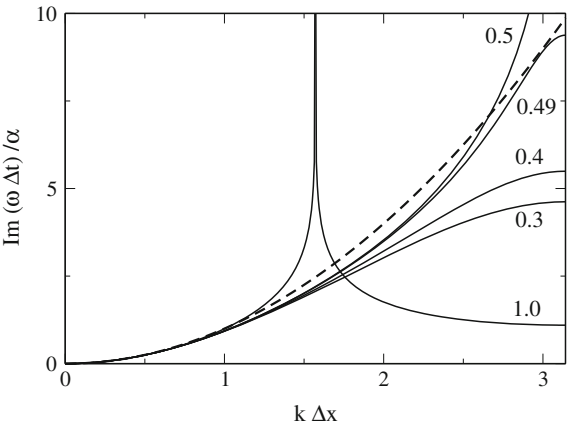


Fig. 21.6 (Dispersion of the Crank–Nicolson method) The dispersion of the Crank–Nicolson method is shown for different values of the Courant number α and compared to the exact dispersion (*dashed curve*). The imaginary part of ω shows a singularity for $\alpha > 1/2$. Above the singularity ω is complex valued. The exact dispersion is approached quite closely for $\alpha \approx 1/2$

21.2.4 Error Order Analysis

Taylor series gives for the exact solution

$$\Delta f_{exact} = \Delta t\, \dot{f}(t,x) + \frac{\Delta t^2}{2}\ddot{f}(t,x) + \frac{\Delta t^3}{6}\frac{\partial^3}{\partial t^3} f(t,x) \cdots$$

$$= \Delta t \left[D f''(t,x) + S(t,x) \right] + \frac{\Delta t^2}{2}\left[D\dot{f}''(t,x) + \dot{S}(t,x) \right] + \cdots \tag{21.48}$$

whereas for the explicit method

$$\Delta f_{expl} = \alpha M f(t,x) + S(t,x)\Delta t$$

$$= D\frac{\Delta t}{\Delta x^2}\left(f(t,x+\Delta x) + f(t,x-\Delta x) - 2f(t,x) \right) + S(t,x)\Delta t.$$

$$= D\frac{\Delta t}{\Delta x^2}\left(\Delta x^2 f''(t,x) + \frac{\Delta x^4}{12} f''''(t,x) + \cdots \right) + S(t,x)\Delta t$$

$$= \Delta f_{exact} + \frac{D\Delta t \Delta x^2}{12} f''''(t,x) - \frac{\Delta t^2}{2}\ddot{f}(t,x) + \cdots \tag{21.49}$$

and for the implicit method

$$\Delta f_{impl} = \alpha M f(t+\Delta t,x) + S(t+\Delta t,x)\Delta t$$

$$= D\frac{\Delta t}{\Delta x^2}\left(f(t+\Delta t,x+\Delta x) + f(t+\Delta t,x-\Delta x) - 2f(t+\Delta t,x) \right)$$

$$
\begin{aligned}
&+S(t+\Delta t,x)\Delta t \\
&= D\frac{\Delta t}{\Delta x^2}\left(\Delta x^2 f''(t,x)+\frac{\Delta x^4}{12}f''''(t,x)+\cdots\right) \\
&+S(t,x)\Delta t+D\frac{\Delta t^2}{\Delta x^2}\left(\Delta x^2 \dot{f}''(t,x)+\frac{\Delta x^4}{12}\dot{f}''''(t,x)+\cdots\right)+\dot{S}(t,x)\Delta t^2 \\
&= \Delta f_{exact}+D\frac{\Delta t\Delta x^2}{12}f''''(t,x)+\frac{1}{2}\Delta t^2\ddot{f}(t,x)+\cdots . \qquad (21.50)
\end{aligned}
$$

The Crank–Nicolson method has higher accuracy in Δt:

$$
\Delta f_{CN}=\frac{\Delta f_{expl}+\Delta f_{impl}}{2}=\frac{D\Delta t\Delta x^2}{12}f''''(t,x)-\frac{\Delta t^3}{6}\frac{\partial^3 f}{\partial t^3}+\cdots . \qquad (21.51)
$$

21.2.5 Finite Element Discretization

In one dimension discretization with finite differences is very similar to discretization with finite elements, if Galerkin's method is applied on a regular grid (Chap. 12). The only difference is the non-diagonal form of the mass-matrix which has to be applied to the time derivative [147]. Implementation of the discretization scheme (12.170) is straightforward. The semi-discrete diffusion equation becomes

$$
\begin{aligned}
&\frac{\partial}{\partial t}\left(\frac{1}{6}f(t,x_{m-1})+\frac{2}{3}f(t,x_m)+\frac{1}{6}f(t,x_{m+1})\right) \\
&=\frac{D}{\Delta x^2}\left(f(t,x_{m+1})+f(t,x_{m-1})-2f(t,x_m)\right)+S(t,x_m) \qquad (21.52)
\end{aligned}
$$

or in matrix form

$$
\left(1+\frac{1}{6}M_2\right)\dot{\mathbf{f}}(t)=\frac{D}{\Delta x^2}M_2\mathbf{f}(t)+\mathbf{S}(t). \qquad (21.53)
$$

This can be combined with the Crank–Nicolson scheme to obtain

$$
\left(1+\frac{1}{6}M_2\right)(\mathbf{f}_{n+1}-\mathbf{f}_n)=\left(\frac{\alpha}{2}M_2\mathbf{f}_n+\frac{\alpha}{2}M_2\mathbf{f}_{n+1}\right)+\frac{\Delta t}{2}(\mathbf{S}_n+\mathbf{S}_{n+1}) \qquad (21.54)
$$

or

$$
\left[1+\left(\frac{1}{6}-\frac{\alpha}{2}\right)M_2\right]\mathbf{f}_{n+1}=\left[1+\left(\frac{1}{6}+\frac{\alpha}{2}\right)M_2\right]\mathbf{f}_n+\frac{\Delta t}{2}(\mathbf{S}_n+\mathbf{S}_{n+1}). \qquad (21.55)
$$

21.3 Split-Operator Method for Multidimensions

The simplest discretization of the Laplace operator in 3 dimensions is given by

$$\Delta f = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right) f(t, x, y, z)$$
$$= \frac{1}{\Delta x^2}(M_x + M_y + M_z)\, f(t, x, y, z) \tag{21.56}$$

where

$$\frac{1}{\Delta x^2} M_x f(t, x, y, z) = \frac{f(t, x + \Delta x, y, z) + f(t, x - \Delta x, y, z) - 2f(t, x, y, z)}{\Delta x^2} \tag{21.57}$$

etc. denote the discretized second derivatives. Generalization of the Crank–Nicolson method for the 3-dimensional problem gives

$$f(t_{n+1}) = \left(1 - \frac{\alpha}{2}M_x - \frac{\alpha}{2}M_y - \frac{\alpha}{2}M_z\right)^{-1} \left(1 + \frac{\alpha}{2}M_x + \frac{\alpha}{2}M_y + \frac{\alpha}{2}M_z\right)\, f(t). \tag{21.58}$$

But now the matrices representing the operators M_x, M_y, M_z are not tridiagonal. To keep the advantages of tridiagonal matrices we use the approximations

$$\left(1 + \frac{\alpha}{2}M_x + \frac{\alpha}{2}M_y + \frac{\alpha}{2}M_z\right) \approx \left(1 + \frac{\alpha}{2}M_x\right)\left(1 + \frac{\alpha}{2}M_y\right)\left(1 + \frac{\alpha}{2}M_z\right) \tag{21.59}$$

$$\left(1 - \frac{\alpha}{2}M_x - \frac{\alpha}{2}M_y - \frac{\alpha}{2}M_z\right) \approx \left(1 - \frac{\alpha}{2}M_x\right)\left(1 - \frac{\alpha}{2}M_y\right)\left(1 - \frac{\alpha}{2}M_z\right) \tag{21.60}$$

and rearrange the factors to obtain

$$f(t_{n+1}) = \left(1 - \tfrac{\alpha}{2}M_x\right)^{-1} \left(1 + \tfrac{\alpha}{2}M_x\right)\left(1 - \tfrac{\alpha}{2}M_y\right)^{-1} \left(1 + \tfrac{\alpha}{2}M_y\right)\left(1 - \tfrac{\alpha}{2}M_z\right)^{-1} \left(1 + \tfrac{\alpha}{2}M_z\right)\, f(t_n) \tag{21.61}$$

which represents successive application of the 1-dimensional scheme for the three directions separately. The last step was possible since the operators M_i and M_j for different directions $i \neq j$ commute. For instance

$$M_x M_y f = M_x\,(f(x, y + \Delta x) + f(x, y - \Delta x) - 2f(x, y))$$
$$= (f(x + \Delta x, y + \Delta y) + f(x - \Delta x, y + \Delta x)$$
$$- 2f(x, y + \Delta x) + f(x + \Delta x, y - \Delta x)$$

$$
\begin{aligned}
&+ f(x - \Delta x, y - \Delta x) - 2f(x, y - \Delta x) \\
&-2f(x + \Delta x, y) - 2f(x - \Delta x, y) + 4f(x, y)) \\
&= M_y M_x f. \qquad (21.62)
\end{aligned}
$$

The Taylor series of (21.58) and (21.61) coincide up to second order with respect to αM_i:

$$
\begin{aligned}
&\left(1 - \frac{\alpha}{2}M_x - \frac{\alpha}{2}M_y - \frac{\alpha}{2}M_z\right)^{-1} \left(1 + \frac{\alpha}{2}M_x + \frac{\alpha}{2}M_y + \frac{\alpha}{2}M_z\right) \\
&= 1 + \alpha(M_x + M_y + M_z) + \frac{\alpha^2}{2}(M_x + M_y + M_z)^2 + O(\alpha^3) \qquad (21.63)
\end{aligned}
$$

$$
\begin{aligned}
&\left(1 - \frac{\alpha}{2}M_x\right)^{-1} \left(1 + \frac{\alpha}{2}M_x\right) \left(1 - \frac{\alpha}{2}M_y\right)^{-1} \left(1 + \frac{\alpha}{2}M_y\right) \left(1 - \frac{\alpha}{2}M_z\right)^{-1} \left(1 + \frac{\alpha}{2}M_z\right) \\
&= \left(1 + \alpha M_x + \frac{\alpha^2 M_x^2}{2}\right)\left(1 + \alpha M_y + \frac{\alpha^2 M_y^2}{2}\right)\left(1 + \alpha M_z + \frac{\alpha^2 M_z^2}{2}\right) + O(\alpha^3) \\
&= 1 + \alpha(M_x + M_y + M_z) + \frac{\alpha^2}{2}(M_x + M_y + M_z)^2 + O(\alpha^3). \qquad (21.64)
\end{aligned}
$$

Hence we have

$$
\begin{aligned}
f_{n+1} &= \left(1 + D\Delta t\left(\Delta + \frac{\Delta x^2}{12}\Delta^2 + \cdots\right) + \frac{D^2\Delta t^2}{2}(\Delta^2 + \cdots)\right) f_n \\
&+ \left(1 + \frac{D\Delta t}{2}\Delta + \cdots\right) \frac{S_{n+1} + S_n}{2}\Delta t \\
&= f_n + \Delta t(D\Delta f_n + S_n) + \frac{\Delta t^2}{2}(D^2\Delta^2 + D\Delta S_n + \dot{S}_n) + O(\Delta t \Delta x^2, \Delta t^3). \qquad (21.65)
\end{aligned}
$$

and the error order is conserved by the split operator method.

Problems

Problem 21.1 Diffusion in 2 Dimensions

In this computer experiment we solve the diffusion equation on a two dimensional grid for

- an initial distribution $f(t = 0, x, y) = \delta_{x,0}\delta_{y,0}$
- a constant source $f(t = 0) = 0,\ S(t, x, y) = \delta_{x,0}\delta_{y,0}$

Compare implicit, explicit and Crank–Nicolson method.

Chapter 22
Nonlinear Systems

Nonlinear problems [261, 262] *are of interest to physicists, mathematicians and also engineers. Nonlinear equations are difficult to solve and give rise to interesting phenomena like indeterministic behavior, multistability or formation of patterns in time and space. In the following we discuss recurrence relations like an iterated function* [263]

$$x_{n+1} = f(x_n) \tag{22.1}$$

systems of ordinary differential equations like population dynamics models [264–266]

$$\begin{aligned} \dot{x}(t) &= f(x, y) \\ \dot{y}(t) &= g(x, y) \end{aligned} \tag{22.2}$$

or partial differential equations like the reaction diffusion equation [265, 267, 268]

$$\frac{\partial}{\partial t} c(x,t) = D \frac{\partial^2}{\partial x^2} c(x,t) + f(c) \tag{22.3}$$

where f and g are nonlinear in the mathematical sense.[1] *We discuss fixed points of the logistic mapping and analyze their stability. A bifurcation diagram visualizes the appearance of period doubling and chaotic behavior as a function of a control parameter. The Ljapunov exponent helps to distinguish stable fixed points and periods from chaotic regions. For continuous-time models, the iterated function is replaced by a system of differential equations. For stable equilibria all eigenvalues of the Jacobian matrix must have a negative real part. We discuss the Lotka–Volterra model, which is the simplest model of predator-prey interactions and the Holling-Tanner model, which incorporates functional response. Finally we allow for spatial inhomogeneity and include diffusive terms to obtain reaction-diffusion systems, which show the*

[1] *Linear functions are additive $f(x+y) = f(x) + f(y)$ and homogeneous $f(\alpha x) = \alpha f(x)$.*

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_22

phenomena of traveling waves and pattern formation. Computer experiments study orbits and bifurcation diagram of the logistic map, periodic oscillations of the Lotka–Volterra model, oscillations and limit cycles of the Holling-Tanner model and finally pattern formation in the diffusive Lotka–Volterra model.

22.1 Iterated Functions

Starting from an initial value x_0 a function f is iterated repeatedly

$$\begin{aligned} x_1 &= f(x_0) \\ x_2 &= f(x_1) \\ &\vdots \\ x_{i+1} &= f(x_i). \end{aligned} \tag{22.4}$$

The sequence of function values $x_0, x_1 \cdots$ is called the orbit of x_0. It can be visualized in a 2-dimensional plot by connecting the points

$$(x_0, x_1) \to (x_1, x_1) \to (x_1, x_2) \to (x_2, x_2) \cdots \to (x_i, x_{i+1}) \to (x_{i+1}, x_{i+1})$$

by straight lines (Fig. 22.1).

22.1.1 Fixed Points and Stability

If the equation

$$x^* = f(x^*) \tag{22.5}$$

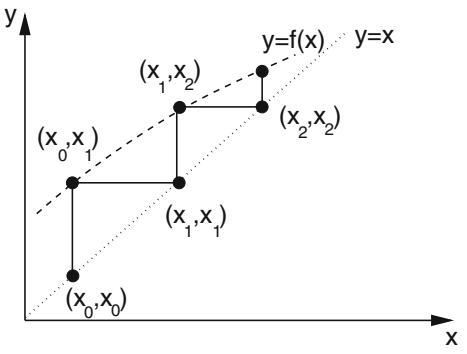


Fig. 22.1 (Orbit of an iterated function) The sequence of points $(x_i, x_{i+1}), (x_{i+1}, x_{i+1})$ is plotted together with the curves $y = f(x)$ (*dashed*) and $y = x$ (*dotted*)

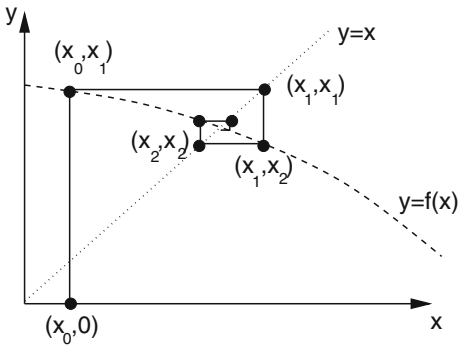


Fig. 22.2 (Attractive fixed point) The orbit of an attractive fixed point converges to the intersection of the curves $y = x$ and $y = f(x)$

has solutions x^*, then these are called fixed points. Consider a point in the vicinity of a fixed point

$$x = x^* + \varepsilon_0 \tag{22.6}$$

and make a Taylor series expansion

$$f(x) = f(x^* + \varepsilon_0) = f(x^*) + \varepsilon_0 f'(x^*) + \cdots = x^* + \varepsilon_1 + \cdots \tag{22.7}$$

with the notation

$$\varepsilon_1 = \varepsilon_0 f'(x^*). \tag{22.8}$$

Repeated iteration gives[2]

$$\begin{aligned} f^{(2)}(x) = f(f(x)) = f(x^* + \varepsilon_1) + \cdots = x^* + \varepsilon_1 f'(x^*) = x^* + \varepsilon_2 \\ \vdots \\ f^{(n)}(x^*) = x^* + \varepsilon_n \end{aligned} \tag{22.9}$$

with the sequence of deviations

$$\varepsilon_n = f'(x^*)\varepsilon_{n-1} = \cdots = \left(f'(x^*)\right)^n \varepsilon_0.$$

The orbit moves away from the fixed point for arbitrarily small ε_0 if $|f'(x^*)| > 1$ whereas the fixed point is attractive for $|f'(x^*)| < 1$ (Fig. 22.2).

Higher order fixed points are defined by iterating $f(x)$ several times. A fixed point of order n simultaneously solves

[2]Here and in the following $f^{(n)}$ denotes an iterated function, not a derivative.

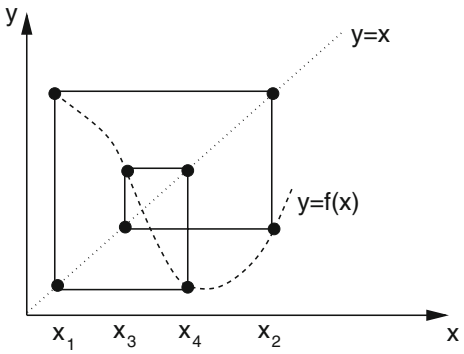


Fig. 22.3 (Periodic orbit) The orbit of an attractive fourth order fixed point cycles through the values $x_1 = f(x_4), x_2 = f(x_1), x_3 = f(x_2), x_4 = f(x_3)$

$$\begin{aligned} f(x^*) &\neq x^* \\ f^{(2)}(x^*) &\neq x^* \\ f^{(n-1)}(x^*) &\neq x^* \\ f^{(n)}(x^*) &= x^*. \end{aligned} \tag{22.10}$$

The iterated function values cycle periodically (Fig. 22.3) through

$$x^* \to f(x^*) \to f^{(2)}(x^*) \cdots f^{(n-1)}(x^*).$$

This period is attractive if

$$|f'(x^*)\, f'(f(x^*))\, f'(f^{(2)}(x^*)) \cdots f'(f^{(n-1)}(x^*))| < 1.$$

22.1.2 The Ljapunov-Exponent

Consider two neighboring orbits with initial values x_0 and $x_0 + \varepsilon_0$. After n iterations the distance is

$$|f(f(\cdots f(x_0))) - f(f(\cdots f(x_0 + \varepsilon_0)))| = |\varepsilon_0| e^{\lambda n} \tag{22.11}$$

with the so called Ljapunov-exponent [269] λ which is useful to characterize the orbit. The Ljapunov-exponent can be determined from

$$\lambda = \lim_{n\to\infty} \frac{1}{n} \ln\left(\frac{|f^{(n)}(x_0 + \varepsilon_0) - f^{(n)}(x_0)|}{|\varepsilon_0|}\right) \tag{22.12}$$

or numerically easier with the approximation

$$|f(x_0+\varepsilon_0)-f(x_0)| = |\varepsilon_0||f'(x_0)|$$

$$\begin{aligned}|f(f(x_0+\varepsilon_0))-f(f(x_0))| &= |(f(x_0+\varepsilon_0)-f(x_0))||f'(x_0+\varepsilon_0)| \\ &= |\varepsilon_0||f'(x_0)||f'(x_0+\varepsilon_0)|\end{aligned} \tag{22.13}$$

$$|f^{(n)}(x_0+\varepsilon_0)-f^{(n)}(x_0)| = |\varepsilon_0||f'(x_0)||f'(x_1)|\cdots|f'(x_{n-1})| \tag{22.14}$$

from

$$\lambda = \lim_{n\to\infty}\frac{1}{n}\sum_{i=0}^{n-1}\ln|f'(x_i)|. \tag{22.15}$$

For a stable fixed point

$$\lambda \to \ln|f'(x^*)| < 0 \tag{22.16}$$

and for an attractive period

$$\lambda \to \ln|f'(x^*)\, f'(f(x^*)\cdots f'(f^{(n-1)}(x^*))| < 0. \tag{22.17}$$

Orbits with $\lambda < 0$ are attractive fixed points or periods. If, on the other hand, $\lambda > 0$, the orbit is irregular and very sensitive to the initial conditions, hence is chaotic.

22.1.3 The Logistic Map

A population of animals is observed yearly. The evolution of the population density N is described in terms of the reproduction rate r by the recurrence relation

$$N_{n+1} = r\, N_n \tag{22.18}$$

where N_n is the population density in year number n. If r is constant, an exponential increase or decrease of N results.

The simplest model for the growth of a population which takes into account that the resources are limited is the logistic model by Verhulst [270]. He assumed that the reproduction rate r depends on the population density N in a simple way (Fig. 22.4)

$$r = r_0\left(1-\frac{N}{K}\right). \tag{22.19}$$

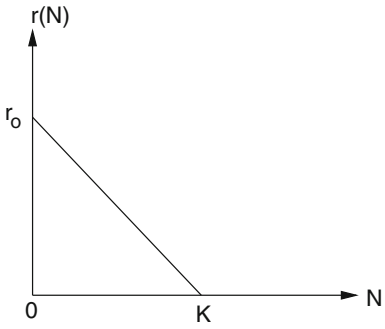


Fig. 22.4 (Reproduction rate of the logistic model) At low densities the growth rate has its maximum value r_0. At larger densities the growth rate declines and reaches $r = 0$ for $N = K$. The parameter K is called carrying capacity

The Verhulst model (22.19) leads to the iterated nonlinear function

$$N_{n+1} = r_0 N_n - \frac{r_0}{K} N_n^2 \tag{22.20}$$

with $r_0 > 0$, $K > 0$. We denote the quotient of population density and carrying capacity by the new variable

$$x_n = \frac{1}{K} N_n \tag{22.21}$$

and obtain an equation with only one parameter, the so called logistic mapping

$$x_{n+1} = \frac{1}{K} N_{n+1} = \frac{1}{K} r_0 N_n \left(1 - \frac{N_n}{K}\right) = r_0 \cdot x_n \cdot (1 - x_n). \tag{22.22}$$

22.1.4 Fixed Points of the Logistic Map

Consider an initial point in the interval

$$0 < x_0 < 1. \tag{22.23}$$

We want to find conditions on r to keep the orbit in this interval. The maximum value of x_{n+1} is found from

$$\frac{\mathrm{d}x_{n+1}}{\mathrm{d}x_n} = r(1 - 2x_n) = 0 \tag{22.24}$$

which gives $x_n = 1/2$ and $\max(x_{n+1}) = r/4$. If $r > 4$ then negative x_n appear after some iterations and the orbit is not bound by a finite interval since

$$\frac{|x_{n+1}|}{|x_n|} = |r|(1 + |x_n|) > 1. \tag{22.25}$$

The fixed point equation

$$x^* = rx^* - rx^{*2} \tag{22.26}$$

always has the trivial solution

$$x^* = 0 \tag{22.27}$$

and a further solution

$$x^* = 1 - \frac{1}{r} \tag{22.28}$$

which is only physically reasonable for $r > 1$, since x should be a positive quantity. For the logistic mapping the derivative is

$$f'(x) = r - 2rx \tag{22.29}$$

which for the first fixed point $x^* = 0$ gives $|f'(0)| = r$. This fixed point is attractive for $0 < r < 1$ and becomes unstable for $r > 1$. For the second fixed point we have $|f'(1 - \frac{1}{r})| = |2 - r|$, which is smaller than one in the interval $1 < r < 3$. For $r < 1$ no such fixed point exists. Finally, for $r_1 = 3$ the first bifurcation appears and higher order fixed points become stable.

Consider the fixed point of the double iteration

$$x^* = r(r(x^* - x^{*^2}) - r^2(x^* - x^{*2})^2). \tag{22.30}$$

All roots of this fourth order equation can be found since we already know two of them. The remaining roots are

$$x^*_{1,2} = \frac{\frac{r+1}{2} \pm \sqrt{r^2 - 2r - 3}}{r}. \tag{22.31}$$

They are real valued if

$$(r - 1)^2 - 4 > 0 \rightarrow r > 3 \quad (\text{or} \quad r < -1). \tag{22.32}$$

For $r > 3$ the orbit oscillates between x_1^* and x_2^* until the next period doubling appears for $r_2 = 1 + \sqrt{6}$. With increasing r more and more bifurcations appear and finally the orbits become chaotic (Fig. 22.5).

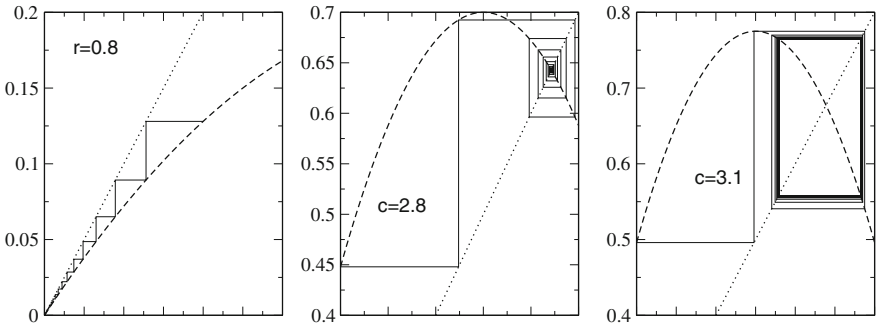


Fig. 22.5 (Orbits of the logistic map) ***Left*** For $0 < r < 1$ the logistic map has the attractive fixed point $x^* = 0$. ***Middle*** In the region $1 < r < 3$ this fixed point becomes unstable and another stable fixed point is at $x^* = 1 - 1/r$. ***Right*** For $3 < r < 1 + \sqrt{6}$ the second order fixed point (22.31) is stable. For larger values of r more and more bifurcations appear

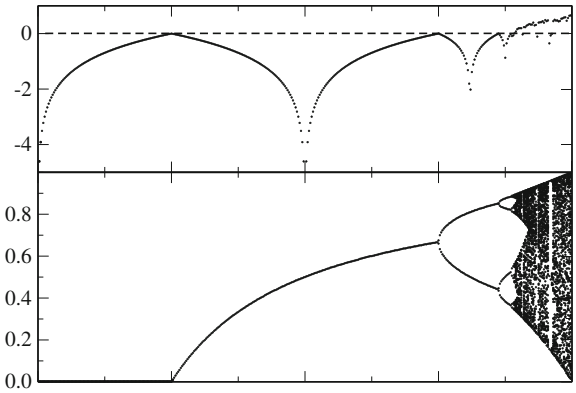


Fig. 22.6 (Bifurcation diagram of the logistic map) For different values of r the function is iterated 1100 times. The first 1000 iterations are dropped to allow the trajectory to approach stable fixed points or periods. The iterated function values $x_{1000} \cdots x_{1100}$ are plotted in a r-x diagram together with the estimate (22.15) of the Ljapunov exponent. The first period doublings appear at $r = 3$ and $r = 1 + \sqrt{6}$. For larger values chaotic behavior is observed and the estimated Ljapunov exponent becomes positive. In some regions motion is regular again with negative Ljapunov exponent

22.1.5 Bifurcation Diagram

The bifurcation diagram visualizes the appearance of period doubling and chaotic behavior as a function of the control parameter r (Fig. 22.6).

22.2 Population Dynamics

If time is treated as a continuous variable, the iterated function has to be replaced by a differential equation

$$\frac{\mathrm{d}N}{\mathrm{d}t} = f(N) \tag{22.33}$$

or, more generally by a system of equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} N_1 \\ N_2 \\ \vdots \\ N_n \end{pmatrix} = \begin{pmatrix} f_1(N_1 \cdots N_n) \\ f_2(N_1 \cdots N_n) \\ \\ f_n(N_1 \cdots N_n) \end{pmatrix}. \tag{22.34}$$

22.2.1 Equilibria and Stability

The role of the fixed points is now taken over by equilibria, which are solutions of

$$0 = \frac{\mathrm{d}N}{\mathrm{d}t} = f(N_{eq}) \tag{22.35}$$

which means roots of $f(N)$. Let us investigate small deviations from equilibrium with the help of a Taylor series expansion. Inserting

$$N = N_{eq} + \xi \tag{22.36}$$

we obtain

$$\frac{\mathrm{d}\xi}{\mathrm{d}t} = f(N_{eq}) + f'(N_{eq})\xi + \cdots \tag{22.37}$$

but since $f(N_{eq}) = 0$, we have approximately

$$\frac{\mathrm{d}\xi}{\mathrm{d}t} = f'(N_{eq})\xi \tag{22.38}$$

with the solution

$$\xi(t) = \xi_0 \exp\left\{f'(N_{eq})t\right\}. \tag{22.39}$$

The equilibrium is only stable if $\mathfrak{Re}\, f'(N_{eq}) < 0$, since then small deviations disappear exponentially. For $\mathfrak{Re}\, f'(N_{eq}) > 0$ deviations will increase, but the

exponential behavior holds only for not too large deviations and saturation may appear. If the derivative $f'(N_{eq})$ has a nonzero imaginary part then oscillations will be superimposed. For a system of equations the equilibrium is defined by

$$\begin{pmatrix} f_1(N_1^{eq} \cdots N_n^{eq}) \\ f_2(N_1^{eq} \cdots N_n^{eq}) \\ \\ f_N(N_1^{eq} \cdots N_n^{eq}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{22.40}$$

and if such an equilibrium exists, linearization gives

$$\begin{pmatrix} N_1 \\ N_2 \\ \vdots \\ N_n \end{pmatrix} = \begin{pmatrix} N_1^{eq} \\ N_2^{eq} \\ \vdots \\ N_n^{eq} \end{pmatrix} + \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{pmatrix} \tag{22.41}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_N \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial N_1} & \frac{\partial f_1}{\partial N_2} & \cdots & \frac{\partial f_1}{\partial N_n} \\ \frac{\partial f_2}{\partial N_1} & \frac{\partial f_2}{\partial N_2} & \cdots & \frac{\partial f_2}{\partial N_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial N_1} & \frac{\partial f_n}{\partial N_2} & \cdots & \frac{\partial f_n}{\partial N_n} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{pmatrix}. \tag{22.42}$$

The equilibrium is stable if all eigenvalues λ_i of the derivative matrix have a negative real part.

22.2.2 The Continuous Logistic Model

The continuous logistic model describes the evolution by the differential equation

$$\frac{dx}{dt} = r_0 x(1-x). \tag{22.43}$$

To find possible equilibria we have to solve

$$x_{eq}(1-x_{eq}) = 0 \tag{22.44}$$

which has the two roots $x_{eq} = 0$ and $x_{eq} = 1$ (Fig. 22.7).

The derivative f' is

$$f'(x) = \frac{\mathrm{d}}{\mathrm{d}x}(r_0 x(1-x)) = r_0(1-2x). \tag{22.45}$$

Since $f'(0) = r_0 > 0$ and $f'(1) = -r_0 < 0$ only the second equilibrium is stable.

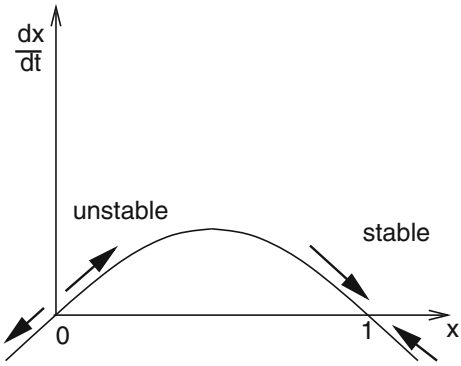


Fig. 22.7 (Equilibria of the logistic model) The equilibrium $x_{eq} = 0$ is unstable since an infinitesimal deviation grows exponentially in time. The equilibrium $x_{eq} = 1$ is stable since initial deviations disappear exponentially

22.3 Lotka–Volterra Model

The model by Lotka [271] and Volterra [272] is the simplest model of predator-prey interactions. It has two variables, the density of prey (H) and the density of predators (P). The overall reproduction rate of each species is given by the difference of the birth rate r and the mortality rate m

$$\frac{\mathrm{d}N}{\mathrm{d}t} = (r - m)N$$

which both may depend on the population densities. The Lotka–Volterra model assumes that the prey mortality depends linearly on the predator density and the predator birth rate is proportional to the prey density

$$m_H = aP \quad r_P = bH \tag{22.46}$$

where a is the predation rate coefficient and b is the reproduction rate of predators per 1 prey eaten. Together we end up with a system of two coupled nonlinear differential equations

$$\frac{\mathrm{d}H}{\mathrm{d}t} = f(H, P) = r_H H - aHP$$

$$\frac{\mathrm{d}P}{\mathrm{d}t} = g(H, P) = bHP - m_P P \tag{22.47}$$

where r_H is the intrinsic rate of prey population increase and m_P the predator mortality rate.

22.3.1 Stability Analysis

To find equilibria we have to solve the system of equations

$$f(H, P) = r_H H - aHP = 0$$

$$g(H, P) = bHP - m_P P = 0. \tag{22.48}$$

The first equation is solved by $H_{eq} = 0$ or by $P_{eq} = r_H/a$. The second equation is solved by $P_{eq} = 0$ or by $H_{eq} = m_P/b$. Hence there are two equilibria, the trivial one

$$P_{eq} = H_{eq} = 0 \tag{22.49}$$

and a nontrivial one

$$P_{eq} = \frac{r_H}{a} \quad H_{eq} = \frac{m_P}{b}. \tag{22.50}$$

Linearization around the zero equilibrium gives

$$\frac{\mathrm{d}H}{\mathrm{d}t} = r_H H + \cdots \quad \frac{\mathrm{d}P}{\mathrm{d}t} = -m_P P + \cdots . \tag{22.51}$$

This equilibrium is unstable since a small prey population will increase exponentially. Now expand around the nontrivial equilibrium:

$$P = P_{eq} + \xi, \quad H = H_{eq} + \eta \tag{22.52}$$

$$\frac{\mathrm{d}\eta}{\mathrm{d}t} = \frac{\partial f}{\partial H}\eta + \frac{\partial f}{\partial P}\xi = (r_H - aP_{eq})\eta - aH_{eq}\xi = -\frac{am_P}{b}\xi \tag{22.53}$$

$$\frac{\mathrm{d}\xi}{\mathrm{d}t} = \frac{\partial g}{\partial H}\eta + \frac{\partial g}{\partial P}\xi = bP_{eq}\eta + (bH_{eq} - m_P)\xi = \frac{br_H}{a}\eta \tag{22.54}$$

or in matrix notation

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\eta\\ \xi\end{pmatrix} = \begin{pmatrix}0 & -\frac{am_P}{b}\\ \frac{br_H}{a} & 0\end{pmatrix}\begin{pmatrix}\eta\\ \xi\end{pmatrix}. \tag{22.55}$$

The eigenvalues are purely imaginary

$$\lambda = \pm i\sqrt{m_H r_P} = \pm \mathrm{i}\omega \tag{22.56}$$

and the corresponding eigenvectors are

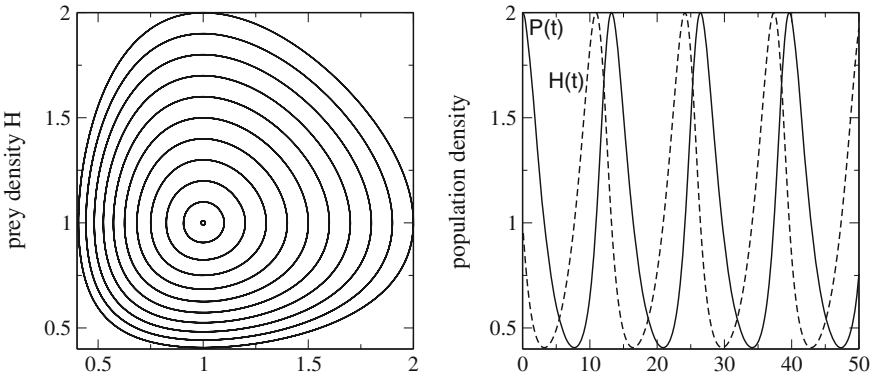


Fig. 22.8 (Lotka Volterra model) The predator and prey population densities show periodic oscillations (***Right***). In the H-P plane the system moves on a closed curve, which becomes an ellipse for small deviations from equilibrium (***Left***)

$$\begin{pmatrix} \mathrm{i}\sqrt{m_H r_p} \\ b r_H / a \end{pmatrix}, \begin{pmatrix} a m_P / b \\ \mathrm{i}\sqrt{m_H r_P} \end{pmatrix}. \tag{22.57}$$

The solution of the linearized equations is then given by

$$\begin{aligned} \xi(t) &= \xi_0 \cos \omega t + \frac{b}{a}\sqrt{\frac{r_P}{m_H}}\eta_0 \sin \omega t \\ \eta(t) &= \eta_0 \cos \omega t - \frac{a}{b}\sqrt{\frac{m_H}{r_P}}\xi_0 \sin \omega t \end{aligned} \tag{22.58}$$

which describes an ellipse in the $\xi - \eta$ plane (Fig. 22.8). The nonlinear equations (22.48) have a first integral

$$r_H \ln P(t) - a\, P(t) - b\, H(t) + m_P \ln H(t) = C \tag{22.59}$$

and therefore the motion in the $H - P$ plane is on a closed curve around the equilibrium which approaches an ellipse for small amplitudes ξ, η.

22.4 Functional Response

Holling [273, 274] studied predation of small mammals on pine sawflies. He suggested a very popular model of functional response. Holling assumed that the predator spends its time on two kinds of activities, searching for prey and prey handling (chasing, killing, eating, digesting). The total time equals the sum of time spent on searching and time spent on handling

$$T = T_{search} + T_{handling}. \tag{22.60}$$

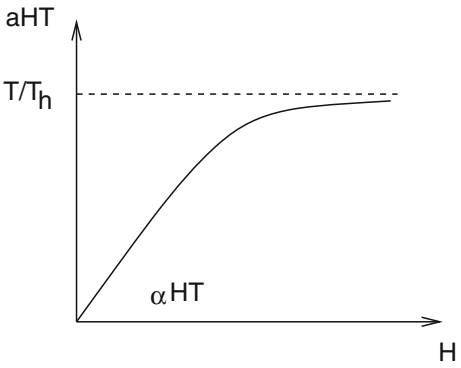


Fig. 22.9 Functional response of Holling's model

Capturing prey is assumed to be a random process. A predator examines an area α per time and captures all prey found there. After spending the time T_{search} the predator examined an area of αT_{search} and captured $H_T = H\alpha T_{search}$ prey. Hence the predation rate is

$$a = \frac{H_T}{HT} = \alpha \frac{T_{search}}{T} = \alpha \frac{1}{1 + T_{handling}/T_{search}}. \tag{22.61}$$

The handling time is assumed to be proportional to the number of prey captured

$$T_{handling} = T_h H \alpha T_{search} \tag{22.62}$$

where T_h is the handling time spent per one prey. The predation rate then is given by

$$a = \frac{\alpha}{1 + \alpha H T_h}. \tag{22.63}$$

At small densities handling time is unimportant and the predation rate is $a_0 = \alpha$ whereas at high prey density handling limits the number of prey captured and the predation rate approaches $a_\infty = \frac{1}{HT_h}$ (Fig. 22.9).

