Philipp O.J. Scherer

Computational Physics

Simulation of Classical and Quantum Systems

Springer

Computational Physics

Philipp O.J. Scherer

Computational Physics

Simulation of Classical and Quantum Systems

Springer

Prof. Dr. Philipp O.J. Scherer
TU München
Physikdepartment T38
85748 München
Germany
philipp.scherer@ph.tum.de

Additional materials to this book can be downloaded from http://extras.springer.com

ISBN 978-3-642-13989-5 e-ISBN 978-3-642-13990-1
DOI 10.1007/978-3-642-13990-1
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2010937781

Cover design: eStudio Calamar S.L., Heidelberg

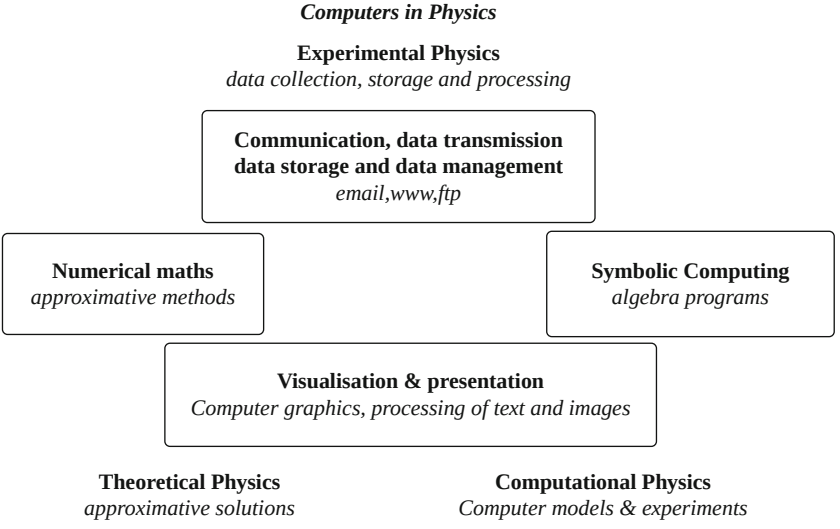
Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

for Christine

Preface

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. It 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 nevertheless 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. The basic algorithms are explained in detail together with limitations due to numerical inaccuracies. The mathematical explanation is supplemented by a large number of 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 one-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-level 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. A large number of computer experiments are provided as Java applets which can be easily run in the web browser. For a deeper insight the source code can be studied and modified with the free "netbeans"[1] environment.

Garching, 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:

- *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* $\Delta\mathbf{x}$*.*
- *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.*
- *Results from more complex operations like square roots or trigonometric functions can have even larger errors since iterations and series expansions have to be truncated after a finite number of 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 IEEE [1] single, double, and quadruple precision numbers are stored as 32, 64, and 128 bits (Table 1.1):

Table 1.1 Binary floating point formats

Format	Sign	Exponent	Hidden bit	Fraction
Float	s	$b_0 \ldots b_7$	1	$a_0 \ldots a_{22}$
Double	s	$b_0 \ldots b_{10}$	1	$a_0 \ldots a_{51}$
Quadruple	s	$b_0 \ldots b_{14}$	1	$a_0 \ldots a_{111}$

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

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_1,

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
$16,383_{10}$	11111111111111_2	$3FFF	Quadruple

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

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

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 to $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

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

Table 1.3 Special double precision numbers

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

[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.

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

$$\$4059000000000000.$$

The exponent is $10000000101_2 - 01111111111_2 = 110_2$ 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 $\Re \to A$

$$x \to r\mathrm{d}(x),$$

with the property

$$|x - r\mathrm{d}(x)| \leq |x - g| \quad \text{for all } g \in A. \tag{1.3}$$

The cases of *exponent overflow* and *exponent underflow* need special attention:
Whenever the exponent b has the maximum possible value $b = b_{\max}$ and $a = 1.11 \cdots 11$ has to be rounded to $a' = 10.00 \cdots 0$, the rounded number is not a machine number and the result is $\pm$inf.

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

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

is given by

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

and the relative error is bounded by

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

The relative machine precision is defined by

$$\varepsilon_{\mathrm{M}} = 2^{-t}, \tag{1.7}$$

and we have

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

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 rounding 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 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:

$$\begin{aligned} y &< 2^{b_x - E} \times 2^{-t},\\ a_y 2^{b_y - b_x} &< 2^{-t}. \end{aligned} \tag{1.11}$$

The mantissa of the exact sum is

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

Rounding to the nearest machine number gives

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

since

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

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

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

Especially for $x = 1$ we have

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

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:

$$\begin{aligned} x_1 &\to rd(x_1) = x_1(1 + \varepsilon_1), \\ x_2 &\to rd(x_2) = x_2(1 + \varepsilon_2). \end{aligned} \tag{1.20}$$

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

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

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

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

and the relative error of the numerical sum is

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

If $x_1 \approx -x_2$ then numerical extinction can produce large relative errors and errors 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

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

with the relative error

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

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

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

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

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

a set of output data are calculated

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

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

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

Starting with $\mathbf{x}$ intermediate results $\mathbf{x}_i = (x_{i1}, \ldots, x_{in_i})$ 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 rounding 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

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

Taylor series expansion gives in first order

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

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 rounding 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

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

The second intermediate result is

$$\begin{aligned} \widetilde{\mathbf{x}_2} &= \left(\varphi^{(2)}(\widetilde{\mathbf{x}_1})\right)(1 + E_2) = \varphi^{(2)}(\mathbf{x}_1 + \Delta \mathbf{x}_1)\,(1 + E_2) \\ &= \mathbf{x}_2(1 + E_2) + \mathrm{D}\varphi^{(2)} \mathrm{D}\varphi^{(1)} \Delta \mathbf{x} + \mathrm{D}\varphi^{(2)} \mathbf{x}_1 E_1, \end{aligned} \tag{1.40}$$

with the error

$$\Delta \mathbf{x}_2 = \mathbf{x}_2 E_2 + \mathrm{D}\varphi^{(2)} \mathrm{D}\varphi^{(1)} \Delta \mathbf{x} + \mathrm{D}\varphi^{(2)} \mathbf{x}_1 E_1 \tag{1.41}$$

Finally the error of the result is

$$\Delta \mathbf{y} = \mathbf{y} E_r + \mathrm{D}\varphi^{(r)} \cdots \mathrm{D}\varphi^{(1)} \Delta \mathbf{x} + \mathrm{D}\varphi^{(r)} \cdots \mathrm{D}\varphi^{(2)} \mathbf{x}_1 E_1 + \cdots + \mathrm{D}\varphi^{(r)} \mathbf{x}_{r-1} E_{r-1}. \tag{1.42}$$

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

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

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]

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

or in case the error of the input data is dominated by the rounding errors $|\Delta x_j| \leq \varepsilon_{\mathrm{M}} |x_j|$

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

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 some initial value $\mathbf{x}_0$ according to an iteration prescription of the type

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

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 of the start values:

$$\mathbf{x}_0 + \Delta\mathbf{x}, \tag{1.49}$$

$$\mathbf{x}_j = A^j\mathbf{x}_0 + A^j\Delta\mathbf{x}. \tag{1.50}$$

The initial errors can be enhanced exponentially if A has at least one eigenvalue[3] 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

$$\mathbf{x}_{j+1} = \varphi(\mathbf{x}_j), \tag{1.51}$$

the error propagates according to

$$\begin{aligned}
&\mathbf{x}_1 = \varphi(\mathbf{x}_0) + \mathrm{D}\varphi\Delta\mathbf{x},\\
&\mathbf{x}_2 = \varphi(\mathbf{x}_1) = \varphi(\varphi(\mathbf{x}_0)) + (\mathrm{D}\varphi)^2\Delta x,\\
&\vdots\\
&\mathbf{x}_j = \varphi(\varphi\cdots\varphi(\mathbf{x}_0)) + (\mathrm{D}\varphi)^j\Delta x.
\end{aligned} \tag{1.52}$$

[3] The eigenvalues of A are solutions of the eigenvalue equation $A\mathbf{x} = \lambda\mathbf{x}$ (9).

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

$$\dot{z} = \mathrm{i}\omega z, \tag{1.53}$$

which obviously has the exact solution

$$z(t) = z_0 \mathrm{e}^{\mathrm{i}\omega t}. \tag{1.54}$$

As a simple algorithm for numerical integration we use the iteration

$$z((j+1)\Delta t) = z_{j+1} = z_j + \mathrm{i}\omega\Delta t z_j = (1 + \mathrm{i}\omega\Delta t) z_j. \tag{1.55}$$

Since

$$|1 + \mathrm{i}\omega\Delta t| = \sqrt{1 + \omega^2\Delta t^2} > 1, \tag{1.56}$$

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 (page 135)

$$\dot{z}\left(t + \frac{\Delta t}{2}\right) = \mathrm{i}\omega z\left(t + \frac{\Delta t}{2}\right) \tag{1.57}$$

and making the approximation (page 136)

$$z\left(t + \frac{\Delta t}{2}\right) \approx \frac{z(t) + z(t + \Delta t)}{2}. \tag{1.58}$$

This gives the implicit equation

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

which can be solved by

$$z_{j+1} = \frac{1 + \frac{\mathrm{i}\omega\Delta t}{2}}{1 - \frac{\mathrm{i}\omega\Delta t}{2}} z_j. \tag{1.60}$$

Now we have

$$\left|\frac{1+\frac{\mathrm{i}\omega\Delta t}{2}}{1-\frac{\mathrm{i}\omega\Delta t}{2}}\right| = \frac{\sqrt{1+\frac{\omega^2\Delta t^2}{4}}}{\sqrt{1+\frac{\omega^2\Delta t^2}{4}}} = 1, \tag{1.61}$$

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

$$z(t+\Delta t) = z(t)\mathrm{e}^{\mathrm{i}\omega\Delta t} = z(t)\left(1+\mathrm{i}\omega\Delta t - \frac{\omega^2\Delta t^2}{2} + \frac{-\mathrm{i}\omega^3\Delta t^3}{6} - \cdots\right) \tag{1.62}$$

is approximated by

$$\begin{aligned} z(t+\Delta t) &\approx z(t)\frac{1+\frac{\mathrm{i}\omega\Delta t}{2}}{1-\frac{\mathrm{i}\omega\Delta t}{2}} \\ &= z(t)\left(1+\frac{\mathrm{i}\omega\Delta t}{2}\right)\left(1+\frac{\mathrm{i}\omega\Delta t}{2} - \frac{\omega^2\Delta t^2}{4} - \frac{\mathrm{i}\omega\Delta t^3}{8} + \cdots\right) &(1.63)\\ &= z(t)\left(1+\mathrm{i}\omega\Delta t - \frac{\omega^2\Delta t^2}{2} + \frac{-\mathrm{i}\omega^3\Delta t^3}{4} - \cdots\right), &(1.64)\end{aligned}$$

which deviates from the exact solution by a term of the order $O(\Delta t^3)$, hence the error order of this algorithm is $O(\Delta t^3)$. Integration up to a total time $T = N\Delta t$ produces a final 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 $x = 2^{-\mathrm{M}}$ 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

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

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.