22.4.1 Holling-Tanner Model

We combine the logistic model with Holling's model for the predation rate [273–275]

$$\frac{dH}{dt} = r_H H \left(1 - \frac{H}{K_H}\right) - aHP = r_H H \left(1 - \frac{H}{K_H}\right) - \frac{\alpha}{1 + \alpha H T_h} HP = f(H, P) \tag{22.64}$$

and assume that the carrying capacity of the predator is proportional to the density of prey

$$\frac{dP}{dt} = r_P P\left(1 - \frac{P}{K_P}\right) = r_P P\left(1 - \frac{P}{kH}\right) = g(H, P). \tag{22.65}$$

Obviously there is a trivial equilibrium with $P_{eq} = H_{eq} = 0$. Linearization gives

$$\frac{\mathrm{d}H}{\mathrm{d}t} = r_H H + \cdots \quad \frac{\mathrm{d}P}{\mathrm{d}t} = r_P P + \cdots \tag{22.66}$$

which shows that this equilibrium is unstable. There is another trivial equilibrium with $P_{eq} = 0, \quad H_{eq} = K_H$. After linearization

$$P = \xi + \ldots \quad H = K_H + \eta + \ldots \tag{22.67}$$

we find

$$\frac{\mathrm{d}\eta}{\mathrm{d}t} = r_H(K_H + \eta)(1 - \frac{K_H + \eta}{K_H}) - \frac{\alpha}{1 + \alpha(K_H + \eta)T_h}(K_H + \eta)\xi + \ldots$$

$$= -r_H\eta - \frac{\alpha}{1 + \alpha K_H T_h} K_H \xi + \ldots \tag{22.68}$$

$$\frac{\mathrm{d}\xi}{\mathrm{d}t} = r_P \xi. \tag{22.69}$$

The eigenvalues of the linearized equations

$$\begin{pmatrix} \dot{\eta} \\ \dot{\xi} \end{pmatrix} = \begin{pmatrix} r_H & -\frac{\alpha}{1+\alpha K_H T_h} K_H \\ 0 & r_P \end{pmatrix} \begin{pmatrix} \eta \\ \xi \end{pmatrix} \tag{22.70}$$

are

$$\lambda = \frac{r_H + r_P}{2} \pm \frac{1}{2}\sqrt{(r_H - r_P)^2} = r_H, r_P. \tag{22.71}$$

Let us now look for nontrivial equilibria. The nullclines (Fig. 22.10) are the curves defined by $\frac{dH}{dt} = 0$ and $\frac{dP}{dt} = 0$, hence by

$$P = \frac{r_H}{\alpha}\left(1 - \frac{H}{K_H}\right)(1 + \alpha H T_h) \tag{22.72}$$

$$P = kH. \tag{22.73}$$

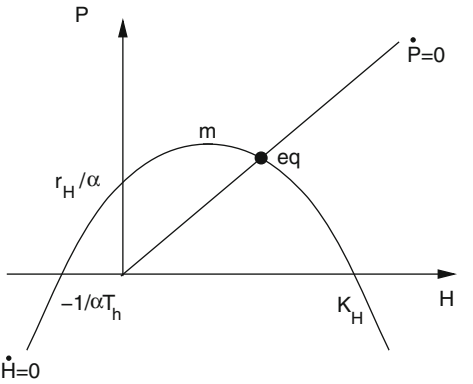


Fig. 22.10 Nullclines of the predator prey model

The H-nullcline is a parabola at

$$H_m = \frac{\alpha T_h - K_H^{-1}}{2\alpha T_h K_H^{-1}} \quad P_m = \frac{(\alpha T_h + K_H^{-1})^2}{4\alpha T_h K_H^{-1}} > 0. \tag{22.74}$$

It intersects the H-axis at $H = K_H$ and $H = -1/\alpha T_h$ and the P-axis at $P = r_H/\alpha$. There is one intersection of the two nullclines at positive values of H and P which corresponds to a nontrivial equilibrium. The equilibrium density H_{eq} is the positive root of

$$r_H \alpha T_h H_{eq}^2 + (r_H + \alpha k K_H - r_H K_H \alpha T_h)\, H_{eq} - r_H K_H = 0. \tag{22.75}$$

It is explicitly given by

$$\begin{aligned} H_{eq} = &-\frac{r_H + \alpha k K_H - r_H K_H \alpha T_h}{2 r_H \alpha T_h} \\ &+ \frac{\sqrt{(r_H + \alpha k K_H - r_H K_H \alpha T_h)^2 + 4 r_H \alpha T_h r_H K_H}}{2 r_H \alpha T_h}. \end{aligned} \tag{22.76}$$

The prey density then follows from

$$P_{eq} = H_{eq} k. \tag{22.77}$$

The matrix of derivatives has the elements

$$\begin{aligned} m_{HP} &= \frac{\partial f}{\partial P} = -\frac{\alpha H_{eq}}{1 + \alpha T_h H_{eq}} \\ m_{HH} &= \frac{\partial f}{\partial H} = r_H \left(1 - 2\frac{H_{eq}}{K_H}\right) - \frac{\alpha k H_{eq}}{1 + \alpha T_h H_{eq}} + \frac{\alpha^2 H_{eq}^2 k T_h}{(1 + \alpha T_h H_{eq})^2} \\ m_{PP} &= \frac{\partial g}{\partial P} = = -r_P \end{aligned}$$

$$m_{PH} = \frac{\partial g}{\partial H} = r_P k \tag{22.78}$$

from which the eigenvalues are calculated as

$$\lambda = \frac{m_{HH} + m_{PP}}{2} \pm \sqrt{\frac{(m_{HH} + m_{PP})^2}{4} - (m_{HH} m_{PP} - m_{HP} m_{PH})}. \tag{22.79}$$

Oscillations appear, if the squareroot is imaginary (Fig. 22.11).

22.5 Reaction-Diffusion Systems

So far we considered spatially homogeneous systems where the density of a population, or the concentration of a chemical agent, depend only on time. If we add spatial inhomogeneity and diffusive motion, new and interesting phenomena like pattern formation or traveling excitations can be observed.

22.5.1 General Properties of Reaction-Diffusion Systems

Reaction-diffusion systems are described by a diffusion equation[3] where the source term depends non-linearly on the concentrations

$$\frac{\partial}{\partial t} \begin{pmatrix} c_1 \\ \vdots \\ c_N \end{pmatrix} = \begin{pmatrix} D_1 & & \\ & \ddots & \\ & & D_N \end{pmatrix} \triangle \begin{pmatrix} c_1 \\ \vdots \\ c_N \end{pmatrix} + \begin{pmatrix} F_1(\{c\}) \\ \vdots \\ F_N(\{c\}) \end{pmatrix}. \tag{22.80}$$

22.5.2 Chemical Reactions

Consider a number of chemical reactions which are described by stoichiometric equations

$$\sum_i \nu_i A_i = 0. \tag{22.81}$$

[3] We consider only the case, that different species diffuse independently and that the diffusion constants do not depend on direction.

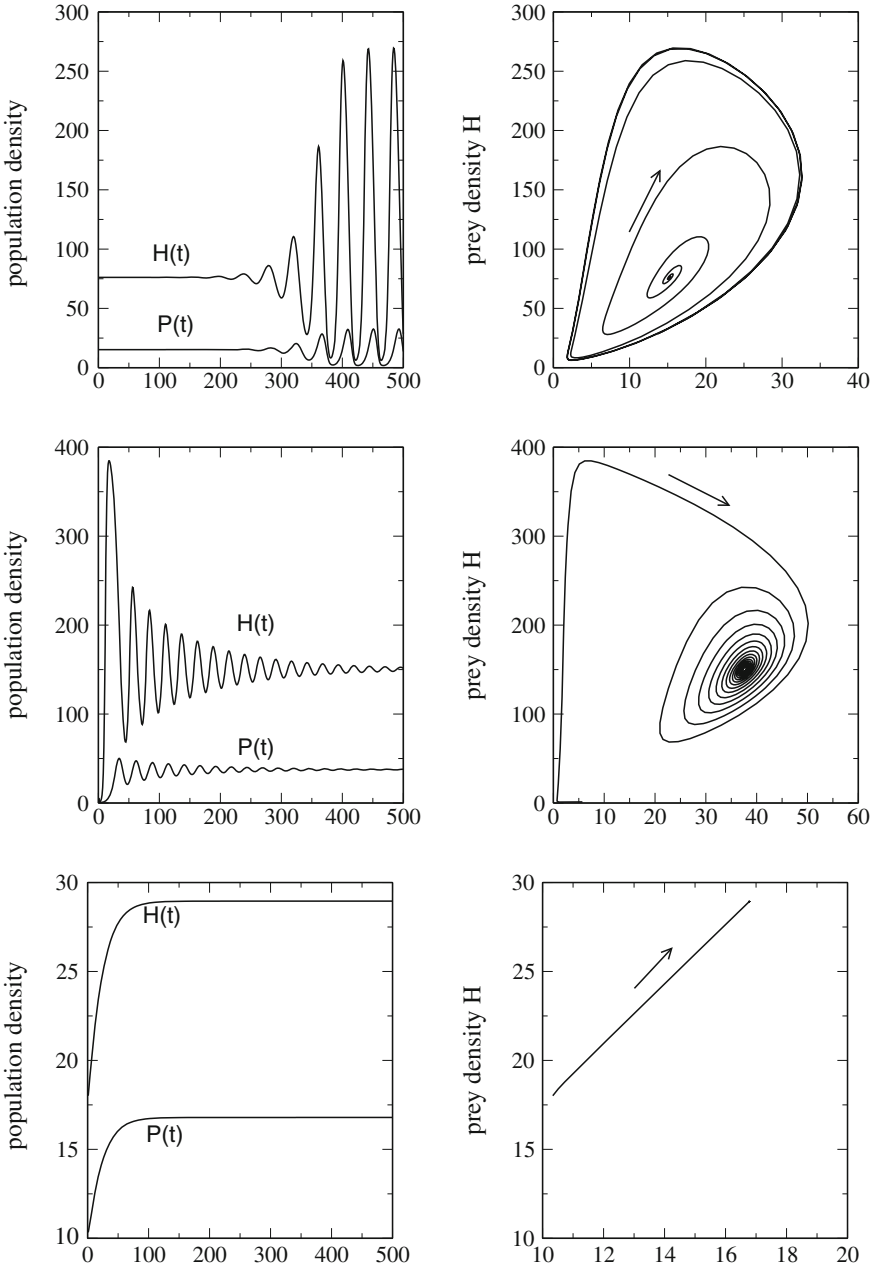


Fig. 22.11 (Holling-Tanner model) ***Top*** evolution from an unstable equilibrium to a limit cycle. ***Middle*** a stable equilibrium is approached with oscillations. ***Bottom*** stable equilibrium without oscillations

The concentration of agent A_i is

$$c_i = c_{i,0} + \nu_i x \tag{22.82}$$

with the reaction variable

$$x = \frac{c_i - c_{i,0}}{\nu_i} \tag{22.83}$$

and the reaction rate

$$r = \frac{\mathrm{d}x}{\mathrm{d}t} = \frac{1}{\nu_i}\frac{\mathrm{d}c_i}{\mathrm{d}t} \tag{22.84}$$

which, in general is a nonlinear function of all concentrations. The total concentration change due to diffusion and reactions is given by

$$\frac{\partial}{\partial t}c_k = D_k \triangle c_k + \sum_j \nu_{kj} r_j = D_k \triangle c_k + F_k(\{c_i\}). \tag{22.85}$$

22.5.3 Diffusive Population Dynamics

Combination of population dynamics (22.2) and diffusive motion gives a similar set of coupled equations for the population densities

$$\frac{\partial}{\partial t}N_k = D_k \triangle N_k + f_k(N_1, N_2, \cdots N_n). \tag{22.86}$$

22.5.4 Stability Analysis

Since a solution of the nonlinear equations is not generally possible we discuss small deviations from an equilibrium solution N_k^{eq}[4] with

$$\frac{\partial}{\partial t}N_k = \triangle N_k = 0. \tag{22.87}$$

Obviously the equilibrium obeys

$$f_k(N_1 \cdots N_n) = 0 \quad k = 1, 2 \cdots n. \tag{22.88}$$

[4] We assume tacitly that such a solution exists.

We linearize the equations by setting

$$N_k = N_k^{eq} + \xi_k \tag{22.89}$$

and expand around the equilibrium

$$\frac{\partial}{\partial t}\begin{pmatrix}\xi_1\\ \xi_2\\ \vdots\\ \xi_n\end{pmatrix} = \begin{pmatrix} D_1 & & \\ & \ddots & \\ & & D_N\end{pmatrix}\begin{pmatrix}\triangle\xi_1\\ \triangle\xi_2\\ \vdots\\ \triangle\xi_n\end{pmatrix} + \begin{pmatrix} \frac{\partial f_1}{\partial N_1} & \frac{\partial f_1}{\partial N_2} & \cdots & \frac{\partial f_1}{\partial N_n}\\ \frac{\partial f_2}{\partial N_1} & \frac{\partial f_2}{\partial N_2} & \cdots & \frac{\partial f_2}{\partial N_n}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial f_n}{\partial N_1} & \frac{\partial f_n}{\partial N_2} & \cdots & \frac{\partial f_n}{\partial N_n}\end{pmatrix}\begin{pmatrix}\xi_1\\ \xi_2\\ \vdots\\ \xi_n\end{pmatrix} + \cdots . \tag{22.90}$$

Plane waves are solutions of the linearized problem.[5] Using the ansatz

$$\xi_j = \xi_{j,0} \mathrm{e}^{\mathrm{i}(\omega t - \mathbf{kx})} \tag{22.91}$$

we obtain

$$i\omega \begin{pmatrix}\xi_1\\ \xi_2\\ \vdots\\ \xi_n\end{pmatrix} = -k^2 D \begin{pmatrix}\xi_1\\ \xi_2\\ \vdots\\ \xi_n\end{pmatrix} + M_0 \begin{pmatrix}\xi_1\\ \xi_2\\ \vdots\\ \xi_n\end{pmatrix} \tag{22.92}$$

where M_0 denotes the matrix of derivatives and D the matrix of diffusion constants. For a stable plane wave solution $\lambda = i\omega$ is an eigenvalue of

$$M_k = M_0 - k^2 D \tag{22.93}$$

with

$$\Re(\lambda) \leq 0. \tag{22.94}$$

If there are purely imaginary eigenvalues for some $\mathbf{k}$ they correspond to stable solutions which are spatially inhomogeneous and lead to formation of certain patterns. Interestingly, diffusion can lead to instabilities even for a system which is stable in the absence of diffusion [276].

[5] Strictly this is true only for an infinite or periodic system.

22.5.5 Lotka Volterra Model with Diffusion

As a simple example we consider again the Lotka Volterra model. Adding diffusive terms we obtain the equations

$$\frac{\partial}{\partial t}\begin{pmatrix} H \\ P \end{pmatrix} = \begin{pmatrix} r_H H - aHP \\ bHP - m_P P \end{pmatrix} + \begin{pmatrix} D_H & \\ & D_P \end{pmatrix} \Delta \begin{pmatrix} H \\ P \end{pmatrix}. \tag{22.95}$$

There are two equilibria

$$H_{eq} = P_{eq} = 0 \tag{22.96}$$

and

$$P_{eq} = \frac{r_H}{a} \quad H_{eq} = \frac{m_P}{b}. \tag{22.97}$$

The Jacobian matrix is

$$M_0 = \frac{\partial}{\partial C} F(C_0) = \begin{pmatrix} r_H - aP_{eq} & -aH_{eq} \\ bP_{eq} & bH_{eq} - m_P \end{pmatrix} \tag{22.98}$$

which gives for the trivial equilibrium

$$M_k = \begin{pmatrix} r_H - D_H k^2 & 0 \\ 0 & -m_P - D_P k^2 \end{pmatrix}. \tag{22.99}$$

One eigenvalue $\lambda_1 = -m_P - D_P k^2$ is negative whereas the second $\lambda_2 = r_H - D_H k^2$ is positive for $k^2 < r_H/D_H$. Hence this equilibrium is unstable against fluctuations with long wavelengths. For the second equilibrium we find:

$$M_k = \begin{pmatrix} -D_H k^2 & -\frac{am_P}{b} \\ \frac{br_H}{a} & -D_P k^2 \end{pmatrix} \tag{22.100}$$

$$\text{tr}\,(M_k) = -(D_H + D_P)k^2$$

$$\det(M_K) = m_P r_H + D_H D_P k^4$$

$$\lambda = -\frac{D_H + D_P}{2} k^2 \pm \frac{1}{2}\sqrt{(D_H - D_P)^2 k^4 - 4m_P r_H}. \tag{22.101}$$

For small k with $k^2 < 2\sqrt{m_P r_H}/|D_H - D_P|$ damped oscillations are expected whereas the system is stable against fluctuations with larger k (Figs. 22.12, 22.13 and 22.14).

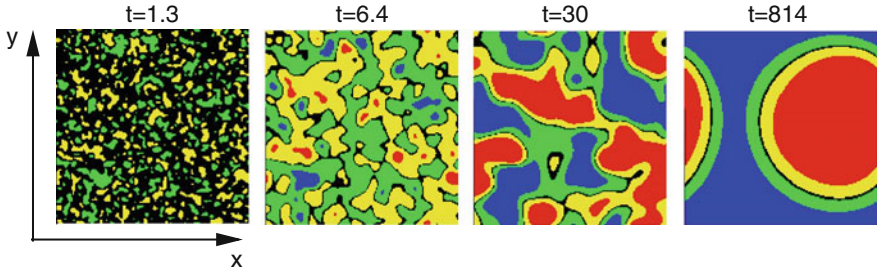


Fig. 22.12 (Lotka–Volterra model with diffusion) The time evolution is calculated for initial random fluctuations. *Colors* indicate the deviation of the predator concentration $P(x, y, t)$ from its average value (*blue*: $\Delta P < -0.1$, *green*: $-0.1 < \Delta P < -0.01$, *black*: $-0.01 < \Delta P < 0.01$, *yellow*: $0.01 < \Delta P < 0.1$, *red*: $\Delta P > 0.1$). Parameters as in Fig. 22.13

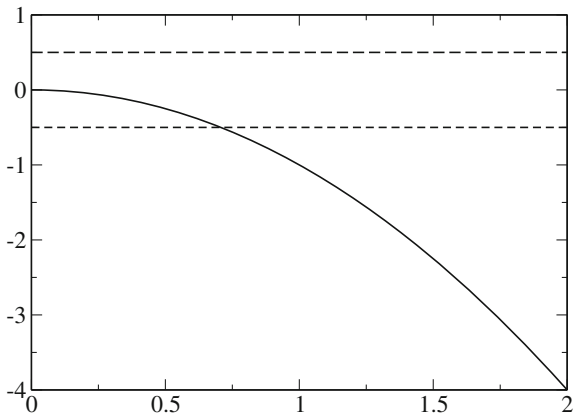


Fig. 22.13 (Dispersion of the diffusive Lotka–Volterra model) Real (*full curve*) and imaginary part (*broken line*) of the eigenvalue λ (22.101) are shown as a function of k. Parameters are $D_H = D_P = 1$, $m_P = r_H = a = b = 0.5$

Problems

Problem 22.1: Orbits of the Iterated Logistic Map

This computer example draws orbits (Fig. 22.5) of the logistic map

$$x_{n+1} = r_0 \cdot x_n \cdot (1 - x_n). \tag{22.102}$$

You can select the initial value x_0 and the variable r.

Problem 22.2: Bifurcation Diagram of the Logistic Map

This computer example generates a bifurcation diagram of the logistic map (Fig. 22.6). You can select the range of r.

Problem 22.3: Lotka–Volterra Model

Equation (22.47) are solved with the improved Euler method (Fig. 22.8). The predictor step uses an explicit Euler step to calculate the values at $t + \Delta t/2$

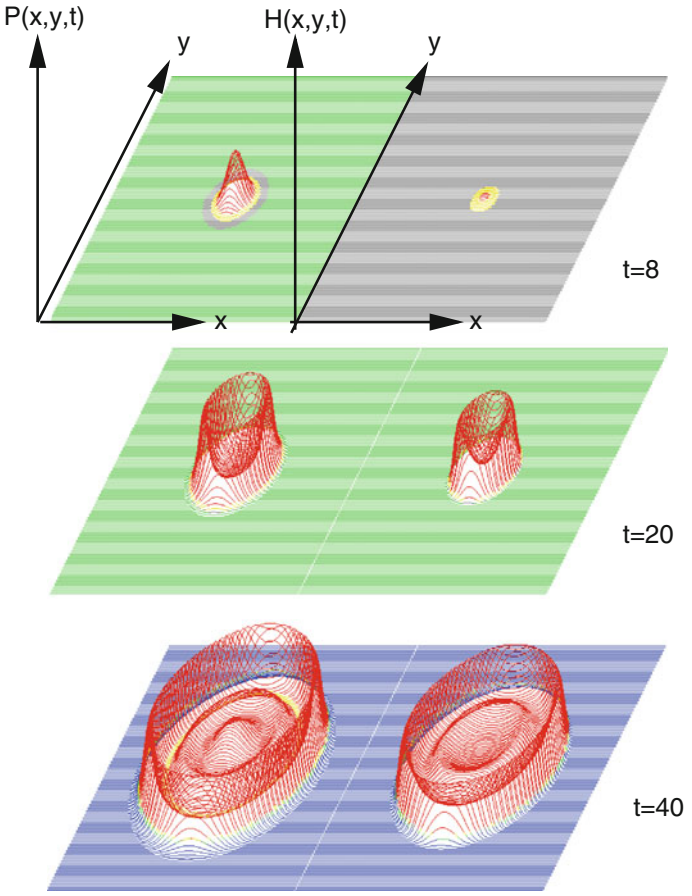


Fig. 22.14 (Traveling waves in the diffusive Lotka–Volterra model) Initially $P(x, y) = P_{eq}$ and $H(x, y)$ is peaked in the center. This leads to oscillations and a sharp wavefront moving away from the excitation. Color code and parameters as in Fig. 22.12

$$H_{pr}(t + \frac{\Delta t}{2}) = H(t) + (r_H H(t) - aH(t)P(t))\,\frac{\Delta t}{2} \tag{22.103}$$

$$P_{pr}(t + \frac{\Delta t}{2}) = P(t) + \left(bH(t)P(t) - m_p P(t)\right)\frac{\Delta t}{2} \tag{22.104}$$

and the corrector step advances time by Δt

$$H(t + \Delta t) = H(t) + \left(r_H H_{pr}(t + \frac{\Delta t}{2}) - aH_{pr}(t + \frac{\Delta t}{2})P_{pr}(t + \frac{\Delta t}{2})\right)\Delta t \tag{22.105}$$

$$P(t+\Delta t) = P(t) + \left(bH_{pr}(t+\frac{\Delta t}{2}) P_{pr}(t+\frac{\Delta t}{2}) - m_p P_{pr}(t+\frac{\Delta t}{2}) \right) \Delta t. \tag{22.106}$$

Problem 22.4: Holling-Tanner Model

The equations of the Holling-Tanner model (22.64), (22.65) are solved with the improved Euler method (see Fig. 22.11). The predictor step uses an explicit Euler step to calculate the values at $t + \Delta t/2$:

$$H_{pr}(t+\frac{\Delta t}{2}) = H(t) + f(H(t), P(t))\frac{\Delta t}{2} \tag{22.107}$$

$$P_{pr}(t+\frac{\Delta t}{2}) = P(t) + g(H(t), P(t))\frac{\Delta t}{2} \tag{22.108}$$

and the corrector step advances time by Δt:

$$H(t+\Delta t) = H(t) + f(H_{pr}(t+\frac{\Delta t}{2}), P_{pr}(t+\frac{\Delta t}{2}))\Delta t \tag{22.109}$$

$$P(t+\Delta t) = P(t) + g(H_{pr}(t+\frac{\Delta t}{2}), P_{pr}(t+\frac{\Delta t}{2}))\Delta t. \tag{22.110}$$

Problem 22.5: Diffusive Lotka–Volterra Model

The Lotka–Volterra model with diffusion (22.95) is solved in 2 dimensions with an implicit method (21.2.2) for the diffusive motion (Figs. 22.12 and 22.14). The split operator approximation (21.3) is used to treat diffusion in x and y direction independently. The equations

$$\begin{pmatrix} H(t+\Delta t) \\ P(t+\Delta t) \end{pmatrix} = \begin{pmatrix} A^{-1}H(t) \\ A^{-1}P(t) \end{pmatrix} + \begin{pmatrix} A^{-1}f(H(t), P(t))\Delta t \\ A^{-1}g(H(t), P(t))\Delta t \end{pmatrix}$$

$$\approx \begin{pmatrix} A_x^{-1}A_y^{-1}\left[H(t) + f(H(t), P(t))\Delta t\right] \\ A_x^{-1}A_y^{-1}\left[P(t) + g(H(t), P(t))\Delta t\right] \end{pmatrix} \tag{22.111}$$

are equivalent to the following systems of linear equations with tridiagonal matrix (5.3):

$$A_y U = H(t) + f(H(t), P(t))\Delta t \tag{22.112}$$

$$U = A_x H(t+\Delta t) \tag{22.113}$$

$$A_y V = P(t) + g(H(t), P(t))\Delta t \tag{22.114}$$

$$V = A_x P(t+\Delta t). \tag{22.115}$$

Periodic boundary conditions are implemented with the method described in Sect. 5.4.

Chapter 23
Simple Quantum Systems

In this chapter we study simple quantum systems. A particle in a one-dimensional potential $V(x)$ is described by a wave packet which is a solution of the partial differential equation [277]

$$\mathrm{i}\hbar\frac{\partial}{\partial t}\psi(x) = H\psi(x) = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}\psi(x) + V(x)\psi(x). \tag{23.1}$$

We discuss two approaches to discretize the second derivative. Finite differences are simple to use but their dispersion deviates largely from the exact relation, except high order differences are used. Pseudo-spectral methods evaluate the kinetic energy part in Fourier space and are much more accurate. The time evolution operator can be approximated by rational expressions like Cauchy's form which corresponds to the Crank-Nicholson method. These schemes are unitary but involve time consuming matrix inversions. Multistep differencing schemes have comparable accuracy but are explicit methods. Best known is second order differencing. Split operator methods approximate the time evolution operator by a product. In combination with finite differences for the kinetic energy this leads to the method of real-space product formula which can be applied to wavefunctions with more than one component, for instance to study transitions between states. In a computer experiment we simulate a one-dimensional wave packet in a potential with one or two minima.

Few-state systems are described with a small set of basis states. Especially the quantum mechanical two-level system is often used as a simple model for the transition between an initial and a final state due to an external perturbation.[1] *Its wavefunction has two components*

$$|\psi> = \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \tag{23.2}$$

which satisfy two coupled ordinary differential equations for the amplitudes $C_{1,2}$ of the two states

[1] *For instance collisions or the electromagnetic radiation field.*

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_23

$$\mathrm{i}\hbar\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}C_1\\C_2\end{pmatrix}=\begin{pmatrix}H_{11}\ H_{12}\\H_{21}\ H_{22}\end{pmatrix}\begin{pmatrix}C_1\\C_2\end{pmatrix}. \tag{23.3}$$

In several computer experiments we study a two-state system in an oscillating field, a three-state system as a model for superexchange, the semiclassical model and the Landau–Zener model for curve-crossing and the ladder model for exponential decay. The density matrix formalism is used to describe a dissipative two-state system in analogy to the Bloch equations for nuclear magnetic resonance. In computer experiments we study the resonance line and the effects of saturation and power broadening. Finally we simulate the generation of a coherent superposition state or a spin flip by applying pulses of suitable duration. This is also discussed in connection with the manipulation of a Qubit represented by a single spin.

23.1 Pure and Mixed Quantum States

Whereas pure states of a quantum system are described by a wavefunction, mixed states are described by a density matrix. Mixed states appear if the exact quantum state is unknown, for instance for a statistical ensemble of quantum states, a system with uncertain preparation history, or if the system is entangled with another system. A mixed state is different from a superposition state. For instance, the superposition

$$|\psi>=C_0|\psi_0>+C_1|\psi_1> \tag{23.4}$$

of the two states $|\psi_0>$ and $|\psi_1>$ is a pure state, which can be described by the density operator

$$\begin{aligned}&|\psi><\psi|=|C_0|^2|\psi_0><\psi_0|+|C_1|^2|\psi_1><\psi_1|\\&+C_0C_1^*|\psi_0><\psi_1|+C_0^*C_1|\psi_1><\psi_0|\end{aligned} \tag{23.5}$$

whereas the density operator

$$\rho=p_0|\psi_0><\psi_0|+p_1|\psi_1><\psi_1| \tag{23.6}$$

describes the mixed state of a system which is in the pure state $|\psi_0>$ with probability p_0 and in the state $|\psi_1>$ with probability $p_1=1-p_0$. The expectation value of an operator A is in the first case

$$\begin{aligned}&<A>=<\psi|A|\psi>=|C_0|^2<\psi_0|A|\psi_0>+|C_1|^2<\psi_1A|\psi_1>\\&+C_0C_1^*<\psi_1|A|\psi_0>+C_0^*C_1<\psi_0|A|\psi_1>\end{aligned} \tag{23.7}$$

and in the second case

$$<A>=p_0<\psi_0|A|\psi_0>+p_1<\psi_1|A|\psi_1>. \tag{23.8}$$

Both can be written in the form

$$< A >= \mathrm{tr}(\rho A). \tag{23.9}$$

23.1.1 Wavefunctions

The time evolution of a quantum system is governed by the time dependent Schroedinger equation [278]

$$\mathrm{i}\hbar \frac{\partial}{\partial t} |\psi >= H|\psi > \tag{23.10}$$

for the wavefunction ψ. The brackets indicate that $|\psi >$ is a vector in an abstract Hilbert space [47]. Vectors can be added

$$|\psi >= |\psi_1 > + |\psi_2 >= |\psi_1 + \psi_2 > \tag{23.11}$$

and can be multiplied with a complex number

$$|\psi >= \lambda |\psi_1 >= |\lambda \psi_1 > . \tag{23.12}$$

Finally a complex valued scalar product of two vectors is defined[2]

$$C =< \psi_1 | \psi_2 > \tag{23.13}$$

which has the properties

$$< \psi_1 | \psi_2 >=< \psi_2 | \psi_1 >^* \tag{23.14}$$

$$< \psi_1 | \lambda \psi_2 >= \lambda < \psi_1 | \psi_2 >=< \lambda^* \psi_1 | \psi_2 > \tag{23.15}$$

$$< \psi | \psi_1 + \psi_2 >=< \psi | \psi_1 > + < \psi | \psi_2 > \tag{23.16}$$

$$< \psi_1 + \psi_2 | \psi >=< \psi_1 | \psi > + < \psi_2 | \psi > . \tag{23.17}$$

[2] If, for instance the wavefunction depends on the coordinates of N particles, the scalar product is defined by $< \psi_n | \psi_{n'} >= \int d^3 r_1 \cdots d^3 r_N \psi_n^*(r_1 \cdots r_N) \psi_{n'}(r_1 \cdots r_N)$.

23.1.2 Density Matrix for an Ensemble of Systems

Consider a thermal ensemble of systems. Their wave functions are expanded with respect to basis functions $|\psi_s>$ as

$$|\psi>=\sum_s C_s|\psi_s> . \tag{23.18}$$

The ensemble average of an operator A is given by

$$\overline{<A>}=\overline{<\psi A\psi>}=\overline{<\sum_{s,s'} C_s^*\psi_s A C_{s'}\psi_{s'}>} \tag{23.19}$$

$$=\sum_{s,s'}\overline{C_s^* C_{s'}}A_{ss'}=\text{tr}\,(\rho A) \tag{23.20}$$

with the density matrix

$$\rho_{s's}=\sum_{s,s'}\overline{C_s^* C_{s'}}. \tag{23.21}$$

The wave function of an N-state system is a linear combination

$$|\psi>=C_1|\psi_1>+C_2|\psi_2>+\cdots C_N|\psi_N> . \tag{23.22}$$

The diagonal elements of the density matrix are the occupation probabilities

$$\rho_{11}=\overline{|C_1|^2}\qquad \rho_{22}=\overline{|C_2|^2}\cdots\qquad \rho_{NN}=\overline{|C_N|^2} \tag{23.23}$$

and the non diagonal elements measure the correlation of two states[3]

$$\rho_{12}=\rho_{21}^*=\overline{C_2^* C_1},\cdots . \tag{23.24}$$

23.1.3 Time Evolution of the Density Matrix

The expansion coefficients of

$$|\psi>=\sum_s C_s|\psi_s> \tag{23.25}$$

can be obtained from the scalar product

$$C_s=<\psi_s|\psi> . \tag{23.26}$$

[3]They are often called the "coherence" of the two states.

Hence we have

$$C_s^* C_{s'} =< \psi|\psi_s >< \psi_{s'}|\psi >=< \psi_{s'}|\psi >< \psi|\psi_s > \tag{23.27}$$

which can be considered to be the s', s matrix element of the projection operator $|\psi >< \psi|$

$$C_s^* C_{s'} = (|\psi >< \psi|)_{s's} \,. \tag{23.28}$$

The thermal average of $|\psi >< \psi|$ is the statistical operator

$$\rho = \overline{|\psi >< \psi|} \tag{23.29}$$

which is represented by the density matrix with respect to the basis functions $|\psi_s >$

$$\rho_{s's} = \overline{|\psi >< \psi|}_{s's} = \overline{C_s^* C_{s'}}. \tag{23.30}$$

From the Schroedinger equation

$$i\hbar|\dot{\psi} >= H|\psi > \tag{23.31}$$

we find

$$-i\hbar < \dot{\psi}| =< H\psi| =< \psi|H \tag{23.32}$$

and hence

$$i\hbar\dot{\rho} = i\hbar\left(\overline{|\dot{\psi} >< \psi|} + \overline{|\psi >< \dot{\psi}|}\right) = \overline{|H\psi >< \psi|} - \overline{|\psi >< H\psi|}. \tag{23.33}$$

Since the Hamiltonian H is identical for all members of the ensemble we end up with the Liouville-von Neumann equation

$$i\hbar\dot{\rho} = H\rho - \rho H = [H, \rho]. \tag{23.34}$$

With respect to a finite basis this becomes explicitly:

$$i\hbar\dot{\rho}_{ii} = \sum_j H_{ij}\rho_{ji} - \rho_{ij}H_{ji} = \sum_{j\neq i} H_{ij}\rho_{ji} - \rho_{ij}H_{ji} \tag{23.35}$$

$$\begin{aligned} i\hbar\dot{\rho}_{ik} &= \sum_j H_{ij}\rho_{jk} - \rho_{ij}H_{jk} \\ &= (H_{ii} - H_{kk})\rho_{ik} + H_{ik}(\rho_{kk} - \rho_{ii}) + \sum_{j\neq i,k}(H_{ij}\rho_{jk} - \rho_{ij}H_{jk}). \end{aligned} \tag{23.36}$$

23.2 Wave Packet Motion in One Dimension

A quantum mechanical particle with mass m_p in a one-dimensional potential $V(x)$ (Fig. 23.1) is described by a complex valued wavefunction $\psi(x)$. We assume that the wavefunction is negligible outside an interval $[a, b]$. This is the case for a particle bound in a potential well i.e. a deep enough minimum of the potential or for a wave-packet with finite width far from the boundaries. Then the calculation can be restricted to the finite interval $[a, b]$ by applying the boundary condition

$$\psi(x) = 0 \quad \text{for } x \leq a \text{ or } x \geq b \tag{23.37}$$

or, if reflections at the boundary should be suppressed, transparent boundary conditions [279].

All observables (quantities which can be measured) of the particle are expectation values with respect to the wavefunction, for instance its average position is

$$< x >=< \psi(x)\, x\, \psi(x) >= \int_a^b dx\, \psi^*(x)\, x\, \psi(x). \tag{23.38}$$

The probability of finding the particle at the position x_0 is given by

$$P(x = x_0) = |\psi(x_0)|^2 . \tag{23.39}$$

For time independent potential $V(x)$ the Schroedinger equation

$$\mathrm{i}\hbar\dot{\psi} = H\psi = \left(-\frac{\hbar^2}{2m_p}\frac{\partial^2}{\partial x^2} + V(x)\right)\psi \tag{23.40}$$

can be formally solved by

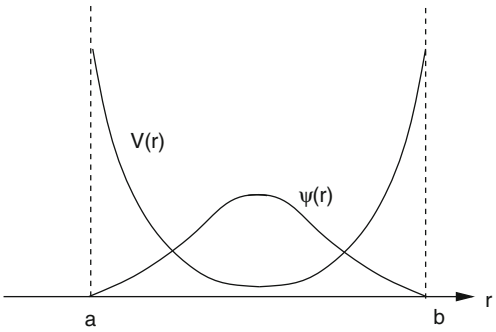


Fig. 23.1 Potential well

$$\psi(t) = U(t, t_0)\psi(t_0) = \exp\left\{-\frac{\mathrm{i}(t - t_0)}{\hbar} H\right\} \psi(t_0). \tag{23.41}$$

If the potential is time dependent, the more general formal solution is

$$\psi(t) = U(t, t_0)\psi(t_0) = \hat{T}_t \exp\left\{-\frac{\mathrm{i}}{\hbar}\int_{t_0}^{t} H(\tau)d\tau\right\} \psi(t_0)$$

$$= \sum_{n=0}^{\infty} \frac{1}{n!}\left(\frac{-\mathrm{i}}{\hbar}\right)^n \int_{t_0}^{t} dt_1 \int_{t_0}^{t} dt_2 \ldots \int_{t_0}^{t} dt_n \, \hat{T}_t \{H(t_1)H(t_2)\ldots H(t_n)\} \tag{23.42}$$

where $\hat{T}_t$ denotes the time ordering operator. The simplest approach for discretization is to divide the time interval $0 \ldots t$ into a sequence of smaller steps

$$U(t, t_0) = U(t, t_{N-1}) \ldots U(t_2, t_1)U(t_1, t_0) \tag{23.43}$$

and to neglect the variation of the Hamiltonian during the small interval $\Delta t = t_{n+1} - t_n$ [280]

$$U(t_{n+1}, t_n) = \exp\left\{-\frac{\mathrm{i}\Delta t}{\hbar} H(t_n)\right\}. \tag{23.44}$$

23.2.1 Discretization of the Kinetic Energy

The kinetic energy

$$T\psi(x) = -\frac{\hbar^2}{2m_p}\frac{\partial^2}{\partial x^2}\psi(x) \tag{23.45}$$

is a nonlocal operator in real space. It is most efficiently evaluated in Fourier space where it becomes diagonal

$$\mathcal{F}[T\psi](k) = \frac{\hbar^2 k^2}{2m_p}\mathcal{F}[\psi](k). \tag{23.46}$$

23.2.1.1 Pseudo-Spectral Methods

The potential energy is diagonal in real space. Therefore, pseudo-spectral (Sect. 12.5.1) methods [281] use a Fast Fourier Transform algorithm (Sect. 7.3.2) to switch between real space and Fourier space. They calculate the action of the Hamiltonian on the wavefunction according to

$$H\psi(x) = V(x)\psi(x) + \mathcal{F}^{-1}\left[\frac{\hbar^2 k^2}{2m_p}\mathcal{F}[\psi](k)\right]. \tag{23.47}$$

23.2.1.2 Finite Difference Methods

In real space, the kinetic energy operator can be approximated by finite differences on a grid, like the simple 3-point expression (3.31)

$$-\frac{\hbar^2}{2m_p}\frac{\psi^n_{m+1} + \psi^n_{m-1} - 2\psi^n_m}{\Delta x^2} + O(\Delta x^2) \tag{23.48}$$

or higher order expressions (3.33)

$$-\frac{\hbar^2}{2m_p}\frac{-\psi^n_{m+2} + 16\psi^n_{m+1} - 30\psi^n_m + 16\psi^n_{m-1} - \psi^n_{m-2}}{12\Delta x^2} + O(\Delta x^4) \tag{23.49}$$

$$-\frac{\hbar^2}{2m_p}\frac{1}{\Delta x^2}\left(\frac{1}{90}\psi^n_{m+3} - \frac{3}{20}\psi^n_{m+2} + \frac{3}{2}\psi^n_{m+1} - \frac{49}{18}\psi^n_m + \frac{3}{2}\psi^n_{m-1} - \frac{3}{20}\psi^n_{m-2} + \frac{1}{90}\psi^n_{m-3}\right) + O(\Delta x^6) \tag{23.50}$$

etc. [282]. However, finite differences inherently lead to deviations of the dispersion relation from (23.46). Inserting $\psi_m = \mathrm{e}^{\mathrm{i}km\Delta x}$ we find

$$E(k) = \frac{\hbar^2}{2m_p}\frac{2(1-\cos(k\Delta x))}{\Delta x^2} \tag{23.51}$$

for the 3-point expression (23.48),

$$E(k) = \frac{\hbar^2}{2m_p}\frac{15 - 16\cos(k\Delta x) + \cos(2k\Delta x)}{6\Delta x^2} \tag{23.52}$$

for the 5-point expression (23.49) and

$$\frac{\hbar^2}{2m_p}\frac{1}{\Delta x^2}\left(\frac{49}{18} - 3\cos(k\Delta x) + \frac{3}{10}\cos(2k\Delta x) - \frac{1}{45}\cos(3k\Delta x)\right) \tag{23.53}$$

for the 7-point expression (23.50). Even the 7-point expression shows large deviations for k-values approaching $k_{\max} = \pi/\Delta x$ (Fig. 23.2). However, it has been shown that not very high orders are necessary to achieve the numerical accuracy of the pseudo-spectral Fourier method [283] and that finite difference methods may be even more efficient in certain applications [284].

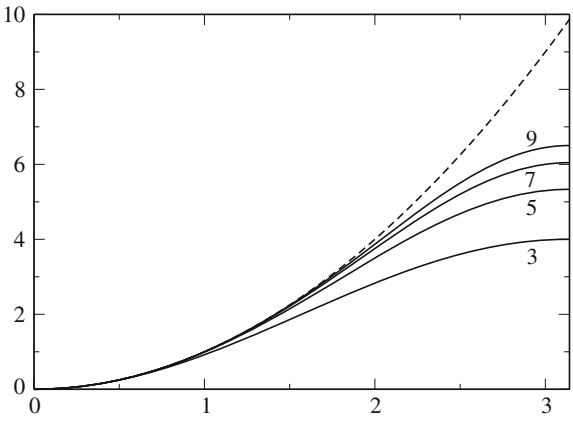


Fig. 23.2 (Dispersion of finite difference expressions) The dispersion relation of finite difference expressions of increasing order (23.48, 23.49, 23.50 and the symmetric 9-point approximation [282]) are compared to the exact dispersion (23.46) of a free particle (*dashed curve*)

23.2.2 Time Evolution

A number of methods have been proposed [280, 285–287] to approximate the short time propagator (23.44). Unitarity is a desirable property since it guaranties stability and norm conservation even for large time steps. However, depending on the application, small deviations from unitarity may be acceptable in return for higher efficiency. The Crank–Nicolson (CN) method [288–290] is one of the first methods which have been applied to the time dependent Schroedinger equation. It is a unitary but implicit method and needs the inversion of a matrix which can become cumbersome in two or more dimensions or if high precision is required. Multistep methods [291, 292], especially second order [293] differencing (SOD) are explicit but only conditionally stable and put limits to the time interval Δt. Split operator methods (SPO) approximate the propagator by a unitary product of operators [294–296]. They are explicit and easy to implement. The real-space split-operator method has been applied to more complex problems like a molecule in a laser field [297]. Polynomial approximations, especially the Chebishev expansion [298, 299], have very high accuracy and allow for large time steps, if the Hamiltonian is time independent. However, they do not provide intermediate results and need many applications of the Hamiltonian. The short time iterative Lanczos (SIL) method [118, 300, 301] is very useful also for time dependent Hamiltonians. Even more sophisticated methods using finite elements and the discrete variable representation are presented for instance in [302, 303]. In the following we discuss three methods (CN,SOD,SPO) which are easy to implement and well suited to solve the time dependent Schroedinger equation for a mass point moving in a one-dimensional potential.

23.2.2.1 Rational Approximation

Taking the first terms of the Taylor expansion

$$U(t_{n+1}, t_n) = \exp\left\{-\frac{\mathrm{i}\Delta t}{\hbar} H\right\} = 1 - \frac{\mathrm{i}\Delta t}{\hbar} H + \cdots \tag{23.54}$$

corresponds to a simple explicit Euler step

$$\psi(t_{n+1}) = \left(1 - \frac{\mathrm{i}\Delta t}{\hbar} H\right) \psi(t_n). \tag{23.55}$$

From the real eigenvalues E of the Hamiltonian we find the eigenvalues of the explicit method

$$\lambda = 1 - \frac{\mathrm{i}\Delta t}{\hbar} E \tag{23.56}$$

which all have absolute values

$$|\lambda| = \sqrt{1 + \frac{\Delta t^2 E^2}{\hbar^2}} > 1. \tag{23.57}$$

Hence the explicit method is not stable.

Expansion of the inverse time evolution operator

$$U(t_n, t_{n+1}) = U(t_{n+1}, t_n)^{-1} = \exp\left\{+\frac{\mathrm{i}\Delta t}{\hbar} H\right\} = 1 + \frac{\mathrm{i}\Delta t}{\hbar} H + \cdots$$

leads to the implicit method

$$\psi(t_{n+1}) = \psi(t_n) - \frac{\mathrm{i}\Delta t}{\hbar} H \psi(t_{n+1}) \tag{23.58}$$

which can be rearranged as

$$\psi(t_{n+1}) = \left(1 + \frac{\mathrm{i}\Delta t}{\hbar} H\right)^{-1} \psi(t_n). \tag{23.59}$$

Now all eigenvalues have absolute values < 1. This method is stable but the norm of the wave function is not conserved. Combination of implicit and explicit method gives a method [289, 290] similar to the Crank–Nicolson method for the diffusion equation (Sect. 21.2.3)

$$\psi(t_{n+1}) - \psi(t_n) = -\frac{\mathrm{i}\Delta t}{\hbar} H \left(\frac{\psi(t_{n+1})}{2} + \frac{\psi(t_n)}{2}\right). \tag{23.60}$$

This equation can be solved for the new value of the wavefunction

$$\psi(t_{n+1}) = \left(1 + \mathrm{i}\frac{\Delta t}{2\hbar}H\right)^{-1}\left(1 - \mathrm{i}\frac{\Delta t}{2\hbar}H\right)\psi(t_n) \tag{23.61}$$

which corresponds to a rational approximation[4] of the time evolution operator (Cayley's form)

$$U(t_{n+1}, t_n) \approx \frac{1 - \mathrm{i}\frac{\Delta t}{2\hbar}H}{1 + \mathrm{i}\frac{\Delta t}{2\hbar}H}. \tag{23.62}$$

The eigenvalues of (23.62) all have an absolute value of

$$|\lambda| = \left|\left(1 + \mathrm{i}\frac{E\,\Delta t}{2\hbar}\right)^{-1}\left(1 - \mathrm{i}\frac{E\,\Delta t}{2\hbar}\right)\right| = \frac{\sqrt{1 + \frac{E^2\Delta t^2}{4\hbar^2}}}{\sqrt{1 + \frac{E^2\Delta t^2}{4\hbar^2}}} = 1. \tag{23.63}$$

It is obviously a unitary operator and conserves the norm of the wavefunction since

$$\left(\frac{1 - \mathrm{i}\frac{\Delta t}{2\hbar}H}{1 + \mathrm{i}\frac{\Delta t}{2\hbar}H}\right)^{\dagger}\left(\frac{1 - \mathrm{i}\frac{\Delta t}{2\hbar}H}{1 + \mathrm{i}\frac{\Delta t}{2\hbar}H}\right) = \left(\frac{1 + \mathrm{i}\frac{\Delta t}{2\hbar}H}{1 - \mathrm{i}\frac{\Delta t}{2\hbar}H}\right)\left(\frac{1 - \mathrm{i}\frac{\Delta t}{2\hbar}H}{1 + \mathrm{i}\frac{\Delta t}{2\hbar}H}\right) = 1 \tag{23.64}$$

as H is Hermitian $H^{\dagger} = H$ and $(1+\mathrm{i}\frac{\Delta t}{2\hbar}H)$ and $(1-\mathrm{i}\frac{\Delta t}{2\hbar}H)$ are commuting operators. From the Taylor series we find the error order

$$\begin{aligned}\left(1 + \mathrm{i}\frac{\Delta t}{2\hbar}H\right)^{-1}\left(1 - \mathrm{i}\frac{\Delta t}{2\hbar}H\right) &= \left(1 - \mathrm{i}\frac{\Delta t}{2\hbar}H - \frac{\Delta t^2}{4\hbar^2}H^2 + \cdots\right)\left(1 - \mathrm{i}\frac{\Delta t}{2\hbar}H\right)\\ = 1 - \frac{\mathrm{i}\Delta t}{\hbar}H - \frac{\Delta t^2}{2\hbar^2}H^2 + \cdots &= \exp\left(-\frac{i\,\Delta t}{\hbar}H\right) + O(\Delta t^3).\end{aligned} \tag{23.65}$$

For practical application we rewrite [304]

$$\begin{aligned}&\left(1 + \mathrm{i}\frac{\Delta t}{2\hbar}H\right)^{-1}\left(1 - \mathrm{i}\frac{\Delta t}{2\hbar}H\right)\\ &= \left(1 + \mathrm{i}\frac{\Delta t}{2\hbar}H\right)^{-1}\left(-1 - \mathrm{i}\frac{\Delta t}{2\hbar}H + 2\right) = -1 + 2\left(1 + \mathrm{i}\frac{\Delta t}{2\hbar}H\right)^{-1}\end{aligned} \tag{23.66}$$

hence

$$\psi(t_{n+1}) = 2\left(1 + \mathrm{i}\frac{\Delta t}{2\hbar}H\right)^{-1}\psi(t_n) - \psi(t_n) = 2\chi - \psi(t_n). \tag{23.67}$$

[4] The Pade approximation (Sect. 2.4.1) of order [1, 1].

$\psi(t_{n+1})$ is obtained in two steps. First we have to solve

$$\left(1 + \mathrm{i}\frac{\Delta t}{2\hbar} H\right) \chi = \psi(t_n). \tag{23.68}$$

Then $\psi(t_{n+1})$ is given by

$$\psi(t_{n+1}) = 2\chi - \psi(t_n). \tag{23.69}$$

We use the finite difference method (Sect. 12.2) on the grid

$$x_m = m\Delta x \qquad m = 0 \cdots M \quad \psi_m^n = \psi(t_n, x_m) \tag{23.70}$$

and approximate the second derivative by

$$\frac{\partial^2}{\partial x^2}\psi(t_n, x_m) = \frac{\psi_{m+1}^n + \psi_{m-1}^n - 2\psi_m^n}{\Delta x^2} + O(\Delta x^2). \tag{23.71}$$

Equation (23.68) then becomes a system of linear equations

$$A \begin{bmatrix} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_M \end{bmatrix} = \begin{bmatrix} \psi_0^n \\ \psi_1^n \\ \vdots \\ \psi_M^n \end{bmatrix} \tag{23.72}$$

with a tridiagonal matrix

$$A = 1 - \mathrm{i}\frac{\hbar\Delta t}{4m_p\Delta x^2} \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{pmatrix} + \mathrm{i}\frac{\Delta t}{2\hbar} \begin{pmatrix} V_0 & & & \\ & V_1 & & \\ & & \ddots & \\ & & & V_M \end{pmatrix}. \tag{23.73}$$

The second step (23.69) becomes

$$\begin{bmatrix} \psi_0^{n+1} \\ \psi_1^{n+1} \\ \vdots \\ \psi_M^{n+1} \end{bmatrix} = 2 \begin{bmatrix} \chi_0 \\ \chi_1 \\ \vdots \\ \chi_M \end{bmatrix} - \begin{bmatrix} \psi_0^n \\ \psi_1^n \\ \vdots \\ \psi_M^n \end{bmatrix}. \tag{23.74}$$

Inserting a plane wave

$$\psi = \mathrm{e}^{\mathrm{i}(kx - \omega t)} \tag{23.75}$$

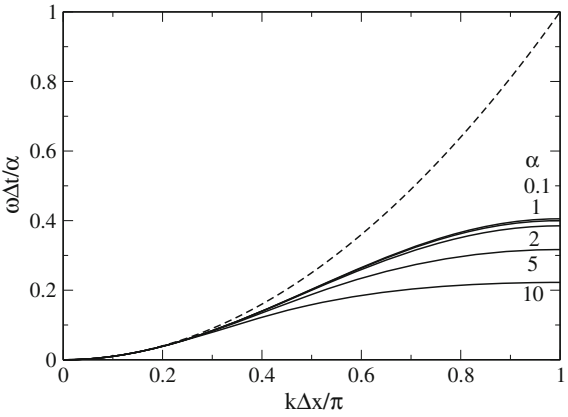


Fig. 23.3 (Dispersion of the Crank–Nicolson method) The dispersion relation of the Crank–Nicolson method (23.95) deviates largely from the exact dispersion (23.98), even for small values of the stability parameter α. The scaled frequency $\omega\Delta t/\alpha$ is shown as a function of $k\Delta x/\pi$ for $\alpha = 0.1, 1, 2, 5, 10$ (*solid curves*) and compared with the exact relation of a free particle $\omega\Delta t/\alpha = (k\Delta x/\pi)^2$ (*dashed curve*)

we obtain the dispersion relation (Fig. 23.3)

$$\frac{2}{\Delta t}\tan(\omega\Delta t/2) = \frac{\hbar}{2m_p}\left(\frac{2}{\Delta x}\sin\frac{k\Delta x}{2}\right)^2 \tag{23.76}$$

which we rewrite as

$$\omega\Delta t = 2\arctan\left[\frac{2\alpha}{\pi^2}\sin^2\frac{k\Delta x}{\pi}\frac{\pi}{2}\right] \tag{23.77}$$

with the dimensionless parameter

$$\alpha = \frac{\pi^2\hbar\Delta t}{2m_p\Delta x^2}. \tag{23.78}$$

For time independent potentials the accuracy of this method can be improved systematically [305] by using higher order finite differences for the spatial derivative (Sect. 23.2.1) and a higher order Pade approximation (Sect. 2.4.1) of order $[M, M]$ for the exponential function

$$\mathrm{e}^z = \prod_{s=1}^{M}\frac{1 - z/z_s^{(M)}}{1 + z/z_s^{(M)*}} + O(z^{2M+1}) \tag{23.79}$$

to approximate the time evolution operator

$$\exp\left(-\frac{\mathrm{i}\Delta t}{\hbar}H\right)=\prod_{s=1}^{M}\frac{1-(i\,\Delta t\,H/\hbar)/z_s^{(M)}}{1+(i\,\Delta t\,H/\hbar)/z_s^{*(M)}}+O((\Delta t)^{2M+1}). \tag{23.80}$$

However, the matrix inversion can become very time consuming in two or more dimensions.

23.2.2.2 Second Order Differencing

Explicit methods avoid the matrix inversion. The method of second order differencing [293] takes the difference of forward and backward step

$$\psi(t_{n-1})=U(t_{n-1},t_n)\psi(t_n) \tag{23.81}$$

$$\psi(t_{n+1})=U(t_{n+1},t_n)\psi(t_n) \tag{23.82}$$

to obtain the explicit two-step algorithm

$$\psi(t_{n+1})=\psi(t_{n-1})+\left[U(t_{n+1},t_n)-U^{-1}(t_n,t_{n-1})\right.\ \psi(t_n). \tag{23.83}$$

The first terms of the Taylor series give the approximation

$$\psi(t_{n+1})=\psi(t_{n-1})-2\frac{\mathrm{i}\Delta t}{\hbar}H\psi(t_n)+O((\Delta t)^3) \tag{23.84}$$

which can also be obtained from the second order approximation of the time derivative [306]

$$H\psi=\mathrm{i}\hbar\frac{\partial}{\partial t}\psi=\frac{\psi(t+\Delta t)-\psi(t-\Delta t)}{2\Delta t}. \tag{23.85}$$

This two-step algorithm can be formulated as a discrete mapping

$$\begin{pmatrix}\psi(t_{n+1})\\ \psi(t_n)\end{pmatrix}=\begin{pmatrix}-2\frac{\mathrm{i}\Delta t}{\hbar}H & 1\\ 1 & 0\end{pmatrix}\begin{pmatrix}\psi(t_n)\\ \psi(t_{n-1})\end{pmatrix} \tag{23.86}$$

with eigenvalues

$$\lambda=-\frac{\mathrm{i}E_s\Delta t}{\hbar}\pm\sqrt{1-\frac{E_s^2\Delta t^2}{\hbar^2}}. \tag{23.87}$$

For sufficiently small time step [280]

$$\Delta t<\frac{\hbar}{\max|E_s|} \tag{23.88}$$

the square root is real,

$$|\lambda|^2 = \frac{E_s^2 \Delta t^2}{\hbar^2} + \left(1 - \frac{E_s^2 \Delta t^2}{\hbar^2}\right) = 1 \tag{23.89}$$

and the method is conditionally stable and has the same error order as the Crank–Nicolson method (Sect. 23.2.2). Its big advantage is that it is an explicit method and does not involve matrix inversions. Generalization to higher order multistep differencing schemes is straightforward [291]. The method conserves [306] the quantities $\Re < \psi(t + \Delta t)|\psi(t) >$ and $\Re < \psi(t + \Delta t)|H|\psi(t) >$ but is not strictly unitary [293]. Consider a pair of wavefunctions at times t_0 and t_1 which obey the exact time evolution

$$\psi(t_1) = \exp\left\{-\frac{\mathrm{i}\Delta t}{\hbar} H\right\} \psi(t_0) \tag{23.90}$$

and apply (23.84) to obtain

$$\psi(t_2) = \left[1 - 2\frac{\mathrm{i}\Delta t}{\hbar} H \exp\left\{-\frac{\mathrm{i}\Delta t}{\hbar} H\right\}\right] \psi(t_0) \tag{23.91}$$

which can be written as

$$\psi(t_2) = \mathcal{L}\,\psi(t_0) \tag{23.92}$$

where the time evolution operator L obeys

$$\mathcal{L}^\dagger \mathcal{L} = \left[1 + 2\frac{\mathrm{i}\Delta t}{\hbar} H \exp\left\{+\frac{\mathrm{i}\Delta t}{\hbar} H\right\}\right]\left[1 - 2\frac{\mathrm{i}\Delta t}{\hbar} H \exp\left\{-\frac{\mathrm{i}\Delta t}{\hbar} H\right\}\right]$$

$$= 1 - 4\frac{\Delta t}{\hbar} H \sin\left\{\frac{\Delta t}{\hbar} H\right\} + 4\left(\frac{\Delta t}{\hbar} H\right)^2 .$$

Expanding the sine function we find the deviation from unitarity [293]

$$\mathcal{L}^\dagger \mathcal{L} - 1 = \frac{2}{3}\left(\frac{\Delta t}{\hbar} H\right)^4 + \cdots = O((\Delta t)^4) \tag{23.93}$$

which is of higher order than the error of the algorithm. Furthermore errors do not accumulate due to the stability of the algorithm (23.89). This also holds for deviations of the starting values from the condition (23.90).

The algorithm (23.84) can be combined with the finite differences method (Sect. 23.2.1)

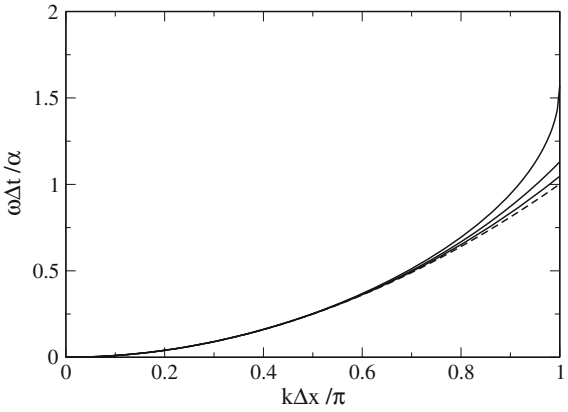


Fig. 23.4 (Dispersion of the Fourier method) The dispersion relation of the SOD-Fourier method (23.95) deviates from the exact dispersion (23.98) only for very high k-values and approaches it for small values of the stability parameter α. The scaled frequency $\omega\Delta t/\alpha$ is shown as a function of $k\Delta x/\pi$ for $\alpha = 0.5, 0.75, 1$ (*solid curves*) and compared with the exact relation of a free particle $\omega\Delta t/\alpha = (k\Delta x/\pi)^2$ (*dashed curve*)

$$\psi_m^{n+1} = \psi_m^{n-1} - 2\frac{\mathrm{i}\Delta t}{\hbar}\left[V_m\psi_m^n - \frac{\hbar^2}{2m_p\Delta x^2}\left(\psi_{m+1}^n + \psi_{m-1}^n - 2\psi_m^n\right)\right] \tag{23.94}$$

or with the pseudo-spectral Fourier method [306]. This combination needs two Fourier transformations for each step but it avoids the distortion of the dispersion relation inherent to the finite difference method. Inserting the plane wave (23.75) into (23.84) we find the dispersion relation (Fig. 23.4) for a free particle ($V = 0$):

$$\omega = \frac{1}{\Delta t}\arcsin\left(\frac{\hbar\Delta t\,k^2}{2m_p}\right) = \frac{1}{\Delta t}\arcsin\left(\alpha\left(\frac{k\Delta x}{\pi}\right)^2\right). \tag{23.95}$$

For a maximum k-value

$$k_{\max} = \frac{\pi}{\Delta x} \tag{23.96}$$

the stability condition (23.88) becomes

$$1 \geq \frac{\Delta t}{\hbar}\frac{\hbar^2 k_{\max}^2}{2m_p} = \alpha. \tag{23.97}$$

For small k the dispersion approximates the exact behavior

$$\omega = \frac{\hbar k^2}{2m_P}. \tag{23.98}$$

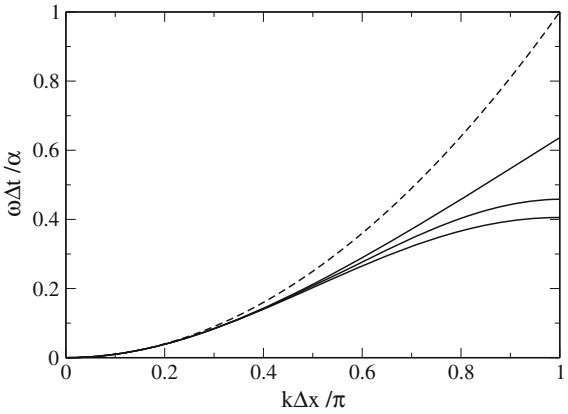


Fig. 23.5 (Dispersion of the finite difference method) The dispersion relation of the SOD-FD method (23.99) deviates largely from the exact dispersion (23.98), even for small values of the stability parameter α. The scaled frequency $\omega\Delta t/\alpha$ is shown as a function of $k\Delta x/\pi$ for $\alpha = \pi^2/4 \approx 2.467,\ 1.85,\ 1.23,\ 0.2$ (*solid curves*) and compared with the exact relation of a free particle $\omega\Delta t/\alpha = (k\Delta x/\pi)^2$ (*dashed curve*)

The finite difference method (23.94), on the other hand, has the dispersion relation (Fig. 23.5)

$$\omega = \frac{1}{\Delta t} \arcsin\left(\frac{4\alpha}{\pi^2} \sin^2\left(\frac{k\Delta x}{2}\right)\right) \tag{23.99}$$

and the stability limit

$$1 = \frac{\Delta t}{\hbar} E_{\max} = \frac{2\hbar\Delta t}{m_p \Delta x^2} = \frac{4\alpha}{\pi^2}. \tag{23.100}$$

The deviation from (23.98) is significant for $k\Delta x/\pi > 0.2$ even for small values of α [306].

23.2.2.3 Split-Operator Methods

The split-operator method approximates the exponential short time evolution operator as a product of exponential operators which are easier tractable. Starting from the Zassenhaus formula [307]

$$e^{\lambda(A+B)} = e^{\lambda A} e^{\lambda B} e^{\lambda^2 C_2} e^{\lambda^3 C_3} \ldots \tag{23.101}$$

$$C_2 = \frac{1}{2}[B, A] \quad C_3 = \frac{1}{6}[C_2, A + 2B] \quad \ldots \tag{23.102}$$

approximants of increasing order can be systematically constructed [295, 308]

$$e^{\lambda(A+B)} = e^{\lambda A} e^{\lambda B} + O(\lambda^2) = e^{\lambda A} e^{\lambda B} e^{\lambda^2 C_2} + O(\lambda^3) \quad \cdots . \tag{23.103}$$

Since these approximants do not conserve time reversibility, often the symmetric expressions

$$e^{\lambda(A+B)} = e^{\lambda A/2} e^{\lambda B} e^{\lambda A/2} + O(\lambda^3) = e^{\lambda A/2} e^{\lambda B/2} e^{\lambda^2 C_3/4} e^{\lambda B/2} e^{\lambda A/2} + O(\lambda^5) \quad \ldots \tag{23.104}$$

are preferred.

Split-Operator-Fourier Method

Dividing the Hamiltonian into its kinetic and potential parts

$$H = T + V = -\frac{\hbar^2}{2m_P}\frac{\partial^2}{\partial x^2} + V(x) \tag{23.105}$$

the time evolution operator can be approximated by the time-symmetric expression

$$U(\Delta t) = \mathrm{e}^{-\frac{\mathrm{i}\Delta t}{2\hbar}T} \mathrm{e}^{-\frac{\mathrm{i}\Delta t}{\hbar}V} \mathrm{e}^{-\frac{\mathrm{i}\Delta t}{2\hbar}T} + O((\Delta t)^3) \tag{23.106}$$

where the exponential of the kinetic energy operator can be easily applied in Fourier space [306, 309]. Combining several steps (23.106) to integrate over a longer time interval, consecutive operators can be combined to simplify the algorithm

$$U(N\Delta t) = U^N(\Delta t) = \mathrm{e}^{-\frac{\mathrm{i}\Delta t}{2\hbar}T} \left(\mathrm{e}^{-\frac{\mathrm{i}\Delta t}{\hbar}V} \mathrm{e}^{-\frac{\mathrm{i}\Delta t}{\hbar}T}\right)^{N-1} \mathrm{e}^{-\frac{\mathrm{i}\Delta t}{\hbar}V} \mathrm{e}^{-\frac{\mathrm{i}\Delta t}{2\hbar}T}. \tag{23.107}$$

Real-Space Product Formulae

Using the discretization (23.48) on a regular grid the time evolution operator becomes the exponential of a matrix

$$U(\Delta t) = \exp\left\{-\mathrm{i}\Delta t \begin{pmatrix} \frac{V_0}{\hbar} + \frac{\hbar}{m_P \Delta x^2} & -\frac{\hbar}{2m_P \Delta x^2} & & \\ -\frac{\hbar}{2m_P \Delta x^2} & \frac{V_1}{\hbar} + \frac{\hbar}{m_P \Delta x^2} & -\frac{\hbar}{2m_P \Delta x^2} & \\ & \ddots & \ddots & \ddots \\ & & -\frac{\hbar}{2m_P \Delta x^2} & \frac{V_M}{\hbar} + \frac{\hbar}{m_P \Delta x^2} \end{pmatrix}\right\}$$

$$= \exp\left\{-\mathrm{i}\Delta t \begin{pmatrix} \gamma_0 + 2\beta & -\beta & & \\ \beta & \gamma_1 + 2\beta & -\beta & \\ & & \ddots & \\ & & -\beta & \gamma_M + 2\beta \end{pmatrix}\right\} \tag{23.108}$$

with the abbreviations

$$\gamma_m = \frac{1}{\hbar} V_m \quad \beta = \frac{\hbar}{2m_P \Delta x^2}. \tag{23.109}$$

The matrix can be decomposed into the sum of two overlapping tridiagonal block matrices [294, 297][5]

$$H_o = \begin{pmatrix} \gamma_0 + 2\beta & -\beta & & & \\ -\beta & \frac{1}{2}\gamma_1 + \beta & & & \\ & & \frac{1}{2}\gamma_2 + \beta & -\beta & \\ & & -\beta & \ddots & \end{pmatrix} = \begin{pmatrix} A_1 & & & \\ & A_3 & & \\ & & \ddots & \\ & & & A_{M-1} \end{pmatrix} \tag{23.110}$$

$$H_e = \begin{pmatrix} 0 & 0 & & & \\ 0 & \frac{1}{2}\gamma_1 + \beta & -\beta & & \\ & -\beta & \frac{1}{2}\gamma_2 + \beta & 0 & \\ & & 0 & \ddots & \end{pmatrix} = \begin{pmatrix} 0 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_{M-2} \\ & & & & 0 \end{pmatrix}. \tag{23.111}$$

The block structure simplifies the calculation of $e^{-\mathrm{i}\Delta t H_o}$ and $e^{-\mathrm{i}\Delta t H_e}$ tremendously since effectively only the exponential functions of 2×2 matrices

$$B_m(\tau) = e^{-\mathrm{i}\tau A_m} \tag{23.112}$$

have to be calculated and the approximation to the time evolution operator

$$U(\Delta t) = e^{-\mathrm{i}\Delta t H_o/2} e^{-\mathrm{i}\Delta t H_e} e^{-\mathrm{i}\Delta t H_o/2}$$

$$= \begin{pmatrix} B_1(\frac{\Delta t}{2}) & & \\ & B_3(\frac{\Delta t}{2}) & \\ & & \ddots \end{pmatrix} \begin{pmatrix} 1 & & \\ & B_2(\Delta t) & \\ & & \ddots \end{pmatrix} \begin{pmatrix} B_1(\frac{\Delta t}{2}) & & \\ & B_3(\frac{\Delta t}{2}) & \\ & & \ddots \end{pmatrix} \tag{23.113}$$

can be applied in real space without any Fourier transformation. To evaluate (23.112) the real symmetric matrix A_m is diagonalized by an orthogonal transformation (Sect. 10.2)

[5] For simplicity only the case of even M is considered.

$$A = R^{-1}\tilde{A}R = R^{-1}\begin{pmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{pmatrix}R \tag{23.114}$$

and the exponential calculated from

$$\begin{aligned} e^{-\mathrm{i}\tau A} &= 1 - \mathrm{i}\tau R^{-1}\tilde{A}R + \frac{(-\mathrm{i}\tau)^2}{2}R^{-1}\tilde{A}R\,R^{-1}\tilde{A}R + \cdots \\ &= R^{-1}\left[1 - \mathrm{i}\tau\tilde{A} + \frac{(-\mathrm{i}\tau)^2}{2}\tilde{A}R + \cdots\right]R \\ &= R^{-1}e^{-\mathrm{i}\tau\tilde{A}}R = R^{-1}\begin{pmatrix}e^{-\mathrm{i}\tau\lambda_1} & \\ & e^{-\mathrm{i}\tau\lambda_2}\end{pmatrix}R. \end{aligned} \tag{23.115}$$

23.2.3 Example: Free Wave Packet Motion

We simulate the free motion ($V = 0$) of a Gaussian wave packet along the x-axis (see Problem 23.1). To simplify the numerical calculation we set $\hbar = 1$ and $m_p = 1$ and solve the time dependent Schroedinger equation

$$\mathrm{i}\frac{\partial}{\partial t}\psi = -\frac{1}{2}\frac{\partial^2}{\partial x^2}\psi \tag{23.116}$$

for initial values given by a Gaussian wave packet with constant momentum

$$\psi_0(x) = \left(\frac{2}{a\pi}\right)^{1/4}\mathrm{e}^{\mathrm{i}k_0 x}\mathrm{e}^{-x^2/a}. \tag{23.117}$$

The exact solution can be easily found. Fourier transformation of (23.117) gives

$$\hat{\psi}_k(t=0) = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} dx\,\mathrm{e}^{-\mathrm{i}kx}\psi_0(x) = \left(\frac{a}{2\pi}\right)^{1/4}\exp\left\{-\frac{a}{4}(k-k_0)^2\right\}. \tag{23.118}$$

Time evolution in k-space is rather simple

$$\mathrm{i}\frac{\partial}{\partial t}\hat{\psi}_k = \frac{k^2}{2}\hat{\psi}_k \tag{23.119}$$

hence

$$\hat{\psi}_k(t) = \mathrm{e}^{-\mathrm{i}k^2 t/2}\hat{\psi}_k(t=0) \tag{23.120}$$

and Fourier back transformation gives the solution of the time dependent Schroedinger equation in real space

$$\psi(t,x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dk\, \mathrm{e}^{\mathrm{i}kx} \hat{\psi}_k(t)$$
$$= \left(\frac{2a}{\pi}\right)^{1/4} \frac{1}{\sqrt{a+2\mathrm{i}t}} \exp\left\{-\frac{(x-\mathrm{i}\frac{ak_0}{2})^2 + \frac{ak_0^2}{4}(a+2\mathrm{i}t)}{a+2\mathrm{i}t}\right\}. \tag{23.121}$$

Finally, the probability density is given by a Gaussian

$$|\psi(t,x)|^2 = \sqrt{\frac{2a}{\pi}} \frac{1}{\sqrt{a^2+4t^2}} \exp\left\{-\frac{2a}{a^2+4t^2}(x-k_0 t)^2\right\} \tag{23.122}$$

which moves with constant velocity k_0 and kinetic energy

$$\int_{-\infty}^{\infty} dx\, \psi^*(x,t) \left(-\frac{\hbar^2}{2}\frac{\partial^2}{\partial x^2}\right) \psi(x,t) = \frac{1}{2}\left(k_0^2 + \frac{1}{a}\right). \tag{23.123}$$

Numerical examples are shown in Figs. 23.6, 23.7 and Table 23.1.

23.3 Few-State Systems

In the following we discuss simple models which reduce the wavefunction to the superposition of a few important states, for instance an initial and a final state which are coupled by a resonant interaction. We approximate the solution of the time dependent Schroedinger equation as a linear combination

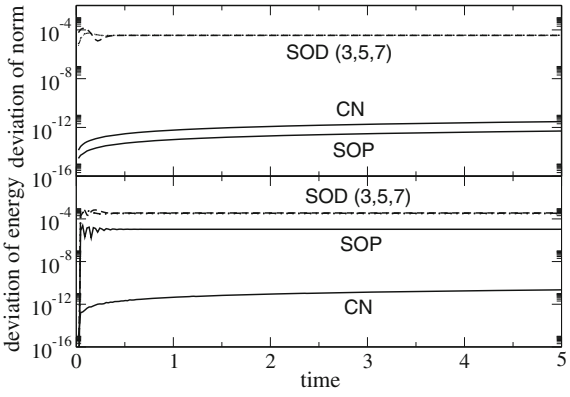


Fig. 23.6 (Conservation of norm and energy) The free motion of a Gaussian wave packet is simulated with the Crank–Nicolson method (CN), the second order differences method (SOD) with 3 point (23.48) 5 point (23.49) and 7-point (23.50) differences and with the real-space split-operator method (SPO). $\Delta t = 10^{-3}$, $\Delta x = 0.1$, $a = 1$, $k_0 = 3.77$

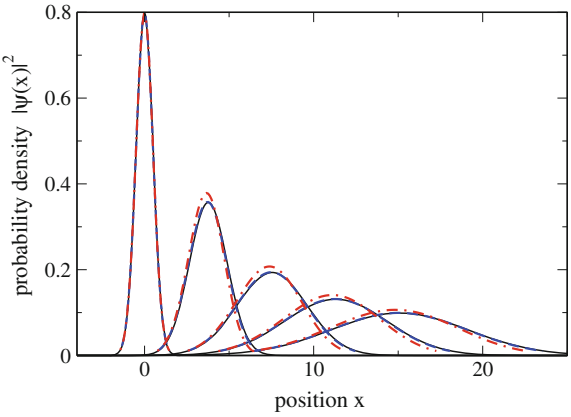


Fig. 23.7 (Free wave-packet motion) The free motion of a Gaussian wave packet is simulated. The probability density is shown for the initial Gaussian wave packet and at later times $t = 1, 2, 3, 4$. Results from the second order differences method with 3 point differences (23.48, *red dash-dotted*) and 5 point differences (23.49, blue dashed) are compared with the exact solution (23.122, thin *black solid line*). $\Delta t = 10^{-3}$, $\Delta x = 0.1$, $a = 1$, $k_0 = 3.77$

Table 23.1 (Accuracy of finite differences methods) The relative error of the kinetic energy (23.123) is shown as calculated with different finite difference methods

Method	E_{kin}	$\frac{E_{kin}-E_{kin}^{exact}}{E_{kin}^{exact}}$
Crank–Nicolson (CN) with 3 point differences	7.48608	-1.6×10^{-2}
Second order differences with 3 point differences (SOD3)	7.48646	-1.6×10^{-2}
Second order differences with 5 point differences (SOD5)	7.60296	-4.6×10^{-4}
Second order differences with 7 point differences (SOD7)	7.60638	-0.9×10^{-5}
Split-operator method (SOP) with 3 point differences	7.48610	-1.6×10^{-2}
Exact	7.60645	

$$|\psi(t)> \approx \sum_{j=1}^{M} C_j(t)|\phi_j> \tag{23.124}$$

of certain basis states $|\phi_1> \cdots |\phi_M>$[6] which are assumed to satisfy the necessary boundary conditions and to be orthonormalized

$$<\phi_i|\phi_j> = \delta_{ij}. \tag{23.125}$$

[6]This basis is usually incomplete.

Applying the method of weighted residuals (Sect. 12.4) we minimize the residual

$$|R>=\mathrm{i}\hbar\sum_j \dot{C}_j(t)|\phi_j> -\sum_j C_j(t)H|\phi_j> \tag{23.126}$$

by choosing the basis functions as weight functions (Sect. 12.4.4) and solving the system of ordinary differential equations

$$0=R_j=<\phi_j|R>=\mathrm{i}\hbar\dot{C}_j-\sum_{j'}<\phi_j|H|\phi_{j'}>C_{j'} \tag{23.127}$$

which can be written

$$\mathrm{i}\hbar\dot{C}_i=\sum_{j=1}^{M}H_{i,j}C_j(t) \tag{23.128}$$

with the matrix elements of the Hamiltonian

$$H_{i,j}=<\phi_i|\,H\,|\phi_j>\,. \tag{23.129}$$

In matrix form (23.128) reads

$$\mathrm{i}\hbar\begin{pmatrix}\dot{C}_1(t)\\ \vdots\\ \dot{C}_M(t)\end{pmatrix}=\begin{pmatrix}H_{1,1}&\cdots&H_{1,M}\\ \vdots&\ddots&\vdots\\ H_{M,1}&\cdots&H_{M,M}\end{pmatrix}\begin{pmatrix}C_1(t)\\ \vdots\\ C_M(t)\end{pmatrix} \tag{23.130}$$

or more symbolically

$$\mathrm{i}\hbar\dot{\mathbf{C}}(t)=\mathbf{H}\mathbf{C}(t). \tag{23.131}$$

If the Hamilton operator does not depend explicitly on time ($H=$ const.) the formal solution of (23.131) is given by

$$\mathbf{C}=\exp\left\{\frac{t}{i\hbar}H\right\}\mathbf{C}(0). \tag{23.132}$$

From the solution of the eigenvalue problem

$$H\mathbf{C}_\lambda=\lambda\mathbf{C}_\lambda \tag{23.133}$$

(eigenvalues λ and corresponding eigenvectors $\mathbf{C}_\lambda$) we build the linear combination

$$\mathbf{C}=\sum_\lambda a_\lambda\mathbf{C}_\lambda e^{\frac{\lambda}{i\hbar}t}. \tag{23.134}$$

The amplitudes a_λ can be calculated from the set of linear equations

$$\mathbf{C}(0) = \sum_\lambda a_\lambda \mathbf{C}_\lambda. \tag{23.135}$$

In the following we use the 4th order Runge–Kutta method to solve (23.131) numerically whereas the explicit solution (23.132) will be used to obtain approximate analytical results for special limiting cases.

A time dependent Hamiltonian $H(t)$ appears in semiclassical models which treat some of the slow degrees of freedom as classical quantities, for instance an electron in the Coulomb field of (slowly) moving nuclei

$$H(t) = T_{el} + \sum_j \frac{-q_j e}{4\pi\varepsilon_0 |\mathbf{r} - \mathbf{R}_j(t)|} + \sum_{j<j'} \frac{q_j q_{j'}}{4\pi\varepsilon_0 |\mathbf{R_j}(t) - \mathbf{R}_{j'}(t)|} \tag{23.136}$$

or in a time dependent electromagnetic field

$$H(t) = T_{el} + V_{el} - e\mathbf{r}\,\mathbf{E}(t). \tag{23.137}$$

23.3.1 Two-State System

The two-state system (Fig. 23.8) (also known as two-level system or TLS) is the simplest model of interacting states and is very often used in physics, for instance in the context of quantum optics, quantum information, spintronics and quantum dots.

Its interaction matrix is

$$H = \begin{pmatrix} E_1 & V \\ V & E_2 \end{pmatrix} \tag{23.138}$$

and the equations of motion are

$$\begin{aligned} i\hbar\dot{C}_1 &= E_1 C_1 + V C_2 \\ i\hbar\dot{C}_2 &= E_2 C_2 + V C_1 \end{aligned}. \tag{23.139}$$

The interaction matrix can be diagonalized by an orthogonal transformation (Sect. 10.2)

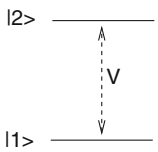


Fig. 23.8 Two-state system model

$$\tilde{H} = RHR^T = \begin{pmatrix} \lambda_1 & \\ & \lambda_2 \end{pmatrix} \tag{23.140}$$

with the rotation matrix

$$R = \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix}. \tag{23.141}$$

The tangent of φ can be determined from (10.2)

$$\tau = \tan\varphi = -\text{sign}\left(\frac{E_2 - E_1}{2V}\right)\left(\left|\frac{E_2 - E_1}{2V}\right| - \sqrt{1 + \left(\frac{E_2 - E_1}{2V}\right)^2}\right) \tag{23.142}$$

from which we find

$$\cos\varphi = \frac{1}{\sqrt{1+\tau^2}} \quad \sin\varphi = \frac{\tau}{\sqrt{1+\tau^2}} \tag{23.143}$$

and the eigenvalues

$$\lambda_1 = E_1 - \tau V \quad \lambda_2 = E_2 + \tau V. \tag{23.144}$$

Finally the solution of (23.139) is given by (23.134)

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = A \begin{pmatrix} 1 \\ \tau \end{pmatrix} e^{\frac{1}{i\hbar}(E_1 - \tau V)t} + B \begin{pmatrix} -\tau \\ 1 \end{pmatrix} e^{\frac{1}{i\hbar}(E_2 + \tau V)t}. \tag{23.145}$$

For initial conditions

$$C_1(0) = 1 \quad C_2(0) = 0 \tag{23.146}$$

solution of (23.135) provides the coefficients

$$A = \frac{1}{1+\tau^2} \quad B = -\frac{\tau}{1+\tau^2} \tag{23.147}$$

and hence the explicit solution

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{1+\tau^2} e^{\frac{1}{i\hbar}(E_1 - \tau V)t} + \frac{\tau^2}{1+\tau^2} e^{\frac{1}{i\hbar}(E_2 + \tau V)t} \\ \frac{\tau}{1+\tau^2}\left(e^{\frac{1}{i\hbar}(E_1 - \tau V)t} - e^{\frac{1}{i\hbar}(E_2 + \tau V)t}\right) \end{pmatrix}. \tag{23.148}$$

The occupation probability of the initial state is

$$C_1^2 = \frac{1+\tau^4}{(1+\tau^2)^2} + \frac{2\tau^2}{(1+\tau^2)^2}\cos\left((E_2 - E_1 + 2\tau V)\frac{t}{\hbar}\right). \tag{23.149}$$

It oscillates with the frequency

$$\hbar\Omega = \sqrt{4V^2 + (E_2 - E_1)^2} \tag{23.150}$$

and reaches a minimum value

$$C_{1min}^2 = \left(\frac{1-\tau^2}{1+\tau^2}\right)^2 = \frac{\Delta E^2\left(|\Delta E| - \sqrt{\Delta E^2 + 4V^2}\right)^2}{\left(4V^2 + \Delta E^2 - |\Delta E|\sqrt{\Delta E^2 + 4V^2}\right)^2} = \frac{\Delta E^2}{\Delta E^2 + 4V^2}. \tag{23.151}$$

Of special interest is the fully resonant limit. $E_1 = E_2$. Addition and subtraction of equations (23.139) here gives

$$i\hbar\frac{\mathrm{d}}{\mathrm{d}t}(C_1 \pm C_2) = (E_1 \pm V)(C_1 \pm C_2) \tag{23.152}$$

with the solution

$$C_1 \pm C_2 = (C_1(0) \pm C_2(0))e^{-it(E_1 \pm V)/\hbar}. \tag{23.153}$$

For initial conditions given by (23.146) the explicit solution is

$$C_1 = e^{-itE_1/\hbar}\cos\frac{Vt}{\hbar} \quad |C_1|^2 = \cos^2\frac{Vt}{\hbar} = \frac{1+\cos\frac{2Vt}{\hbar}}{2} \tag{23.154}$$

$$C_2 = -ie^{-itE_1/\hbar}\sin\frac{Vt}{\hbar} \quad |C_2|^2 = \sin^2\frac{Vt}{\hbar} = \frac{1-\cos\frac{2Vt}{\hbar}}{2}. \tag{23.155}$$

At resonance the two-state system oscillates between the two states with the period

$$T = \frac{\pi\hbar}{V}. \tag{23.156}$$

Numerical results are shown in Fig. 23.9.