Problem 1.3 Truncation Error

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

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

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

Chapter 2
Interpolation

Experiments usually produce a discrete set of data points. If additional data points are needed, for instance to draw a continuous curve or to change the sampling frequency of audio or video signals, interpolation methods are necessary. But interpolation is also helpful to develop more sophisticated numerical methods for the calculation of numerical derivatives and integrals.

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:

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

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

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

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

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

Important examples are

- polynomials

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

- trigonometric functions

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

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_2,

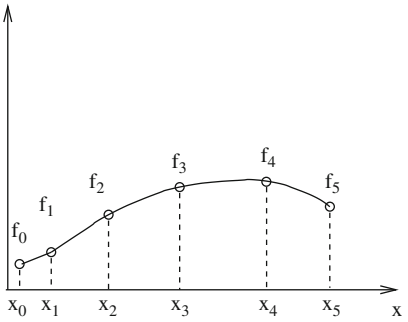


Fig. 2.1 Interpolating function

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

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

Important examples for nonlinear interpolating functions are

- rational functions

$$\frac{a_0 + a_1 x + \cdots + a_n x^n}{b_0 + b_1 x + \cdots + b_m x^m} \tag{2.7}$$

- exponential functions

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

2.2 Polynomial Interpolation

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

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

2.2.1 Lagrange Polynomials

Lagrange polynomials [3] are defined as

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

They are of degree n and have the property

$$L_i(x_k) = \delta_{i,k}. \tag{2.11}$$

The interpolating polynomial is given in terms of Lagrange polynomials by

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

2.2.2 Newton's Divided Differences

For numerical purposes the method of divided differences [4] has advantages. We rewrite

$$f(x) = f(x_0) + \frac{f(x) - f(x_0)}{x - x_0}(x - x_0). \tag{2.13}$$

With the first-order divided difference

$$[x, x_0] = \frac{f(x) - f(x_0)}{x - x_0}, \tag{2.14}$$

this becomes

$$[x, x_0] = [x_1, x_0] + \frac{[x, x_0] - [x_1, x_0]}{x - x_1}(x - x_1), \tag{2.15}$$

and with the second-order divided difference

$$\begin{aligned}[x, x_0, x_1] &= \frac{[x, x_0] - [x_1, x_0]}{x - x_1} = \frac{f(x) - f(x_0)}{(x - x_0)(x - x_1)} - \frac{f(x_1) - f(x_0)}{(x_1 - x_0)(x - x_1)} \\ &= \frac{f(x)}{(x - x_0)(x - x_1)} + \frac{f(x_1)}{(x_1 - x_0)(x_1 - x)} + \frac{f(x_0)}{(x_0 - x_1)(x_0 - x)},\end{aligned} \tag{2.16}$$

we have

$$f(x) = f(x_0) + (x - x_0)[x_1, x_0] + (x - x_0)(x - x_1)[x, x_0, x_1]. \tag{2.17}$$

Higher order divided differences are defined recursively by

$$[x_1 x_2 \ldots x_{r-1} x_r] = \frac{[x_1 x_2 \ldots x_{r-1}] - [x_2 \ldots x_{r-1} x_r]}{x_1 - x_r}. \tag{2.18}$$

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

$$[x_1 x_2 \dots x_r] = \sum_{k=1}^{r} \frac{f(x_k)}{\prod_{i \neq k}(x_k - x_i)}. \tag{2.19}$$

Finally we have

$$f(x) = p(x) + q(x) \tag{2.20}$$

with a polynomial of degree n

$$\begin{aligned} p(x) = f(x_0) + [x_1, x_0](x - x_0) + [x_2 x_1 x_0](x - x_0)(x - x_1) + \cdots \\ \cdots + [x_n x_{n-1} \dots x_0](x - x_0)(x - x_1) \cdots (x - x_{n-1}), \end{aligned} \tag{2.21}$$

and the function

$$q(x) = [x x_n \cdots x_0](x - x_0) \cdots (x - x_n). \tag{2.22}$$

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

2.2.3 Interpolation Error

The error of the interpolation can be estimated with 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 of the x_i with

$$q(\overline{x}) = \prod_{i=0}^{n} (\overline{x} - x_i) \frac{f^{(n+1)}(\xi)}{(n+1)!}. \tag{2.23}$$

From a discussion of the function

$$\omega(x) = \prod_{i=0}^{n} (\overline{x} - x_i), \tag{2.24}$$

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)

$$f(x) = \sin(x) \quad x_i = 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}, 2\pi. \tag{2.25}$$

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

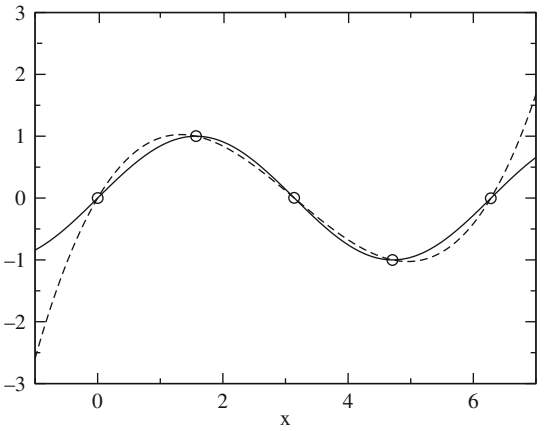


Fig. 2.2 Interpolating polynomial. The interpolated function (*solid curve*) and the interpolating polynomial (*broken curve*) for the example (2.25) are compared

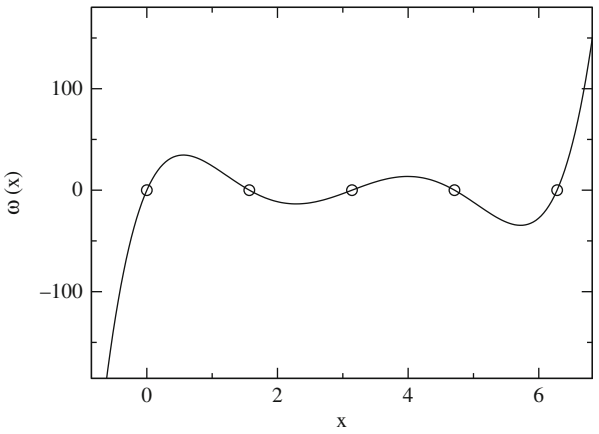


Fig. 2.3 Interpolation error. The polynomial $\omega(x)$ is shown for the example (2.25). Its roots x_i are given by the x values of the sample points (*circles*). Inside the interval $x_0 \ldots x_4$ the absolute value of ω is bounded by $|\omega(x)| \leq 35$ whereas outside the interval it increases very rapidly

$$|f(x) - p(x)| \leq |\omega(x)| \frac{1}{120} \leq \frac{35}{120} \approx 0.3, \tag{2.26}$$

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

Algorithm

The divided differences are arranged in the following way:

$$\begin{array}{llll} f_0 & & & \\ f_1 & [x_0x_1] & & \\ \vdots & \vdots & \ddots & \\ f_{n-1} & [x_{n-2}x_{n-1}] & [x_{n-3}x_{n-2}x_{n-1}] & \\ f_n & [x_{n-1}x_n] & [x_{n-2}x_{n-1}x_n] & \cdots\ [x_0x_1\cdots x_{n-1}x_n] \end{array} . \tag{2.27}$$

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 [5] 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 $x_0 \ldots x_k$, which will be denoted as $P_{0,1,\ldots,k}(x)$. Obviously

$$P_{0,1,\ldots,k}(x) = \frac{(x - x_0)P_{1\ldots k}(x) - (x - x_k)P_{0\ldots k-1}(x)}{x_k - x_0}, \tag{2.28}$$

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

$$\frac{(x - x_0)f(x) - (x - x_k)f(x)}{x_k - x_0} = f(x). \tag{2.29}$$

For $x = x_0$ we have

$$\frac{-(x_0 - x_k)f(x)}{x_k - x_0} = f(x), \tag{2.30}$$

and finally for $x = x_k$

$$\frac{(x_k - x_0) f(x)}{x_k - x_0} = f(x). \tag{2.31}$$

Algorithm

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

$$\begin{array}{lllll} P_0 & & & & \\ P_1 & P_{01} & & & \\ P_2 & P_{12} & P_{012} & & \\ \vdots & \vdots & \vdots & \ddots & \\ P_n & P_{n-1,n} & P_{n-2,n-1,n} & \cdots & P_{01\cdots n} \end{array} . \tag{2.32}$$

The first column contains the function values $P_i(x) = f_i$. The value $P_{0,1\ldots,n}$ can be calculated using a one-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];
```

2.3 Spline Interpolation

Polynomials are not well suited for interpolation over a larger range. Often spline functions are superior which are piecewise defined polynomials [6, 7]. The simplest case is a linear spline which just connects the sampling points by straight lines:

$$p_i(x) = y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i}(x - x_i), \tag{2.33}$$

$$s(x) = p_i(x) \quad \text{where } x_i \le x < x_{i+1}. \tag{2.34}$$

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

$$p_i(x) = \alpha_i + \beta_i(x - x_i) + \gamma_i(x - x_i)^2 + \delta_i(x - x_i)^3. \tag{2.35}$$

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

$$
\begin{aligned}
& i = 0, \ldots, n-1, \\
& p_i(x_{i+1}) = p_{i+1}(x_{i+1}), && (2.36)\\
& p_i'(x_{i+1}) = p_{i+1}'(x_{i+1}), && (2.37)\\
& p_i''(x_{i+1}) = p_{i+1}''(x_{i+1}). && (2.38)
\end{aligned}
$$

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 $s''(x_0) = s''(x_n)$, $s'(x_0) = s'(x_n)$, or given derivative values $s'(x_0)$ and $s'(x_n)$ are often used. The second derivative is a linear function [2]

$$p_i''(x) = 2\gamma_i + 6\delta_i(x - x_i), \tag{2.39}$$

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

$$p_i''(x) = M_{i+1}\frac{(x - x_i)}{h_{i+1}} + M_i\frac{(x_{i+1} - x)}{h_{i+1}} \quad i = 0 \ldots n-1, \tag{2.40}$$

since

$$p_i''(x_i) = M_i\frac{x_{i+1} - x_i}{h_{i+1}} = s''(x_i), \tag{2.41}$$

$$p_i''(x_{i+1}) = M_{i+1}\frac{(x_{i+1} - x_i)}{h_{i+1}} = s''(x_{i+1}). \tag{2.42}$$

Integration gives with the two constants A_i and B_i

$$p_i'(x) = M_{i+1}\frac{(x - x_i)^2}{2h_{i+1}} - M_i\frac{(x_{i+1} - x)^2}{2h_{i+1}} + A_i \tag{2.43}$$

$$p_i(x) = M_{i+1}\frac{(x - x_i)^3}{6h_{i+1}} + M_i\frac{(x_{i+1} - x)^3}{6h_{i+1}} + A_i(x - x_i) + B_i. \tag{2.44}$$

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

$$M_i\frac{h_{i+1}^2}{6} + B_i = y_i, \tag{2.45}$$

$$M_{i+1}\frac{h_{i+1}^2}{6} + A_i h_{i+1} + B_i = y_{i+1}, \tag{2.46}$$

and hence

$$B_i = y_i - M_i\frac{h_{i+1}^2}{6}, \tag{2.47}$$

$$A_i = \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{h_{i+1}}{6}(M_{i+1} - M_i). \tag{2.48}$$

Now the polynomial is

$$\begin{aligned} p_i(x) &= \frac{M_{i+1}}{6h_{i+1}}(x-x_i)^3 - \frac{M_i}{6h_{i+1}}(x-x_i-h_{i+1})^3 + A_i(x-x_i) + B_i \\ &= (x-x_i)^3\left(\frac{M_{i+1}}{6h_{i+1}} - \frac{M_i}{6h_{i+1}}\right) + \frac{M_i}{6h_{i+1}}3h_{i+1}(x-x_i)^2 \\ &\quad +(x-x_i)\left(A_i - \frac{M_i}{6h_{i+1}}3h_{i+1}^2\right) + B_i + \frac{M_i}{6h_{i+1}}h_{i+1}^3. \end{aligned} \tag{2.49}$$

Comparison with

$$p_i(x) = \alpha_i + \beta_i(x-x_i) + \gamma_i(x-x_i)^2 + \delta_i(x-x_i)^3 \tag{2.50}$$

gives

$$\alpha_i = B_i + \frac{M_i}{6}h_{i+1}^2 = y_i, \tag{2.51}$$

$$\beta_i = A_i - \frac{h_{i+1}M_i}{2} = \frac{y_{i+1}-y_i}{h_{i+1}} - h_{i+1}\frac{M_{i+1}+2M_i}{6}, \tag{2.52}$$

$$\gamma_i = \frac{M_i}{2}, \tag{2.53}$$

$$\delta_i = \frac{M_{i+1}-M_i}{6h_{i+1}}. \tag{2.54}$$

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

$$p_i'(x) = M_{i+1}\frac{(x-x_i)^2}{2h_{i+1}} - M_i\frac{(x_{i+1}-x)^2}{2h_{i+1}} + \frac{y_{i+1}-y_i}{h_{i+1}} - \frac{h_{i+1}}{6}(M_{i+1}-M_i), \tag{2.55}$$

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

$$\begin{aligned} &M_i\frac{h_i}{2} + \frac{y_i-y_{i-1}}{h_i} - \frac{h_i}{6}(M_i-M_{i-1}) \\ &\quad = -M_i\frac{h_{i+1}}{2} + \frac{y_{i+1}-y_i}{h_{i+1}} - \frac{h_{i+1}}{6}(M_{i+1}-M_i), \end{aligned} \tag{2.56}$$

$$M_i\frac{h_i}{3} + M_{i-1}\frac{h_i}{6} + M_i\frac{h_{i+1}}{3} + M_{i+1}\frac{h_{i+1}}{6} = \frac{y_{i+1}-y_i}{h_{i+1}} - \frac{y_i-y_{i-1}}{h_i}, \tag{2.57}$$

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

$$\lambda_i = \frac{h_{i+1}}{h_i + h_{i+1}}, \tag{2.58}$$

$$\mu_i = 1 - \lambda_i = \frac{h_i}{h_i + h_{i+1}}, \tag{2.59}$$

$$d_i = \frac{6}{h_i + h_{i+1}} \left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i} \right), \tag{2.60}$$

we have

$$\mu_i M_{i-1} + 2M_i + \lambda_i M_{i+1} = d_i \quad i = 1 \ldots n-1. \tag{2.61}$$

We define for natural boundary conditions

$$\lambda_0 = 0 \quad \mu_n = 0 \quad d_0 = 0 \quad d_n = 0, \tag{2.62}$$

and in case of given derivative values

$$\lambda_0 = 1 \quad \mu_n = 1 \quad d_0 = \frac{6}{h_1} \left(\frac{y_1 - y_0}{h_1} - y_0' \right) \quad d_n = \frac{6}{h_n} \left(y_n' - \frac{y_n - y_{n-1}}{h_n} \right). \tag{2.63}$$

The system of equation has the form

$$\begin{bmatrix} 2 & \lambda_0 & & & & \\ \mu_1 & 2 & \lambda_1 & & & \\ & \mu_2 & 2 & \lambda_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mu_{n-1} & 2 & \lambda_{n-1} \\ & & & & \mu_n & 2 \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ M_2 \\ \vdots \\ M_{n-1} \\ M_n \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix}. \tag{2.64}$$

For periodic boundary conditions we define

$$\lambda_n = \frac{h_1}{h_1 + h_n} \quad \mu_n = 1 - \lambda_n \quad d_n = \frac{6}{h_1 + h_n} \left(\frac{y_1 - y_n}{h_1} - \frac{y_n - y_{n-1}}{h_n} \right), \tag{2.65}$$

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

$$\begin{bmatrix} 2 & \lambda_1 & & & & \mu_1 \\ \mu_2 & 2 & \lambda_2 & & & \\ & \mu_3 & 2 & \lambda_3 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mu_{n-1} & 2 & \lambda_{n-1} \\ \lambda_n & & & & \mu_n & 2 \end{bmatrix} \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ \vdots \\ M_{n-1} \\ M_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_{n-1} \\ d_n \end{bmatrix}. \tag{2.66}$$

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

2.4 Multivariate Interpolation

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

$$\begin{aligned} p(x_i + h_x, y_i + h_y) &= p(x_i + h_x, y_i) + h_y \frac{p(x_i + h_x, y_{i+1}) - p(x_i + h_x, y_i)}{y_{i+1} - y_i} \\ &= f(x_i, y_i) + h_x \frac{f(x_{i+1}, y_i) - f(x_i, y_i)}{x_{i+1} - x_i} \\ &+ h_y \frac{f(x_i, y_{i+1}) + h_x \frac{f(x_{i+1}, y_{i+1}) - f(x_i, y_{i+1})}{x_{i+1} - x_i} - f(x_i, y_i) - h_x \frac{f(x_{i+1}, y_i) - f(x_i, y_i)}{x_{i+1} - x_i}}{y_{i+1} - y_i}, \end{aligned} \tag{2.67}$$

which can be written as a two-dimensional polynomial

$$p(x_i + h_x, y_i + h_y) = a_{00} + a_{10} h_x + a_{10} h_y + a_{11} h_x h_y, \tag{2.68}$$

with

$$\begin{aligned} a_{00} &= f(x_i, y_i), \\ a_{10} &= \frac{f(x_{i+1}, y_i) - f(x_i, y_i)}{x_{i+1} - x_i}, \\ a_{01} &= \frac{f(x_i, y_{i+1}) - f(x_i, y_i)}{y_{i+1} - y_i}, \\ a_{11} &= \frac{f(x_{i+1}, y_{i+1}) - f(x_i, y_{i+1}) - f(x_{i+1}, y_i) + f(x_i, y_i)}{(x_{i+1} - x_i)(y_{i+1} - y_i)}. \end{aligned} \tag{2.69}$$

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:[2]

$$f_{i,j} = f(ih_x, jh_y) \quad \text{with } 0 \le i < N_x \quad 0 \le j < N_y. \tag{2.70}$$

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

[2] A typical task of image processing.

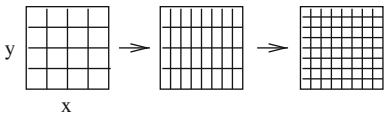


Fig. 2.4 Bispline interpolation

First perform spline interpolation in x-direction for each data row j to calculate new data sets (Fig. 2.4)

$$f_{i',j} = s(x_{i'}, f_{ij}, 0 \le i < N_x) \quad 0 \le j \le N_y \quad 0 \le i' < N'_x \tag{2.71}$$

and then interpolate in y-direction to obtain the final high resolution data:

$$f_{i',j'} = s(y_{j'}, f_{i'j}, 0 \le j < N_y) \quad 0 \le i' < N'_x \quad 0 \le j' < N'_y. \tag{2.72}$$

Problems

Problem 2.1 Polynomial Interpolation