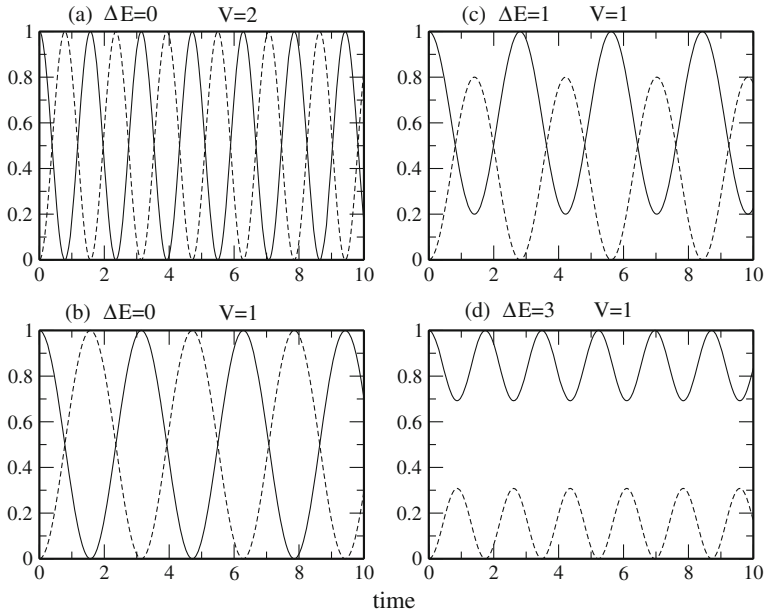
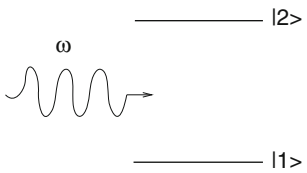


Fig. 23.9 (Numerical simulation of a two-state system) The equations of motion of the two-state system (23.139) are integrated with the 4th order Runge–Kutta method. For two resonant states the occupation probability of the initial state shows oscillations with the period (23.156) proportional to V^{-1}. With increasing energy gap $E_2 - E_1$ the amplitude of the oscillations decreases

Fig. 23.10 Two-state system in an oscillating field



23.3.2 Two-State System with Time Dependent Perturbation

Consider now a 2-state system with an oscillating perturbation (Fig. 23.10) (for instance an atom or molecule in a laser field)

$$H = \begin{pmatrix} E_1 & V(t) \\ V(t) & E_2 \end{pmatrix} \qquad V(t) = V_0 \cos \omega t. \tag{23.157}$$

The equations of motion are

$$\begin{aligned} \mathrm{i}\hbar \dot{C}_1 &= E_1 C_1 + V(t) C_2 \\ \mathrm{i}\hbar \dot{C}_2 &= V(t) C_1 + E_2 C_2 \end{aligned}. \tag{23.158}$$

After the substitutions

$$\begin{aligned} C_1 &= \mathrm{e}^{\frac{E_1}{\mathrm{i}\hbar}t} u_1 \\ C_2 &= \mathrm{e}^{\frac{E_2}{\mathrm{i}\hbar}t} u_2 \end{aligned} \tag{23.159}$$

$$\omega_{21} = \frac{E_2 - E_1}{\hbar} \tag{23.160}$$

they become

$$\begin{aligned} \mathrm{i}\hbar\dot{u}_1 &= V(t)\mathrm{e}^{\frac{E_2-E_1}{\mathrm{i}\hbar}t} u_2 = \frac{V_0}{2}\left(\mathrm{e}^{-\mathrm{i}(\omega_{21}-\omega)t} + \mathrm{e}^{-\mathrm{i}(\omega_{21}+\omega)t}\right) u_2 \\ \mathrm{i}\hbar\dot{u}_2 &= V(t)e^{\frac{E_1-E_2}{\mathrm{i}\hbar}t} u_1 = \frac{V_0}{2}\left(\mathrm{e}^{\mathrm{i}(\omega_{21}-\omega)t} + \mathrm{e}^{\mathrm{i}(\omega_{21}+\omega)t}\right) u_1 \end{aligned}. \tag{23.161}$$

At larger times the system oscillates between the two states.[7] Applying the rotating wave approximation for $\omega \approx \omega_{21}$ we neglect the fast oscillating perturbation

$$\mathrm{i}\hbar\dot{u}_1 = \frac{V_0}{2}\mathrm{e}^{-\mathrm{i}(\omega_{21}-\omega)t} u_2 \tag{23.162}$$

$$\mathrm{i}\hbar\dot{u}_2 = \frac{V_0}{2}\mathrm{e}^{\mathrm{i}(\omega_{21}-\omega)t} u_1 \tag{23.163}$$

and substitute

$$u_1 = a_1\mathrm{e}^{-\mathrm{i}(\omega_{21}-\omega)t} \tag{23.164}$$

to have

$$\mathrm{i}\hbar(\dot{a_1} - a_1\mathrm{i}(\omega_{21} - \omega))\mathrm{e}^{-\mathrm{i}(\omega_{21}-\omega)t} = \frac{V_0}{2}\mathrm{e}^{-\mathrm{i}(\omega_{21}-\omega)t} u_2 \tag{23.165}$$

$$\mathrm{i}\hbar\dot{u}_2 = \frac{V_0}{2}\mathrm{e}^{\mathrm{i}(\omega_{21}-\omega)t}\mathrm{e}^{-\mathrm{i}(\omega_{21}-\omega)t} a_1 \tag{23.166}$$

or

$$\mathrm{i}\hbar\dot{a}_1 = \hbar(\omega - \omega_{21})a_1 + \frac{V_0}{2}u_2 \tag{23.167}$$

$$\mathrm{i}\hbar\dot{u}_2 = \frac{V_0}{2}a_1 \tag{23.168}$$

[7] So called Rabi oscillations.

which shows that the system behaves approximately like a two-state system with a constant interaction $V_0/2$ and an energy gap $\hbar(\omega_{21} - \omega) = E_2 - E_1 - \hbar\omega$ (a comparison with a full numerical calculation is shown in Fig. 23.11).

23.3.3 Superexchange Model

The concept of superexchange was originally formulated for magnetic interactions [310] and later introduced to electron transfer theory [311]. It describes an indirect interaction through high energy intermediates (Fig. 23.12). In the simplest case, we have to consider two isoenergetic states i and f which do not interact directly but via coupling to an intermediate state v.
The interaction matrix is

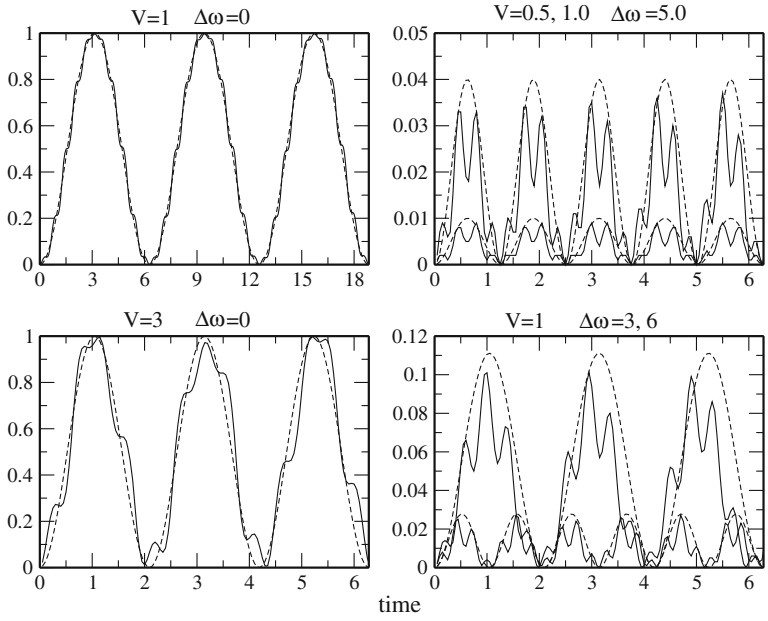


Fig. 23.11 (Simulation of a two-state system in an oscillating field) The equations of motion (23.158) are integrated with the 4th order Runge–Kutta method. At resonance the system oscillates between the two states with the frequency $V/\hbar$. The *dashed curves* show the corresponding solution of a two-state system with constant coupling (Sect. 23.3.1)

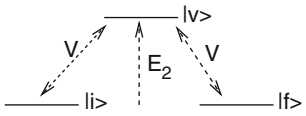


Fig. 23.12 Superexchange model

$$H = \begin{pmatrix} 0 & V_1 & 0 \\ V_1 & E_2 & V_2 \\ 0 & V_2 & 0 \end{pmatrix}. \tag{23.169}$$

For simplification we choose $V_1 = V_2$.

Let us first consider the special case of a resonant intermediate state $E_2 = 0$:

$$H = \begin{pmatrix} 0 & V & 0 \\ V & 0 & V \\ 0 & V & 0 \end{pmatrix}. \tag{23.170}$$

Obviously one eigenvalue is $\lambda_1 = 0$ and the corresponding eigenvector is

$$\mathbf{C}_1 = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}. \tag{23.171}$$

The two remaining eigenvalues are solutions of

$$0 = \det \begin{vmatrix} -\lambda & V & 0 \\ V & -\lambda & V \\ 0 & V & -\lambda \end{vmatrix} = \lambda(-\lambda^2 + 2V^2) \tag{23.172}$$

which gives

$$\lambda_{2,3} = \pm\sqrt{2}V. \tag{23.173}$$

The eigenvectors are

$$\mathbf{C}_{2,3} = \begin{pmatrix} 1 \\ \pm\sqrt{2} \\ 1 \end{pmatrix}. \tag{23.174}$$

From the initial values

$$\mathbf{C}(0) = \begin{pmatrix} a_1 + a_2 + a_3 \\ \sqrt{2}a_2 - \sqrt{2}a_3 \\ -a_1 + a_2 + a_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{23.175}$$

the amplitudes are calculated as

$$a_1 = \frac{1}{2}\; a_2 = a_3 = \frac{1}{4} \tag{23.176}$$

and finally the solution is

$$\mathbf{C} = \frac{1}{2}\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} + \frac{1}{4}\begin{pmatrix} 1 \\ \sqrt{2} \\ 1 \end{pmatrix} e^{\frac{1}{i\hbar}\sqrt{2}Vt} + \frac{1}{4}\begin{pmatrix} 1 \\ -\sqrt{2} \\ 1 \end{pmatrix} e^{-\frac{1}{i\hbar}\sqrt{2}Vt}$$

$$= \begin{pmatrix} \frac{1}{2} + \frac{1}{2}\cos\frac{\sqrt{2}V}{\hbar}t \\ \frac{\sqrt{2}}{2} i \sin\frac{\sqrt{2}V}{\hbar}t \\ -\frac{1}{2} + \frac{1}{2}\cos\frac{\sqrt{2}V}{\hbar}t \end{pmatrix}. \tag{23.177}$$

Let us now consider the case of a distant intermediate state $V \ll |E_2|$. $\lambda_1 = 0$ and the corresponding eigenvector still provide one solution. The two other eigenvalues are approximately given by

$$\lambda_{2,3} = \pm\sqrt{\frac{E_2^2}{4} + 2V^2} + \frac{E_2}{2} \approx \frac{E_2}{2} \pm \frac{E_2}{2}(1 + \frac{4V^2}{E_2^2}) \tag{23.178}$$

$$\lambda_2 \approx E_2 + \frac{2V^2}{E_2} \quad \lambda_3 \approx -\frac{2V^2}{E_2} \tag{23.179}$$

and the eigenvectors by

$$\mathbf{C}_2 \approx \begin{pmatrix} 1 \\ \frac{E_2}{V} + \frac{2V}{E_2} \\ 1 \end{pmatrix} \qquad \mathbf{C}_3 \approx \begin{pmatrix} 1 \\ -\frac{2V}{E_2} \\ 1 \end{pmatrix}. \tag{23.180}$$

From the initial values

$$\mathbf{C}(0) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} a_1 + a_2 + a_3 \\ a_2\lambda_2 + a_3\lambda_3 \\ -a_1 + a_2 + a_3 \end{pmatrix} \tag{23.181}$$

we calculate the amplitudes

$$a_1 = \frac{1}{2} \qquad a_2 \approx \frac{V^2}{E_2^2} \qquad a_3 \approx \frac{1}{2}\left(1 - \frac{2V^2}{E_2^2}\right) \tag{23.182}$$

and finally the solution

$$\mathbf{C} \approx \begin{pmatrix} \frac{1}{2}(1 + e^{-\frac{1}{i\hbar}\frac{2V^2}{E_2}t}) \\ \frac{V}{E_2} e^{\frac{1}{i\hbar}E_2 t} - \frac{2V}{E_2} e^{-\frac{1}{i\hbar}\frac{2V^2}{E_2}t} \\ \frac{1}{2}(-1 + e^{-\frac{1}{i\hbar}\frac{2V^2}{E_2}t}) \end{pmatrix}. \tag{23.183}$$

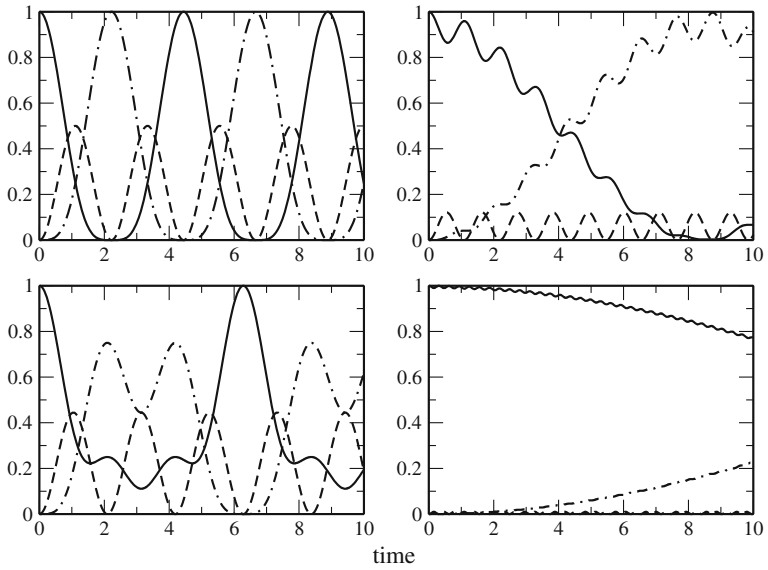


Fig. 23.13 (Numerical simulation of the superexchange model) The equations of motion for the model (23.169) are solved numerically with the 4th order Runge–Kutta method. The energy gap is varied to study the transition from the simple oscillation with $\omega = \sqrt{2}V/\hbar$ (23.177) to the effective two-state system with $\omega = V_{eff}/\hbar$ (23.184). Parameters are $V_1 = V_2 = 1$, $E_1 = E_3 = 0$, $E_2 = 0, 1, 5, 20$. The occupation probability of the initial (*solid curves*), virtual intermediate (*dashed curves*) and final (*dash-dotted curves*) state are shown

The occupation probability of the initial state is

$$|C_1|^2 = \frac{1}{4}|1 + e^{-\frac{1}{i\hbar}\frac{2V^2}{E_2}t}|^2 = \cos^2\left(\frac{V^2}{\hbar E_2}t\right) \tag{23.184}$$

which shows that the system behaves like a 2-state system with an effective interaction of

$$V_{eff} = \frac{V^2}{E_2}. \tag{23.185}$$

Numerical results are shown in Fig. 23.13.

23.3.4 Ladder Model for Exponential Decay

For time independent Hamiltonian the solution (23.132) of the Schroedinger equation is a sum of oscillating terms and the quantum recurrence theorem [312] states that the

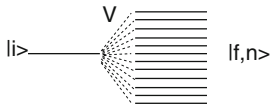


Fig. 23.14 Ladder model

system returns to the initial state arbitrarily closely after a certain time T_r. However, if the initial state is coupled to a larger number of final states, the recurrence time can become very long and an exponential decay observed over a large period. The ladder model [313, 314] considers an initial state $|0>$ interacting with a manifold of states $|1> \ldots |n>$, which do not interact with each other and are equally spaced (Fig. 23.14)

$$H = \begin{pmatrix} 0 & V & \cdots & V \\ V & E_1 & & \\ \vdots & & \ddots & \\ V & & & E_n \end{pmatrix} \quad E_j = E_1 + (j-1)\Delta E. \tag{23.186}$$

The equations of motion are

$$\mathrm{i}\hbar\dot{C}_0 = V\sum_{j=1}^{n} C_j$$

$$\mathrm{i}\hbar\dot{C}_j = E_j C_j + V C_0. \tag{23.187}$$

For the special case $\Delta E = 0$ we simply have

$$\ddot{C}_0 = -\frac{V^2}{\hbar^2} n C_0 \tag{23.188}$$

with an oscillating solution

$$C_0 \sim \cos\left(\frac{V\sqrt{n}}{\hbar}t\right). \tag{23.189}$$

Here the n states act like one state with an effective coupling of $V\sqrt{n}$.
For the general case $\Delta E \neq 0$ we substitute

$$C_j = u_j \mathrm{e}^{\frac{E_j}{\mathrm{i}\hbar}t} \tag{23.190}$$

and have

$$\mathrm{i}\hbar\dot{u}_j \mathrm{e}^{\frac{E_j}{\mathrm{i}\hbar}t} = VC_0. \tag{23.191}$$

Integration gives

$$u_j = \frac{V}{\mathrm{i}\hbar}\int_{t_0}^{t} \mathrm{e}^{-\frac{E_j}{\mathrm{i}\hbar}t'} C_0(t')dt' \tag{23.192}$$

and therefore

$$C_j = \frac{V}{\mathrm{i}\hbar}\int_{t_0}^{t} e^{\mathrm{i}\frac{E_j}{\hbar}(t'-t)} C_0(t')dt'. \tag{23.193}$$

With the definition

$$E_j = j * \hbar\Delta\omega \tag{23.194}$$

we have

$$\dot{C}_0 = \frac{V}{\mathrm{i}\hbar}\sum_{j=1}^{n} C_j = -\frac{V^2}{\hbar^2}\sum_j \int_{t_0}^{t} \mathrm{e}^{\mathrm{i}j\Delta\omega(t'-t)} C_0(t')dt'. \tag{23.195}$$

We replace the sum by an integral over the continuous variable

$$\omega = j\Delta\omega \tag{23.196}$$

and extend the integration range to $-\infty \cdots \infty$. Then the sum becomes approximately a delta function

$$\sum_{j=-\infty}^{\infty} \mathrm{e}^{\mathrm{i}j\Delta\omega(t'-t)}\Delta j \;\rightarrow\; \int_{-\infty}^{\infty} \mathrm{e}^{\mathrm{i}\omega(t'-t)}\frac{d\omega}{\Delta\omega} = \frac{2\pi}{\Delta\omega}\delta(t-t') \tag{23.197}$$

and the final result is an exponential decay law

$$\dot{C}_0 = -\frac{2\pi V^2}{\hbar^2\Delta\omega}C_0 = -\frac{2\pi V^2}{\hbar}\rho(E)C_0 \tag{23.198}$$

with the density of final states

$$\rho(E) = \frac{1}{\hbar\Delta\omega} = \frac{1}{\Delta E}. \tag{23.199}$$

Numerical results are shown in Fig. 23.15.

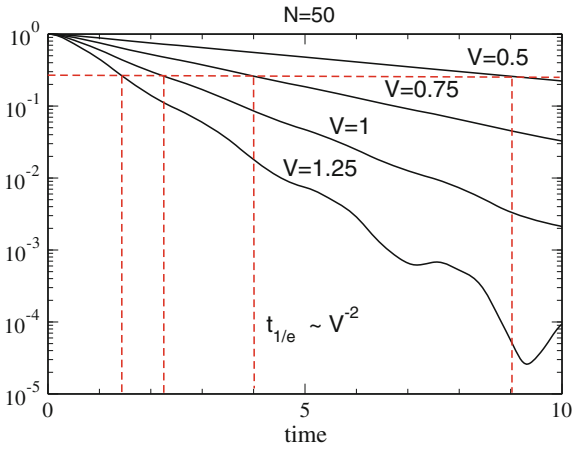


Fig. 23.15 (Numerical solution of the ladder model) The time evolution of the ladder model (23.187) is calculated with the 4th order Runge–Kutta method for N = 50 states and different values of the coupling V

23.3.5 Semiclassical Curve Crossing

In the following we study simple models for the transition between two electronic states along a nuclear coordinate Q.[8] Within the crude diabatic model the wavefunction takes the form

$$\Psi = \begin{pmatrix} \chi_1(Q,t) \\ \chi_2(Q,t) \end{pmatrix} \tag{23.200}$$

where the two components refer to the two electronic states.

The nuclear wavefunctions $\chi_{1,2}$ obey a system of coupled equations (M is the reduced mass corresponding to the nuclear coordinate)

$$i\hbar\dot{\Psi} = H\Psi = \left[-\frac{\hbar^2}{2M}\frac{\partial^2}{\partial Q^2} + \begin{pmatrix} E_1(Q) & V(Q) \\ V(Q) & E_2(Q) \end{pmatrix} \right] \Psi. \tag{23.201}$$

Here $E_{1,2}(Q)$ are the diabatic potential energy surfaces which cross at a point Q_c and $V(Q)$ is the coupling matrix element in the diabatic basis.

According to Ehrenfest's theorem, the average position

$$\overline{Q}(t) = \int \left[|\chi_1(Q,t)|^2 + |\chi_2(Q,t)|^2 \right] Q\, dQ \tag{23.202}$$

[8] For a diatomic molecule, e.g. the nuclear coordinate is simply the distance R of the two nuclei.

obeys an equation of motion which looks very similar to its classical counterpart

$$M\frac{d^2}{dt^2}\overline{Q}(t) = \overline{F} = -\frac{\partial}{\partial Q}V_{eff}$$

$$= \int dQ \begin{pmatrix} \chi_1(Q,t)^* & \chi_2(Q,t)^* \end{pmatrix} \begin{pmatrix} -\frac{\partial}{\partial Q}E_1(Q) & -\frac{\partial}{\partial Q}V(Q) \\ -\frac{\partial}{\partial Q}V(Q) & -\frac{\partial}{\partial Q}E_2(Q) \end{pmatrix} \begin{pmatrix} \chi_1(Q,t) \\ \chi_2(Q,t) \end{pmatrix}. \tag{23.203}$$

The semiclassical approach approximates both nuclear wavefunctions as one and the same narrow wave packet centered at the classical position $Q(t) = \overline{Q}(t)$. Equation (23.203) then becomes

$$M\frac{\partial^2}{\partial t^2}Q(t) = -\frac{\partial}{\partial Q} \begin{pmatrix} a(t)^* & b(t)^* \end{pmatrix} \begin{pmatrix} E_1(Q(t)) & V(Q(t)) \\ V(Q(t)) & E_2(Q(t)) \end{pmatrix} \begin{pmatrix} a(t) \\ b(t) \end{pmatrix}.$$

Substitution of

$$\chi_1(Q,t) = \chi_2(Q,t) = \phi(Q,t) \tag{23.204}$$

in (23.200)

$$\Psi = \begin{pmatrix} a(t) \\ b(t) \end{pmatrix} \phi(Q,t) \tag{23.205}$$

and taking the average over Q, which in fact means to replace Q by $Q(t)$, the semiclassical approximation is obtained:

$$i\hbar \begin{pmatrix} \dot{a}(t) \\ \dot{b}(t) \end{pmatrix} = \begin{pmatrix} E_1(Q(t)) & V(Q(t)) \\ V(Q(t)) & E_2(Q(t)) \end{pmatrix} \begin{pmatrix} a(t) \\ b(t) \end{pmatrix}. \tag{23.206}$$

In Problem 23.5 we compare the solutions of (23.201) and (23.206). The two wave packets are propagated with the split-operator-Fourier transform method Sect. 23.2.2. For a small time step Δt the propagator is approximated as a product

$$\exp\left\{\frac{\Delta t}{\mathrm{i}\hbar}H\right\}$$

$$= \exp\left\{i\Delta t\frac{\hbar}{4M}\frac{\partial^2}{\partial Q^2}\right\} \exp\left\{\frac{\Delta t}{\mathrm{i}\hbar} \begin{pmatrix} E_1(Q) & V(Q) \\ V(Q) & E_2(Q) \end{pmatrix}\right\} \exp\left\{i\Delta t\frac{\hbar}{4M}\frac{\partial^2}{\partial Q^2}\right\} + \cdots \tag{23.207}$$

where the kinetic energy part is evaluated in Fourier space and the potential energy part requires diagonalization of a 2×2 matrix for each grid point. From the resulting wavefunction the average position $\overline{Q}(t)$ is calculated which is needed to define the trajectory for the semiclassical approximation. Equation 23.206 is then solved with the Runge–Kutta method. The initial wavefunction is a Gaussian wave packet on one of the diabatic surfaces with constant momentum (as in 23.117). Figure 23.16 shows an example from Problem 23.5.

23.3.6 *Landau–Zener Model*

This model describes crossing of two states, for instance for colliding atoms or molecules [315, 316]. It is assumed that in the vicinity of the crossing point the interaction V is constant and the time dependency of the energy gap is linearized (Fig. 23.17)

$$V(Q(t)) = V \tag{23.208}$$

$$\Delta E(t) = E_2(Q(t)) - E_1(Q(t)) = \Delta E_0 + vt. \tag{23.209}$$

The Hamiltonian matrix of the Landau–Zener model is

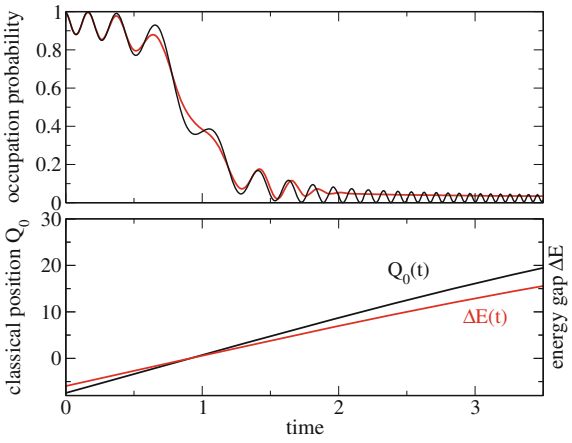


Fig. 23.16 (Semiclassical approximation of a curve Crossing) The crossing between two states is simulated for coupling $V = 1.23$, velocity $= 12$ and slope $= 0.4$. (Problem 23.5). ***Top*** the semiclassical approximation (*black*) reproduces the occupation probability from the full quantum calculation (*red*) quite accurately. Generally, it shows more pronounced oscillations than the quantum calculation with wave packets of finite width. ***Bottom*** If the initial velocity is large enough, acceleration is not important and the classical position (*black*) as well as the energy gap (*red*) become linear functions of time

$$H = \begin{pmatrix} 0 & V \\ V & \Delta E(t) \end{pmatrix}. \tag{23.210}$$

For small interaction V or large velocity $\frac{\partial}{\partial t}\Delta E = \dot{Q}\frac{\partial}{\partial Q}\Delta E$ the transition probability can be calculated with perturbation theory to give

$$P = \frac{2\pi V^2}{\hbar \frac{\partial}{\partial t}\Delta E}. \tag{23.211}$$

This expression becomes invalid for small velocities. Here the system stays on the adiabatic potential surface, i.e. $P \to 1$. Landau and Zener found the following expression which is valid in both limits:

$$P_{LZ} = 1 - \exp\left(-\frac{2\pi V^2}{\hbar \frac{\partial}{\partial t}\Delta E}\right). \tag{23.212}$$

In case of collisions multiple crossing of the interaction region has to be taken into account (Fig. 23.18).

Numerical results from Problem 23.6 are shown in Fig. 23.19.

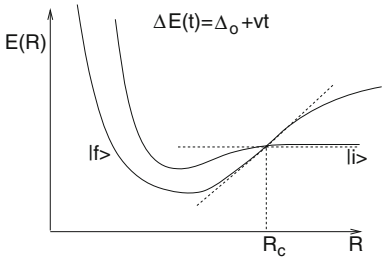


Fig. 23.17 Slow atomic collision

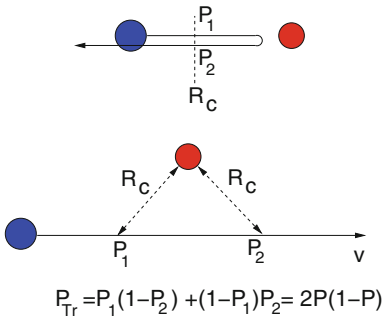


Fig. 23.18 Multiple passage of the interaction region

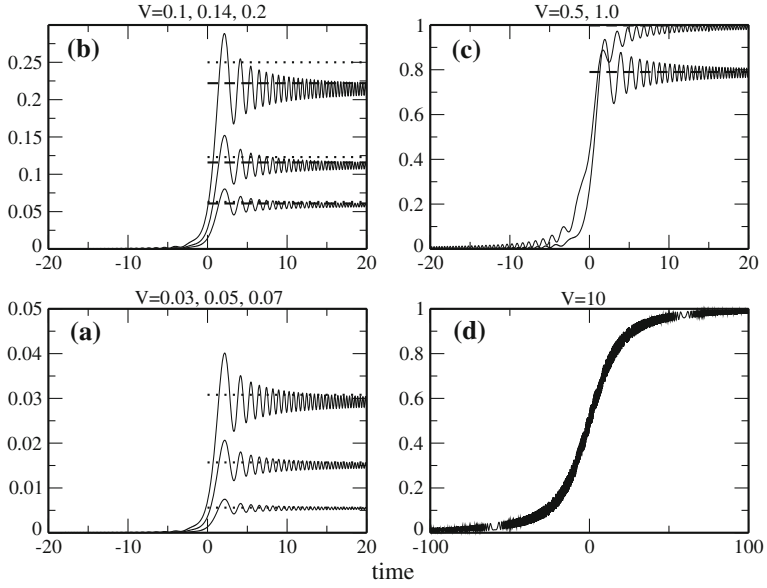


Fig. 23.19 (Numerical solution of the Landau–Zener model) Numerical calculations (*solid curves*) are compared with the Landau–Zener probability (23.212, *dashed lines*) and the approximation (23.211, *dotted lines*) The velocity is $\mathrm{d}\Delta E/\mathrm{d}t = 1$. (Problem 23.6)

23.4 The Dissipative Two-State System

A two-state quantum system coupled to a thermal bath serves as a model for magnetic resonance phenomena, coherent optical excitations [317, 318] and, quite recently, for a Qubit, the basic element of a future quantum computer [319, 320]. Its quantum state can not be described by a single wavefunction. Instead mixed quantum states have to be considered which can be conveniently described within the density matrix formalism [277].

23.4.1 Equations of Motion for a Two-State System

The equations of motion for a two-state system are

$$i\hbar\dot{\rho}_{11} = H_{12}\rho_{21} - \rho_{12}H_{21} \tag{23.213}$$

$$i\hbar\dot{\rho}_{22} = H_{21}\rho_{12} - \rho_{21}H_{12} \tag{23.214}$$

$$i\hbar\dot{\rho}_{12} = (H_{11} - H_{22})\rho_{12} + H_{12}(\rho_{22} - \rho_{11}) \tag{23.215}$$

$$-i\hbar\dot{\rho}_{21} = (H_{11} - H_{22})\rho_{21} + H_{21}(\rho_{22} - \rho_{11}) \tag{23.216}$$

which can be arranged as a system of linear equations[9]

$$i\hbar \begin{pmatrix} \dot{\rho}_{11} \\ \dot{\rho}_{22} \\ \dot{\rho}_{12} \\ \dot{\rho}_{21} \end{pmatrix} = \begin{pmatrix} 0 & 0 & -H_{21} & H_{12} \\ 0 & 0 & H_{21} & -H_{12} \\ -H_{12} & H_{12} & H_{11} - H_{22} & 0 \\ H_{21} & -H_{21} & 0 & H_{22} - H_{11} \end{pmatrix} \begin{pmatrix} \rho_{11} \\ \rho_{22} \\ \rho_{12} \\ \rho_{21} \end{pmatrix}. \tag{23.217}$$

23.4.2 The Vector Model

The density matrix is Hermitian

$$\rho_{ij} = \rho_{ji}^* \tag{23.218}$$

its diagonal elements are real valued and due to conservation of probability

$$\rho_{11} + \rho_{22} = \text{const.} \tag{23.219}$$

Therefore the four elements of the density matrix can be specified by three real parameters, which are usually chosen as

$$x = 2\Re\rho_{21} \tag{23.220}$$

$$y = 2\Im\rho_{21} \tag{23.221}$$

$$z = \rho_{11} - \rho_{22} \tag{23.222}$$

and satisfy the equations

$$\frac{\mathrm{d}}{\mathrm{d}t} 2\Re(\rho_{21}) = -\frac{1}{\hbar}\left((H_{11} - H_{22})2\Im(\rho_{21}) + 2\Im(H_{12})(\rho_{11} - \rho_{22})\right) \tag{23.223}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} 2\Im(\rho_{21}) = \frac{1}{\hbar}\left((H_{11} - H_{22})2\Re(\rho_{21}) - 2\Re(H_{12})(\rho_{11} - \rho_{22})\right) \tag{23.224}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}(\rho_{11} - \rho_{22}) = \frac{2}{\hbar}\left(\Im(H_{12})2\Re(\rho_{21}) + \Re H_{12}2\Im(\rho_{21})\right). \tag{23.225}$$

[9] The matrix of this system corresponds to the Liouville operator.

Together they form the Bloch vector

$$\mathbf{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{23.226}$$

which is often used to visualize the time evolution of a two-state system [321]. In terms of the Bloch vector the density matrix is given by

$$\begin{pmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{pmatrix} = \begin{pmatrix} \frac{1+z}{2} & \frac{x-iy}{2} \\ \frac{x+iy}{2} & \frac{1-z}{2} \end{pmatrix} = \frac{1}{2}(1 + \mathbf{r}\boldsymbol{\sigma}) \tag{23.227}$$

with the Pauli matrices

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{pmatrix}, \quad \sigma_z = \begin{pmatrix} 1 & \\ & -1 \end{pmatrix}. \tag{23.228}$$

From (23.223–23.225) we obtain the equation of motion

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -y \frac{H_{11}-H_{22}}{\hbar} - z \frac{2\Im(H_{12})}{\hbar} \\ x \frac{H_{11}-H_{22}}{\hbar} - z \frac{2\Re(H_{12})}{\hbar} \\ x \frac{2\Im(H_{12})}{\hbar} + y \frac{2\Re(H_{12})}{\hbar} \end{pmatrix} \tag{23.229}$$

which can be written as a cross product

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{r} = \boldsymbol{\omega} \times \mathbf{r} \tag{23.230}$$

with

$$\boldsymbol{\omega} = \begin{pmatrix} \frac{2}{\hbar}\Re H_{12} \\ -\frac{2}{\hbar}\Im H_{12} \\ \frac{1}{\hbar}(H_{11} - H_{22}) \end{pmatrix}. \tag{23.231}$$

Any normalized pure quantum state of the two-state system can be written as [322]

$$|\psi> = \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \cos\frac{\theta}{2}\begin{pmatrix} 1 \\ 0 \end{pmatrix} + \mathrm{e}^{\mathrm{i}\phi}\sin\frac{\theta}{2}\begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{23.232}$$

corresponding to the density matrix

$$\rho = \begin{pmatrix} \cos^2\frac{\theta}{2} & \mathrm{e}^{-\mathrm{i}\phi}\sin\frac{\theta}{2}\cos\frac{\theta}{2} \\ \mathrm{e}^{\mathrm{i}\phi}\sin\frac{\theta}{2}\cos\frac{\theta}{2} & \sin^2\frac{\theta}{2} \end{pmatrix}. \tag{23.233}$$

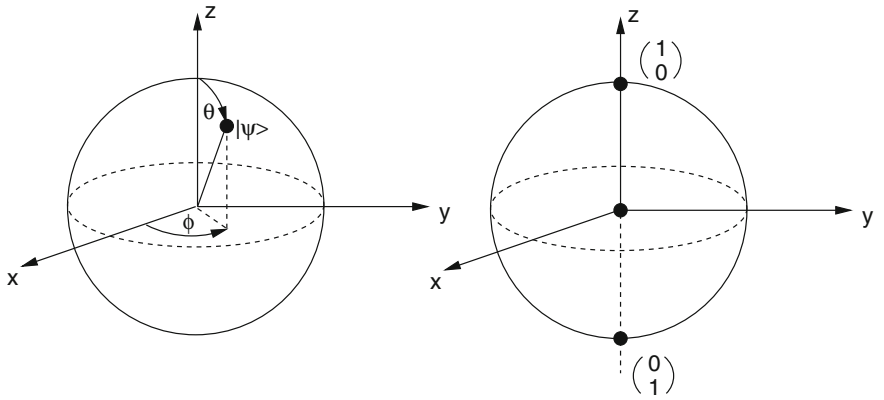


Fig. 23.20 (Bloch sphere) ***Left*** Any pure quantum state of a two-state system can be represented by a point on the Bloch sphere. ***Right*** The poles represent the basis states. Mixed quantum states correspond to the interior of the sphere, the central point represents the fully mixed state

The Bloch vector

$$\mathbf{r} = \begin{pmatrix} \cos\phi \sin\theta \\ \sin\phi \sin\theta \\ \cos\theta \end{pmatrix} \tag{23.234}$$

represents a point on the unit sphere (the Bloch sphere, Fig. 23.20). Mixed states correspond to the interior of the Bloch sphere with the fully mixed state $\rho = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}$ represented by the center of the sphere (Fig. 23.20).

23.4.3 The Spin-1/2 System

An important example of a two-state system is a particle with spin $\frac{1}{2}$. Its quantum state can be described by a two-component vector

$$\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = C_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + C_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{23.235}$$

where the two unit vectors are eigenvectors of the spin component in z-direction corresponding to the eigenvalues $s_z = \pm\frac{\hbar}{2}$. The components of the spin operator are given by the Pauli matrices

$$S_i = \frac{\hbar}{2}\sigma_i \tag{23.236}$$

and have expectation values

$$< \mathbf{S} >= \frac{\hbar}{2} \left(C_1^* \; C_2^* \right) \begin{pmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \hbar \begin{pmatrix} \frac{C_1^* C_2 + C_2^* C_1}{2} \\ \frac{C_1^* C_2 - C_2^* C_1}{2i} \\ \frac{|C_1|^2 - |C_2|^2}{2} \end{pmatrix}. \tag{23.237}$$

The ensemble average for a system of spin-$\frac{1}{2}$ particles is given by the Bloch vector

$$\overline{< \mathbf{S} >} = \hbar \begin{pmatrix} \frac{\rho_{21} + \rho_{12}}{2} \\ \frac{\rho_{21} - \rho_{12}}{2i} \\ \frac{\rho_{11} - \rho_{22}}{2} \end{pmatrix} = \frac{\hbar}{2} \mathbf{r}. \tag{23.238}$$

The Hamiltonian of a spin-$\frac{1}{2}$ particle in a magnetic field $\mathbf{B}$ is

$$H = -\gamma \frac{\hbar}{2} \boldsymbol{\sigma} \mathbf{B} = -\gamma \frac{\hbar}{2} \begin{pmatrix} B_z & B_x - \mathrm{i} B_y \\ B_x + \mathrm{i} B_y & -B_z \end{pmatrix} \tag{23.239}$$

from which the following relations are obtained

$$\gamma B_x = -\frac{2}{\hbar} \Re H_{12} \tag{23.240}$$

$$\gamma B_y = \frac{2}{\hbar} \Im H_{12} \tag{23.241}$$

$$\gamma B_z = -\frac{H_{11} - H_{22}}{\hbar} \tag{23.242}$$

$$\boldsymbol{\omega} = -\gamma \mathbf{B}. \tag{23.243}$$

The average magnetization

$$\mathbf{m} = \gamma \overline{< \mathbf{S} >} = \gamma \frac{\hbar}{2} \mathbf{r} \tag{23.244}$$

obeys the equation of motion

$$\frac{\mathrm{d}}{\mathrm{d}t} \mathbf{m} = -\gamma \mathbf{B} \times \mathbf{m}. \tag{23.245}$$

23.4.4 Relaxation Processes - The Bloch Equations

Relaxation of the nuclear magnetization due to interaction with the environment was first described phenomenologically by Bloch in 1946 [323]. A more rigorous

description was given later [324, 325] and also applied to optical transitions [326]. Recently electron spin relaxation has attracted much interest in the new field of spintronics [327] and the dissipative two-state system has been used to describe the decoherence of a Qubit [328].