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

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

a linear spline which connects successive points by straight lines

$$s_i(x) = a_i + b_i(x - x_i) \quad \text{for } x_i \le x \le x_{i+1}$$

or a cubic spline

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

with natural boundary conditions

$$s''(x_n) = s''(x_0) = 0$$

(a) Interpolate the following data in the range (Table 2.1)
$-1.5 < x < 0$.
(b) Now add some more sample points (Table 2.2)
for $-1.5 < x < 4.5$
(c) 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.

Table 2.1 Zener diode voltage/current data

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

Table 2.2 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

(d) Investigate the oscillatory behavior for a discontinuous pulse or step function as given by the following data table (Table 2.3):

Table 2.3 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

Problem 2.3 Two-Dimensional Interpolation

This computer experiment uses bilinear interpolation or bicubic spline interpolation to interpolate the data table (Table 2.4)
on a finer grid $\Delta x = \Delta y = 0.1$.

Table 2.4 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

Chapter 3
Numerical Differentiation

For more complex problems analytical derivatives are not always available and have to be approximated by numerical methods. If numerical precision plays a role a simple difference quotient is not sufficient and more accurate methods are necessary which will be discussed in this chapter.

3.1 Simple Forward Difference

The simplest method approximates the derivative by the quotient of finite differences

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

The truncation error can be estimated from the Taylor series expansion

$$\begin{aligned}\frac{f(x+h)-f(x)}{h} &= \frac{f(x)+hf'(x)+\frac{h^2}{2}f''(x)+\cdots-f(x)}{h}\\ &= f'(x)+\frac{h}{2}f''(x)+\cdots .\end{aligned} \tag{3.2}$$

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

$$\begin{aligned}\widetilde{\Delta f} &= fl_-(f(x+h)(1+\varepsilon_1),\, f(x)(1+\varepsilon_2))\\ &= (\Delta f + f(x+h)\varepsilon_1 - f(x)\varepsilon_2)(1+\varepsilon_3)\\ &= \Delta f + \Delta f\varepsilon_3 + f(x+h)\varepsilon_1 - f(x)\varepsilon_2 + \cdots ,\end{aligned} \tag{3.3}$$

$$\begin{aligned}fl_\div(\widetilde{\Delta f}, h(1+\varepsilon_4)) &= \frac{\Delta f + \Delta f\varepsilon_3 + f(x+h)\varepsilon_1 - f(x)\varepsilon_2}{h(1+\varepsilon_4)}(1+\varepsilon_5)\\ &= \frac{\Delta f}{h}(1+\varepsilon_5-\varepsilon_4+\varepsilon_3)+\frac{f(x+h)}{h}\varepsilon_1-\frac{f(x)}{h}\varepsilon_2.\end{aligned} \tag{3.4}$$

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_3,

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

$$\frac{\left|\frac{\widetilde{\Delta f}}{\Delta x} - \frac{\Delta f}{\Delta x}\right|}{\frac{\Delta f}{\Delta x}} \leq 3\varepsilon_{\mathrm{M}} + \left|\frac{f(x)}{\frac{\Delta f}{\Delta x}}\right| 2\frac{\varepsilon_{\mathrm{M}}}{h}. \tag{3.5}$$

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

$$\frac{h}{2}|f''(x)| = |f(x)|\frac{2\varepsilon_{\mathrm{M}}}{h}. \tag{3.6}$$

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

$$h_0 = \sqrt{\varepsilon_{\mathrm{M}}} \approx 10^{-8}$$

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

3.2 Symmetrical Difference Quotient

Accuracy is much higher if a symmetrical difference quotient is used (Fig. 3.1):

$$\begin{aligned}\frac{\Delta f}{\Delta x} &= \frac{f(x+\frac{h}{2}) - f(x-\frac{h}{2})}{h} \\ &= \frac{f(x) + \frac{h}{2}f'(x) + \frac{h^2}{8}f''(x) + \cdots - (f(x) - \frac{h}{2}f'(x) + \frac{h^2}{8}f''(x) + \cdots)}{h} \\ &= f'(x) + \frac{h^2}{24}f'''(x) + \cdots .\end{aligned} \tag{3.7}$$

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

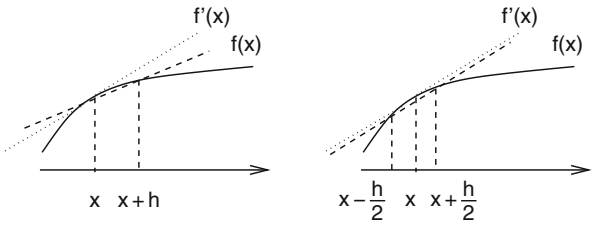


Fig. 3.1 Difference quotient. The symmetric difference quotient (*right side*) approximates the derivative (*dotted*) much more accurately than the single-sided difference quotient (*left side*)

$$\frac{h^2}{24}|f'''(x)| = |f(x)|\frac{2\varepsilon_{\mathrm{M}}}{h}, \tag{3.8}$$

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

$$h_0 = \sqrt[3]{48\varepsilon_{\mathrm{M}}} \approx 10^{-5}. \tag{3.9}$$

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:

$$D(h) = \frac{f(x+h) - f(x-h)}{2h} = f'(x) + \frac{h^2}{3!}f'''(x) + \frac{h^4}{5!}f^{(5)}(x) + \cdots. \tag{3.10}$$

The extrapolation method [8] uses a series of step widths, e.g.,

$$h_{i+1} = \frac{h_i}{2}, \tag{3.11}$$

and calculates an estimate of $D(0)$ by polynomial interpolation. 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:

$$p(h) = D_0\frac{h^2 - \frac{h_0^2}{4}}{h_0^2 - \frac{h_0^2}{4}} + D_1\frac{h^2 - h_0^2}{\frac{h_0^2}{4} - h_0^2}. \tag{3.12}$$

Extrapolation for $h = 0$ gives

$$p(0) = -\frac{1}{3}D_0 + \frac{4}{3}D_1. \tag{3.13}$$

Taylor series expansion shows

$$\begin{aligned} p(0) &= -\frac{1}{3}\left(f'(x) + \frac{h_0^2}{3!}f'''(x) + \frac{h_0^4}{5!}f^{(5)}(x) + \cdots\right) + \\ &\quad + \frac{4}{3}\left(f'(x) + \frac{h_0^2}{4\cdot 3!}f'''(x) + \frac{h_0^4}{16\cdot 5!}f^{(5)}(x) + \cdots\right) \qquad (3.14) \\ &= f'(x) - \frac{1}{4}\frac{h_0^4}{5!}f^{(5)}(x) + \cdots \qquad (3.15) \end{aligned}$$

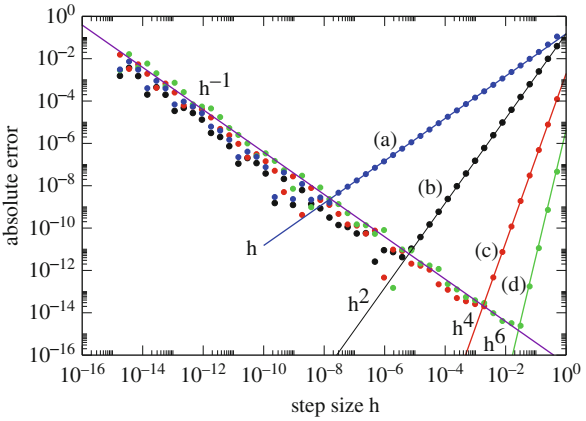


Fig. 3.2 Numerical differentiation. The derivative $\frac{\mathrm{d}}{\mathrm{d}x}\sin(x)$ is calculated numerically using algorithms with increasing error order (3.1(a), 3.7(b), 3.13(c), 3.17(d)). For very small step sizes the error increases as h^{-1} due to rounding errors

that the error order is $O(h_0^4)$. For three step widths $h_0 = 2h_1 = 4h_2$ we obtain the polynomial of second order (in h^2) (Fig. 3.2)

$$p(h) = D_0 \frac{(h^2 - \frac{h_0^2}{4})(h^2 - \frac{h_0^2}{16})}{(h_0^2 - \frac{h_0^2}{4})(h_0^2 - \frac{h_0^2}{16})} + D_1 \frac{(h^2 - h_0^2)(h^2 - \frac{h_0^2}{16})}{(\frac{h_0^2}{4} - h_0^2)(\frac{h_0^2}{4} - \frac{h_0^2}{16})} + D_2 \frac{(h^2 - h_0^2)(h^2 - \frac{h_0^2}{4})}{(\frac{h_0^2}{16} - h_0^2)(\frac{h_0^2}{16} - \frac{h_0^2}{4})} \tag{3.16}$$

and the improved expression

$$\begin{aligned} p(0) &= D_0 \frac{\frac{1}{64}}{\frac{3}{4} \cdot \frac{15}{16}} + D_1 \frac{\frac{1}{16}}{\frac{-3}{4} \cdot \frac{3}{16}} + D_2 \frac{\frac{1}{4}}{\frac{-15}{16} \cdot \frac{-3}{16}} = \\ &= \frac{1}{45} D_0 - \frac{4}{9} D_1 + \frac{64}{45} D_2 = f'(x) + O(h_0^6). \end{aligned} \tag{3.17}$$

Often used is the following series of step widths:

$$h_i^2 = \frac{h_0^2}{2^i}. \tag{3.18}$$

The Neville method

$$P_{i\ldots k}(h^2) = \frac{(h^2 - \frac{h_0^2}{2^i}) P_{i+1\ldots k}(h^2) - (h^2 - \frac{h_0^2}{2^k}) P_{i\ldots k-1}(h^2)}{\frac{h_0^2}{2^k} - \frac{h_0^2}{2^i}} \tag{3.19}$$

gives for $h = 0$

$$P_{i\ldots k} = \frac{P_{i\ldots k-1} - 2^{k-i} P_{i+1\ldots k}}{1 - 2^{k-i}} \tag{3.20}$$

which can be written as

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

and can be calculated according to the following scheme:

$$\begin{array}{llll} P_0 = D(h^2) & P_{01} & P_{012} & P_{0123} \\ P_1 = D(\frac{h^2}{2}) & P_{12} & P_{123} & \\ P_2 = D(\frac{h^2}{4}) & P_{23} & & \\ \vdots & \vdots & \vdots & \ddots \end{array} . \tag{3.22}$$

Here the values of the polynomials are arranged in matrix form

$$P_{i\cdots k} = T_{i,k-i} = T_{i,j} \tag{3.23}$$

with the recursion formula

$$T_{i,j} = T_{i+1,j-1} + \frac{T_{i,j-1} - T_{i+1,j}}{1 - 2^j}. \tag{3.24}$$

3.4 Higher Derivatives

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

$$x_n = x_0 + nh = \ldots, x_0 - 2h, x_0 - h, x_0, x_0 + h, x_0 + 2h, \ldots. \tag{3.25}$$

From the second-order polynomial

$$\begin{aligned} p(x) = {} & y_{-1} \frac{(x - x_0)(x - x_1)}{(x_{-1} - x_0)(x_{-1} - x_1)} + y_0 \frac{(x - x_{-1})(x - x_1)}{(x_0 - x_{-1})(x_0 - x_1)} \\ & + y_1 \frac{(x - x_{-1})(x - x_0)}{(x_1 - x_{-1})(x_1 - x_0)} = \\ = {} & y_{-1} \frac{(x - x_0)(x - x_1)}{2h^2} + y_0 \frac{(x - x_{-1})(x - x_1)}{-h^2} \\ & + y_1 \frac{(x - x_{-1})(x - x_0)}{2h^2} \end{aligned} \tag{3.26}$$

we calculate the derivatives

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

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

which are evaluated at x_0:

$$f'(x_0) \approx p'(x_0) = -\frac{1}{2h}y_{-1} + \frac{1}{2h}y_1 = \frac{f(x_0 + h) - f(x_0 - h)}{2h}, \tag{3.29}$$

$$f''(x_0) \approx p''(x_0) = \frac{f(x_0 - h) - 2f(x_0) + f(x_0 + h)}{h^2}. \tag{3.30}$$

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

$$f'(x_0) \approx \frac{f(x_0 - 2h) - 8f(x_0 - h) + 8f(x_0 + h) - f(x_0 + 2h)}{12h}, \tag{3.31}$$

$$f''(x_0) \approx \frac{-f(x_0 - 2h) + 16f(x_0 - h) - 30f(x_0) + 16f(x_0 + h) - f(x_0 + 2h)}{12h^2}, \tag{3.32}$$

$$f'''(x_0) \approx \frac{-f(x_0 - 2h) + 2f(x_0 - h) - 2f(x_0 + h) + f(x_0 + 2h)}{2h^3}, \tag{3.33}$$

$$f^{(4)}(x_0) \approx \frac{f(x_0 - 2h) - 4f(x_0 - h) + 6f(x_0 + h) - 4f(x_0 + h) + f(x_0 + 2h)}{h^4}. \tag{3.34}$$

3.5 More Dimensions

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 samples 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

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

and the Laplace operator

$$\begin{aligned}\nabla^2 f(x_0, y_0) &\approx \nabla^2 p(x_0, y_0)\\ &= \frac{1}{h^2}\left(f(x_0, y_0 + h) + f(x_0, y_0 - h) + f(x_0, y_0 + h) + f(x_0, y_0 - h) - 4f(x_0, y_0)\right).\end{aligned} \tag{3.40}$$

Problems

Problem 3.1 Numerical Differentiation

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

(a) the single-sided difference quotient

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

(b) the symmetrical difference quotient

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

(c) higher order approximations which can be derived using the extrapolation method

$$\begin{aligned}&-\frac{1}{3}D_h f(x) + \frac{4}{3}D_{h/2} f(x)\\ &\frac{1}{45}D_h f(x) - \frac{4}{9}D_{h/2} f(x) + \frac{64}{45}D_{h/4} f(x)\end{aligned}$$

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 Coulombic interaction between two electrons. Integration is also the elementary step in solving equations of motion. In general a definite integral can be approximated numerically as the weighted average over a finite number of function values:

$$\int_a^b f(x)\mathrm{d}x \approx \sum_{x_i} w_i f(x_i). \tag{4.1}$$

Specific sets of sample points x_i and weight factors w_i are known as "integral rules."

4.1 Equidistant Sample Points

For equidistant points

$$x_i = a + ih \quad i = 0 \ldots n \quad h = \frac{b-a}{n}, \tag{4.2}$$

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

$$p(x) = \sum_{i=0}^{n} f_i \prod_{k=0,k\neq i}^{n} \frac{x - x_k}{x_i - x_k}. \tag{4.3}$$

Integration of the polynomial gives

$$\int_a^b p(x)\mathrm{d}x = \sum_{i=0}^{n} f_i \int_a^b \prod_{k=0,k\neq i}^{n} \frac{x - x_k}{x_i - x_k}\mathrm{d}x. \tag{4.4}$$

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_4,

After substituting

$$\begin{aligned} x &= a + hs, \\ x - x_k &= h(s - k), \\ x_i - x_k &= (i - k)h, \end{aligned} \tag{4.5}$$

we have

$$\int_a^b \prod_{k=0, k\neq i}^{n} \frac{x - x_k}{x_i - x_k} \mathrm{d}x = \int_0^n \prod_{k=0, k\neq i}^{n} \frac{s - k}{i - k} h \mathrm{d}s = h\alpha_i, \tag{4.6}$$

and hence

$$\int_a^b p(x)\mathrm{d}x = (b - a) \sum_{i=0}^{n} f_i \alpha_i. \tag{4.7}$$

The α_i are weight factors for the function values f_i.

4.1.1 Newton–Cotes Rules

For $n = 1$ the polynomial is

$$p(x) = f_0 \frac{x - x_1}{x_0 - x_1} + f_1 \frac{x - x_0}{x_1 - x_0}, \tag{4.8}$$

and the integral is

$$\begin{aligned} \int_a^b p(x)\mathrm{d}x &= f_0 \int_0^1 \frac{s - 1}{0 - 1} h \mathrm{d}s + f_1 \int_0^1 \frac{s - 0}{1 - 0} h \mathrm{d}s \\ &= -f_0 h \left(\frac{(1 - 1)^2}{2} - \frac{(0 - 1)^2}{2} \right) + f_1 h \left(\frac{1^2}{2} - \frac{0^2}{2} \right) \\ &= h \frac{f_0 + f_1}{2}, \end{aligned} \tag{4.9}$$

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

$$2h \frac{f_0 + 4f_1 + f_2}{6}. \tag{4.10}$$

Larger n gives further integration rules:

$$\begin{aligned} 3h\,\tfrac{f_0+3f_1+3f_2+f_3}{8} &\quad \text{3/8 rule} \\ 4h\,\tfrac{7f_0+32f_1+12f_2+32f_3+7f_4}{90} &\quad \text{Milne rule} \\ 5h\,\tfrac{19f_0+75f_1+50f_2+50f_3+75f_4+19f_5}{288} & \\ 6h\,\tfrac{41f_0+216f_1+27f_2+272f_3+27f_4+216f_5+41f_6}{840} &\quad \text{Weddle rule} \end{aligned} \tag{4.11}$$

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

4.1.2 Newton–Cotes Expressions for an Open Interval

If the function has a singularity at the end of the interval, it is more convenient to compute the integral from only interior points

$$x_i = a + ih \quad i = 1, 2, \ldots, N \quad h = \frac{b-a}{N+1}. \tag{4.12}$$

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

$$\int_a^b f(x)\mathrm{d}x \approx 2hf_1 = (b-a)\,f\left(\frac{a+b}{2}\right). \tag{4.13}$$

The next two are

$$\frac{3h}{2}(f_1 + f_2), \tag{4.14}$$

$$\frac{4h}{3}(2f_1 - f_2 + 2f_3). \tag{4.15}$$

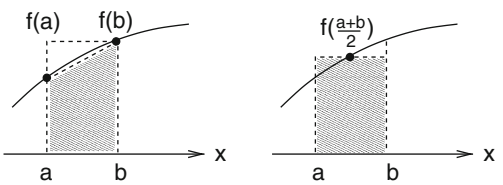


Fig. 4.1 Trapezoidal rule and midpoint rule. The trapezoidal rule (*left*) approximates the integral by the average of the function values at the boundaries. The midpoint rule (*right*) evaluates the function in the center of the interval and has the same error order

4.1.3 Composite Newton–Cotes Formulas

Let us divide the integration range into intervals

$$[x_i, x_{i+1}] \quad x_i = a + ih \quad i = 0 \ldots n \tag{4.16}$$

and use the trapezoidal rule for each interval:

$$I_i = \frac{h}{2}(f(x_i) + f(x_{i+1})). \tag{4.17}$$

This gives the composite trapezoidal rule

$$T_s = h\left(\frac{f(a)}{2} + f(a+h) + \cdots + f(b-h) + \frac{f(b)}{2}\right), \tag{4.18}$$

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:

$$\begin{aligned} S = \frac{h}{3}(&f(a) + 4f(a+h) + 2f(a+2h) + 4f(a+3h) + \cdots \\ &\cdots + 2f(b-2h) + 4f(b-h) + f(b)), \end{aligned} \tag{4.19}$$

with error order $O(h^4)$. (The number of sample points must be even!)
Repeated application of the midpoint rule gives the composite midpoint rule

$$S = 2h(f(a+h) + f(a+3h) + \cdots + f(b-h)), \tag{4.20}$$

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

4.1.4 Extrapolation Method (Romberg Integration)

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

$$\int_{x_0}^{x_n} f(x)\mathrm{d}x - T_s = \alpha_2 h^2 + \alpha_4 h^4 + \cdots + \alpha_{2m-2} h^{2m-2} + O(h^{2m}). \tag{4.21}$$

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

$$\int f(x)\mathrm{d}x - T_s(h) = \alpha_2 h^2 + \alpha_4 h^4 + \cdots , \tag{4.22}$$

$$\int f(x)\mathrm{d}x - T_s(h/2) = \alpha_2 \frac{h^2}{4} + \alpha_4 \frac{h^4}{16} + \cdots , \tag{4.23}$$

$$\int f(x)\mathrm{d}x - \frac{4T_s(h/2) - T_s(h)}{3} = -\alpha_4 \frac{h^4}{4} + \cdots \ . \tag{4.24}$$

More generally, for the series of step widths

$$h_k = \frac{h_0}{2^k}, \tag{4.25}$$