23.4.4.1 Phenomenological Description

In thermal equilibrium the density matrix is given by a canonical distribution

$$\rho^{eq} = \frac{\mathrm{e}^{-\beta H}}{\mathrm{tr}(\mathrm{e}^{-\beta H})} \tag{23.246}$$

which for a two-state system without perturbation

$$H_0 = \begin{pmatrix} \frac{\Delta}{2} & \\ & -\frac{\Delta}{2} \end{pmatrix} \tag{23.247}$$

becomes

$$\rho^{eq} = \begin{pmatrix} \frac{\mathrm{e}^{-\beta\Delta/2}}{\mathrm{e}^{\beta\Delta/2}+\mathrm{e}^{-\beta\Delta/2}} & \\ & \frac{\mathrm{e}^{\beta\Delta/2}}{\mathrm{e}^{\beta\Delta/2}+\mathrm{e}^{-\beta\Delta/2}} \end{pmatrix} \tag{23.248}$$

where, as usually $\beta = 1/k_B T$. If the energy gap is very large $\Delta \gg k_B T$ like for an optical excitation, the equilibrium state is the state with lower energy[10]

$$\rho^{eq} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \tag{23.249}$$

The phenomenological model assumes that deviations of the occupation difference from its equilibrium value

$$\rho_{11}^{eq} - \rho_{22}^{eq} = -\tanh\left(\frac{\Delta}{2k_B T}\right) \tag{23.250}$$

decay exponentially with a time constant T_1 (for NMR this is the spin-lattice relaxation time)

$$\frac{d}{dt}_{|Rel}(\rho_{11} - \rho_{22}) = -\frac{1}{T_1}\left[(\rho_{11} - \rho_{22}) - (\rho_{11}^{eq} - \rho_{22}^{eq}\right) \quad . \tag{23.251}$$

[10] We assume $\Delta \geq 0$, such that the equilibrium value of $z = \rho_{11} - \rho_{22}$ is negative. Eventually, the two states have to be exchanged.

The coherence of the two states decays exponentially with a time constant T_2 which is closely related to T_1 in certain cases[11] but can be much smaller than T_1 if there are additional dephasing mechanisms. The equation

$$\frac{d}{dt}_{|Rel}\rho_{12} = -\frac{1}{T_2}\rho_{12} \tag{23.252}$$

describes the decay of the transversal polarization due to spatial and temporal differences of different spins (spin-spin relaxation), whereas for an optical excitation or a Qubit it describes the loss of coherence of a single two-state system due to interaction with its environment.

The combination of (23.245) and the relaxation terms (23.251, 23.252) gives the Bloch equations [323] which were originally formulated to describe the time evolution of the macroscopic polarization

$$\frac{\mathrm{d}\mathbf{m}}{\mathrm{d}t} = -\gamma\mathbf{B}\times\mathbf{m} - R(\mathbf{m}-\mathbf{m}_{eq}) \qquad R = \begin{pmatrix} \frac{1}{T_2} & 0 & 0 \\ 0 & \frac{1}{T_2} & 0 \\ 0 & 0 & \frac{1}{T_1} \end{pmatrix}. \tag{23.253}$$

For the components of the Bloch vector they read explicitly

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -1/T_2 & -\frac{1}{\hbar}(H_{11}-H_{22}) & -\frac{2}{\hbar}\Im H_{12} \\ \frac{1}{\hbar}(H_{11}-H_{22}) & -1/T_2 & -\frac{2}{\hbar}\Re H_{12} \\ \frac{2}{\hbar}\Im H_{12} & \frac{2}{\hbar}\Re H_{12} & -1/T_1 \end{pmatrix}\begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ z_{eq}/T_1 \end{pmatrix}. \tag{23.254}$$

23.4.5 The Driven Two-State System

The Hamiltonian of a two-state system (for instance an atom or molecule) in an oscillating electric field $E\mathrm{e}^{-\mathrm{i}\omega_f t}$ with energy splitting Δ and transition dipole moment μ is

$$H = \begin{pmatrix} \frac{\Delta}{2} & -\mu E\mathrm{e}^{-\mathrm{i}\omega_f t} \\ -\mu E\mathrm{e}^{\mathrm{i}\omega_f t} & -\frac{\Delta}{2} \end{pmatrix}. \tag{23.255}$$

The corresponding magnetic field

$$B_x = \frac{2}{\gamma\hbar}\mu E\cos\omega_f t \tag{23.256}$$

[11] For instance $T_2 = 2T_1$ for pure radiative damping.

$$B_y = \frac{2}{\gamma\hbar}\mu E \sin\omega_f t \tag{23.257}$$

$$B_z = -\frac{\Delta}{\gamma\hbar} \tag{23.258}$$

is that of a typical NMR experiment with a constant component along the z-axis and a rotating component in the $xy-$plane.

23.4.5.1 Free Precession

Consider the special case $B_z = \text{const}$, $B_x = B_y = 0$. The corresponding Hamiltonian matrix is diagonal

$$H = \begin{pmatrix} \frac{\hbar\Omega_0}{2} & 0 \\ 0 & -\frac{\hbar\Omega_0}{2} \end{pmatrix} \tag{23.259}$$

with the Larmor-frequency

$$\Omega_0 = \frac{\Delta}{\hbar} = -\gamma B_0. \tag{23.260}$$

The equations of motion for the density matrix are

$$\frac{\partial}{\partial t}(\rho_{11} - \rho_{22}) = -\frac{(\rho_{11} - \rho_{22}) - (\rho_{11}^{eq} - \rho_{22}^{eq})}{T_1} \tag{23.261}$$

$$\mathrm{i}\hbar\frac{\partial}{\partial t}\rho_{12} = \hbar\Omega_0\rho_{12} - \mathrm{i}\hbar\frac{1}{T_2}\rho_{12} \tag{23.262}$$

with the solution

$$(\rho_{11} - \rho_{22}) = (\rho_{11}^{eq} - \rho_{22}^{eq}) + [(\rho_{11}(0) - \rho_{22}(0)) - (\rho_{11}^{eq} - \rho_{22}^{eq})]\mathrm{e}^{-t/T_1} \tag{23.263}$$

$$\rho_{12} = \rho_{12}(0)\mathrm{e}^{-\mathrm{i}\Omega_0 t - t/T_2}. \tag{23.264}$$

The Bloch vector

$$\mathbf{r} = \begin{pmatrix} (x_0\cos\Omega_0 t - y_0\sin\Omega_0 t)\mathrm{e}^{-t/T_2} \\ (y_0\cos\Omega_0 t + x_0\sin\Omega_0 t)\mathrm{e}^{-t/T_2} \\ z^{eq} + (z_0 - z^{eq})\mathrm{e}^{-t/T_1} \end{pmatrix} \tag{23.265}$$

is subject to damped precession around the $z-$axis with the Larmor frequency (Fig. 23.21).

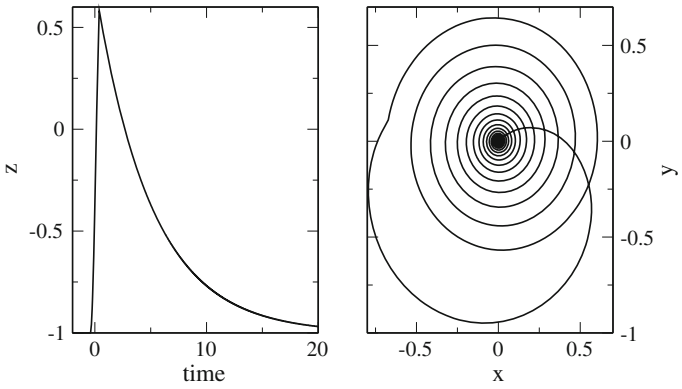


Fig. 23.21 (Free precession) The Bloch equations (23.254) are numerically solved with the 4th order Runge Kutta method. After excitation with a short resonant pulse the free precession is observed. ***Left*** The occupation difference $z = \rho_{11} - \rho_{22}$ decays exponentially to its equilibrium value. ***Right*** In the xy-plane the Bloch vector moves on a spiral towards the equilibrium position $(x = 0, y = 0)$

23.4.5.2 Stationary Solution for Monochromatic Excitation

For the two-state system (23.255) with

$$H_{11} - H_{22} = \Delta = \hbar\Omega_0 \tag{23.266}$$

$$H_{12} = V_0(\cos\omega_f t - \mathrm{i}\sin\omega_f t) \tag{23.267}$$

the solution of the Bloch equations (23.253)

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} -1/T_2 & -\Omega_0 & \frac{2V_0}{\hbar}\sin\omega_f t \\ \Omega_0 & -1/T_2 & -\frac{2V_0}{\hbar}\cos\omega_f t \\ -\frac{2V_0}{\hbar}\sin\omega_f t & \frac{2V_0}{\hbar}\cos\omega_f t & -1/T_1 \end{pmatrix}\begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ z_{eq}/T_1 \end{pmatrix} \tag{23.268}$$

can be found explicitly [317]. We transform to a coordinate system which rotates around the z-axis (Sect. 14.3 on page 330) with angular velocity ω_f

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos(\omega_f t) & \sin(\omega_f t) & 0 \\ -\sin(\omega_f t) & \cos(\omega_f t) & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = A(t)\begin{pmatrix} x \\ y \\ z \end{pmatrix}. \tag{23.269}$$

Then

$$\frac{d}{dt}\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \dot{A}\begin{pmatrix} x \\ y \\ z \end{pmatrix} + A\frac{d}{dt}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = (\dot{A}A^{-1} + AKA^{-1})\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} + A\begin{pmatrix} 0 \\ 0 \\ \frac{z_{eq}}{T_1} \end{pmatrix} \tag{23.270}$$

with

$$K = \begin{pmatrix} -1/T_2 & -\Omega_0 & \frac{2V_0}{\hbar}\sin\omega_f t \\ \Omega_0 & -1/T_2 & -\frac{2V_0}{\hbar}\cos\omega_f t \\ -\frac{2V_0}{\hbar}\sin\omega_f t & \frac{2V_0}{\hbar}\cos\omega_f t & -1/T_1 \end{pmatrix}. \tag{23.271}$$

The matrix products are

$$\dot{A}A^{-1} = W = \begin{pmatrix} 0 & \omega_f & 0 \\ -\omega_f & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad AKA^{-1} = \begin{pmatrix} -1/T_2 & -\Omega_0 & 0 \\ \Omega_0 & -1/T_2 & -\frac{2V_0}{\hbar} \\ 0 & \frac{2V_0}{\hbar} & -1/T_1 \end{pmatrix} \tag{23.272}$$

and the equation of motion simplifies to

$$\begin{pmatrix} \dot{x}' \\ \dot{y}' \\ \dot{z}' \end{pmatrix} = \begin{pmatrix} -\frac{1}{T_2} & \omega_f - \Omega_0 & 0 \\ \Omega_0 - \omega_f & -\frac{1}{T_2} & -\frac{2V_0}{\hbar} \\ 0 & \frac{2V_0}{\hbar} & -\frac{1}{T_1} \end{pmatrix}\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{z^{eq}}{T_1} \end{pmatrix}. \tag{23.273}$$

For times short compared to the relaxation times the solution is approximately given by harmonic oscillations. The generalized Rabi frequency Ω_R follows from [329]

$$\mathrm{i}\Omega_R x' = (\omega_f - \Omega_0)y' \tag{23.274}$$

$$\mathrm{i}\Omega_R y' = (\Omega_0 - \omega_f)x' - \frac{2V_0}{\hbar}z' \tag{23.275}$$

$$\mathrm{i}\Omega_R z' = \frac{2V_0}{\hbar}y' \tag{23.276}$$

as

$$\Omega_R = \sqrt{(\Omega_0 - \omega_f)^2 + \left(\frac{2V_0}{\hbar}\right)^2}. \tag{23.277}$$

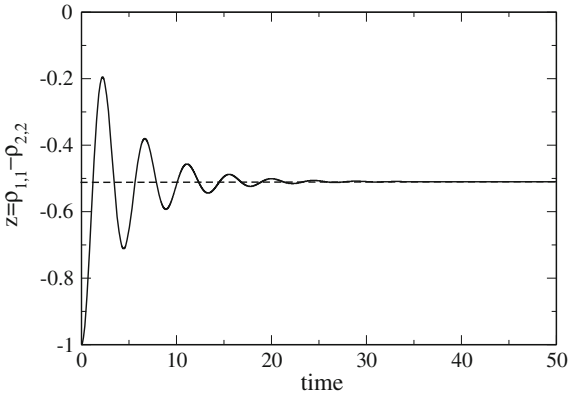


Fig. 23.22 (Monochromatic Excitation) The Bloch equations are solved numerically with the 4th order Runge–Kutta method for a monochromatic perturbation with $\omega = 4$, $V_0 = 0.5$. Parameters of the two-state system are $\omega_0 = 5$, $z_{eq} = -1.0$ and $T_1 = T_2 = 5.0$. The occupation difference $z = \rho_{11} - \rho_{22}$ initially shows Rabi oscillations which disappear at larger times where the stationary value $z = -0.51$ is reached

At larger times these oscillations are damped and the stationary solution is approached (Fig. 23.22) which is given by

$$\frac{z^{eq}}{1 + 4\frac{V_0^2}{\hbar^2} T_1 T_2 + T_2^2(\omega_f - \Omega_0)^2} \begin{pmatrix} 2T_2^2 \frac{V_0}{\hbar}(\Omega_0 - \omega_f) \\ -2T_2 \frac{V_0}{\hbar} \\ 1 + T_2^2(\omega_f - \Omega_0)^2 \end{pmatrix}. \tag{23.278}$$

The occupation difference

$$z = \rho_{11} - \rho_{22} = z^{eq} \left(1 - \frac{4\frac{V_0^2}{\hbar^2} T_1 T_2}{1 + 4\frac{V_0^2}{\hbar^2} T_1 T_2 + T_2^2(\omega_f - \Omega_0)^2} \right) \tag{23.279}$$

has the form of a Lorentzian. The line width increases for higher intensities (power broadening)

$$\Delta\omega = \frac{1}{T_2} \sqrt{1 + 4\frac{V_0^2}{\hbar^2} T_1 T_2} \tag{23.280}$$

and the maximum

$$\frac{z(\Omega_0)}{z^{eq}} = \frac{1}{1 + 4\frac{V_0^2}{\hbar^2} T_1 T_2} \tag{23.281}$$

approaches zero (saturation) (Figs. 23.23, 23.24).

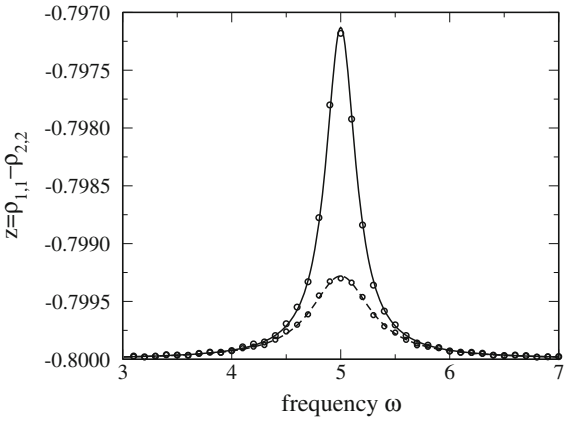
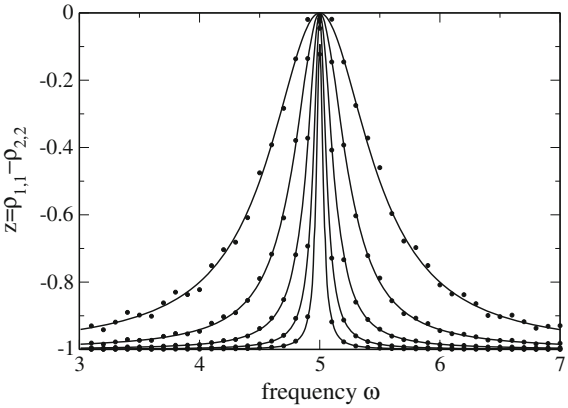


Fig. 23.23 (Resonance line) The equations of motion of the two-state system including relaxation terms are integrated with the 4th order Runge–Kutta method until a steady state is reached. Parameters are $\omega_0 = 5$, $z_{eq} = -0.8$, $V = 0.01$ and $T_1 = T_2 = 3.0, 6.9$. The change of the occupation difference is shown as a function of frequency (*circles*) and compared with the steady state solution (23.278)

Fig. 23.24 (Power saturation and broadening) The resonance line is investigated as a function of the coupling strength V and compared with the stationary solution (23.278) to observe the broadening of the line width (23.280). Parameters are $\omega_0 = 5$, $z_{eq} = -1.0$, $T_1 = T_2 = 100$ and $V = 0.5, 0.25, 0.125, 0.0625, 0.03125$



23.4.5.3 Excitation by a Resonant Pulse

For a resonant pulse with real valued envelope $V_0(t)$ and initial phase angle Φ_0

$$H_{12} = V_0(t)\, \mathrm{e}^{-\mathrm{i}(\Omega_0 t + \Phi_0)}$$

the equation of motion in the rotating system is

$$\begin{pmatrix} \dot{x}' \\ \dot{y}' \\ \dot{z}' \end{pmatrix} = \begin{pmatrix} -\frac{1}{T_2} & 0 & -\frac{2V_0(t)}{\hbar}\sin\Phi_0 \\ 0 & -\frac{1}{T_2} & -\frac{2V_0(t)}{\hbar}\cos\Phi_0 \\ \frac{2V_0(t)}{\hbar}\sin\Phi_0 & \frac{2V_0(t)}{\hbar}\cos\Phi_0 & -\frac{1}{T_1} \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{z^{eq}}{T_1} \end{pmatrix}. \tag{23.282}$$

If the relaxation times are large compared to the pulse duration this describes approximately a rotation around an axis in the $xy-$plane (compare with 14.24)

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{r}' \approx W(t)\mathbf{r}' = \frac{2V_0(t)}{\hbar}W_0\mathbf{r}' \tag{23.283}$$

$$W_0 = \begin{pmatrix} 0 & 0 & -\sin\Phi_0 \\ 0 & 0 & -\cos\Phi_0 \\ \sin\Phi_0 & \cos\Phi_0 & 0 \end{pmatrix}. \tag{23.284}$$

Since the axis is time independent, a formal solution is given by

$$\mathbf{r}'(t) = \mathrm{e}^{W\int_0^t \frac{2V_0(t')}{\hbar}dt'}\mathbf{r}'(0) = \mathrm{e}^{W_0\Phi(t)}\mathbf{r}'(0) \tag{23.285}$$

with the phase angle

$$\Phi(t) = \int_{t_0}^{t} \frac{2V_0(t')}{\hbar}dt'. \tag{23.286}$$

Now, since

$$W_0^2 = \begin{pmatrix} -\sin^2\Phi_0 & -\sin\Phi_0\cos\Phi_0 & 0 \\ -\sin\Phi_0\cos\Phi_0 & -\cos^2\Phi_0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \tag{23.287}$$

$$W_0^3 = -W_0 \tag{23.288}$$

$$W_0^4 = -W_0^2 \tag{23.289}$$

the Taylor series of the exponential function in (23.285) can be summed up

$$\mathrm{e}^{W_0\Phi} = 1 + \Phi W_0 + \frac{1}{2}\Phi^2 W_0^2 + \frac{1}{3!}\Phi^3 W_0^3 + \cdots$$

$$= 1 + W_0^2\left(\frac{\Phi^2}{2} - \frac{\Phi^4}{4!} + \cdots\right) + W_0\left(\Phi - \frac{\Phi^3}{3!} + \cdots\right)$$

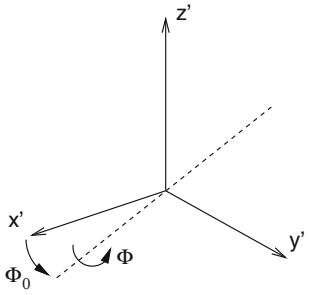


Fig. 23.25 (Rotation of the Bloch vector by a resonant pulse) A resonant pulse rotates the Bloch vector by the angle Φ around an axis in the $x'y'$-plane

$$= 1 + W_0^2\,(1 - \cos\Phi) + W_0 \sin\Phi$$

$$= \begin{pmatrix} 1 - \sin^2\Phi_0\,(1 - \cos\Phi) & -\sin\Phi_0 \cos\Phi_0\,(1 - \cos\Phi) & -\sin\Phi_0 \sin\Phi \\ -\sin\Phi_0 \cos\Phi_0\,(1 - \cos\Phi) & 1 - \cos^2\Phi_0\,(1 - \cos\Phi) & -\cos\Phi_0 \sin\Phi \\ \sin\Phi_0 \sin\Phi & \cos\Phi_0 \sin\Phi & \cos\Phi \end{pmatrix}$$

$$= \begin{pmatrix} \cos\Phi_0 & \sin\Phi_0 & 0 \\ -\sin\Phi_0 & \cos\Phi_0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi & \cos\Phi \end{pmatrix} \begin{pmatrix} \cos\Phi_0 & -\sin\Phi_0 & 0 \\ \sin\Phi_0 & \cos\Phi_0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{23.290}$$

The result is a rotation about the angle Φ around an axis in the $xy-$plane determined by Φ_0 (Fig. 23.25), especially around the $x-$axis for $\Phi_0 = 0$ and around the $y-$axis for $\Phi_0 = \frac{\pi}{2}$.

After a $\pi-$pulse ($\Phi = \pi$) the z-component changes its sign

$$\mathbf{r}' = \begin{pmatrix} \cos(2\Phi_0) & -\sin(2\Phi_0) & 0 \\ -\sin(2\Phi_0) & -\cos(2\Phi_0) & 0 \\ 0 & 0 & -1 \end{pmatrix} \mathbf{r}(0). \tag{23.291}$$

The transition between the two basis states $z = -1$ and $z = 1$ corresponds to a spin flip (Fig. 23.26). On the other hand, a $\pi/2-$pulse transforms the basis states into a coherent mixture

$$\mathbf{r}' = \begin{pmatrix} 1 - \sin^2\Phi_0 & -\sin\Phi_0 \cos\Phi_0 & -\sin\Phi_0 \\ -\sin\Phi_0 \cos\Phi_0 & 1 - \cos^2\Phi_0 & -\cos\Phi_0 \\ \sin\Phi_0 & \cos\Phi_0 & 0 \end{pmatrix} \mathbf{r}(0). \tag{23.292}$$

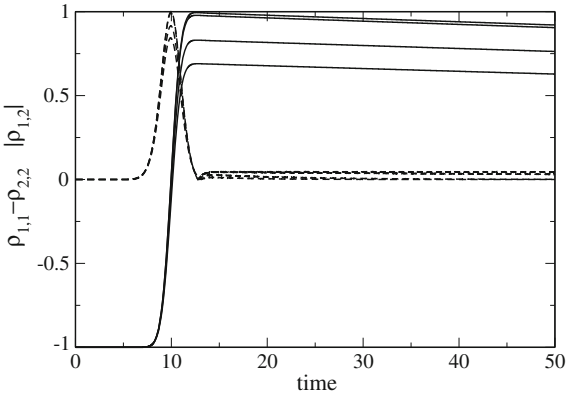


Fig. 23.26 (Spin flip by a π-pulse) The equations of motion of the Bloch vector (23.268) are solved with the 4th order Runge–Kutta method for an interaction pulse with a Gaussian shape. The pulse is adjusted to obtain a spin flip. The influence of dephasing processes is studied. $T_1 = 1000, t_p = 1.8, V_0 = 0.25$. The occupation difference $\rho_{11} - \rho_{22} = z$ (*solid curves*) and the coherence $|\rho_{12}| = \frac{1}{2}\sqrt{x^2 + y^2}$ (*broken curves*) are shown for several values of the dephasing time $T_2 = 5, 10, 100, 1000$

23.4.6 Elementary Qubit Manipulation

Whereas a classical bit can be only in one of two states

$$\text{either} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{or} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{23.293}$$

the state of a Qubit is a quantum mechanical superposition

$$|\psi> = C_0 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + C_1 \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \tag{23.294}$$

The time evolution of the Qubit is described by a unitary transformation

$$|\psi> \rightarrow U|\psi> \tag{23.295}$$

which is represented by a complex 2×2 unitary matrix that has the general form (see also Sect. 14.15)

$$U = \begin{pmatrix} \alpha & \beta \\ -\mathrm{e}^{\mathrm{i}\varphi}\beta^* & \mathrm{e}^{\mathrm{i}\varphi}\alpha^* \end{pmatrix} \quad |\alpha|^2 + |\beta|^2 = 1, \quad \det U = \mathrm{e}^{\mathrm{i}\varphi}. \tag{23.296}$$

The Bloch vector is transformed with an orthogonal matrix A, which can be found from (23.227) and the transformed density matrix $U\rho U^{-1}$

$$\mathbf{r} \to A\mathbf{r} \quad A = \begin{pmatrix} \Re\left((\alpha^2-\beta^2)\mathrm{e}^{-\mathrm{i}\varphi}\right) & \Im\left((\alpha^2+\beta^2)\mathrm{e}^{-\mathrm{i}\varphi}\right) & -2\Re\left(\alpha\beta\mathrm{e}^{-\mathrm{i}\varphi}\right) \\ \Im\left((\beta^2-\alpha^2)\mathrm{e}^{-\mathrm{i}\varphi}\right) & \Re\left((\alpha^2+\beta^2)\mathrm{e}^{-\mathrm{i}\varphi}\right) & 2\Im\left(\alpha\beta\mathrm{e}^{-\mathrm{i}\varphi}\right) \\ 2\Re(\alpha^*\beta) & 2\Im(\alpha^*\beta) & (|\alpha|^2-|\beta|^2) \end{pmatrix}. \tag{23.297}$$

Any single Qubit transformation can be realized as a sequence of rotations around just two axes [318, 319, 330]. In the following we consider some simple transformations, so called quantum gates [331].

23.4.6.1 Pauli-gates

Of special interest are the gates represented by the Pauli matrices $U = \sigma_i$ since any complex 2×2 matrix can be obtained as a linear combination of the Pauli matrices and the unit matrix (Sect. 14.15). For all three of them $\det U = -1$ and $\varphi = \pi$.

The X-gate

$$U_X = \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{23.298}$$

corresponds to rotation by π radians around the $x-$axis (23.291 with $\Phi_0 = 0$)

$$A_X = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}. \tag{23.299}$$

It is also known as NOT-gate since it exchanges the two basis states. Similarly, the Y-gate

$$U_Y = \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

rotates the Bloch vector by π radians around the $y-$axis (23.291 with $\Phi_0 = \pi/2$)

$$A_Y = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \tag{23.300}$$

and the Z-gate

$$U_Z = \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{23.301}$$

by π radians around the $z-$axis

$$A_Z = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{23.302}$$

This rotation can be replaced by two successive rotations in the $xy-$plane

$$A_Z = A_X A_Y. \tag{23.303}$$

The corresponding transformation of the wavefunction produces an overall phase shift of $\pi/2$ since the product of the Pauli matrices is $\sigma_x \sigma_y = \mathrm{i}\sigma_z$, which is not relevant for observable quantities.

23.4.6.2 Hadamard Gate

The Hadamard gate is a very important ingredient for quantum computation. It transforms the basis states into coherent superpositions and vice versa. It is described by the matrix

$$U_H = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \tag{23.304}$$

with $\det U_H = -1$ and

$$A_H = \begin{pmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \tag{23.305}$$

which can be obtained as the product

$$A_H = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \tag{23.306}$$

of a rotation by π radians around the $x-$axis and a second rotation by $\pi/2$ radians around the y-axis. The first rotation corresponds to the X-gate and the second to (23.292) with $\Phi_0 = \pi/2$

$$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}. \tag{23.307}$$

Problems

Problem 23.1 Wave Packet Motion

In this computer experiment we solve the Schroedinger equation for a particle in the potential $V(x)$ for an initially localized Gaussian wave packet $\psi(t = 0, x) \sim \exp(-a(x - x_0)^2)$. The potential is a box, a harmonic parabola or a fourth order double well. Initial width and position of the wave packet can be varied.

- Try to generate the stationary ground state wave function for the harmonic oscillator
- Observe the dispersion of the wave packet for different conditions and try to generate a moving wave packet with little dispersion.
- Try to observe tunneling in the double well potential

Problem 23.2 Two-state System

In this computer experiment a two-state system is simulated. Amplitude and frequency of an external field can be varied as well as the energy gap between the two states (see Fig. 23.9).

- Compare the time evolution at resonance and away from it

Problem 23.3 Three-state System

In this computer experiment a three-state system is simulated.

- Verify that the system behaves like an effective two-state system if the intermediate state is higher in energy than initial and final states (see Fig. 23.13).

Problem 23.4 Ladder Model

In this computer experiment the ladder model is simulated. The coupling strength and the spacing of the final states can be varied.

- Check the validity of the exponential decay approximation (see Fig. 23.15)

Problem 23.5 Semiclassical Approximation

In this computer experiment we study the crossing between two states along a nuclear coordinate. The time dependent Schrödinger equation for a wave packet approaching the crossing region is solved numerically and compared to the semiclassical approximation.

- Study the accuracy of the semiclassical approximation for different values of coupling and initial velocity

Problem 23.6 Landau–Zener Model

This computer experiment simulates the Landau Zener model. The coupling strength and the nuclear velocity can be varied (see Fig. 23.19).

- Try to find parameters for an efficient crossing of the states.

Problem 23.7 Resonance Line

In this computer experiment a two-state system with damping is simulated. The resonance curve is calculated from the steady state occupation probabilities (see Figs. 23.23, 23.24).

- Study the dependence of the line width on the intensity (power broadening).

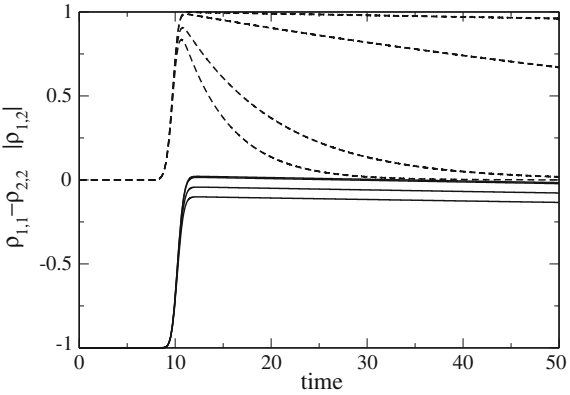


Fig. 23.27 (Generation of a coherent mixture by a $\pi/2$-pulse) The equations of motion of the Bloch vector (23.268) are solved with the 4th order Runge–Kutta method for an interaction pulse with a Gaussian shape. The pulse is adjusted to obtain a coherent mixture. The influence of dephasing processes is studied. $T_1 = 1000, t_p = 0.9, V_0 = 0.25$. The occupation difference $\rho_{11} - \rho_{22} = z$ (*solid curves*) and the coherence $|\rho_{12}| = \frac{1}{2}\sqrt{x^2 + y^2}$ (*broken curves*) are shown for several values of the dephasing time $T_2 = 5, 10, 100, 1000$

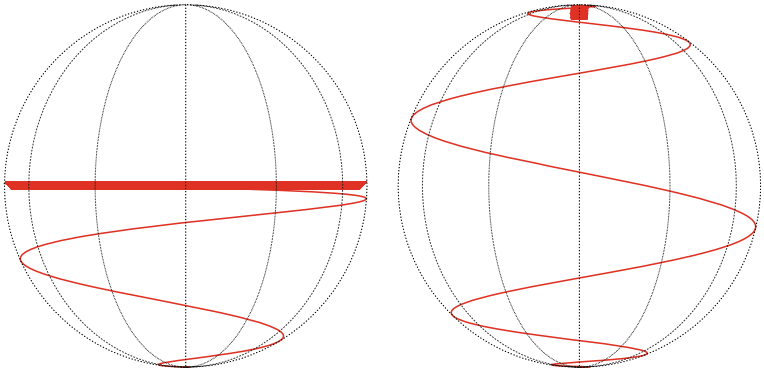


Fig. 23.28 (Motion of the Bloch vector during $\pi/2$ and π pulses) The trace of the Bloch vector is shown in the laboratory system. ***Left*** $\pi/2-$pulse as in Fig. 23.27 with $T_2 = 1000$. ***Right*** $\pi-$pulse as in Fig. 23.26 with $T_2 = 1000$

Problem 23.8 Spin Flip

The damped two-state system is now subject to an external pulsed field (see Figs. 23.26, 23.27, 23.28).

- Try to produce a coherent superposition state ($\pi/2$ pulse) or a spin flip (π pulse).
- Investigate the influence of decoherence.

Chapter 24
Variational Methods for Quantum Systems

The variational principle states, that the energy expectation value of any trial function is bounded from below by the exact ground state energy. Therefore, the ground state can be approximated by minimizing the energy of a trial function which involves certain parameters that have to be optimized. In this chapter we study two different kinds of quantum systems. First we apply the variational principle to one- and two-electron systems and calculate the ground state energy of the Helium atom and the Hydrogen molecule. If the trial function treats electron correlation explicitly, the calculation of the energy involves unseparable multidimensional integrals which can be efficiently evaluated with the variational quantum Monte Carlo method. In a second series of computer experiments we study models with a large number of variational parameters. We simulate excitons in a molecular aggregate which are coupled to internal vibrations. The number of parameters increases with the system size up to several hundred and the optimization requires efficient strategies. We use several kinds of trial functions to study the transition from a delocalized to a localized state.

The variational principle is a very valuable tool to approximate the groundstate energy and wavefunction. Consider the representation of the Hamiltonian in a complete basis of eigenfunctions [277]

$$H = \sum_n |\psi_n > E_n < \psi_n| \tag{24.1}$$

with the groundstate energy

$$E_0 \leq E_n \tag{24.2}$$

and a trial function with some adjustable parameters

$$\psi_{trial}(\lambda). \tag{24.3}$$

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7_24

The expectation value of the Hamiltonian

$$< \psi_{trial} H \psi_{trial} >= \sum_n | < \psi_{trial} | \psi_n > |^2 E_n \geq E_0 \sum_n | < \psi_{trial} | \psi_n > |^2$$

$$= E_0 < \psi_{trial} \left[\sum_n |\psi_n >< \psi_n| \right] \psi_{trial} >= E_0 |\psi_{trial}|^2. \tag{24.4}$$

Hence the energy expectation value is bounded from below by the groundstate energy

$$\frac{< \psi_{trial} H \psi_{trial} >}{|\psi_{trial}|^2} \geq E_0. \tag{24.5}$$

For the exact groundstate

$$\frac{< \psi_0 | H | \psi_0 >}{|\psi_0|^2} = E_0$$

and the variance

$$\sigma_E^2 = \frac{< \psi_0 | H^2 | \psi_0 >}{|\psi_0|^2} - \left(\frac{< \psi_0 | H | \psi_0 >}{|\psi_0|^2} \right)^2 = 0. \tag{24.6}$$

Now, let us try to find an approximate solution of the eigenvalue problem

$$H\psi = E_0 \psi \tag{24.7}$$

by optimizing the trial function. The residual is

$$R = H\psi_{trial} - E_0 \psi_{trial} \tag{24.8}$$

and, applying Galerkin's method (p. 272) we minimize the scalar product

$$< \psi_{trial} R >=< \psi_{trial} H \psi_{trial} > - E_0 < \psi_{trial} | \psi_{trial} > \tag{24.9}$$

where the trial function should be normalized. Alternatively, we divide by the squared norm and minimize

$$\frac{< \psi_{trial} H \psi_{trial} >}{< \psi_{trial} | \psi_{trial} >} - E_0. \tag{24.10}$$

Hence the "best" trial function is found by minimizing the energy with respect to the parameters λ.

Now, assume that the groundstate is normalized

$$|\psi_0|^2 = 1 \tag{24.11}$$

and choose the normalization of the trial function such that

$$\psi_{trial} = \psi_0 + \rho \tag{24.12}$$

$$< \psi_0 | \rho >= 0. \tag{24.13}$$

Then,

$$\frac{< \psi_{trial} H \psi_{trial} >}{|\psi_{trial}|^2} = \frac{E_0 + < \rho H \rho >}{1 + |\rho|^2} = E_0 + O(|\rho|^2) \tag{24.14}$$

the accuracy of the energy is of second order in $|\rho|$. From

$$\frac{< \psi_{trial} H^2 \psi_{trial} >}{|\psi_{trial}|^2} = \frac{E_0^2 + < \rho H^2 \rho >}{1 + |\rho|^2} \tag{24.15}$$

we find that the variance of the energy

$$\begin{aligned}
\sigma_E^2 &= \frac{< \psi_{trial} H^2 \psi_{trial} >}{|\psi_{trial}|^2} - \left(\frac{< \psi_{trial} H \psi_{trial} >}{|\psi_{trial}|^2} \right)^2 \\
&= \frac{E_0^2 + < \rho H^2 \rho >}{1 + |\rho|^2} - \left(\frac{E_0 + < \rho H \rho >}{1 + |\rho|^2} \right)^2 \\
&\approx E_0^2 (1 - |\rho|^2) + < \rho H^2 \rho > - E_0^2 (1 - 2|\rho|^2) - 2E_0 < \rho H \rho > \\
&\approx E_0^2 < \rho | \rho >^2 + < \rho H | H \rho > - 2E_0 < \rho H \rho > \\
&\approx |(H - E_0)\rho|^2
\end{aligned} \tag{24.16}$$

is also second order in $|\rho|$. It is bounded from below by zero. Therefore, Quantum Monte Carlo methods often minimize the variance instead of the energy for which the lower bound is unknown.

24.1 Variational Quantum Monte Carlo Simulation of Atomic and Molecular Systems

Electron structure calculations for atoms and molecules beyond the self consistent field level (Hartree Fock uses one Slater determinant as a trial function, MCSCF methods a combination of several) need an explicit treatment of electron correlation. This can be achieved by expanding the wavefunction into a large number of

configurations (CI method) or, alternatively, by using trial functions which depend explicitly on the electron-electron distances. Very popular [332, 333] are factors of the Jastrow pair-correlation [334] type

$$\exp\left\{\sum_{i<j} U(r_{ij})\right\} \tag{24.17}$$

where in the simplest case

$$U(r_{ij}) = \frac{\alpha r_{ij}}{1+\beta r_{ij}} \tag{24.18}$$

has the form of a Pade approximant. Wavefunctions including a Jastrow factor do not factorize and make it necessary to apply Monte Carlo integration methods to calculate the energy expectation value (see p. 205). For the computer simulation of two-electron systems we use trial functions of the type

$$\psi = \mathrm{e}^{-\kappa r_{1a}}\mathrm{e}^{-\kappa r_{2b}}\mathrm{e}^{\alpha r_{12}/(1+\beta r_{12})} \tag{24.19}$$

which are products of two 1s-orbitals centered at the (possibly same) positions $\mathbf{r}_{a,b}$ and a Jastrow factor. In the following, we abbreviate

$$u = 1+\beta r_{12}. \tag{24.20}$$

Starting with the derivatives

$$\frac{\partial}{\partial x_1}\psi = -\frac{\kappa x_{1a}}{r_{1a}}\psi + \frac{\alpha x_{12}}{r_{12}u^2}\psi \tag{24.21}$$

$$\begin{aligned}\frac{\partial^2}{\partial x_1^2}\psi &= \left[-\frac{\kappa x_{1a}}{r_{1a}} + \frac{\alpha x_{12}}{r_{12}u^2}\right]^2\psi \\ &+ \left[-\frac{\kappa}{r_{1a}} + \frac{\kappa x_{1a}^2}{r_{1a}^3}\right]\psi + \left[\frac{\alpha}{r_{12}u^2} - \frac{\alpha x_{12}^2}{r_{12}^3u^2} - 2\frac{\alpha\beta x_{12}^2}{r_{12}^2u^3}\right]\psi\end{aligned} \tag{24.22}$$

we calculate the kinetic energy

$$\begin{aligned}T\psi = -\frac{1}{2}(\nabla_1^2+\nabla_2^2)\psi &= \left[-\kappa^2 - \frac{\alpha^2}{u^4} + \frac{\alpha\kappa}{u^2}\left(\frac{\mathbf{r}_{1a}}{r_{1a}} - \frac{\mathbf{r}_{2b}}{r_{2b}}\right)\frac{\mathbf{r}_{12}}{r_{12}}\right]\psi \\ &+ \left[\frac{\kappa}{r_{1a}} + \frac{\kappa}{r_{2b}} - \frac{2\alpha}{r_{12}u^2} + 2\frac{\alpha\beta}{u^3}\right]\psi.\end{aligned} \tag{24.23}$$

For short electron-electron distance

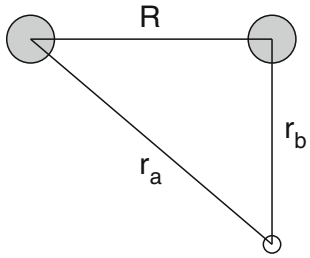


Fig. 24.1 Geometry of H_2^+

$$T\psi \to \left[-\kappa^2 - \alpha^2 + \alpha\kappa\left(\frac{\mathbf{r}_{1a}}{r_{1a}} - \frac{\mathbf{r}_{2b}}{r_{2b}}\right)\frac{\mathbf{r}_{12}}{r_{12}} + \frac{\kappa}{r_{1a}} + \frac{\kappa}{r_{2b}} + 2\alpha\beta - \frac{2\alpha}{r_{12}}\right]\psi. \tag{24.24}$$

A choice of $\alpha = 1/2$ cancels the divergent Coulomb repulsion at $r_{12} \to 0$ and fulfills the electron-electron cusp condition [333, 335]. More complicated Jastrow factors also allow to fulfill the electron-nuclei cusp conditions.