the Neville method gives the recursion for the interpolating polynomial

$$P_{i\ldots k}(h^2) = \frac{\left(h^2 - \frac{h_0^2}{2^{2i}}\right) P_{i+1\ldots k}(h^2) - \left(h^2 - \frac{h_0^2}{2^{2k}}\right) P_{i\ldots k-1}(h^2)}{\frac{h_0^2}{2^{2k}} - \frac{h_0^2}{2^{2i}}}, \tag{4.26}$$

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

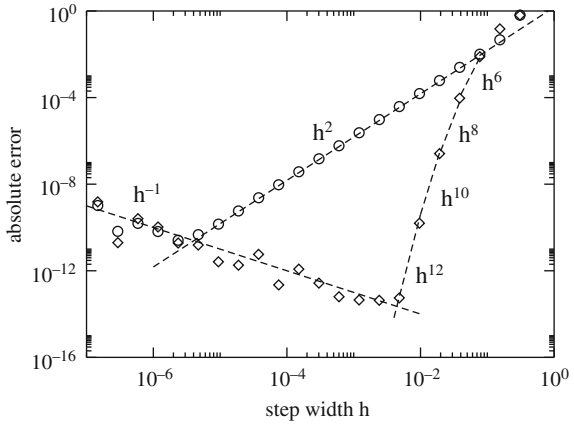


Fig. 4.2 Romberg integration The integral $\int_0^{\pi^2} \sin(x^2)\mathrm{d}x$ is calculated numerically. *Circles* show the absolute error of the composite trapezoidal rule (4.18) for the step size sequence $h_{i+1} = h_i/2$. *Diamonds* show the absolute error of the extrapolated value (4.27). The 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}

$$P_{i\ldots k} = \frac{2^{-2k} P_{i\ldots k-1} - 2^{-2i} P_{i+1\cdots k}}{2^{-2k} - 2^{-2i}} = \frac{P_{i\ldots k-1} - 2^{2k-2i} P_{i+1\ldots k}}{1 - 2^{2k-2i}}$$
$$= P_{i+1\ldots k} + \frac{P_{i\ldots k-1} - P_{i+1\ldots k}}{1 - 2^{2k-2i}}. \tag{4.27}$$

The polynomial values can again be arranged in matrix form

$$\begin{matrix} P_0 & P_{01} & P_{012} & \cdots \\ P_1 & P_{12} & & \\ P_2 & & & \\ \vdots & & & \end{matrix}, \tag{4.28}$$

with

$$T_{i,j} = P_{i\cdots i+j} \tag{4.29}$$

and the recursion formula

$$T_{i,0} = P_i = T_s\left(\frac{h_0}{2^i}\right), \tag{4.30}$$

$$T_{i,j} = T_{i+1,j-1} + \frac{T_{i,j-1} - T_{i+1,j-1}}{1 - 2^{2j}}. \tag{4.31}$$

4.2 Optimized Sample Points

The accuracy of the integration can be improved by optimizing the sample point positions.

4.2.1 Clenshaw–Curtis Expressions

Here the sample points are chosen as the roots

$$x_i = \cos\frac{(2-i)\pi}{2N} \tag{4.32}$$

or as the extrema

$$x_i = \cos\frac{(i-1)\pi}{N-1} \tag{4.33}$$

of the Tschebyscheff polynomials:

$$T_0 = 1 \quad T_n(x) = \frac{\cos(n \arccos(x))}{2^{n-1}}. \tag{4.34}$$

This leads to integration rules of arbitrary order which have only positive weights and are therefore numerically stable.

4.2.2 Gaussian Integration

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

$$\int_a^b f(x)\mathrm{d}x \approx I = \sum_{I=1}^{N} f(x_i)w_i \tag{4.35}$$

and determine the $2n$ parameters x_i and w_i such that a polynomial of order $2n-1$ is integrated exactly. We restrict the integration interval to $[-1, 1]$. The general case $[a, b]$ is then given by a simple change of variables. A scalar product for functions on the interval $[-1, 1]$ is defined by

$$\langle fg\rangle = \int_{-1}^{1} f(x)g(x)\mathrm{d}x, \tag{4.36}$$

and an orthogonal system of polynomials can be found using the Gram–Schmid method:

$$P_0 = 1, \tag{4.37}$$

$$P_1 = x - \frac{P_0}{\langle P_0 P_0\rangle}\int_{-1}^{1} x P_0(x)\mathrm{d}x = x, \tag{4.38}$$

$$\begin{aligned} P_2 &= x^2 - \frac{P_1}{\langle P_1 P_1\rangle}\int_{-1}^{1} x^2 P_1(x)\mathrm{d}x - \frac{P_0}{\langle P_0 P_0\rangle}\int_{-1}^{1} x^2 P_0(x)\mathrm{d}x \\ &= x^2 - \frac{1}{3}, \end{aligned} \tag{4.39}$$

$$\begin{aligned} P_n &= x^n - \frac{P_{n-1}}{\langle P_{n-1} P_{n-1}\rangle}\int_{-1}^{1} x^n P_{n-1}(x)\mathrm{d}x \\ &\quad - \frac{P_{n-2}}{\langle P_{n-2} P_{n-2}\rangle}\int_{-1}^{1} x^n P_{n-2}(x)\mathrm{d}x - \cdots . \end{aligned} \tag{4.40}$$

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

$$\widetilde{p}(x) = \sum_{j=1}^{n} L_j(x)p(x_j). \tag{4.41}$$

Then $p(x)$ can be written as

$$p(x) = \widetilde{p}(x) + (x - x_1)(x - x_2) \cdots (x - x_n) q(x). \tag{4.42}$$

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

$$(x - x_1)(x - x_2) \cdots (x - x_n) = P_n(x). \tag{4.43}$$

Then we have

$$\int_{-1}^{1} (x - x_1)(x - x_2) \cdots (x - x_n) q(x) \mathrm{d}x = 0, \tag{4.44}$$

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

$$\int_{-1}^{1} p(x) \mathrm{d}x = \int_{-1}^{1} \widetilde{p}(x) \mathrm{d}x = \int_{-1}^{1} \sum_{j=1}^{n} p(x_j) L_j(x) \mathrm{d}x = \sum_{j=1}^{n} w_j p(x_j), \tag{4.45}$$

with the weight factors

$$w_j = \int_{-1}^{1} L_j(x) \mathrm{d}x. \tag{4.46}$$

Example The second-order Legendre polynomial

$$P_2(x) = x^2 - \frac{1}{3} \tag{4.47}$$

has two roots

$$x_{1,2} = \pm \sqrt{\frac{1}{3}}. \tag{4.48}$$

The Lagrange polynomials are

$$L_1 = \frac{x - \sqrt{\frac{1}{3}}}{-\sqrt{\frac{1}{3}} - \sqrt{\frac{1}{3}}}, \quad L_2 = \frac{x + \sqrt{\frac{1}{3}}}{\sqrt{\frac{1}{3}} + \sqrt{\frac{1}{3}}}, \tag{4.49}$$

and the weights

$$w_1 = \int_{-1}^{1} L_1 \mathrm{d}x = -\frac{\sqrt{3}}{2}\left(\frac{x^2}{2} - \sqrt{\frac{1}{3}}x\right)_{-1}^{1} = 1, \tag{4.50}$$

$$w_2 = \int_{-1}^{1} L_2 \mathrm{d}x = \frac{\sqrt{3}}{2}\left(\frac{x^2}{2} + \sqrt{\frac{1}{3}}x\right)_{-1}^{1} = 1. \tag{4.51}$$

This gives the integral rule

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx f\left(-\sqrt{\frac{1}{3}}\right) + f\left(\sqrt{\frac{1}{3}}\right). \tag{4.52}$$

For a general integration interval we substitute

$$x = \frac{a+b}{2} + \frac{b-a}{2}u \tag{4.53}$$

and find the approximation

$$\begin{aligned}\int_a^b f(x)\mathrm{d}x &= \int_{-1}^{1} f\left(\frac{a+b}{2} + \frac{b-a}{2}u\right)\frac{b-a}{2}\mathrm{d}u \\ &\approx \frac{b-a}{2}\left(f\left(\frac{a+b}{2} - \frac{b-a}{2}\sqrt{\frac{1}{3}}\right) + f\left(\frac{a+b}{2} + \frac{b-a}{2}\sqrt{\frac{1}{3}}\right)\right).\end{aligned} \tag{4.54}$$

The next higher order Gaussian rule is given by

$$n = 3 : w_1 = w_3 = 5/9,\, w_2 = 8/9,\, x_3 = -x_1 = 0.77459\ldots,\, x_2 = 0. \tag{4.55}$$

Besides these Gaussian (Legendre) expressions further integral rules can be obtained by using other sets of orthogonal polynomials, for instance Laguerre, Hermite, or Jacobi polynomials.

Problems

Problem 4.1 Romberg integration

Use the trapezoidal rule

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

with the step sequence

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

and calculate the elements of the triangular matrix

$$T(i,0) = T(h_i)$$
$$T(i,k) = T(i+1,k-1) + \frac{T(i,k-1) - T(i+1,k-1)}{1 - \frac{h_i^2}{h_{i+k}^2}}$$

to obtain the approximations

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

(a) calculate

$$\int_0^{\pi^2} \sin(x^2)\mathrm{d}x = 0.6773089370468890331\ldots$$

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

$$\int_\varepsilon^1 \frac{\mathrm{d}x}{\sqrt{x}}$$

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

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

Chapter 5
Systems of Inhomogeneous Linear Equations

Many problems in physics involve systems of linear equations

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

or shortly

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

If the dimension of the system is not too large standard methods like Gaussian elimination are sufficient. However, many applications involve very large dimensions and require special iterative methods.

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 $2 \ldots n$

$$\begin{pmatrix} \mathbf{a}_1^t \\ \mathbf{a}_2^t \\ \vdots \\ \mathbf{a}_n^t \end{pmatrix} \rightarrow \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 - l_{21}\mathbf{a}_1^t \\ \vdots \\ \mathbf{a}_n^t - l_{n1}\mathbf{a}_1^t \end{pmatrix}, \tag{5.3}$$

which can be written as a multiplication

$$A^{(1)} = L_1 A, \tag{5.4}$$

with the Frobenius matrix

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_5,

$$L_1 = \begin{pmatrix} 1 & & & \\ -l_{21} & 1 & & \\ -l_{31} & & 1 & \\ \vdots & & & \ddots \\ -l_{n1} & & & 1 \end{pmatrix} \qquad l_{i1} = \frac{a_{i1}}{a_{11}}. \tag{5.5}$$

The result has the form

$$A^{(1)} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n-1} & a_{1n} \\ 0 & a_{22}^{(1)} & \dots & a_{2n-1}^{(1)} & a_{2n}^{(1)} \\ 0 & a_{32}^{(1)} & \dots & \dots & a_{3n}^{(1)} \\ 0 & \vdots & & & \vdots \\ 0 & a_{n2}^{(1)} & \dots & \dots & a_{nn}^{(1)} \end{pmatrix}. \tag{5.6}$$

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

$$A^{(2)} = L_2 A^{(1)} = L_2 L_1 A, \tag{5.7}$$

with

$$L_2 = \begin{pmatrix} 1 & & & \\ 0 & 1 & & \\ 0 & -l_{32} & 1 & \\ \vdots & \vdots & & \ddots \\ 0 & -l_{n2} & & 1 \end{pmatrix} \qquad l_{i2} = \frac{a_{i2}^{(1)}}{a_{22}^{(1)}}. \tag{5.8}$$

The result is

$$A^{(2)} = \begin{pmatrix} a_{11}^{(2)} & a_{12}^{(2)} & a_{13}^{(2)} & \dots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \dots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(2)} & \dots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & a_{n3}^{(2)} & \dots & a_{nn}^{(2)} \end{pmatrix}. \tag{5.9}$$

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

$$A^{(n-1)} = L_{n-1} A^{(n-2)}, \tag{5.10}$$

$$L_{n-1} = \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & -l_{n,n-1} & 1 \end{pmatrix} \qquad l_{n,n-1} = \frac{a_{n,n-1}^{(n-2)}}{a_{n-1,n-1}^{(n-2)}}, \tag{5.11}$$

$$A^{(n-1)} = L_{n-1} L_{n-2} \dots L_2 L_1 A = U, \tag{5.12}$$

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ & u_{22} & u_{23} & \dots & u_{2n} \\ & & u_{33} & \dots & u_{3n} \\ & & & \ddots & \vdots \\ & & & & u_{nn} \end{pmatrix}. \tag{5.13}$$

The transformed system of equations

$$U\mathbf{x} = \mathbf{y} \qquad \mathbf{y} = L_{n-1}L_{n-1}\dots L_2 L_1 \mathbf{b} \tag{5.14}$$

can be solved easily by backward substitution:

$$x_n = \frac{1}{u_{nn}} y_n, \tag{5.15}$$

$$x_{n-1} = \frac{y_{n-1} - x_n u_{n-1,n}}{u_{n-1,n-1}}, \tag{5.16}$$

$$\vdots \tag{5.17}$$

Alternatively the matrices L_i can be inverted:

$$L_1^{-1} = \begin{pmatrix} 1 & & & \\ l_{21} & 1 & & \\ l_{31} & & 1 & \\ \vdots & & & \ddots \\ l_{n1} & & & 1 \end{pmatrix} \cdots L_{n-1}^{-1} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \\ & & l_{n,n-1} & 1 \end{pmatrix}. \tag{5.18}$$

This gives

$$A = L_1^{-1} L_2^{-1} \dots L_{n-1}^{-1} U. \tag{5.19}$$

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

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

$$\vdots$$

$$L = L_1^{-1} L_2^{-1} \dots L_{n-1}^{-1} = \begin{pmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ \vdots & \vdots & \ddots & & \\ l_{n-1,1} & l_{n-1,2} & \dots & 1 & \\ l_{n1} & l_{n2} & \dots & l_{n,n-1} & 1 \end{pmatrix}. \tag{5.20}$$

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

$$A = LU, \tag{5.21}$$

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

$$A\mathbf{x} = LU\mathbf{x} = \mathbf{b} \tag{5.22}$$

in two steps

$$L\mathbf{y} = \mathbf{b}, \tag{5.23}$$

which can be solved from the top

$$y_1 = b_1, \tag{5.24}$$
$$y_2 = b_2 - l_{21} y_1, \tag{5.25}$$
$$\vdots \tag{5.26}$$

and

$$U\mathbf{x} = \mathbf{y}, \tag{5.27}$$

which can be solved from the bottom

$$x_n = \frac{1}{u_{nn}} y_n, \tag{5.28}$$
$$x_{n-1} = \frac{y_{n-1} - x_n u_{n-1,n}}{u_{n-1,n-1}}, \tag{5.29}$$
$$\vdots \tag{5.30}$$

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

$$P \begin{pmatrix} 1 \\ \vdots \\ N \end{pmatrix} = \begin{pmatrix} i_1 \\ \vdots \\ i_N \end{pmatrix}. \tag{5.31}$$

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

$$\frac{a_{ik}}{\sum_j |a_{ij}|} \tag{5.32}$$

as the pivoting element.

5.1.2 Direct LU Decomposition

LU decomposition can be also performed in a different order [9]. For symmetric positive definite matrices there exists the simpler and more efficient Cholesky method which decomposes the matrix into the product LL^t of a lower triangular matrix and its transpose [10].

5.2 QR Decomposition

The Gaussian elimination method can become numerically unstable. An alternative method to solve a system of linear equations uses the decomposition [11]

$$A = QR, \tag{5.33}$$

with a unitary matrix $Q^H Q = 1$ (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 $Q^H = Q^{-1}$:

$$QR\mathbf{x} = A\mathbf{x} = \mathbf{b}, \tag{5.34}$$

$$R\mathbf{x} = Q^H \mathbf{b}. \tag{5.35}$$

Such a system with upper triangular matrix is easily solved (see (5.27)).

QR decomposition can be achieved by a series of unitary transformations (Householder reflections [2] or Givens rotations [11]) or simpler by Gram–Schmidt orthogonalization [2, 11]:

From the decomposition $A = QR$ we have

$$a_{ik} = \sum_{j=1}^{k} q_{ij} r_{jk}, \tag{5.36}$$

$$\mathbf{a}_k = \sum_{j=1}^{k} r_{jk} \mathbf{q}_j, \tag{5.37}$$

which gives the kth column vector $\mathbf{a}_k$ of A as a linear combination of the orthonormal vectors $\mathbf{q}_1 \dots \mathbf{q}_k$. 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:

$$r_{11} := |a_1|, \tag{5.38}$$

$$\mathbf{q}_1 := \frac{\mathbf{a}_1}{r_{11}}. \tag{5.39}$$

For $k = 2, \dots, n$

$$r_{ik} := \mathbf{q}_i \mathbf{a}_k \quad i = 1 \dots k-1 \tag{5.40}$$

$$\mathbf{b}_k := \mathbf{a}_k - r_{1k}\mathbf{q}_1 - \dots - r_{k-1,k}\mathbf{q}_{k-1}, \tag{5.41}$$

$$r_{kk} := |\mathbf{b}_k|, \tag{5.42}$$

$$\mathbf{q}_k := \frac{\mathbf{b}_k}{r_{kk}}. \tag{5.43}$$

Obviously now

$$\mathbf{a}_k = r_{kk}\mathbf{q}_k + r_{k-1,k}\mathbf{q}_{k-1} + \dots + r_{1k}\mathbf{q}_1, \tag{5.44}$$

since as per definition

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

and

$$r_{kk}^2 = |\mathbf{b}_k|^2 = |\mathbf{a}_k|^2 + r_{1k}^2 + \dots + r_{k-1,k}^2 - 2r_{1k}^2 - \dots - 2r_{k-1,k}^2. \tag{5.46}$$

Hence,

$$\mathbf{q}_k \mathbf{a}_k = \frac{1}{r_{kk}}(\mathbf{a}_k - r_{1k}\mathbf{q}_1 \dots r_{k-1,k}\mathbf{q}_{k-1})\mathbf{a}_k = \frac{1}{r_{kk}}(|a_k|^2 - r_{1k}^2 - \dots - r_{k-1,k}^2) = r_{kk}. \tag{5.47}$$

Orthogonality gives

$$\mathbf{q}_i \mathbf{a}_k = 0 \quad i = k+1 \dots n. \tag{5.48}$$

In matrix notation we have finally

$$A = (\mathbf{a}_1 \dots \mathbf{a}_n) = (\mathbf{q}_1 \dots \mathbf{q}_n) \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ & r_{22} & \dots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \end{pmatrix}. \tag{5.49}$$

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

$$\mathbf{b}_k \to \mathbf{b}_k - (\mathbf{q}_1\mathbf{b}_k)\mathbf{q}_1 - \cdots - (\mathbf{q}_{k-1}\mathbf{b}_k)\mathbf{q}_{k-1}. \tag{5.50}$$

5.3 Linear Equations with Tridiagonal Matrix

Linear equations with the form

$$b_1x_1 + c_1x_2 = r_1, \tag{5.51}$$
$$a_ix_{i-1} + b_ix_i + c_ix_{i+1} = r_i \quad i = 2\ldots(n-1),$$
$$a_nx_{n-1} + b_nx_n = r_n \tag{5.52}$$

can be solved very easily with a specialized Gaussian elimination method.[1] 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

$$\beta_2x_2 + c_2x_3 = \rho_2, \tag{5.53}$$

with the abbreviations

$$\beta_2 = b_2 - \frac{c_1a_2}{b_1}, \quad \rho_2 = r_2 - \frac{r_1a_2}{b_1}. \tag{5.54}$$

We iterate this procedure

$$\beta_ix_i + c_ix_{i+1} = \rho_i, \tag{5.55}$$
$$\beta_i = b_i - \frac{c_{i-1}a_i}{\beta_{i-1}}, \quad \rho_i = r_i - \frac{\rho_{i-1}a_i}{\beta_{i-1}}, \tag{5.56}$$

until we reach the nth equation, which becomes simply

$$\beta_nx_n = \rho_n, \tag{5.57}$$
$$\beta_n = b_n - \frac{c_{n-1}a_n}{\beta_{n-1}}, \quad \rho_n = r_n - \frac{\rho_{n-1}a_n}{\beta_{n-1}}. \tag{5.58}$$

Now we immediately have

$$x_n = \frac{\rho_n}{\beta_n}, \tag{5.59}$$