24.1.1 The Simplest Molecule: H_2^+

As a first example (Problem 24.1), we consider an electron moving in the Coulomb field of two protons (Fig. 24.1). Applying the Born-Oppenheimer approximation the protons are kept fixed at a distance R. In atomic units,[1] the Hamiltonian is

$$H = T + V = -\frac{1}{2}\nabla^2 - \frac{1}{r_a} - \frac{1}{r_b} + \frac{1}{R}. \tag{24.25}$$

This eigenvalue problem can be solved exactly (using elliptic coordinates) and is also a popular example for the variational method.

As a trial wavefunction we use the linear combination of two hydrogen-like 1 s orbitals

$$\varphi_a = \sqrt{\frac{\kappa^3}{\pi}}\mathrm{e}^{-\kappa r_a} \quad \varphi_b = \sqrt{\frac{\kappa^3}{\pi}}\mathrm{e}^{-\kappa r_b} \tag{24.26}$$

which are solutions for the problem with two nuclear charges κ at infinite distance. At finite distances, the variational parameter κ is a measure of the effective nuclear charge. For large distance $\kappa = 1$ as for a single proton whereas at short distances the optimum value approaches $\kappa = 2$ as for the He^+ ion.

Since the problem is highly symmetric,we take a symmetric combination

[1] i.e. setting $a_B = 4\pi\varepsilon_0\hbar^2/e^2m_e = 1$ and $\hbar^2/m_e = 1$.

$$\varphi_{trial} = \frac{1}{\sqrt{2(1 \pm S)}} [\varphi_a \pm \varphi_b] \tag{24.27}$$

where the overlap integral can be calculated using elliptic coordinates

$$\begin{aligned} r_a &= \frac{R}{2}(\lambda + \mu) \quad r_b = \frac{R}{2}(\lambda - \mu) \\ S &= \int\int\int \varphi_a \varphi_b dV = 2\pi \int_1^\infty d\lambda \int_{-1}^1 d\mu \quad 2\kappa^3(\lambda^2 - \mu^2)\frac{R^3}{8} \mathrm{e}^{-\kappa R\lambda} \\ &= \mathrm{e}^{-\kappa R}(1 + \kappa R + \frac{\kappa^2 R^2}{3}). \end{aligned} \tag{24.28}$$

The action of the Hamiltonian is

$$H\varphi_a = -\frac{1}{2}\left(\kappa^2 - \frac{2\kappa}{r_a}\right)\varphi_a + \left[\frac{1}{R} - \frac{1}{r_a} - \frac{1}{r_b}\right]\varphi_a = \left[-\frac{1}{r_b} + \frac{\kappa - 1}{r_a} + \left(-\frac{\kappa^2}{2} + \frac{1}{R}\right)\right]\varphi_a \tag{24.29}$$

$$H\varphi_b = -\frac{1}{2}\left(\kappa^2 - \frac{2\kappa}{r_b}\right)\varphi_b + \left[\frac{1}{R} - \frac{1}{r_a} - \frac{1}{r_b}\right]\varphi_b = \left[-\frac{1}{r_a} + \frac{\kappa - 1}{r_b} + \left(-\frac{\kappa^2}{2} + \frac{1}{R}\right)\right]\varphi_b \tag{24.30}$$

$$H\varphi_{trial} = \frac{1}{\sqrt{2(1 \pm S)}}\left[(\frac{\kappa}{r_a} - \frac{\kappa^2}{2})\varphi_a \pm \left(\frac{\kappa}{r_b} - \frac{\kappa^2}{2}\right)\varphi_b\right] + \left[\frac{1}{R} - \frac{1}{r_a} - \frac{1}{r_b}\right]\varphi_{trial}$$

from which we obtain the local energy

$$E_{loc} = \left[\frac{1}{R} - \frac{1}{r_a} - \frac{1}{r_b} - \frac{\kappa^2}{2}\right] + \frac{\frac{\kappa}{r_a}\varphi_a \pm \frac{\kappa}{r_b}\varphi_b}{\varphi_a \pm \varphi_b}. \tag{24.31}$$

For comparison, we calculate the expectation value of the energy

$$< \varphi_{trial} H \varphi_{trial} >= \frac{1}{2(1 \pm S)} [H_{aa} + H_{bb} \pm H_{ab} \pm H_{ba}] = \frac{H_{aa} \pm H_{ab}}{1 \pm S} \tag{24.32}$$

with the matrix elements

$$H_{aa} = H_{bb} = \frac{1}{R} - \frac{\kappa^2}{2} - \int \frac{\varphi_a^2}{r_b} dV + (\kappa - 1)\int \frac{\varphi_a^2}{r_a} dV \tag{24.33}$$

$$H_{ab} = H_{ba} = \left(\frac{1}{R} - \frac{\kappa^2}{2}\right) S - \int \frac{\varphi_a \varphi_b}{r_a} dV + (\kappa - 1)\int \frac{\varphi_a \varphi_b}{r_b}. \tag{24.34}$$

The integrals can be evaluated in elliptic coordinates

$$\int \frac{\varphi_b^2}{r_b} dV = \int \frac{\varphi_a^2}{r_a} dV = 2\kappa^3 \int_1^\infty d\lambda \int_{-1}^1 d\mu \frac{R^2}{4} (\lambda - \mu) \mathrm{e}^{-\kappa R(\lambda+\mu)/2} = \kappa \tag{24.35}$$

$$\int \frac{\varphi_b^2}{r_a} dV = \int \frac{\varphi_a^2}{r_b} dV = 2\kappa^3 \int_1^\infty d\lambda \int_{-1}^1 d\mu \frac{R^2}{4} (\lambda + \mu) \mathrm{e}^{-\kappa R(\lambda+\mu)/2}$$
$$= \frac{1}{R} - \mathrm{e}^{-2\kappa R} (\kappa + \frac{1}{R}) \tag{24.36}$$

$$\int \frac{\varphi_a \varphi_b}{r_b} dV = \int \frac{\varphi_a \varphi_b}{r_a} dV = 2\kappa^3 \int_1^\infty d\lambda \int_{-1}^1 d\mu \frac{R^2}{4} (\lambda - \mu) \mathrm{e}^{-\kappa R\lambda}$$
$$= \mathrm{e}^{-\kappa R} \left(\kappa + \kappa^2 R\right). \tag{24.37}$$

In our computer experiment (Problem 24.1), we first keep $\kappa = 1$ fixed and use the variational MC method to calculate the expectation value of the energy. Figure 24.2 compares the results with the exact value (24.32). Next we vary κ and determine the optimum value at each point R by minimizing $E(R, \kappa)$. Figure 24.3 shows the κ-dependence for several points. The optimized κ-values (Fig. 24.4) lead to lower energies, especially at short distances. The equilibrium is now at $R_0 = 2.0$ Bohr with a minimum energy of -0.587 a.u. instead of 2.5 Bohr and -0.565 a.u. for $\kappa = 1$. (Exact values are 2.00 Bohr and -0.603 a.u. [277]).

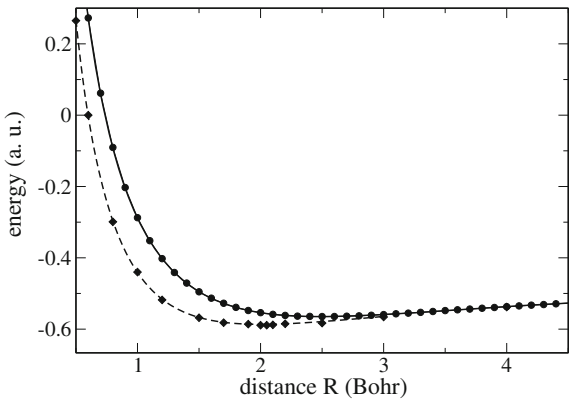


Fig. 24.2 (Adiabatic groundstate energy of H_2^+) The potential energy curve of H_2^+ is calculated with the variational method. *Circles* show the results from MC integration for a maximum step length of 0.5 Bohr and averages over 2×10^7 samples for a fixed effective charge $\kappa = 1$. The *solid curve* shows the results of the exact integration (24.32) for comparison. *Diamonds* show the MC results after optimizing $\kappa(R)$ at each point. The *dashed curve* shows the results of the exact integration (24.32) where $\kappa(R)$ was determined by solving $\frac{\partial}{\partial \kappa} < \varphi_{trial} H \varphi_{trial} >= 0$ numerically

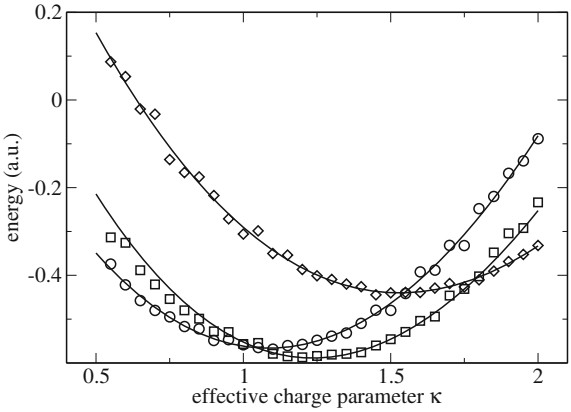


Fig. 24.3 (Optimization of the variational parameter κ for H_2^+) The groundstate energy from MC integration is shown as a function of κ for $R = 1$ (*diamonds*), $R = 2$ (*squares*) and $R = 3$ (*circles*). The *curves* show a fit with a cubic polynomial which helps to find the minima

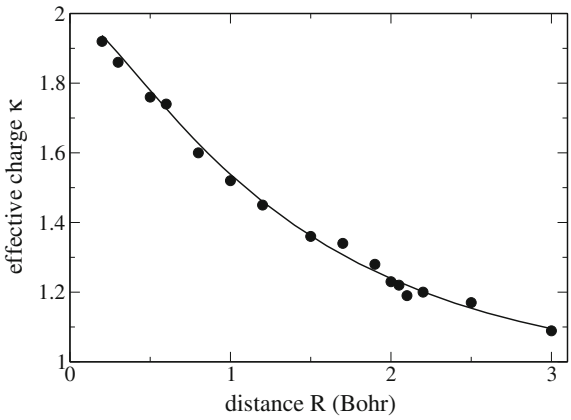


Fig. 24.4 (Optimized effective charge parameter for H_2^+) The variational parameter κ is optimized by minimizing the MC energy as shown in Fig. 24.3 (*circles*). The *curve* shows the exact values obtained by minimizing (24.32) numerically

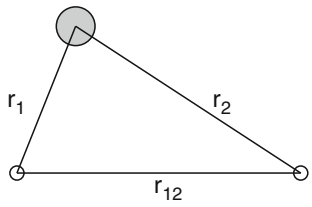


Fig. 24.5 Geometry of He

24.1.2 The Simplest Two-Electron System: The Helium Atom

The Helium atom (Fig. 24.5) is the simplest "many-electron" system where electron-electron interaction has to be taken into account. The electronic Hamiltonian reads in atomic units

$$H = -\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{2}{r_1} - \frac{2}{r_2} + \frac{1}{r_{12}}. \tag{24.38}$$

Without electron-electron interaction, the singlet groundstate would be simply given in terms of hydrogen-like 1s-orbitals as

$$\psi_0 = \frac{2^3}{\pi} e^{-2r_1} e^{-2r_2} \frac{1}{\sqrt{2}} (\uparrow(1)\downarrow(2) - \uparrow(2)\downarrow(1)) \tag{24.39}$$

with

$$\left[-\frac{1}{2}\nabla_1^2 - \frac{1}{2}\nabla_2^2 - \frac{2}{r_1} - \frac{2}{r_2}\right]\psi_0 = -2\left(1 - \frac{1}{r_1}\right)\psi_0 - 2\left(1 - \frac{1}{r_2}\right)\psi_0 - \frac{2}{r_1}\psi_0 - \frac{2}{r_2}\psi_0 = -4\psi_0. \tag{24.40}$$

For the variational treatment (Problem 24.2) we use a trial wavefunction with a variable exponent to take the partial shielding of the central charge into account

$$\psi_{trial} = \frac{\kappa^3}{\pi} e^{-\kappa r_1} e^{-\kappa r_2} \frac{1}{\sqrt{2}} (\uparrow(1)\downarrow(2) - \uparrow(2)\downarrow(1)) \tag{24.41}$$

where the antisymmetric spin function accounts for the Pauli principle.

Then,

$$H\psi_{trial} = -\frac{1}{2}\left(\kappa^2 - \frac{2\kappa}{r_1} + \kappa^2 - \frac{2\kappa}{r_2}\right)\psi_{trial} + \left(\frac{1}{r_{12}} - \frac{2}{r_1} - \frac{2}{r_2}\right)\psi_{trial} \tag{24.42}$$

$$E_{loc} = \frac{1}{r_{12}} - \kappa^2 + \frac{\kappa - 2}{r_1} + \frac{\kappa - 2}{r_2}. \tag{24.43}$$

The integration can be performed analytically [277]. First we calculate

$$\left(\frac{\kappa^3}{\pi}\right)^2 \int e^{-2\kappa r_1} e^{-2\kappa r_2} \frac{1}{r_1} dV_1 dV_2 = \frac{\kappa^3}{\pi} \int e^{-2\kappa r} \frac{1}{r} dV = \kappa. \tag{24.44}$$

The integral of the electron-electron interaction is

$$\begin{aligned}
&\left(\frac{\kappa^3}{\pi}\right)^2 \int e^{-2\kappa r_1} e^{-2\kappa r_2} \frac{1}{r_{12}} dV_1 dV_2 \\
&= \left(\frac{\kappa^3}{\pi}\right)^2 \int_0^\infty r_1^2 dr_1 e^{-2\kappa r_1} \int_0^\infty r_2^2 dr_2 e^{-2\kappa r_2} \int d\Omega_1 \int d\Omega_2 \frac{1}{r_{12}} \\
&= \left(\frac{\kappa^3}{\pi}\right)^2 \int_0^\infty r_1^2 dr_1 e^{-2\kappa r_1} \int_0^\infty r_2^2 dr_2 e^{-2\kappa r_2} \frac{(4\pi)^2}{\max(r_1, r_2)}
\end{aligned}$$

$$= \left(\frac{\kappa^3}{\pi}\right)^2 \int_0^\infty r_1^2 dr_1 \mathrm{e}^{-2\kappa r_1} (4\pi)^2 \left[\frac{1}{r_1}\int_0^{r_1} r_2^2 dr_2 \mathrm{e}^{-2\kappa r_2} + \int_{r_1}^\infty r_2 dr_2 \mathrm{e}^{-2\kappa r_2}\right]$$
$$= 16\kappa^6 \int_0^\infty r_1^2 dr_1 \mathrm{e}^{-2\kappa r_1} \left[\frac{1-\mathrm{e}^{-2\kappa r_1}}{4r_1\kappa^3} - \frac{\mathrm{e}^{-2\kappa r_1}}{4\kappa^2}\right]$$
$$= \frac{5}{8}\kappa. \tag{24.45}$$

Together, we obtain

$$< \psi_{trial} H \psi_{trial} >= -\kappa^2 + \frac{5}{8}\kappa + 2(\kappa - 2)\kappa = \kappa^2 + \left(\frac{5}{8} - 4\right)\kappa \tag{24.46}$$

which has its minimum at (Figs. 24.6 and 24.7)

$$\kappa_{min} = 2 - \frac{5}{16} \approx 1.688 \tag{24.47}$$

with the value

$$\min_\kappa < \psi_{trial} H \psi_{trial} >= -\frac{729}{256} \approx -2.848.$$

Next, we consider a (not normalized) trial wavefunction of the Slater-Jastrow type (24.19)

$$\psi_{trial} = \mathrm{e}^{-\kappa r_1} \mathrm{e}^{-\kappa r_2} \mathrm{e}^{\alpha r_{12}/(1+\beta r_{12})} \frac{1}{\sqrt{2}} \left(\uparrow(1)\downarrow(2) - \uparrow(2)\downarrow(1)\right). \tag{24.48}$$

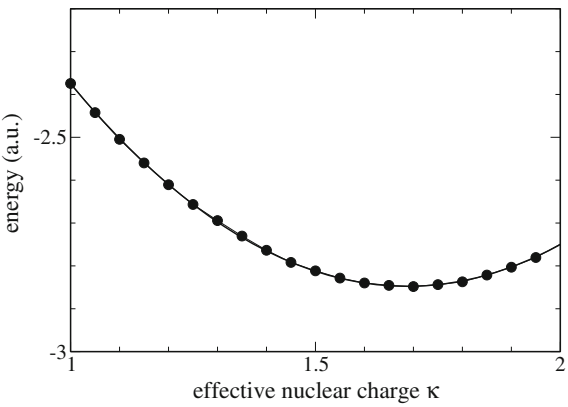


Fig. 24.6 (Optimization of the effective charge for the Helium atom) The groundstate energy of the Helium atom was calculated with MC integration. The *circles* show the average over 10^7 points. The *curve* shows the exact result (24.46) for comparison. The optimum value is $\kappa = 1.688$

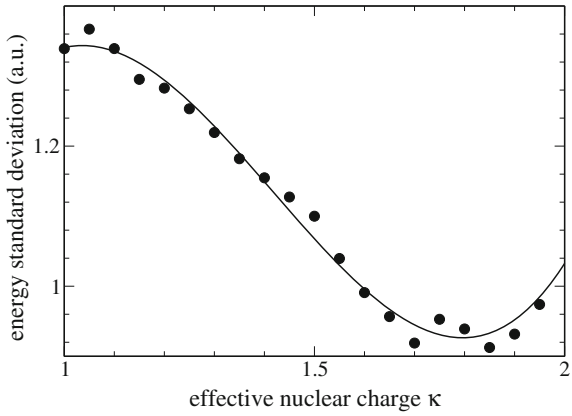


Fig. 24.7 (Standard deviation of the MC energy) The *Circles* show the standard deviation of the MC energy for Helium. Its minimum between $\kappa = 1.7 \cdots 1.9$ is close to the minimum of the energy (Fig. 24.6). The *curve* shows a cubic polynomial fit

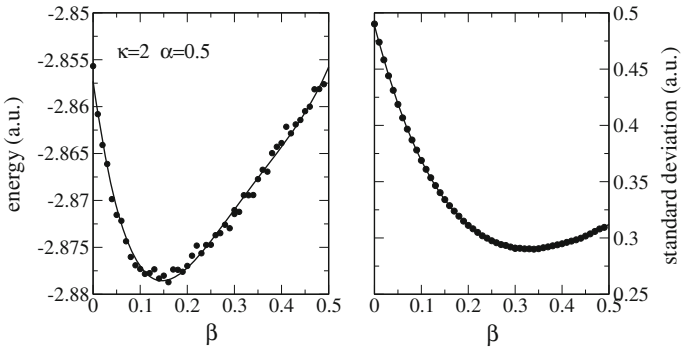


Fig. 24.8 (Variation of β) The groundstate of the Helium atom is approximated with the Slater-Jastrow wavefunction (24.48). Singularities of the potential energy are removed by using $\kappa = 2$ and $\alpha = 1/2$. Each point represents an average over 10^7 samples. ***Left*** The energy minimum of -2.879 is found at $\beta = 0.15$. ***Right*** the standard deviation has a minimum value of 0.29 at $\beta = 0.35$

From (24.23) with $\mathbf{r}_a = \mathbf{r}_b$ we find the local energy

$$E_{loc} = \frac{\kappa - 2}{r_1} + \frac{\kappa - 2}{r_2} + \frac{1}{r_{12}}\left(1 - \frac{2\alpha}{u^2}\right) + \frac{2\alpha\beta}{u^3} - \kappa^2 - \frac{\alpha^2}{u^4} + \frac{\kappa\alpha}{u^2}\left(\frac{\mathbf{r}_1}{r_1} - \frac{\mathbf{r}_2}{r_2}\right)\frac{(\mathbf{r}_1 - \mathbf{r}_2)}{r_{12}}. \tag{24.49}$$

With fixed values $\alpha = 1/2$ and $\kappa = 2$ all singularities in the local energy are removed, but this also reduces the flexibility of the test function. The energy minimum of -2.879 is found at $\beta = 0.15$ (Fig. 24.8). A further improvement can be achieved by varying the exponent κ together with β. The minimum now is -2.885 at $\kappa = 1.91$ (Fig. 24.9). If we drop the cusp condition and vary all three parameters we find a

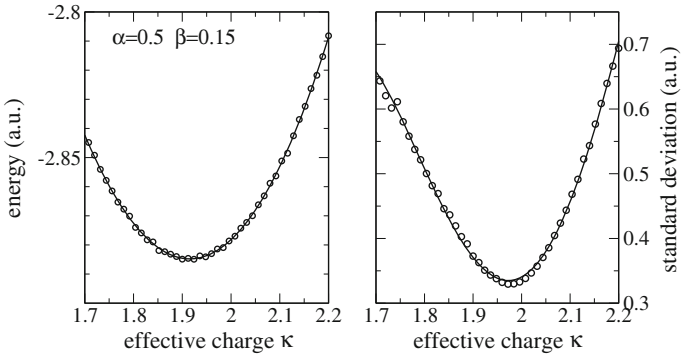


Fig. 24.9 (Variation of κ) The groundstate of the Helium atom is approximated with the Slater-Jastrow wavefunction (24.48). From Fig. 24.8 the optimized value of $\beta = 0.15$ is taken, $\alpha = 1/2$. Each point represents an average over 10^7 samples. ***Left*** The energy minimum of -2.885 is found at $\kappa = 1.91$. ***Right*** the standard deviation has a minimum value of 0.33 at $\kappa = 1.98$

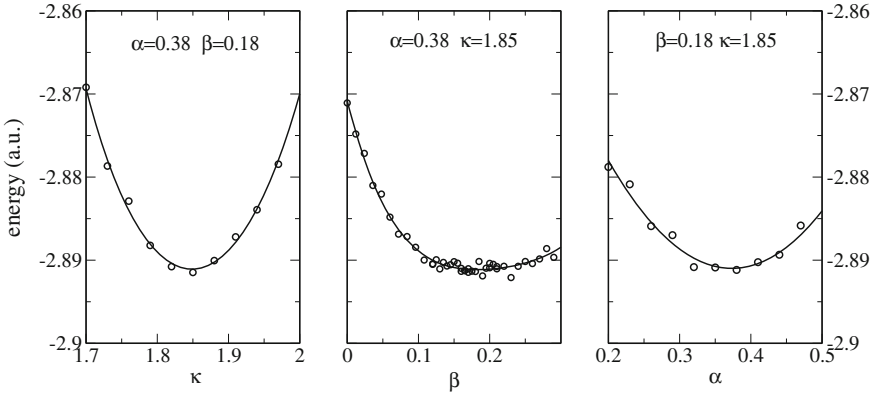


Fig. 24.10 (Variation of all parameters) The groundstate of the Helium atom is approximated with the Slater-Jastrow wavefunction (24.48). Variation of all three parameters gives a lowest energy of -2.891 for $\alpha = 0.38$, $\beta = 0.18$, $\kappa = 1.85$

slightly smaller value of -2.891 with a standard variation of $\sigma = 0.36$ (Fig. 24.10). More sophisticated trial wavefunctions reproduce the exact value of -2.903724 even more accurately [336, 337].

24.1.3 The Hydrogen Molecule H_2

The Helium atom can be considered as the limiting case of the H_2 molecule for zero distance (neglecting nuclear Coulomb repulsion). At finite distance R the one-

electron factors of the wavefunction have to be symmetrized.[2] We use a trial function (we omit the singlet spin function and do not normalize the wavefunction)

$$\begin{aligned}\psi &= C\left[\varphi_a(r_1)\varphi_b(r_2)+\varphi_b(r_1)\varphi_a(r_2)\right]+(1-C)\left[\varphi_a(r_1)\varphi_a(r_2)+\varphi_b(r_1)\varphi_b(r_2)\right]\\ &= C\psi_{VB}+(1-C)\psi_{ionic}\end{aligned} \tag{24.50}$$

which combines covalent and ionic configurations

$$\psi_{VB} = (\varphi_a(r_1)\varphi_b(r_2)+\varphi_b(r_1)\varphi_a(r_2)) \tag{24.51}$$

$$\psi_{ionic} = (\varphi_a(r_1)\varphi_a(r_2)+\varphi_b(r_1)\varphi_b(r_2)) \tag{24.52}$$

and includes as special cases

- the Heitler-London or valence-bond ansatz ($C = 1$) ψ_{VB}
- the Hund-Mulliken-Bloch or molecular orbital method where the symmetric MO is doubly occupied ($C = 0.5$)

$$\begin{aligned}\psi_{MO}^{++} &= (\varphi_a(r_1)+\varphi_b(r_1))\,(\varphi_a(r_2)+\varphi_b(r_2))\\ &= \psi_{VB}+\psi_{ionic}\end{aligned} \tag{24.53}$$

- the Heitler-London method augmented by ionic contributions $C = (1+\lambda)^{-1}$

$$\psi = \psi_{VB}+\lambda\psi_{ionic} \tag{24.54}$$

- the MCSCF ansatz which mixes two determinants ($C = 1 - C_d$)

$$\begin{aligned}\psi &= \psi_{MO}^{++}+C_d\psi_{MO}^{--}\\ &= (\varphi_a(r_1)+\varphi_b(r_1))\,(\varphi_a(r_2)+\varphi_b(r_2))+C_d\,(\varphi_a(r_1)-\varphi_b(r_1))\,(\varphi_a(r_2)-\varphi_b(r_2))\\ &= (1-C_d)\psi_{VB}+(1+C_d)\psi_{ionic}.\end{aligned} \tag{24.55}$$

The molecular orbital method corresponds to the Hartree–Fock method which is very popular in molecular physics. At large distance it fails to describe two separate hydrogen atoms with an energy of -1 au properly. In the bonding region it is close to the valence bond method which has the proper asymptotic limit. Both predict an equilibrium around $R = 1.6$ (Fig. 24.11).

To improve the results we vary the effective charge κ and the configuration mixing C (Fig. 24.12). Optimization of κ lowers the energy especially at small internuclear distances where the effective charge reaches a value of 2 as for the Helium atom (Fig. 24.13). The minimum of the potential curve now is found at a much more reasonable value of $R = 1.4$. Variation of the configuration mixing lowers the energy

[2] We consider only singlet states with antisymmetric spin part.

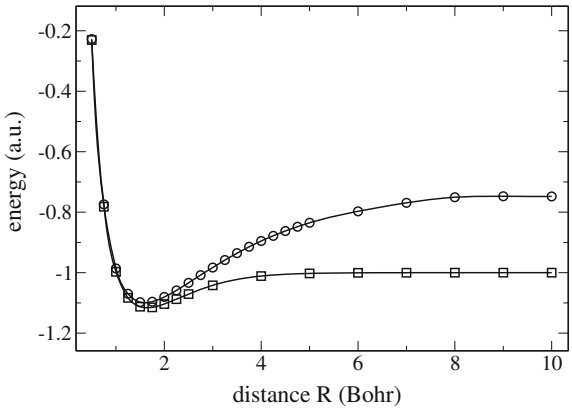


Fig. 24.11 (Comparison of Heitler-London and Hund-Mulliken-Bloch energies for H_2) The MO method (*circles*) fails to describe the asymptotic behaviour at large distances properly. In the bonding region it is close to the VB method (*squares*). Both predict a minimum around $R = 1.6$

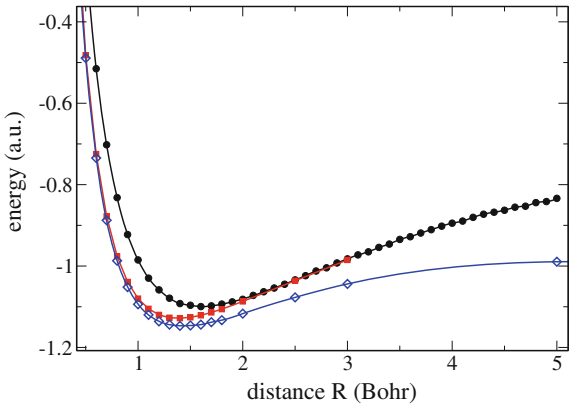


Fig. 24.12 (Optimization of effective charge κ and configuration mixing C) Starting from the MO energy (*black circles*) optimization of κ (red squares) and C (*blue diamonds*) lower the energy considerably and shift the potential minimum from 1.6 to 1.4 Bohr (see Problem 24.3)

mostly at larger distances where the proper limit is now obtained. For our computer experiment (Problem 24.3) we include a Jastrow factor into the trial function

$$\psi = \left\{C\left[\mathrm{e}^{-\kappa r_{1a}-\kappa r_{2b}} + \mathrm{e}^{-\kappa r_{1b}-\kappa r_{2a}}\right] + (1-C)\left[\mathrm{e}^{-\kappa r_{1a}-\kappa r_{2a}} + \mathrm{e}^{-\kappa r_{1b}-\kappa r_{2b}}\right]\right\} \times \exp\left\{\frac{\alpha r_{12}}{1+\beta r_{12}}\right\} \tag{24.56}$$

and vary κ, β and C to minimize the expectation value of the local energy

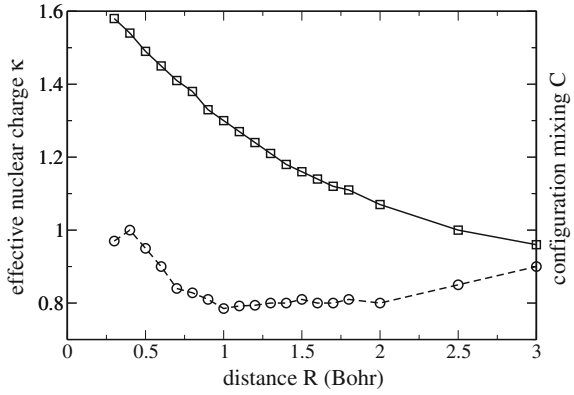


Fig. 24.13 (Optimized values of κ and C) The effective charge (*squares*) approaches $\kappa = 2$ at very short distances corresponding to the He atom. Configuration mixing (*circles*) is most important in the bonding region. At large distances the valence bond wavefunction (C = 1) provides the lowest energy. At very small distances the two configurations become equivalent making the mixing meaningless as $R \to 0$

$$
\begin{aligned}
E_{loc} = {} & \frac{1}{r_{12}}\left(1-\frac{2\alpha}{u^2}\right) - \frac{1}{r_{1a}} - \frac{1}{r_{1b}} - \frac{1}{r_{2a}} - \frac{1}{r_{2b}} + \frac{2\alpha\beta}{u^3} - \kappa^2 - \frac{\alpha^2}{u^4} \\
& + C\left[\frac{\kappa}{r_{1a}} + \frac{\kappa}{r_{2b}} + \frac{\kappa\alpha}{u^2}\left(\frac{\mathbf{r}_{1a}}{r_{1a}} - \frac{\mathbf{r}_{2b}}{r_{2b}}\right)\frac{(\mathbf{r}_1-\mathbf{r}_2)}{r_{12}}\right] \mathrm{e}^{\alpha r_{12}/u - \kappa r_{1a} - \kappa r_{2b}}\psi^{-1} \\
& + C\left[\frac{\kappa}{r_{1b}} + \frac{\kappa}{r_{2a}} + \frac{\kappa\alpha}{u^2}\left(\frac{\mathbf{r}_{1b}}{r_{1b}} - \frac{\mathbf{r}_{2a}}{r_{2a}}\right)\frac{(\mathbf{r}_1-\mathbf{r}_2)}{r_{12}}\right] \mathrm{e}^{\alpha r_{12}/u - \kappa r_{1b} - \kappa r_{2a}}\psi^{-1} \\
& + (1-C)\left[\frac{\kappa}{r_{1a}} + \frac{\kappa}{r_{2a}} + \frac{\kappa\alpha}{u^2}\left(\frac{\mathbf{r}_{1a}}{r_{1a}} - \frac{\mathbf{r}_{2a}}{r_{2a}}\right)\frac{(\mathbf{r}_1-\mathbf{r}_2)}{r_{12}}\right] \mathrm{e}^{\alpha r_{12}/u - \kappa r_{1a} - \kappa r_{2a}}\psi^{-1} \\
& + (1-C)\left[\frac{\kappa}{r_{1b}} + \frac{\kappa}{r_{2b}} + \frac{\kappa\alpha}{u^2}\left(\frac{\mathbf{r}_{1b}}{r_{1b}} - \frac{\mathbf{r}_{2b}}{r_{2b}}\right)\frac{(\mathbf{r}_1-\mathbf{r}_2)}{r_{12}}\right] \mathrm{e}^{\alpha r_{12}/u - \kappa r_{1b} - \kappa r_{2b}}\psi^{-1}.
\end{aligned}
\tag{24.57}
$$

In the bonding region the energy is lowered by further 0.01 au with a minimum value of -1.16 au (Fig. 24.14). This effect is small as part of the correlation is already included in the two-determinant ansatz. More sophisticated trial functions or larger CI expansions give –1.174 a.u. quite close to the exact value [333].

24.2 Exciton-Phonon Coupling in Molecular Aggregates

In this section we simulate excitons in a molecular aggregate which are coupled to internal vibrations of the molecular units. Molecular aggregates are of considerable interest for energy transfer in artificial [338] and biological systems [339]. Even

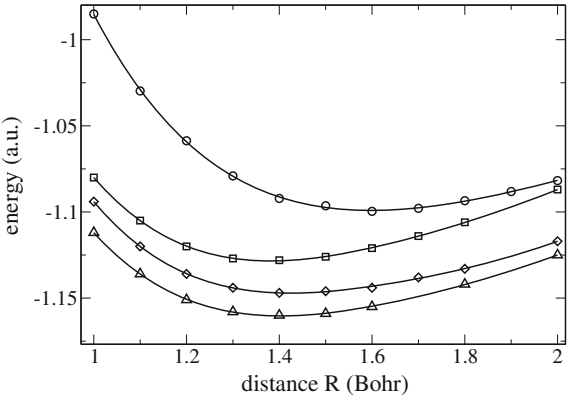


Fig. 24.14 (Optimization of the Slater-Jastrow wavefunction for H_2) Optimization of the Jastrow factor lowers the energy by further 0.01 au (*triangles*). *Circles* show the MO energy, squares the MO energy with optimized exponent κ and diamonds the optimized MCSCF energies as in Fig. 24.13

simple trial functions involve a large number of parameters which have to be optimized and require efficient strategies to minimize the energy. We consider a finite periodic system like in the light harvesting complex of photosynthesis. An optical excitation on the n-th molecule is denoted by the state $|n>$. It can be transferred to the neighboring molecules by the excitonic coupling V and is coupled to the vibrational coordinate q_n. (For simplicity, we consider only one internal vibration per molecule). The model Hamiltonian reads in dimensionless units (periodic b.c. imply that $|0> \equiv |N>$ and $|N+1> \equiv |1>$)

$$H = \sum_{mn} |m> H_{mn} <n|$$
$$= \frac{\lambda^2}{2} + \sum_n \left(-\frac{1}{2}\frac{\partial^2}{\partial q_n^2} + \frac{1}{2} q_n^2 \right) + \sum_{n=1}^{N} |n> \lambda q_n <n| + |n> V <n+1| + |n> V <n-1|. \tag{24.58}$$

Due to the N-fold degeneracy of the excited states, a simple Born-Oppenheimer wavefunction is not adequate. Instead we consider a sum of N Born-Oppenheimer products

$$\Psi = \sum_n |n> \Phi_n(q_1, \ldots q_N). \tag{24.59}$$

We use the variational principle to approximate the lowest eigenstate. Obviously, the number of variational parameters will rapidly increase with the system size. Even if we introduce only one parameter for each unit, e.q. a shift of the potential minimum, this requires N^2 parameters for the aggregate.

The Hamiltonian (24.58) can be brought to a more convenient form by a unitary transformation with

$$S = \sum_n |n > G^n < n| \tag{24.60}$$

where the translation operator G transforms the nuclear coordinates according to

$$G q_n G^{-1} = q_{n+1}. \tag{24.61}$$

The transformed Hamiltonian then reads

$$= \frac{\lambda^2}{2} + \sum_n \left(-\frac{1}{2}\frac{\partial^2}{\partial q_n^2} + \frac{1}{2} q_n^2 \right) + \lambda q_0 + |n > V G < n+1| + |n > V G^{-1} < n-1|. \tag{24.62}$$

Delocalized exciton states

$$|k >= \frac{1}{\sqrt{N}} \sum_n \mathrm{e}^{\mathrm{i}kn} |n > \tag{24.63}$$

transform the Hamiltonian into N independent exciton modes

$$\tilde{H} = \sum_k |k > H_k < k| \tag{24.64}$$

with

$$H_k = \frac{\lambda^2}{2} + \sum_n \left(-\frac{1}{2}\frac{\partial^2}{\partial q_n^2} + \frac{1}{2} q_n^2 \right) + \lambda q_0 + V \mathrm{e}^{\mathrm{i}k} G + V \mathrm{e}^{-\mathrm{i}k} G^{-1}. \tag{24.65}$$

Hence, we conclude that the eigenfunctions of the Hamiltonian H have the general form

$$\Psi = \frac{1}{\sqrt{N}} \sum_n \mathrm{e}^{\mathrm{i}kn} |n > G^n \Phi_k$$

where Φ_k is an eigenfunction of H_k and the number of parameters has been reduced by a factor of N (since for each k, only one function Φ_k is involved).

In the following we study the lowest exciton state, which for $V < 0$ is the lowest eigenfunction[3] for $k = 0$. Hence, the wavefunction of interest has the form

$$\Psi = \frac{1}{\sqrt{N}} \sum_n |n > G^n \Phi \tag{24.66}$$

and can be chosen real valued.

[3]This is the case of the so called J-aggregates [338] for which the lowest exciton state is strongly allowed.

24.2.1 Molecular Dimer

To begin with, let us consider a dimer (N = 2) consisting of two identical molecules in a symmetric arrangement. The model Hamiltonian reads in matrix form

$$H = -\frac{1}{2}\frac{\partial^2}{\partial q_1^2} - \frac{1}{2}\frac{\partial^2}{\partial q_2^2} + \begin{bmatrix} \frac{1}{2}(q_1+\lambda)^2 + \frac{1}{2}q^2 & V \\ V & \frac{1}{2}q_1^2 + \frac{1}{2}(q_2+\lambda)^2 \end{bmatrix} \tag{24.67}$$

and can be considerably simplified by introducing delocalized vibrations

$$q_\pm = \frac{q_1 \pm q_2}{\sqrt{2}} \tag{24.68}$$

which separates the symmetric mode q_+

$$H = \left(-\frac{1}{2}\frac{\partial^2}{\partial q_+^2} + \frac{1}{2}(q_+ + \frac{\lambda}{\sqrt{2}})^2\right) - \frac{1}{2}\frac{\partial^2}{\partial q_-^2} + \begin{bmatrix} \frac{1}{2}\left(q_- + \frac{\lambda}{\sqrt{2}}\right)^2 & V \\ V & \frac{1}{2}\left(q_- - \frac{\lambda}{\sqrt{2}}\right)^2 \end{bmatrix}. \tag{24.69}$$

The lowest eigenfunction of the symmetric oscillation is

$$\Phi_{0+} = \pi^{-1/4} \exp\left\{-\frac{1}{2}(q_+ + \frac{\lambda}{\sqrt{2}})^2\right\} \tag{24.70}$$

with the eigenvalue (the zero point energy)

$$E_{0+} = \frac{1}{2}. \tag{24.71}$$

Hence, for the dimer we may consider a simplified Hamiltonian with only one vibration

$$H = \frac{1}{2} - \frac{1}{2}\frac{\partial^2}{\partial q^2} + \begin{bmatrix} \frac{1}{2}\left(q + \frac{\lambda}{\sqrt{2}}\right)^2 & V \\ V & \frac{1}{2}\left(q - \frac{\lambda}{\sqrt{2}}\right)^2 \end{bmatrix}. \tag{24.72}$$

According to (24.66) the $k = 0$ eigenstates have the form

$$\Psi = \frac{1}{\sqrt{2}}\Phi|1> + \frac{1}{\sqrt{2}}G\Phi|2> = \frac{1}{\sqrt{2}}\Phi(q)|1> + \frac{1}{\sqrt{2}}\Phi(-q)|2>. \tag{24.73}$$

For the dimer problem, the eigenstates can be calculated numerically by diagonalization of the Hamiltonian equation (24.72) in the basis of harmonic oscillator states [340].

Dressed Exciton

The simplest trial function which is well known as "dressed exciton" or "mean field" ansatz [341, 342], uses a Gaussian representing the groundstate of a displaced (and possibly distorted) harmonic oscillator

$$\Phi = \left(\frac{2\kappa}{\pi}\right)^{1/4} \mathrm{e}^{-\kappa(q_- + \alpha)^2} \quad G\Phi = \left(\frac{2\kappa}{\pi}\right)^{1/4} \mathrm{e}^{-\kappa(q_- - \alpha)^2} \tag{24.74}$$

with the energy expectation value

$$E_{MF}(\kappa, \lambda) = < \Psi H \Psi > = \frac{1}{2} + (\frac{1}{8\kappa} + \frac{\kappa}{2}) + \frac{1}{2}(\alpha - \frac{\lambda}{\sqrt{2}})^2 + V\mathrm{e}^{-2\kappa\alpha^2} \tag{24.75}$$

for which the first and second derivatives are easily found

$$\frac{\partial E_{MF}}{\partial \alpha} = \alpha - \frac{\lambda}{\sqrt{2}} - 4V\alpha\kappa\mathrm{e}^{-2\kappa\alpha^2} \tag{24.76}$$

$$\frac{\partial E_{MF}}{\partial \kappa} = \frac{1}{2} - \frac{1}{8\kappa^2} - 2V\alpha^2\mathrm{e}^{-2\kappa\alpha^2} \tag{24.77}$$

$$\frac{\partial^2 E_{MF}}{\partial \alpha^2} = 1 + 4V\kappa\mathrm{e}^{-2\kappa\alpha^2}\left[4\kappa\alpha^2 - 1\right] \tag{24.78}$$

$$\frac{\partial^2 E_{MF}}{\partial \kappa^2} = \frac{1}{4\kappa^3} + 4V\alpha^4\mathrm{e}^{-2\kappa\alpha^2} \tag{24.79}$$

$$\frac{\partial^2 E_{MF}}{\partial \kappa \partial \alpha} = 4V\alpha\mathrm{e}^{-2\kappa\alpha^2}\left[2\kappa\alpha^2 - 1\right]. \tag{24.80}$$

In the limit of vanishing "dressed" coupling $V\mathrm{e}^{-2\kappa\alpha^2} \approx 0$, corresponding to the so called self trapped state, the lowest energy is found for $\alpha = \lambda/\sqrt{2}, \quad \kappa = 1/2$

$$\min E_{MF}(V\mathrm{e}^{-2\kappa\alpha^2} \to 0) = 1 \tag{24.81}$$

which is the zero point energy of the two dimer modes. In the limit of vanishing exciton-phonon coupling $\lambda = 0$ (the fully delocalized state) the energy is minimized for $\alpha = 0, \quad \kappa = 1/2$

$$\min E_{MF}(\lambda \to 0) = V + 1. \tag{24.82}$$

For the general case we apply the Newton-Raphson method (p. 124) to locate the minimum. It is quite important to use a reasonable starting point to ensure conver-

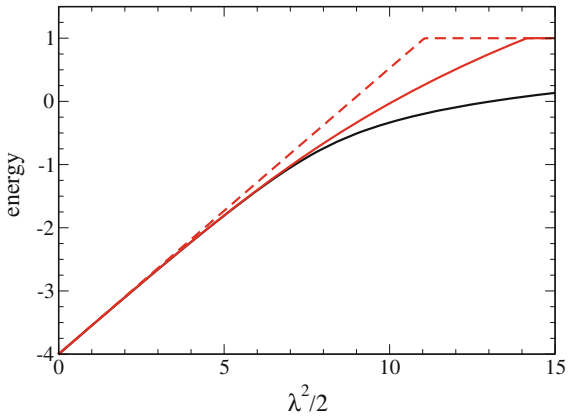


Fig. 24.15 (Variational solutions for the dimer) The lowest excitation energy of the dimer Hamiltonian is shown as a function of the reorganization energy $\lambda^2/2$. The mean field ansatz (*red curves*) predicts a sharp transition to the self-trapped state and deviates largely for $\lambda^2/2 > 5$. Variation of the exponent κ improves the agreement in the transition region considerably (*full red curve*) as compared to the standard treatment with fixed $\kappa = 1/2$ (*dashed red curve*). The *black curve* shows the numerically exact solution for comparison

gence to the lowest energy.[4] In Problem 24.4, we search for an approximate minimum on a coarse grid first. Figure 24.15 shows the calculated energy minimum for strong excitonic coupling $V = -5$ as a function of λ^2. For small values of the exciton-phonon coupling, the numerically exact values are reproduced quite closely. For larger values the mean field ansatz predicts a rapid transition to a so called self-trapped state [343] with $\alpha = \lambda/\sqrt{2}$ and a very small Franck-Condon factor $F = \exp(-2\kappa\alpha^2) \approx 0$ (Figs. 24.16, 24.17). In this region, the deviation from the numerical exact result is appreciable, especially if only α is varied and $\kappa = 1/2$ kept fixed.

Solitonic Solution

In the region of large exciton-phonon coupling a simple ansatz similar to Davydov's soliton [344] is quite successful (Fig. 24.18) which breaks the symmetry of the system and uses a trial function

$$\Psi_{sol} = (\varphi_1|1> + \varphi_2|2>)\Phi_{\alpha_1,\alpha_2}(q_1, q_2) \tag{24.83}$$

with two mixing amplitudes with the constraint

$$\varphi_1^2 + \varphi_2^2 = 1 \tag{24.84}$$

[4] In the transition region, the energy may converge to an unstable state, depending on the starting point.

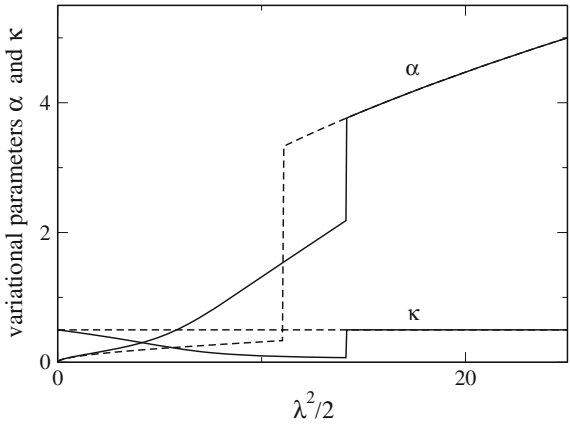


Fig. 24.16 (Optimized parameters of the mean field ansatz) The optimized parameters for $V = -5$ show a sharp transition to the self trapped state. *Full curves* optimization of α and κ. *Dashed curves* optimization of α for fixed $\kappa = 1/2$

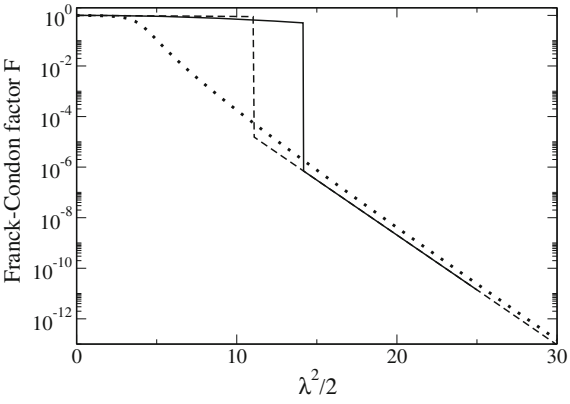


Fig. 24.17 (Franck-Condon factor of the mean field ansatz) The transition to the self trapped state shows also up in the Franck-Condon factor $F = \exp\{-2\kappa\alpha^2\}$ which is shown in a semi-logarithmic plot. *Full curve* optimization of α and κ. *Dashed curve* optimization of α for fixed $\kappa = 1/2$. The *dotted curve* shows the numerically exact result for comparison

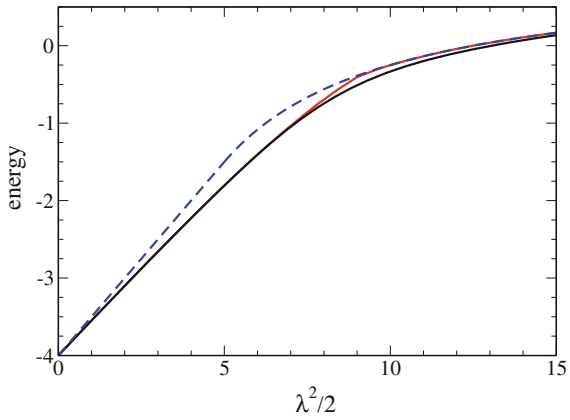


Fig. 24.18 (Variational solutions for the dimer) The soliton approach (*dashed blue curve*) works quite well for large but also for very weak exciton-phonon coupling. The delocalized soliton interpolates between mean field and soliton results and describes the transition quite well (*red curve*). The *black curve* shows the numerically exact solution for comparison

and the same vibrational function for both states (in the self trapped region distortion of the oscillator is not important)

$$\Phi_{\alpha_1,\alpha_2}(q_1,q_2) = \frac{1}{\sqrt{\pi}} \mathrm{e}^{-(q_1+\alpha_1)^2/2} \mathrm{e}^{-(q_2+\alpha_2)^2/2}. \tag{24.85}$$

The energy expectation value is

$$\begin{aligned} E_{sol}(\varphi_1,\varphi_2,\alpha_1,\alpha_2) &= < \Psi H \Psi > \\ &= \varphi_1^2 \left[1 + \frac{(\alpha_1-\lambda)^2}{2} + \frac{\alpha_2^2}{2} \right] + \varphi_2^2 \left[1 + \frac{(\alpha_2-\lambda)^2}{2} + \frac{\alpha_1^2}{2} \right] + 2V\varphi_1\varphi_2 \\ &= 1 + \frac{1}{2}(\alpha_1 - \varphi_1^2\lambda)^2 + \frac{1}{2}(\alpha_2 - \varphi_2^2\lambda)^2 + \lambda^2\varphi_1^2\varphi_2^2 + 2V\varphi_1\varphi_2 \end{aligned} \tag{24.86}$$

and for the optimized values

$$\alpha_i^o = \varphi_i^2 \lambda \tag{24.87}$$

it becomes

$$E_{sol}(\varphi_1,\varphi_2,\alpha_1^o,\alpha_2^o) = 1 + \lambda^2\varphi_1^2\varphi_2^2 + 2V\varphi_1\varphi_2 = 1 + \frac{\lambda^2}{4}\left(2\varphi_1\varphi_2 + \frac{2V}{\lambda^2} \right)^2 - \frac{V^2}{\lambda^2}. \tag{24.88}$$

Alternatively, using symmetrized coordinates we obtain

$$E_{sol}(\varphi_1,\varphi_2,\alpha_+,\alpha_-) = 1 + \frac{1}{2}\left(\alpha_+ - \frac{\lambda}{\sqrt{2}} \right)^2 + \frac{1}{2}\left(\alpha_- - \frac{\lambda}{\sqrt{2}}(\varphi_1^2 - \varphi_2^2) \right)^2 + \lambda^2\varphi_1^2\varphi_2^2 + 2V\varphi_1\varphi_2 \tag{24.89}$$

and optimum values

$$\alpha_+^o = \frac{\lambda}{\sqrt{2}} \tag{24.90}$$

$$\alpha_-^o = \frac{\lambda}{\sqrt{2}}(\varphi_1^2 - \varphi_2^2). \tag{24.91}$$

Since $|2\varphi_1\varphi_2| \leq 1$, the minimum for large exciton-phonon coupling is at the bottom of the parabola

$$\min E_{sol} = 1 - \frac{V^2}{\lambda^2} \quad \text{for } |V| < \lambda^2/2 \tag{24.92}$$

whereas in the opposite case it is found for $2\varphi_1\varphi_2 = 1$ (V is assumed to be negative)

$$\min E_{sol} = 1 + V + \frac{\lambda^2}{4} \quad \text{for } |V| > \lambda^2/2. \tag{24.93}$$

The transition between the two regions is continuous with

$$\min E_{sol} = 1 + \frac{V}{2} \quad \text{for } |V| = \lambda^2/2. \tag{24.94}$$

Delocalized Soliton Ansatz

Mean field and soliton ansatz can be combined by delocalizing the solitonic wave function [345]. The energies of the trial function

$$\Psi_{sol} = (\varphi_1|1> + \varphi_2|2>)\Phi \tag{24.95}$$

and its mirror image

$$\Psi'_{sol} = (\varphi_2|1> + \varphi_1|2>)G\Phi \tag{24.96}$$

are degenerate. Hence delocalization of the trial function

$$\Psi_{delsol} = |1> [\varphi_1\Phi + \varphi_2 G\Phi] + |2> [\varphi_2\Phi + \varphi_1 G\Phi] \tag{24.97}$$

is expected to lower the energy further and ensures the proper form of (24.73). Its norm is

$$< \Psi_{delsol}|\Psi_{delsol} >= 2(1 + 2\varphi_1\varphi_2 F) \tag{24.98}$$

with the Franck-Condon factor

$$F =< \Phi|G|\Phi >= \mathrm{e}^{-\kappa(\alpha_1-\alpha_2)^2} = \mathrm{e}^{-2\kappa\alpha_-^2}. \tag{24.99}$$

The expectation value of the Hamiltonian simplifies due to symmetry

$$< \Psi_{delsol} H \Psi_{delsol} >= 2\varphi_1^2 < \Phi H_1 \Phi > +2\varphi_2^2 < \Phi H_2 \Phi > +4\varphi_1\varphi_2 < \Phi H_1 G\Phi >$$

$$+2V\,[F + 2\varphi_1\varphi_2]\,. \tag{24.100}$$

Finally, varying only the antisymmetric mode, the energy is

$$E_{delsol} = \frac{\kappa}{2} + \frac{1}{8\kappa} + \frac{\lambda^2}{2} + \frac{\frac{\alpha_-^2}{2} - (\varphi_1^2 - \varphi_2^2)\alpha_-\lambda + 2\varphi_1\varphi_2\left[-2F\kappa^2\alpha_-^2 + V\right] + VF}{1 + 2\varphi_1\varphi_2 F}. \tag{24.101}$$

In Problem 24.5 we locate the minimum energy with the steepest descent (6.2.5) or the conjugate gradient (6.2.5) method

24.2.2 Larger Aggregates

The variational methods for the dimer can be generalized for larger systems [345]. The mean field ansatz for the lowest k = 0 state becomes

$$\Psi_{MF} = \frac{1}{\sqrt{N}} \sum_n |n> G^n \Phi \tag{24.102}$$

with

$$\Phi = \prod_{n=1}^{N} \pi^{-1/4} \mathrm{e}^{-(q_n+\alpha_n)^2/2}. \tag{24.103}$$

The energy is

$$\begin{aligned} E_{MF} &= \frac{1}{N} \sum_n < \Phi G^{-n} H_n G^n \Phi > + V < \Phi G \Phi > + V < \Phi G^{-1} > \\ &= < \Phi H_0 \Phi > + 2VF \\ &= \frac{N}{2} + \frac{\lambda^2}{2} - \alpha_0 \lambda + \sum_n \frac{\alpha_n^2}{2} + 2VF \end{aligned} \tag{24.104}$$

and its gradient

$$\frac{\partial E_{MF}}{\partial \alpha_n} = -\lambda \delta_{n,0} + \alpha_n - VF(2\alpha_n - \alpha_{n+1} - \alpha_{n-1}).$$

In Problem 24.5 we locate the minimum energy with the steepest descent (6.2.5) or the conjugate gradient (6.2.5) method. As for the dimer, the mean field method shows a rapid transition to the self-trapped state. The starting point is quite important as in the vicinity of the transition the gradient based methods eventually converge to a metastable state (Fig. 24.19).

The soliton wavefunction (corresponding to Davydov's D1 soliton) for the aggregate is

$$\Psi_{sol} = \sum_n \varphi_n |n> \Phi \tag{24.105}$$

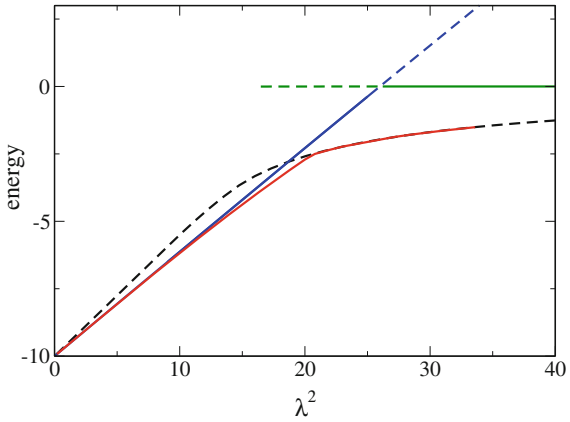


Fig. 24.19 (Variational solutions for a 10-mer) The lowest energy of a periodic 10-mer is calculated for $V = -5$ (see Problem 24.5). The mean field wavefunction gives (*green* and *blue curves*) reasonable values for small values of λ^2 and predicts a rapid transition to the self trapped state. Approaching the transition from below or above the calculation may end up in a metastable state (*dashed green* and *blue curves*). The solitonic wavefunction (*dashed black curve*) provides lower energies at larger values of λ^2 and a much smoother transition to the self trapped state. The delocalized soliton (*red curve*) gives the lowest energy at all values of λ^2. The zero point energy has been subtracted

with the constraint

$$\sum_n \varphi_n^2 = 1 \tag{24.106}$$

where Φ is given by (24.103). The energy is

$$\begin{aligned} E_{sol} &= <\Psi_{sol}|H|\Psi_{sol}> = \sum_n \varphi_n^2 < \Phi|H_n|\Phi > + V\sum_n(\varphi_n\varphi_{n+1} + \varphi_n\varphi_{n-1}) \\ &= \frac{N}{2} + \sum_n \frac{\alpha_n^2}{2} + \frac{\lambda^2}{2} - \lambda\sum_n \varphi_n^2\alpha_n + V\sum_n(\varphi_n\varphi_{n+1} + \varphi_n\varphi_{n-1}) \\ &= \frac{N}{2} + \sum_n \frac{(\alpha_n - \lambda\varphi_n^2)^2}{2} + \frac{\lambda^2}{2}\left(1 - \sum_n \varphi_n^4\right) + V\sum_n(\varphi_n\varphi_{n+1} + \varphi_n\varphi_{n-1}) \end{aligned} \tag{24.107}$$

with the optimum displacements

$$\alpha_n^o = \lambda\varphi_n^2. \tag{24.108}$$

The energy functional becomes

$$E_{sol}(\varphi_n, \alpha_n^o) = \frac{N}{2} + \frac{\lambda^2}{2}\left(1 - \sum_n \varphi_n^4\right) + 2V \sum_n \varphi_n \varphi_{n+1} \tag{24.109}$$

and its gradient

$$\frac{\partial E_{sol}}{\partial \varphi_n} = -2\lambda^2 \varphi_n^3 + 2V(\varphi_{n-1} + \varphi_{n+1}). \tag{24.110}$$

In Problem 24.5 we locate the minimum energy by varying the φ_n under the constraint (24.106). At larger exciton-phonon coupling, the energy of the soliton wavefunction is much lower in energy than the mean field result and the transition to the self-trapped state is smoother. At small exciton-phonon coupling, the mean field ansatz is lower in energy (Fig. 24.19).

Similar to the dimer case, the solitonic wavefunction can be delocalized by combining the N degenerate mirror images

$$\sum_n \varphi_n |n+m> G^m \Phi \quad m = 1 \cdots N \tag{24.111}$$

into the trial function

$$\Psi_{delsol} = \frac{1}{\sqrt{N}} \sum_m \mathrm{e}^{\mathrm{i}km} \sum_n \varphi_n |n+m> G^m \Phi$$

$$= \frac{1}{\sqrt{N}} \sum_{nn'} \mathrm{e}^{\mathrm{i}k(n'-n)} \varphi_n |n'> G^{n'-n} \Phi = \frac{1}{\sqrt{N}} \sum_{n'} \mathrm{e}^{\mathrm{i}kn'} |n'> G^{n'} \sum_n \mathrm{e}^{-\mathrm{i}kn} \varphi_n G^{-n} \Phi. \tag{24.112}$$

From the squared norm

$$< \Psi_{delsol} | \Psi_{delsol} > = \sum_{nn'} \varphi_n \varphi_{n'} < \Phi G^{n-n'} \Phi > = \sum_{nn'} \varphi_n \varphi_{n'} F_{n-n'} \tag{24.113}$$

and the expectation value

$$< \Psi_{delsol} H \Psi_{delsol} > = \frac{1}{N} \sum_{m,n,m',n'} < \Phi G^{-m} < n+m | \varphi_n H \varphi_{n'} | m'+n' > G^{m'} \Phi >$$

$$= \frac{1}{N} \sum_{m,n,m',n'} < \Phi G^{-m} \varphi_n H_{n+m} \varphi_{n'} G^{m+n-n'} \Phi >$$

$$+ \frac{1}{N} \sum_{m,n,m',n'} V < \Phi G^{-m} \varphi_n \varphi_{n'} G^{m+n-n'+1} \Phi >$$

$$+\frac{1}{N}\sum_{m,n,n'} V < \Phi G^{-m}\varphi_n\varphi_{n'}G^{m+n-n'-1}\Phi >$$
$$=\sum_{n,n'}\varphi_n\varphi_{n'} < \Phi G^n H_0 G^{-n'}\Phi >$$
$$+V\sum_{n,n'}\varphi_n\varphi_{n'} < \Phi G^{n-n'+1}\Phi > +V\sum_{n,n'}\varphi_n\varphi_{n'} < \Phi G^{n-n'-1}\Phi > \tag{24.114}$$

we obtain the energy of the $k = 0$ state

$$E_{delsol} = \frac{N}{2} + \frac{\lambda^2}{2}$$
$$+\left(\sum_{nn'}\varphi_n\varphi_{n'}\frac{1}{2}\left[-\lambda(\alpha_{n'}+\alpha_{n'-n}) + \sum_m \alpha_m\alpha_{m+n'-n}\right]F_{n'-n}\right.$$
$$\left.+V\sum_{nn'}\varphi_n\varphi_{n'}(F_{n'-n+1}+F_{n'-n-1})\right)\left(\sum_{nn'}\varphi_n\varphi_{n'}F_{n-n'}\right)^{-1} \tag{24.115}$$

with the Franck-Condon factors

$$F_k =< \Phi G^k\Phi >= \mathrm{e}^{-\sum_m(\alpha_m-\alpha_{m+k})^2/4} = \mathrm{e}^{-\sum_m(\alpha_m^2-\alpha_m\alpha_{m+k})/2}. \tag{24.116}$$

The results for longer aggregates are qualitatively similar to the dimer. The delocalized soliton interpolates between mean field and soliton wave functions and shows a smooth transition (Fig. 24.19).

Problems

In the first three computer experiments, we use the variational quantum Monte Carlo method to calculate the groundstate energy. The Metropolis algorithm with N_w walkers is used to evaluate the integral

$$E(\kappa,R) = \frac{<\psi_\kappa H\psi_\kappa>}{<\psi_\kappa\psi_\kappa>} = \int d^3r \quad \frac{|\psi_\kappa(\mathbf{r})|^2}{\int|\psi_\kappa(\mathbf{r}')|^2d^3r'}E_{loc}(\mathbf{r}).$$

Adjust the maximum trial step to obtain an acceptance ration of about 1 and study the influence of the number of walkers on the statistical error.

Problem 24.1

Optimize the effective nuclear charge κ for the hydrogen molecular ion H_2^+ as a function of R and determine the equilibrium bond length. The trial function has the form

$$\psi_{trial} = \sqrt{\frac{\kappa^3}{\pi}} e^{-\kappa r_a} + \sqrt{\frac{\kappa^3}{\pi}} e^{-\kappa r_b}.$$

Problem 24.2

For the Helium atom we use a trial wavefunction of the Slater-Jastrow type

$$\psi_{trial} = e^{-\kappa r_1} e^{-\kappa r_2} e^{\alpha r_{12}/(1+\beta r_{12})} \frac{1}{\sqrt{2}} (\uparrow (1) \downarrow (2) - \uparrow (2) \downarrow (1))$$

to find the optimum parameters α, β, κ.

Problem 24.3

In this computer experiment we study the hydrogen molecule H_2. The trial function has the form

$$\psi_{trial} = \left\{ C \left[e^{-\kappa r_{1a} - \kappa r_{2b}} + e^{-\kappa r_{1b} - \kappa r_{2a}} \right] + (1 - C) \left[e^{-\kappa r_{1a} - \kappa r_{2a}} + e^{-\kappa r_{1b} - \kappa r_{2b}} \right] \right\} \times \exp \left\{ \frac{\alpha r_{12}}{1 + \beta r_{12}} \right\}.$$

Optimize the parameters κ, β, C as a function of R and determine the equilibrium bond length.

Problem 24.4

In this computer experiment we simulate excitons in a molecular dimer coupled to molecular vibrations. The energy of the lowest exciton state is calculated with the dressed exciton trial function including a frequency change of the vibration

$$\psi_{trial} = \frac{1}{\sqrt{2}} |1> \left(\frac{2\kappa}{\pi} \right)^{1/4} e^{-\kappa (q_- + \alpha)^2} + \frac{1}{\sqrt{2}} |2> \left(\frac{2\kappa}{\pi} \right)^{1/4} e^{-\kappa (q_- - \alpha)^2}.$$

The parameters κ, α are optimized with the Newton-Raphson method. Vary the exciton coupling V and the reorganization energy $\lambda^2/2$ and compare with the numerically exact values.

Problem 24.5

In this computer experiment we simulate excitons in a molecular aggregate coupled to molecular vibrations. The energy of the lowest exciton state is calculated with different kinds of trial functions

- the dressed exciton

$$\Psi_{MF} = \frac{1}{\sqrt{N}} \sum_n |n> G^n \prod_{n=1}^{N} \pi^{-1/4} \mathrm{e}^{-(q_n+\alpha_n)^2/2}$$

- the soliton

$$\Psi_{sol} = \sum_n \varphi_n |n> \prod_{n=1}^{N} \pi^{-1/4} \mathrm{e}^{-(q_n+\alpha_n)^2/2}$$

- the delocalized soliton

$$\Psi_{delsol} = \frac{1}{\sqrt{N}} \sum_m \sum_n \varphi_n |n+m> G^m \prod_{n=1}^{N} \pi^{-1/4} \mathrm{e}^{-(q_n+\alpha_n)^2/2}.$$

The system size can be varied from a dimer (N=2) up to chains of 100 molecules. The N equilibrium shifts α_n and the N excitonic amplitudes φ_n are optimized with the methods of steepest descent or conjugate gradients. The optimized parameters are shown graphically. Vary the exciton coupling V and the reorganization energy $\lambda^2/2$ and study the transition from a delocalized to a localized state. Compare the different trial functions.

Appendix A: Performing the Computer Experiments

The computer experiments are realized as Java programs which can be run on any platform if a Java runtime environment (JRE) is installed. They are written in a C-like fashion which improves the readability for readers who are not so familiar with object oriented programming. The source code can be studied most conveniently with the netbeans environment which is open source and allows quick generation of graphical user interfaces. The screenshot in Fig. A.1 shows an example.

After downloading and unzipping the zipped file from **extras.springer.com** you have two options.

Run a Program Directly

Open the directory CP-examples in your file manager. If the JRE is installed properly you can start any one of the programs by simply clicking onto it. Under Linux, you can alternatively start it in a console window with e.g.

```
java -jar CPexample.jar
```

Figure A.2 shows a screenshot from computer exercise 23.4 (ladder model for exponential decay).

Open a Program with the Netbeans Environment

If you have the netbeans environment installed, you can import any of the programs as a separate project by opening the corresponding folder in the directory CP-examples/NBprojects/. You may have a look at the source code and compile and run it

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7

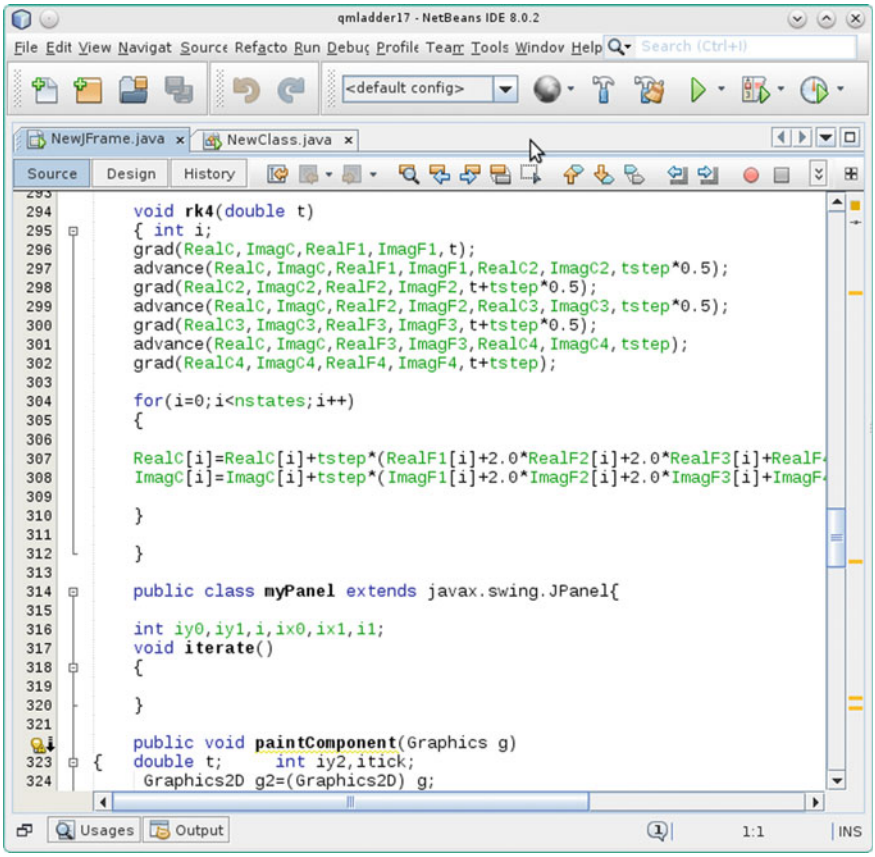


Fig. A.1 Screenshot of the source code

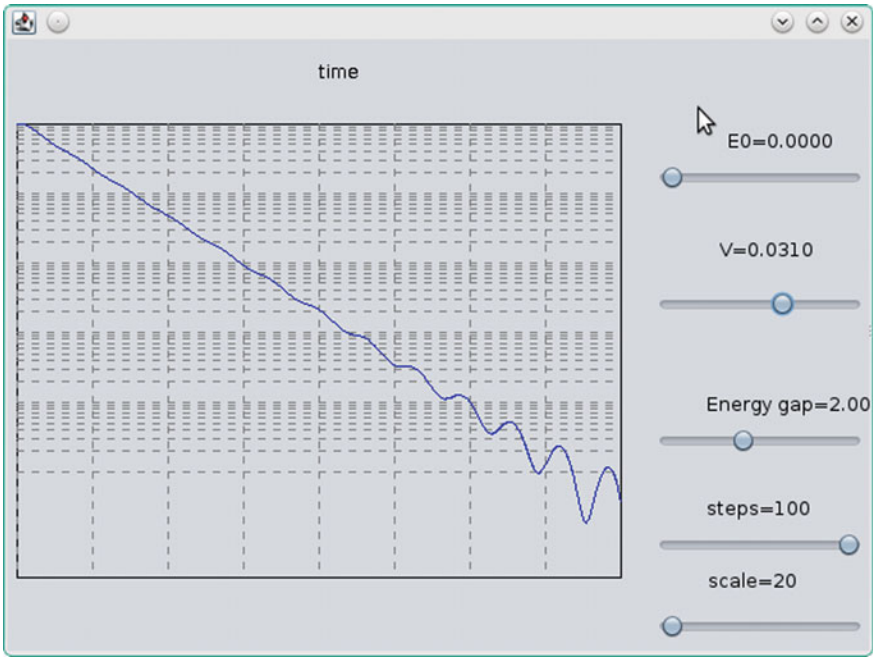


Fig. A.2 Screenshot of computer experiment 23.4

Appendix B: Methods and Algorithms

Purpose	Method	Comments	Pages
Interpolation	Lagrange polynomial	Explicit form, easy to evaluate	19
	Barycentric Lagrange polynomial	For evaluation at many points	19
	Newton's divided differences	New points added easily	21
	Neville method	For evaluation at one point	22
	Spline interpolation	Smoother, less oscillatory	22
	Rational interpolation	Smoother, less oscillatory, often less coefficients necessary	28, 32
	Pade approximation	Often better than Taylor series	29
	Barycentric rational interpolation	Easy to evaluate	30
	Rational interpolation without poles	Alternative to splines, analytical	34
	Multivariate interpolation	Multidimensional	35
	Trigonometric interpolation	Periodic functions	132
Differentiation	One-sided difference quotient	Low error order	39
	Central difference quotient	Higher error order	41
	Extrapolation	High accuracy	41
	Higher derivatives	Finite difference methods	43
	Partial derivatives	Finite difference methods	45

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7

Purpose	Method	Comments	Pages
Integration	Newton-Cotes formulas	Equally spaced points	49
	Trapezoidal rule	Simple, closed interval	49
	Midpoint rule	Simple, open interval	50
	Simpson's rule	More accurate	49
	Composite Newton-Cotes rules	For larger intervals	50
	Extrapolation (Romberg)	High accuracy	51
	Clenshaw-Curtis expressions	Suitable for adaptive and multidimensional quadrature	53
	Gaussian integration	High accuracy if polynomial approximation possible	53
	Monte Carlo integration	High dimensional integrals	202
Linear equations	Gaussian elimination (LU reduction)	Standard method for linear equations and matrix inversion	64
	QR decomposition	Numerically more stable	69
	Iterative solution	Large sparse systems	78
	Richardson iteration	Simplest iterative method	79
	Jacobi relaxation	Iterative matrix-splitting method, converges for diagonally dominant matrices, parallel computation possible	80
	Gauss-Seidel relaxation	Iterative matrix-splitting method, converges for symmetric positive definite or diagonal dominant matrices, no extra storage	81
	Chessboard (black-red)	Two independent subgrids, especially for Poisson equation	402
	Damping and Successive over-relaxation	Speeds up convergence for proper relaxation parameter	81

Purpose	Method	Comments	Pages
	Multigrid method	Fast convergence but more complicated	402
	Conjugate gradients method (CG)	Krylov space method for symmetric positive definite matrices, preconditioning often necessary	86
	General minimum residual method (GMRES)	Krylov space method for nonsymmetric systems	89
	special LU decomposition	Tridiagonal linear equations	75
	Sherman-Morrison formula	Cyclic tridiagonal systems	77
Root finding	Bisection	Reliable but slow continuous functions	98
	Regula falsi (false position)	Speed and robustness between bisection and interpolation	99
	Newton-Raphson	Continuous derivative necessary, converges fast if starting point is close to a root	100
	Interpolation (secant)	No derivative necessary, but slower than Newton	101
	Inverse interpolation	Mainly used by combined methods	102
	Dekker's combined method	Combination of bisection and secant method	106
	Brent's combined method	Combination of bisection, secant, and quadratic inverse interpolation methods, very popular	107
	Chandrupatla's combined method	Uses quadratic interpolation whenever possible, faster than Brent's method, especially for higher order roots	109
Multidimensional root finding	Newton-Raphson	Needs full Hessian	124
	Quasi-Newton (Broyden)	Hessian not needed, no matrix inversion	125
Function Minimization	Ternary search	No gradient needed, very simple, for unimodal functions	115
	Golden section search (Brent)	Faster than ternary search but more complicated	116

Purpose	Method	Comments	Pages
Multidimensional minimization	Steepest descent	Simple but slow	122
	Conjugate gradients	Faster than steepest descent	124
	Newton-Raphson	Fast, if starting point close to minimum, needs full Hessian	124
	Quasi-Newton (BFGS, DFP)	Hessian not needed, very popular	125
Fourier transformation	Görtzel's algorithm	Efficient if only some Fourier components are needed	136
	Fast Fourier transform	Much faster than direct discrete Fourier transform	138
Time-Frequency Analysis	Short Time Fourier Transform (STFT)	Constant resolution for all frequencies, often used for audio signals	145
	Gabor transform	STFT with Gaussian window represents signal by elementary signals localized in time and frequency	156
	Discrete STFT	Reduced redundancy, still invertible	153
	Continuous Wavelet transform	Constant relative frequency resolution, better time resolution for high frequencies,very time consuming convolution integral	158
	Discrete Wavelet Transform	Uses orthogonal or biorthogonal wavelets, fast scalar product	
	Multiresolution analysis	Represents a signal by a basic approximation and a series of details with increasing resolution	164
	Fast wavelet transform	Recursive filtering, very fast	178
Random numbers	Linear congruent mapping (LC)	Simple pseudo-random number generator	197
	Xorshift	Fast, maximum possible period	197
	Multiply with carry (MWC)	Similar to LC but uses a varying carry	198
	Complementary multiply with carry (CMWC)	Improves MWC, passes many tests	199

Purpose	Method	Comments	Pages
	RN with given distribution	Inverse of cumulative distribution function needed	199
	Random points on unit sphere	Random directions	200
	Gaussian RN (Box-Muller)	Gaussian random numbers	201
Thermodynamic average	Simple sampling	Inefficient	206
	Importance sampling	Samples preferentially important configurations	207
	Metropolis algorithm	Generates configurations according to a canonical distribution	207
Eigenvalue problems	Direct solution	Only for very small dimension	214
	Tridiagonal matrices	Explicit solutions for some special tridiagonal matrices	217
	Jacobi	Simple but not very efficient	214
	Power iteration	Finds dominant eigenvector	225
	QL and QR	Efficient power iteration method for not too large matrices, especially in combination with tridiagonalization by Householder transformations	228
	Lanczos	Iterative method for very large matrices or if only a few eigenvalues are needed	230
	Singular value decomposition (SVD)	Generalization for arbitrary matrices	242
Data fitting	Least square fit	Fit a model function to a set of data	236
	Linear least square fit with normal equations	Simple but less accurate	237
	Linear fit with orthogonalisation	Better numerical stability	239
	Linear fit with SVD	Expensive but more reliable, also for rank deficient matrices	248
	Low rank matrix approximation	Data compression, total linear least squares	245

Purpose	Method	Comments	Pages
Discretization	Method of lines	Continuous time, discretized space	261
	Eigenvector expansion		
	Finite differences	Simplest discretization, uniform grids	259
	Finite volumes	Partial differential equations with a divergence term (conservation laws), flux conservative, allows unstructured meshes and discontinuous material parameters	265
	Finite elements	Very flexible and general discretization method but also more complicated	277
	Spectral methods	Expansion with global basis functions, mostly polynomials and Fourier sums, less expensive than finite elements but not as accurate for discontinuous material parameters and complicated geometries	273
	Dual grid	For finite volumes	265, 409
	Weighted residuals	General method to determine the expansion coefficients	270
	Point collocation	Simplest criterion, often used for nonlinear problems and spectral methods	271
	Sub-domains	More general than finite volumes	271
	Least square	Popular for computational fluid dynamics and electrodynamics	272
	Galerkin	Most widely used criterion, leads often to symmetric matrices	273
	Fourier pseudo-spectral method	Very useful whenever a Laplacian is involved, reduces dispersion	273
	Boundary elements	If the Green's function is available	286

Purpose	Method	Comments	Pages
Time evolution	Explicit forward Euler	Low error order and unstable, mainly used as predictor step	292
	Implicit backward Euler	Low error order but stable, used for stiff problems and as corrector step	295
	Improved Euler (Heun, predictor-corrector)	Higher error order	296
	Nordsieck predictor-corrector	Implicit method, has been used for molecular dynamics	298
	Gear predictor-corrector	Optimized for molecular dynamics	300
	Explicit Runge Kutta (2nd, 3rd, 4th)	General and robust methods, easy step size and quality control	301
	Extrapolation (Gragg-Bulirsch-Stoer)	Very accurate and very slow	305
	Explicit Adams-Bashforth	High error order but not self-starting, for smooth functions, can be used as predictor	306
	Implicit Adams-Moulton	Better stability than explicit method, can be used as corrector	306
	Backward differentiation (Gear)	Implicit, especially for stiff problems	307
	Linear multistep predictor-corrector	General class, includes Adams-Bashforth-Moulton and Gear methods	309
	Verlet integration	Symplectic, time reversible, for molecular dynamics	310
	Position Verlet	Less popular	312
	Velocity Verlet	Often used	313
	Stoermer-Verlet	If velocities are not needed	313
	Beeman's method	Velocities more accurate than for Stoermer-Verlet	315
	Leapfrog	Simple but two different grids	317, 317, 471
	Crank-Nicolson	Implicit, stable, diffusion and Schroedinger equation	486, 474
	FTBS, Lax-Friedrich	simple methods for advection	434, 436
	Lax-Wendroff	Hyperbolic differential equations	472
	Taylor-Galerkin Lax-Wendroff	highly accurate for advection	449
	Two-step	Differential equation with second order time derivative	464

Purpose	Method	Comments	Pages
	Reduction to a first order equation	Derivatives treated as additional variables	467
	Two-variable	Transforms wave equation into a system of two first order equations	470
	Split operator	Approximates an operator by a product	490, 311, 533
Unitary time evolution	Rational approximation	Implicit,unitary	526
	Second order differencing	Explicit, not exactly unitary	530
	Split operator Fourier	Low dispersion, needs fast Fourier transformation	533
	Real space product formula	Fast but less accurate, useful for wavepackets in coupled states	534
Rotation	Reorthogonalization	Restore orthogonality of rotation matrix	293
	Quaternions	Optimum parametrization of the rotation matrix	343
	Euler angles	Numerical singularities	343
	Explicit method	Low accuracy, reorthogonalization needed	335
	Implicit method	Higher accuracy, orthogonal transformation	338
Molecular dynamics	Force field gradients	Needed for molecular dynamics	361
	Normal mode analysis	Small amplitude motion around an equilibrium	364
	Behrendsen thermostat	Simple method to control temperature	371
	Langevin dynamics	Brownian motion	395
Many body quantum systems	Variational Quantum Monte-Carlo method (VQMC)	Calculates energy for non separable trial wavefunctions	205, 577

References

1. IEEE 754-2008: Standard for Floating-Point Arithmetic, IEEE Standards Association (2008)
2. J. Stoer, R. Bulirsch, *in Introduction to Numerical Analysis*, 3rd edn. (Springer, New York, 2010). ISBN 978-1441930064
3. H. Jeffreys, B.S. Jeffreys, in *Lagrange's Interpolation Formula. §9.011 in: Methods of Mathematical Physics*, 3rd ed. (Cambridge University Press, Cambridge, 1988), p. 260
4. J.-P. Berrut, L.N. Trefethen, Barycentric Lagrange interpolation. SIAM Rev. **46**(3), 501–517 (2004)
5. H. Jeffreys, B.S. Jeffreys, in *Divided Differences §9.012 in: Methods of Mathematical Physics*, 3rd edn. (Cambridge University Press, Cambridge, 1988), pp. 260–264
6. W. Werner, Math. Comput. **43**, 205 (1984)
7. E.H. Neville, Indian Math. Soc. **20**, 87 (1933)
8. C.T.H. Baker, Numer. Math. **15**, 315 (1970)
9. I.J. Schoenberg, Q. Appl, Math. **4**(45–99), 112–141 (1946)
10. G. Nürnberger, *in Approximation by Spline Functions* (Springer, Berlin, 1989). ISBN 3-540-51618-2
11. M.S. Floater, K. Hormann, Numer. Math. **107**, 315 (2007)
12. J.P. Berrut, R. Baltensperger, H.D. Mittelmann, Int. Ser. Numer. Math. **151**, 27 (2005)
13. C. Schneider, W. Werner, Math. Comput. **47**, 285 (1986)
14. G.A. Baker Jr., P. Graves-Morris, *in Padé Approximants* (Cambridge University Press, New York, 1996)
15. J.P. Berrut, Comput. Math. Appl. **14**, 1 (1988)
16. L.F. Richardson, Philos. Trans. R. Soc. Lond. Ser. A **210**, 307–357 (1911)
17. L.N. Trefethen, SIAM Rev. **50**, 67–87 (2008)
18. M. Novelinkova, in *WDS'11 Proceedings of Contributed Papers, Part I* (2011), p. 67. ISBN 978-80-7378-184-2
19. C.W. Clenshaw, A.R. Curtis, Numer. Math. **2**, 197 (1960)
20. J. Waldvogel, B.I.T. Numer, Math. **46**, 195 (2006)
21. A. Sommariva, *submitted to Comput* (Math, Appl, 2012)
22. G.H. Golub, J.H. Welsch, Math. Comput. **23**, 221 (1969)
23. R.Z. Iqbal, Master Thesis, School of Mathematics (University of Birmingham, 2008)
24. H. Wilf, *in Mathematics for the Physical Sciences* (Wiley, New York, 1962)
25. M.R. Hestenes, E. Stiefel, J. Res. Natl. Bur. Stand. **49**, 435 (1952)
26. Y. Saad, M. Schultz, SIAM, J. Sci. Stat. Comput. **7**, 856–869 (1986)
27. R. Fletcher, Conjugate gradient methods for indefinite systems, in *Numerical Analysis, vol. 506*, Lecture Notes in Mathematics, ed. by G.A. Watson (Springer, Berlin, 1976), pp. 73–89

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7

28. H.A. Van der Vorst, SIAM, J. Sci. Stat. Comput. **13**(2), 631–644 (1992)
29. Y. Saad, *in Iterative Methods for Sparse Linear Systems*, 2nd edn. (SIAM, Philadelphia, 2003), p. 194
30. C. Paige, S.I.A.M.J. Saunders, Numer. Anal. **12**, 617–629 (1975)
31. R. Freund, N. Nachtigal, Numer. Math. **60**, 315–339 (1991)
32. C. Paige, B. Parlett, H. van der Vorst, Numer. Linear Algebra Appl. **29**, 115–134 (1995)
33. C. Brezinski, H. Sadok, Appl. Numer. Math. **11**, 443–473 (1993)
34. M.H. Gutknecht, Acta Numer. **6**, 271–397 (1997)
35. S.-L. Zhang, SIAM, J. Sci. Stat. Comput. **18**, 537–551 (1997)
36. W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *LU decomposition and its applications, in Numerical Recipes, the Art of Scientific Computing*, 3rd edn. (Cambridge University Press, Cambridge, 2007), pp. 48–55
37. W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Cholesky decomposition, in Numerical Recipes, the Art of Scientific Computing*, 3rd edn. (Cambridge University Press, Cambridge, 2007), pp. 100–101
38. L.N. Trefethen, D. Bau III, *in Numerical Linear Algebra (Society for Industrial and Applied Mathematics* (SIAM, Philadelphia, 1997)
39. G.H. Golub, C.F. Van Loan, *in Matrix Computations*, 3rd edn. (Johns Hopkins, Baltimore, 1976). ISBN 978-0-8018-5414-9
40. J.W. Daniel, W.B. Gragg, L. Kaufmann, G.W. Stewart, Math. Comput. **30**, 772 (1976)
41. W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Cyclic tridiagonal systems, in Numerical Recipes, the Art of Scientific Computing*, 3rd edn. (Cambridge University Press, Cambridge, 2007), p. 79
42. J. Sherman, W.J. Morrison, Ann. Math. Stat. **20**, 621 (1949)
43. I.N. Bronshtein, K.A. Semendyayev, *in Handbook of Mathematics*, 3rd edn. (Springer, New York, 1997), p. 892
44. W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Relaxation methods for boundary value problems, in Numerical Recipes, the Art of Scientific Computing*, 3rd edn. (Cambridge University Press, Cambridge, 2007), pp. 1059–1065
45. H. Jeffreys, B.S. Jeffreys, *in Methods of Mathematical Physics*, 3rd edn. (Cambridge University Press, Cambridge, 1988), pp. 305–306
46. J.H. Wilkinson, J. ACM **8**, 281 (1961)
47. D. Hilbert, L. Nordheim, J. von Neumann, John. Math. Ann. **98**, 1 (1927)
48. A. Bermann, S. Gueron, Math. Gaz. **86**, 274 (2002)
49. B. Beckermann, Numer. Math. **85**, 553 (2000)
50. I.R. Savage, E. Lukacs, in *Contributions to the Solution of Systems of Linear Equations and the Determination of Eigenvalues.* National Bureau of Standards, Applied Mathematics Series, vol. 39, ed. by O. Taussky (1954)
51. R.L. Burden, J.D. Faires, *in Numerical Analysis* (Brooks Cole Publishing Company, Boston, 2010)
52. J.A. Ford, Improved Algorithms of Illinois-type for the Numerical Solution of Nonlinear Equations. Technical Report CSM-257. University of Essex Press (1995)
53. J.Y. Tjalling, Historical development of the Newton-Raphson method. SIAM Rev. **37**, 531 (1995)
54. J.M. Papakonstantinou, A historical development of the (n+1)-point secant method, M.A. thesis, Rice University Electronic Theses and Dissertations, 2007
55. D.E. Muller, Math. Tables Aids Comput. **10**, 208 (1956)
56. R.P. Brent, Comput. J. **14**, 422 (1971)
57. T.R. Chandrupatla, Adv. Eng. Softw. **28**, 145 (1997)
58. T.J. Dekker, Finding a zero by means of successive linear interpolation, in *Constructive Aspects of the Fundamental Theorem of Algebra*, ed. by B. Dejon, P. Henrici (Wiley-Interscience, London, 1969)
59. J.C.P. Bus, T.J. Dekker, A.C.M. Trans, Math. Softw. **1**, 330 (1975)

60. R.P. Brent, *Algorithms for Minimization without Derivatives* (Prentice-Hall, Englewood Cliffs, 1973)
61. A. Quarteroni, R. Sacco, F. Saleri, *Numerical Mathematics*, 2nd edn. (Springer, Berlin, 2007)
62. C.G. Broyden, Math. Comput. **19**, 577 (1965)
63. W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Section 10.2. golden section search in one dimension, in *Numerical Recipes: The Art of Scientific Computing*, 3rd edn. (Cambridge University Press, New York, 2007). ISBN 978-0-521-88068-8
64. J. Kiefer, Proc. Am. Math. Soc. **4**, 502–506 (1953)
65. R. Fletcher, C. Reeves, Comput. J. **7**, 149 (1964)
66. C.G. Broyden, J. Inst. Math. Appl. **6**, 76 (1970)
67. R. Fletcher, Comput. J. **13**, 317 (1970)
68. D. Goldfarb, Math. Comput. **24**, 23 (1970)
69. D.F. Shanno, Math. Comput. **24**, 647 (1970)
70. Z. Wang, B.R. Hunt, Appl. Math. Comput. **16**, 19 (1985)
71. N. Ahmed, T. Natarajan, K.R. Rao, IEEE Trans. Comput. **C-23**, 90 (1974)
72. K.R. Rao, P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications* (Academic Press, Boston, 1990)
73. F.j. Harris, Proc. IEEE **66**, 51 (1978)
74. G. Goertzel, Am. Math. Mon. **65**, 34 (1958)
75. E.I. Jury, *in Theory and Application of the Z-Transform Method* (Krieger Publishing Company, New York, 1973). ISBN 0-88275-122-0
76. P. Duhamel, M. Vetterli, Signal Process. **19**, 259 (1990)
77. H.J. Nussbaumer, *in Fast Fourier Transform and Convolution Algorithms* (Springer, Berlin, 1990)
78. J.W. Cooley, J.W. Tukey, Math. Comput. **19**, 297 (1965)
79. D. Gabor, J. Inst. Electr. Eng. **93**, 429 (1946)
80. M.J. Bastiaans, Appl. Opt. **33**, 5241 (1994)
81. M.J. Bastiaans, Proc. IEEE **68**, 538 (1980)
82. R.S. Orr, I.E.E.E. Trans, Sign. Proc. **41**, 122 (1993)
83. S. Qian, I.E.E.E. Trans, Signal Proc. **41**, 2429 (1993)
84. J. Ashmead, Quanta **1**, 58 (2012)
85. A. Grossmann, J. Morlet, SIAM. J. Math. Anal. **15**, 723 (1984)
86. S.G. Mallat, Trans. Am. Math. Soc. **315**, 69 (1989)
87. S.G. Mallat, I.E.E.E. Trans, Pattern Anal. Mach. Intell. **11**, 674 (1989)
88. A. Haar, Math. Ann. **69**, 331 (1910)
89. I. Daubechies, A. Grossmann, Y. Meyer, J. Math. Phys. **27**, 1271 (1986)
90. Y. Meyer, *in Wavelets and Operators*, vol. I (Cambridge University Press, Cambridge, 1995)
91. V. Vermehren Valenzuela, H.M. de Oliveira, OALib J. (2015), arXiv.org/abs/1502.00161v1
92. D. Esteban, C. Galand, in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (1977)
93. I. Daubechies, Commun. Pure Appl. Math. **16**, 909 (1988)
94. N. Metropolis, S. Ulam, J. Am. Stat. Assoc. **44**, 335 (1949)
95. G.S. Fishman, in *Monte Carlo: Concepts, Algorithms, and Applications* (Springer, New York, 1995, 1996). ISBN 038794527X
96. C.P. Robert, G.C. Casella, *in Monte Carlo Statistical Methods*, 2nd edn. (Springer, New York, 2004). ISBN 0387212396
97. R.E. Caflisch, in *Monte Carlo and Quasi-Monte Carlo Methods*. Acta Numerica, vol. 7 (Cambridge University Press, Cambridge, 1998), pp. 1–49
98. R.D. Richtmyer, *in Principles of Modern Mathematical Physics*, vol. I (Springer, New York, 1978)
99. H. Pollard, Q. J. Pure Appl. Math. **49**, 1 (1920)
100. J. Rice, *in Mathematical Statistics and Data Analysis*, 2nd edn. (Duxbury Press, Belmont, 1995). ISBN 0-534-20934-3
101. True random number service at RANDOM.ORG

102. J. Kelsey, B. Schneier, N. Ferguson, *Yarrow-160 notes on the design and analysis of the yarrow cryptographic pseudorandom number generator, in Sixth Annual Workshop on Selected Areas in Cryptography* (Springer, Berlin, 1999)
103. N. Ferguson, B. Schneier, *in Practical Cryptography* (Wiley, New York, 2003)
104. The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness, http://stat.fsu.edu/pub/diehard/
105. G. Marsaglia, J. Mod. Appl. Stat. Methods **1**, 2 (2003)
106. The TestU01 website, http://simul.iro.umontreal.ca/testu01/tu01.html
107. P. L'Ecuyer, R. Simard, A.C.M. Trans, Math. Softw. **33**, 22 (2007)
108. M. Matsumoto, T. Nishimura, A.C.M. Trans, Modeling Comput. Simul. **8**, 3 (1998)
109. K. Entacher, A.C.M. Trans, Modeling Comput. Simul. **8**, 61 (1998)
110. G.E.P. Box, M.E. Muller, Ann. Math. Stat. **29**, 610 (1958)
111. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, J. Chem. Phys. **21**, 1087 (1953)
112. E. Bonomi, J.-L. Lutton, SIAM Rev. **26**, 551 (1984)
113. B.N. Parlett, *in The Symmetric Eigenvalue Problem* (Society for Industrial and Applied Mathematics, Philadelphia, 1998)
114. H.W. Chang, S.E. Liu, R. Burridge, Linear Algebra Appl. **430**, 999 (2009)
115. S. Kouachi, Electron. J. linear Algebra **15**, 115 (2006)
116. W.C. Yueh, Appl. Math. E-Notes **5**, 66 (2005)
117. J.G.F. Francis, Comput. J. **4**, 265 (1961); **4**, 332 (1962)
118. C. Lanczos, J. Res. Natl Bur. Stand. **45**, 255 (1950)
119. H.D. Simon, Linear Algebra Appl. **61**, 101 (1984)
120. B.N. Parlett, D.S. Scott, Math. Comput. **33**, 217 (1979)
121. D. Calvetti, L. Reichel, D.C. Sorensen, Electron. Trans. Numer. Anal. **2**, 1 (1994)
122. D.C. Sorenson, NASA contractor report 198342 (1996)
123. J. Wolberg, *in Data Analysis Using the Method of Least Squares: Extracting the Most Information from Experiments* (Springer, Berlin, 2005)
124. J.A. Richards, *in Remote Sensing Digital Image Analysis* (Springer, Berlin, 1993)
125. A.E. Garcia, Phys. Rev. Lett. **86**, 2696 (1992)
126. T.D. Romo et al., Proteins **22**, 311 (1995)
127. D.P. Derrarr et al., *in A Practical Approach to Microarray Data Analysis* (Kluwer, Norwell, 2003)
128. DGESVD routine from the freely available software package LAPACK, http://www.netlib.org/lapack
129. G. Golub, W. Kahan, J.S.I.A.M. Numer, Anal. B **2**, 205 (1965)
130. G.W. Stewart, SIAM Rev. **35**, 551 (1993)
131. J. Mandel, Am. Stat. **36**, 15 (1982)
132. A. Ben-Israel, T.N.E. Greville, *in Generalized Inverses* (Springer, Berlin, 2003). ISBN 0-387-00293-6
133. J. Peiro, S. Sherwin, in *Handbook of Materials Modeling*, vol. 1, ed. by S. Yip (Springer, Berlin, 2005), pp. 1–32
134. H.P. Langtangen, *in Computational Partial Differential Equations: Numerical Methods and Diffpack Programming* (Springer, Berlin, 2003)
135. W. Ames, *in Numerical Methods for Partial Differential Equations* (Academic Press Inc., New York, 1992)
136. F. John, Duke Math. J. **4**, 300 (1938)
137. D.J. Acheson, *in Elementary Fluid Dynamics* (Oxford University Press, Oxford, 1990)
138. B.G. Galerkin, On electrical circuits for the approximate solution of the Laplace equation. Vestn. Inzh. **19**, 897–908 (1915)
139. R. Eymard, T. Galloue, R. Herbin, in *Finite Volume Methods. Handbook of Numerical Analysis*, ed. by P.G. Ciarlet, J.L. Lions (Elsevier, Amsterdam, 2000), pp. 713–1020
140. E.L. Ortiz, SIAM. J. Numer. Anal. **6**, 480 (1969)
141. B.-N. Jiang, *in The Least-Squares Finite Element Method* (Springer, Berlin, 1998)

142. P. Bochev, M. Gunzburger, *in Least Squares Finite Element Methods* (Springer, Berlin, 2009)
143. C.A.J. Fletcher, *in Computational Galerkin Methods* (Springer, Berlin, 1984)
144. J.P. Boyd, *in Chebyshev and Fourier Spectral Methods* (Dover Publications, New York, 2001)
145. J. Fish, T. Belytschko, *in A First Course in Finite Elements* (Wiley, New York, 2007)
146. S.S. Rao, *in The Finite Element Method in Engineering* (Elsevier, Amsterdam, 2011)
147. C.A.J. Fletcher, in *Computational Techniques for Fluid Dynamics*, vol. I, 2nd edn. (Springer, Berlin, 1991)
148. G. Beer, I. Smith, C. Duenser, *in The Boundary Element Method with Programming: For Engineers and Scientists* (Springer, Berlin, 2008)
149. L.C. Wrobel, M.H. Aliabadi, *in The Boundary Element Method* (Wiley, New Jersey, 2002)
150. G. Wunsch, *in Feldtheorie* (VEB Technik, Berlin, 1973)
151. A. Nordsieck, Math. Comput. **16**, 22 (1962)
152. C.W. Gear, *in Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice-Hall, Englewood Cliffs, 1971)
153. G.J. Martyna, M.E. Tuckerman, J. Chem. Phys. **102**, 8071 (1995)
154. J.C. Butcher, *in The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods* (Wiley, New York, 1987)
155. W.B. Gragg, SIAM. J. Numer. Anal. **2**, 384 (1965)
156. L.F. Shampine, IMA. J. Numer. Anal. **3**, 383 (1983)
157. L.F. Shampine, L.S. Baca, Numer. Math. **41**, 165 (1983)
158. C.W. Gear, I.E.E.E. Trans, Circuit Theory **18**, 89–95 (1971)
159. C.W. Gear, Math. Comput. **21**, 146 (1967)
160. I.P. Omelyan, I.M. Mryglod, R. Folk, Comput. Phys. Commun. **151**, 272 (2003)
161. S.-H. Tsai et al., Braz. J. Phys. **34**, 384 (2004)
162. M. Tuckerman, B.J. Berne, J. Chem. Phys. **97**, 1990 (1992)
163. M.P. Allen, D.J. Tildesley, *in Computer Simulation of Liquids* (Oxford University Press, Oxford, 1989). ISBN 0-19-855645-4
164. L. Verlet, Phys. Rev. **159**, 98 (1967)
165. E. Hairer, C. Lubich, G. Wanner, Acta Numer. **12**, 399 (2003)
166. W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, in Section 17.4. second-order conservative equations, in *Numerical Recipes: The Art of Scientific Computing*, 3rd edn. (Cambridge University Press, Cambridge, 2007)
167. P. Schofield, Comput. Phys. Commun. **5**, 17 (1973)
168. D. Beeman, J. Comput. Phys. **20**, 130 (1976)
169. R. Sonnenschein, A. Laaksonen, E. Clementi, J. Comput. Chem. **7**, 645 (1986)
170. B.K.P. Horn, H.M. Hilden, S. Negahdaripour, J. Opt. Soc. Am. A **5**, 1127 (1988)
171. I.P. Omelyan, Phys. Rev. **58**, 1169 (1998)
172. I.P. Omelyan, Comput. Phys. Commun. **109**, 171 (1998)
173. H. Goldstein, *inKlassische Mechanik* (Akademische Verlagsgesellschaft, Frankfurt/Main, 1974)
174. I.P. Omelyan, Comput. Phys. **12**, 97 (1998)
175. D.C. Rapaport, *in The Art of Molecular Dynamics Simulation* (Cambridge University Press, Cambridge, 2004). ISBN 0-521-44561-2
176. D. Frenkel, B. Smit, *in Berend: Understanding Molecular Simulation: From Algorithms to Applications* (Academic Press, San Diego, 2002)
177. J.M. Haile, *in Molecular Dynamics Simulation: Elementary Methods* (Wiley, New York, 2001). ISBN 0-471-18439-X
178. A. Leach, *in Molecular Modelling: Principles and Applications*, 2nd edn. (Prentice Hall, Publisher, Harlow, 2001). ISBN 978-0582382107
179. T. Schlick, *in Molecular Modeling and Simulation* (Springer, Berlin, 2002). ISBN 0-387-95404-X
180. R. Car, M. Parrinello, Phys. Rev. Lett. **55**, 2471 (1985)
181. T. Kühne, M. Krack, F. Matthias, M.P. Mohamed, Phys. Rev. Lett. **98**, 066401 (2007)

182. W.D. Cornell, P. Cieplak, C.I. Bayly, I.R. Gould, K.M. Merz Jr., D.M. Ferguson, D.C. Spellmeyer, T. Fox, J.W. Caldwell, P.A. Kollman, J. Am. Chem. Soc. **117**, 5179 (1995)
183. A.D. MacKerell Jr., B. Brooks, C.L. Brooks III, L. Nilsson, B. Roux, Y. Won, M. Karplus, CHARMM: the energy function and its parameterization with an overview of the program, in *The Encyclopedia of Computational Chemistry*, vol. 1, ed. by P.v.R. Schleyer, et al. (Wiley, Chichester, 1998), pp. 271–277
184. W.F. van Gunsteren, S.R. Billeter, A.A. Eising, P.H. Hünenberger, P. Krüger, A.E. Mark, W.R.P. Scott, I.G. Tironi, *Biomolecular Simulation: The GROMOS96 Manual and User Guide*. vdf Hochschulverlag AG an der ETH Zürich and BIOMOS b.v.: Zürich, Groningen, 1996
185. M. Christen, P.H. Hünenberger, D. Bakowies, R. Baron, R. Bürgi, D.P. Geerke, T.N. Heinz, M.A. Kastenholz, V. Kräutler, C. Oostenbrink, C. Peter, D. Trzesniak, W.F. van Gunsteren, J. Comput. Chem. **26**, 1719 (2005)
186. IUPAC-IUB Commission on Biochemical Nomenclature, Biochemsistry **9**, 3471 (1970)
187. B.R. Brooks, C.L. Brooks III, A.D. Mackerell Jr., L. Nilsson, R.J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A.R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuzcera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R.W. Pastor, C.B. Post, J.Z. Pu, M. Schaefer, B. Tidor, R.M. Venable, H.L. Woodcock, X. Wu, W. Yang, D.M. York, M. Karplus, J. Comput. Chem. **30**, 1545 (2009)
188. B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, M. Karplus, J. Comput. Chem. **4**, 187 (1983)
189. K.J. Bowers, E. Chow, H. Xu, R.U. Dror, M.P. Eastwood, B.A. Gregersen, J.L. Klepeis, I. Kolossvary, M.A. Moraes, F.D. Sacerdoti, J.K. Salmon, Y. Shan, D.E. Shaw, Scalable algorithms for molecular dynamics simulations on commodity clusters, in SC. Conference. Proceedings of the ACM/IEEE **2006**, 43 (2006)
190. A.K. Rappe, C.J. Casewit, K.S. Colwell, W.A. Goddar, W.M. Skiff, J. Am. Chem. Soc. **115**, 10024 (1992)
191. E.J. Bautista, J.M. Seminario, Int. J. Quantum Chem. **108**, 180 (2008)
192. R.E. Tuzun, D.W. Noid, B.G. Sumpter, Macromol. Theory Simul. **5**, 771 (1996)
193. R.E. Tuzun, D.W. Noid, B.G. Sumpter, J. Comput. Chem. **18**, 1804 (1997)
194. C. Lavor, Phys. D **227**, 135 (2007)
195. V.S. Allured, C.M. Kelly, C.R. Landis, J. Am. Chem. Soc. **113**, 1 (1991)
196. T. Schlick, J. Comput. Chem. **10**, 951 (1989)
197. S. Chynoweth, U.C. Klomp, L.E. Scales, Comput. Phys. Commun. **62**, 297 (1991)
198. H. Bekker, H.J.C. Berendsen, W.F. van Gunsteren, J. Comput. Chem. **16**, 527 (1995)
199. J.P. Hansen, L. Verlet, Phys. Rev. **184**, 151 (1969)
200. J.J. Nicolas, K.E. Gubbins, W.B. Streett, D.J. Tildesley, Mol. Phys. **37**, 1429 (1979)
201. J.K. Johnson, J.A. Zollweg, K.E. Gubbins, Mol. Phys. **78**, 591 (1993)
202. H. Watanabe, N. Ito, C.K. Hu, J. Chem. Phys. **136**, 204102 (2012)
203. C. Muguruma, Y. Okamoto, M. Mikami, Croat. Chemica Acta **80**, 203 (2007)
204. S.A. Khrapak, M. Chaudhuri, G.E. Morfill, Phys. Rev. B **82**, 052101 (2010)
205. B. Smit, J. Chem. Phys. **96**, 8639 (1992)
206. H.J.C. Berendsen, J.P.M. Postma, W.F. van Gunsteren, A. DiNola, J.R. Haak, J. Chem. Phys. **81**, 3684 (1984)
207. P.H. Huenenberger, Adv. Polym. Sci. **173**, 105 (2005)
208. F. Schwabl, *in Statistical Mechanics* (Springer, Berlin, 2003)
209. J.R. Errington, P.G. Debenedetti, S. Torquato, J. Chem. Phys. **118**, 2256 (2003)
210. H. Risken, *in The Fokker-Planck Equation* (Springer, Berlin, 1989)
211. D. Levesque, L. Verlet, Phys. Rev. A **2**, 2514 (1970)
212. T. Tsang, H. Tang, Phys. Rev. A **15**, 1696 (1977)
213. E. Ising, Beitrag zur Theorie des Ferromagnetismus. Z. Phys. **31**, 253–258 (1925). doi:10.1007/BF02980577
214. K. Binder, in *"Ising Model" Encyclopedia of Mathematics* (Suppl.), vol. 2, ed. by R. Hoksbergen (Kluwer Academic Publishers, Dordrecht, 2000), pp. 279–281
215. L. Onsager, Phys. Rev. **65**, 117 (1944)

216. B.M. McCoy, T.T. Wu, *in The Two-Dimensional Ising Model* (Harvard University Press, Cambridge, 1973). ISBN 0674914406
217. K. Pearson, The problem of the Random Walk. Nature **72**, 294 (1905)
218. A.A. Markov, Theory of Algorithms. [Translated by Jacques J. Schorr-Kon and PST staff] Imprint Moscow, Academy of Sciences of the USSR, 1954 [Jerusalem, Israel Program for Scientific Translations, 1961; available from Office of Technical Services, United States Department of Commerce] Added t.p. in Russian Translation of Works of the Mathematical Institute, Academy of Sciences of the USSR, v. 42. Original title: Teoriya algorifmov. [QA248.M2943 Dartmouth College library. U.S. Dept. of Commerce, Office of Technical Services, number OTS 60-51085.] (1954)
219. A.A. Markov, Extension of the limit theorems of probability theory to a sum of variables connected in a chain reprinted in Appendix B of: R. Howard, in *Dynamic Probabilistic Systems, volume 1: Markov Chains* (Wiley, New York, 1971)
220. G. Schaftenaar, J.H. Noordik, J. Comput.-Aided Mol. Design **14**, 123 (2000)
221. W.L. Mattice, U.W. Suter, *in Conformational Theory of Large Molecules* (Wiley Interscience, New York, 1994). ISBN 0-471-84338-5
222. R. Brown, Philos. Mag. **4**, 161 (1828)
223. A. Einstein, Annalen der Physik **17**, 549 (1905)
224. A. Einstein, *in Investigations on the Theory of Brownian Movement* (Dover, New York, 1956)
225. S. Yang, M.K. Gobbert, Appl. Math. Lett. **22**, 325 (2009)
226. Y. Zhu, A.C. Cangellaris, in *Multigrid Finite Element Methods for Electromagnetic Field Modeling* (Wiley, New York, 2006), p. 132 ff. ISBN 0471741108
227. M.T. Heath, in *§11.5.7 Multigrid Methods. Scientific Computing: An Introductory Survey* (McGraw-Hill Higher Education, New York, 2002), p. 478 ff. ISBN 007112229X
228. G. Skollermo, Math. Comput. **29**, 697 (1975)
229. P.N. Swarztrauber, R.A. Sweet, in *Handbook of Fluid Dynamics and Fluid Machinery*, ed. by J.A. Schetz, A.E. Fuhs (Wiley, New York, 1996)
230. M. Oevermann, R. Klein, J. Comput. Phys. **219**, 749 (2006)
231. M. Oevermann, C. Scharfenberg, R. Klein, submitted to J. Comput. Phys
232. R.E. Bruccoleri, J. Novotny, M.E. Davis, K.A. Sharp, J. Comput. Chem. **18**, 268 (1997)
233. P. Debye, E. Hueckel, Phys. Z. **24**, 185 (1923)
234. G.L. Gouy, J. Phys. **9**, 457 (1910)
235. D.L. Chapman, Philos. Mag. **25**, 475 (1913)
236. F. Fogolari, A. Brigo, H. Molinari, J. Mol. Recognit. **15**, 377 (2002)
237. A. Nicholls, B. Honig, J. Comput. Chem. **12**, 435 (1990)
238. A.H. Boschitsch, M.O. Fenley, H.X. Zhou, J. Phys. Chem. B **106**, 2741 (2002)
239. A.H. Juffer et al., J. Phys. Chem. B **101**, 7664 (1997)
240. J.S. Bader et al., J. Chem. Phys. **106**, 2372 (1997)
241. T. Simonson, Rep. Prog. Phys. **66**, 737 (2003)
242. J.G. Kirkwood, J. Chem. Phys. **2**, 351 (1934)
243. R. Courant, K. Friedrichs, H. Lewy, Math. Ann. **100**, 32 (1928)
244. Y. Li, C. Trenchea, J. Comput. Phys. **259**, 23 (2014)
245. P.D. Williams, Mon. Weather Rev. **139**, 1996 (2011)
246. R.J. LeVeque, *in Finite-Volume Methods for Hyperbolic Problems* (Cambridge University Press, Cambridge, 2004)
247. S.K. Godunov, Math. Sbornik **47**, 271 (1959) (translated US Joint Publ. Res. Service, JPRS 7225 Nov. 29, 1960)
248. E.F. Toro, *in Riemann Solvers and Numerical Methods for Fluid Dynamics* (Springer, Berlin, 1999)
249. J. Donea, Int. J. Numer. Meth. Eng. **20**, 101 (1984)
250. J. Donea, L. Quartapelle, V. Selmin, J. Comput. Phys. **70**, 463 (1987)
251. B. Roig, J. Comput. Appl. Math. **204**, 95 (2007)
252. E.F. Toro, S.J. Billet, College of Aeronautics Report 9312 (1993). ISBN 1-871564-65-4

253. H. Ibach, H. Lüth, *in Solid-State Physics: An Introduction to Principles of Materials Science (Advanced Texts in Physics)* (Springer, Berlin, 2003)
254. L. Lynch, in *Numerical Integration of Linear and Nonlinear Wave Equations*. Bachelor Thesis, Florida Atlantic University (Digital Commons@University of Nebraska-Lincoln, 2004)
255. S.A. Teukolsky, Phys. Rev. D **61**, 067503 (2000)
256. Y.A. Cengel, in *Heat Transfer-A Practical Approach*, 2nd edn. (McGraw Hill Professional, New York, 2003), p. 26. ISBN 0072458933, 9780072458930
257. J. Fourier, in *The Analytical Theory of Heat* (Cambridge University Press, Cambridge University Press, 1878) (Cambridge University Press, Cambridge, 2009; ISBN 978-1-108-00178-6)
258. A. Fick, Philos. Mag. **10**, 30 (1855)
259. J. Crank, P. Nicolson, Proc. Camb. Philos. Soc. **43**, 50 (1947)
260. J.W. Thomas, *in Numerical Partial Differential Equations: Finite Difference Methods, Texts in Applied Mathematics*, vol. 22 (Springer, New York, 1995)
261. D. Hinrichsen, A.J. Pritchard, *in Mathematical Systems Theory I - Modelling, State Space Analysis, Stability and Robustness* (Springer, Berlin, 2005). ISBN 0-978-3-540-441250
262. H.K. Khalil, *in Nonlinear Systems* (Prentice Hall, Englewood Cliffs, 2001). ISBN 0-13-067389-7
263. I. Vasile, *in Istratescu in Fixed Point Theory, An Introduction* (D. Reidel Publishing Company's, London, 1981)
264. S.H. Strogatz, in *Nonlinear dynamics and Chaos: Applications to Physics, Biology, Chemistry, and Engineering* (Perseus, 2001). ISBN 0-7382-0453-6
265. J.D. Murray, in *Mathematical Biology: I. An Introduction*, 3rd edn., vol. 2 (Springer, Berlin, 2002). ISBN 0-387-95223-3
266. E. Renshaw, in *Modelling Biological Populations in Space and Time* (C.U.P., 1991). ISBN 0-521-44855-7
267. P. Grindrod, *in Patterns and Waves: The Theory and Applications of Reaction-Diffusion Equations* (Clarendon Press, Oxford, 1991)
268. P.C. Fife, *in Mathematical Aspects of Reacting and Diffusing Systems* (Springer, Berlin, 1979)
269. A.M. Lyapunov, *in Stability of Motion* (Academic Press, New York, 1966)
270. P.F. Verhulst, Mémoires de l'Académie Royale des Sciences et Belles Lettres de Bruxelles. 18, Bruxelles (1845), pp. 1–42
271. A.J. Lotka, *in Elements of Physical Biology* (Williams and Wilkins, Baltimore, 1925)
272. V. Volterra, Mem. R. Accad. Naz. dei Lincei **2**, 31 (1926)
273. C.S. Holling, Can. Entomol. **91**, 293 (1959)
274. C.S. Holling, Can. Entomol. **91**, 385 (1959)
275. J.T. Tanner, Ecology **56**, 855 (1975)
276. A.M. Turing, Philos. Trans. R. Soc. B **237**, 37 (1952)
277. F. Schwabl, *in Quantum Mechanics*, 4th edn. (Springer, Berlin, 2007)
278. E. Schrödinger, Phys. Rev. **28**, 1049 (1926)
279. X. Antoine, A. Arnold, C. Besse, M. Erhardt, A. Schädle, Commun. Comput. Phys. **4**, 729 (2008)
280. C. Leforestier, R.H. Bisseling, C. Cerjan, M.D. Feit, R. Friesner, A. Guldberg, A. Hammerich, G. Jolicard, W. Karrlein, H.-D. Meyer, N. Lipkin, O. Roncero, R. Kosloff, J. Comput. Phys. **94**, 59 (1991)
281. B. Fornberg, Geophysics **52**, 4 (1987)
282. B. Fornberg, Math. Comput. **51**, 699 (1988)
283. R. Guantes, S.C. Farantos, J. Chem. Phys. **111**, 10827 (1999)
284. R. Guantes, A. Nezis, S.C. Farantos, J. Chem. Phys. **111**, 10836 (1999)
285. A. Castro, M.A.L. Marques, A. Rubio, J. Chem. Phys. **121**, 3425 (2004)
286. Z.A. Anastassi, T.E. Simos, Phys. Rep. **482–483**, 1 (2009)
287. K. Yabana, G.F. Bertsch, Phys. Rev. B **54**, 4484 (1996)
288. A. Goldberg, H.M. Schey, J.L. Schwartz, Am. J. Phys. **35**, 177 (1967)
289. E.A. McCullough Jr., R.E. Wyatt, J. Chem. Phys. **51**, 1253 (1969)
290. E.A. McCullough Jr., R.E. Wyatt, J. Chem. Phys. **54**, 3592 (1971)

291. T. Iitaka, Phys. Rev. E **49**, 4684 (1994)
292. T. Iitaka, N. Carjan, T. Strottman, Comput. Phys. Commun. **90**, 251 (1995)
293. A. Askar, A.S. Cakmak, J. Chem. Phys. **68**, 2794 (1978)
294. H. De Raedt, Comput. Phys. Rep. **7**, 1 (1987)
295. H. De Raedt, B. De Raedt, Phys. Rev. A **28**, 3575 (1983)
296. A.D. Bandrauk, H. Shen, Chem. Phys. Lett. **176**, 428 (1991)
297. L.A. Collins, J.D. Kress, R.B. Walker, Comput. Phys. Commun. **114**, 15 (1998)
298. C. Cerjan, K.C. Kulander, Comput. Phys. Commun. **63**, 529 (1991)
299. H. TalEzer, R. Kosloff, J. Chem. Phys. **81**, 3967 (1984)
300. R. Chen, H. Guo, J. Chem. Phys. **111**, 9944 (1999)
301. T.J. Park, J.C. Light, J. Chem. Phys. **85**, 5870 (1986)
302. B.I. Schneider, J.A. Collins, J. Non-Cryst, Solids **351**, 1551 (2005)
303. I. Allonso-Mallo, N. Reguera, J. Comput. Phys. **220**, 409 (2006)
304. S.E. Koonin, C.M. Dawn, *in Computational Physics* (Perseus Books, New York, 1990). ISBN 978-0201127799
305. W. van Dijk, F.M. Toyama, Phys. Rev. E **75**, 036707 (2007)
306. D. Kosloff, R. Kosloff, J. Comput. Phys. **52**, 35 (1983)
307. W. Magnus, Commun. Appl. Math. **7**, 649 (1954)
308. M. Suzuki, Commun. Math. Phys. **51**, 183 (1976)
309. M.D. Feit, J.A. Fleck Jr., A. Steiger, J. Comput. Phys. **47**, 412 (1982)
310. P.W. Anderson, Phys. Rev. **79**, 350 (1950)
311. J. Halpern, L.E. Orgel, Discuss. Faraday Soc. **29**, 32 (1960)
312. P. Bocchieri, A. Loinger, Phys. Rev. **107**, 337 (1957)
313. M. Bixon, J. Jortner, J. Chem. Phys. **48**, 715 (1986)
314. B.I. Stepanov, V.P. Grobkovskii, *in Theory of Luminescence* (Butterworth Publications, London, 1986)
315. L. Landau, Phys. Soviet Union **2**, 46–51 (1932)
316. C. Zener, Proc. R. Soc. Lond. A **137**(6), 696–702 (1932)
317. A. Yariv, *in Quantum Electronics* (Wiley, New York, 1975)
318. L.M.K. Vandersypen, I.L. Chuang, Rev. Mod. Phys. **76**, 1037 (2004)
319. M. Nielsen, I. Chuang, *in Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000)
320. L. Diosi, *in A Short Course in Quantum Information Theory* (Springer, Berlin, 2007)
321. R.P. Feynman, F.L. Vernon, R.W. Hellwarth, J. Appl. Phys. **28**, 49 (1957)
322. S.E. Economou, T.L. Reinecke, in *Optical Generation and Control of Quantum Coherence in Semiconductor Nanostructures*, ed. by G. Slavcheva, P. Roussignol (Springer, Berlin, 2010), p. 62
323. F. Bloch, Nucl. Induction Phys. Rev. **70**, 460 (1946)
324. D. Pines, C.P. Slichter, Phys. Rev. **100**, 1014 (1955)
325. R.K. Wangsness, F. Bloch, Phys. Rev. **89**, 728 (1953)
326. S.J. Nettel, A. Kempicki, Am. J. Phys. **47**, 987 (1979)
327. I. Zutic, J. Fabian, S. Das, Sarma. Rev. Mod. Phys. **76**, 323 (2004)
328. G. Burkard, R.H. Koch, D.P. DiVincenzo, Phys. Rev. B **69**, 64503 (2004)
329. M. Fox, *in Quantum Optics, An Introduction* (Oxford University Press, Oxford, 2006)
330. F.-Y. Hong, S.-J. Xiong, Chin. J. Phys. **46**, 379 (2008)
331. C.P. Williams, *in Explorations in Quantum Computing* (Springer, Berlin, 2011)
332. C.J. Umrigar, K.G. Wilson, W. Wilkins, Phys. Rev. Lett. **60**, 1719 (1988)
333. P.J. Reynolds, D.M. Ceperley, B.J. Alder, J. Chem. Phys. **77**, 5593 (1982)
334. R. Jastrow, Phys. Rev. **98**, 1479 (1955)
335. T. Kato, Commun. Pure Appl. Math. **10**, 151 (1957)
336. N.D. Drummond, M.D. Towler, R.J. Needs, Phys. Rev. B **70**, 235119 (2004)
337. H.J. Flad, A. Savin, J. Chem. Phys. **103**, 691 (1995)
338. T. Kobayashi, *in J-Aggregates* (World Scientific, Singapore, 1996)
339. R.J. Cogdell, A. Gall, J. Köhler, Q. Rev, Biophysics **39**, 227 (2006)

340. P.O.J. Scherer, E.W. Knapp, S.F. Fischer, Chem. Phys. Lett. **106**, 101 (1984)
341. R.E. Merrifield, J. Chem. Phys. **40**, 445 (1964)
342. P.O.J. Scherer, S.F. Fischer, Chem. Phys. **86**, 269 (1984)
343. R. Kunz, K. Timpmann, J. Southall, R.J. Cogdell, A. Freiberg, J. Köhler, J. Phys. Chem. B **116**, 11017 (2012)
344. A.S. Davydov, Physica **3D**, 1 (1981)
345. G. Venzl, S.F. Fischer, Phys. Rev. B **32**, 6437 (1985)

Index

P.O.J. Scherer, *Computational Physics*, Graduate Texts in Physics,
DOI 10.1007/978-3-319-61088-7

S

T

U