and backward substitution gives

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

$$x_{i-1} = \frac{\rho_{i-1} - c_{i-1}x_i}{\beta_{i-1}} \tag{5.60}$$

and finally

$$x_1 = \frac{r_1 - c_1 x_2}{\beta_2}. \tag{5.61}$$

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

$$L = \begin{pmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & l_3 & 1 & & \\ & & \ddots & \ddots & \\ & & & l_n & 1 \end{pmatrix} \qquad U = \begin{pmatrix} \beta_1 & c_1 & & & \\ & \beta_2 & c_2 & & \\ & & \beta_3 & c_3 & \\ & & & \ddots & \\ & & & & \beta_n \end{pmatrix} \tag{5.62}$$

gives

$$LU = \begin{pmatrix} \beta_1 & c_1 & & & & & \\ \ddots & \ddots & & & & & \\ & \ddots & \ddots & & & & \\ & l_i\beta_{i-1} & (l_i c_{i-1} + \beta_i) & c_i & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & & & \\ & & & & l_n\beta_{n-1} & (l_n c_{n-1} + \beta_n) \end{pmatrix}, \tag{5.63}$$

which coincides with the matrix

$$A = \begin{pmatrix} b_1 & c_1 & & & & & \\ a_2 & \ddots & \ddots & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & a_i & b_i & c_i & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & & & a_n & b_n \end{pmatrix} \tag{5.64}$$

if we choose

$$l_i = \frac{a_i}{\beta_{i-1}} \tag{5.65}$$

since then from (5.56)

$$b_i = \beta_i + l_i c_{i-1} \tag{5.66}$$

and

$$l_i \beta_{i-1} = a_i. \tag{5.67}$$

5.4 Cyclic Tridiagonal Systems

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

$$A = \begin{pmatrix} b_1 & c_1 & & & & a_1 \\ a_2 & \ddots & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & a_i & b_i & c_i & \\ & & & \ddots & \ddots & \ddots \\ & & & & a_{n-1} & b_{n-1} & c_{n-1} \\ c_n & & & & & a_n & b_n \end{pmatrix}. \tag{5.68}$$

The system of equations

$$A\mathbf{x} = \mathbf{r} \tag{5.69}$$

can be reduced to a tridiagonal system [12] with the help of the Sherman–Morrison formula [13], which states that if A_0 is an invertible matrix and $\mathbf{u}$, $\mathbf{v}$ are vectors and

$$1 + \mathbf{v}^{\mathrm{T}} A_0^{-1} \mathbf{u} \neq 0, \tag{5.70}$$

then the inverse of the matrix[2]

$$A = A_0 + \mathbf{u}\mathbf{v}^{\mathrm{T}} \tag{5.71}$$

is given by

$$A^{-1} = A_0^{-1} - \frac{A_0^{-1}\mathbf{u}\mathbf{v}^{\mathrm{T}} A_0^{-1}}{1 + \mathbf{v}^T A_0^{-1}\mathbf{u}}. \tag{5.72}$$

We choose

$$\mathbf{u}\mathbf{v}^{\mathrm{T}} = \begin{pmatrix} \alpha \\ 0 \\ \vdots \\ 0 \\ c_n \end{pmatrix} \begin{pmatrix} 1 & 0 & \cdots & 0 & \frac{a_1}{\alpha} \end{pmatrix} = \begin{pmatrix} \alpha & & & a_1 \\ & & & \\ & & & \\ c_n & & & \frac{a_1 c_n}{\alpha} \end{pmatrix}. \tag{5.73}$$

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

Then

$$A_0 = A - \mathbf{u}\mathbf{v}^{\mathrm{T}} = \begin{pmatrix} (b_1-\alpha) & c_1 & & & & 0 \\ a_2 & \ddots & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & a_i & b_i & c_i & \\ & & & \ddots & \ddots & \ddots \\ & & & & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & & & & a_n & (b_n - \frac{a_1 c_n}{\alpha}) \end{pmatrix} \tag{5.74}$$

is tridiagonal. The free parameter α has to be chosen such that the diagonal elements do not become too small. We solve the system (5.69) by solving the two tridiagonal systems

$$\begin{aligned} A_0\mathbf{x}_0 &= \mathbf{r} \\ A_0\mathbf{q} &= \mathbf{u} \end{aligned} \tag{5.75}$$

and compute $\mathbf{x}$ from

$$\mathbf{x} = A^{-1}\mathbf{r} = A_0^{-1}\mathbf{r} - \frac{(A_0^{-1}\mathbf{u})\mathbf{v}^{\mathrm{T}}(A_0^{-1}\mathbf{r})}{1+\mathbf{v}^{\mathrm{T}}(A_0^{-1}\mathbf{u})} = \mathbf{x}_0 - \mathbf{q}\frac{\mathbf{v}^{\mathrm{T}}\mathbf{x}_0}{1+\mathbf{v}^{\mathrm{T}}\mathbf{q}}. \tag{5.76}$$

5.5 Iterative Solution of Inhomogeneous Linear Equations

5.5.1 General Treatment

We want to solve

$$A\mathbf{x} = \mathbf{b} \tag{5.77}$$

iteratively. To that end we divide the matrix A into two (non-singular) parts [2]

$$A = A_1 + A_2 \tag{5.78}$$

and rewrite (5.77) as

$$A_1\mathbf{x} = \mathbf{b} - A_2\mathbf{x}, \tag{5.79}$$

which we use to define the iteration

$$\mathbf{x}^{(n+1)} = \Phi(\mathbf{x}^{(n)}), \tag{5.80}$$

$$\Phi(\mathbf{x}) = -A_1^{-1}A_2\mathbf{x} + A_1^{-1}\mathbf{b}. \tag{5.81}$$

A fixed point $\mathbf{x}$ of this equation fulfills

$$\mathbf{x}_{\text{fp}} = \Phi(\mathbf{x}_{\text{fp}}) = -A_1^{-1}A_2\mathbf{x}_{\text{fp}} + A_1^{-1}\mathbf{b} \tag{5.82}$$

and is obviously a solution of (5.77). The iteration can be written as

$$\begin{aligned}\mathbf{x}^{(n+1)} &= -A_1^{-1}(A - A_1)\mathbf{x}^{(n)} + A_1^{-1}\mathbf{b} \\ &= (E - A_1^{-1}A)\mathbf{x}^{(n)} + A_1^{-1}\mathbf{b} = \mathbf{x}^{(n)} - A_1^{-1}(A\mathbf{x}^{(n)} - \mathbf{b})\end{aligned} \tag{5.83}$$

or

$$A_1(\mathbf{x}^{(n+1)} - \mathbf{x}^{(n)}) = -(A\mathbf{x}^{(n)} - \mathbf{b}). \tag{5.84}$$

5.5.2 Jacobi Method

Jacobi divides the matrix A into its diagonal and two triangular matrices [14]:

$$A = L + U + D. \tag{5.85}$$

For A_1 the diagonal part is chosen

$$A_1 = D \tag{5.86}$$

giving

$$\mathbf{x}^{(n+1)} = -D^{-1}(A - D)\mathbf{x}^{(n)} + D^{-1}\mathbf{b}, \tag{5.87}$$

which reads explicitly

$$x_i^{(n+1)} = -\frac{1}{a_{ii}}\sum_{j\neq i} a_{ij}x_j^{(n)} + \frac{1}{a_{ii}}b_i. \tag{5.88}$$

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 $O(N^2)$.

5.5.3 Gauss–Seidel Method

With

$$A_1 = D + L, \tag{5.89}$$

the iteration becomes

$$(D+L)\mathbf{x}^{(n+1)} = -U\mathbf{x}^{(n)} + \mathbf{b}, \tag{5.90}$$

which has the form of a system of equations with triangular matrix [15]. It reads explicitly

$$\sum_{j\le i} a_{ij}x_j^{(n+1)} = -\sum_{j>i} a_{ij}x_j^{(n)} + b_i. \tag{5.91}$$

Forward substitution starting from x_1 gives

$$\begin{aligned}
i=1:\quad & x_1^{(n+1)} = \frac{1}{a_{11}}\left(-\sum_{j\ge 2} a_{ij}x_j^{(n)} + b_1\right),\\
i=2:\quad & x_2^{(n+1)} = \frac{1}{a_{22}}\left(-a_{21}x_1^{(n+1)} - \sum_{j\ge 3} a_{ij}x_j^{(n)} + b_2\right),\\
i=3:\quad & x_3^{(n+1)} = \frac{1}{a_{33}}\left(-a_{31}x_1^{(n+1)} - a_{32}x_2^{(n+1)} - \sum_{j\ge 4} a_{ij}x_j^{(n)} + b_3\right),\\
&\vdots\\
& x_i^{(n+1)} = \frac{1}{a_{ii}}\left(-\sum_{j<i} a_{ij}x_j^{(n+1)} - \sum_{j>i} a_{ij}x_j^{(n)} + b_i\right).
\end{aligned} \tag{5.92}$$

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.

5.5.4 Damping and Successive Over-Relaxation

Convergence can be improved [16] by combining old and new values. Starting from the iteration

$$A_1\mathbf{x}^{(n+1)} = (A_1 - A)\mathbf{x}^{(n)} + \mathbf{b}, \tag{5.93}$$

we form a linear combination with

$$\mathrm{D}\mathbf{x}^{(n+1)} = \mathrm{D}\mathbf{x}^{(n)} \tag{5.94}$$

giving the new iteration equation

$$((1-\omega)\mathrm{D}+\omega A_1)\mathbf{x}^{(n+1)} = ((1-\omega)\mathrm{D}+\omega A_1-\omega A)\mathbf{x}^{(n)}+\omega\mathbf{b}. \tag{5.95}$$

In case of the Jacobi method with $\mathrm{D} = A_1$ we have

$$\mathrm{D}\mathbf{x}^{(n+1)} = (\mathrm{D}-\omega A)\mathbf{x}^{(n)}+\omega\mathbf{b}, \tag{5.96}$$

or explicitly

$$x_i^{(n+1)} = (1-\omega)x_i^{(n)} + \frac{\omega}{a_{ii}}\left(-\sum_{j\neq i} a_{ij}x_j^{(n)} + b_i\right). \tag{5.97}$$

The changes are damped ($0 < \omega < 1$) or exaggerated[3] ($1 < \omega < 2$).

In case of the Gauss–Seidel method with $A_1 = D + L$ the new iteration (5.95) is

$$(\mathrm{D}+\omega L)\mathbf{x}^{(n+1)} = (\mathrm{D}+\omega L-\omega A)\mathbf{x}^{(n)}+\omega\mathbf{b} = (1-\omega)\mathrm{D}\mathbf{x}^{(n)}-\omega U\mathbf{x}^{(n)}+\omega\mathbf{b} \tag{5.98}$$

or explicitly

$$x_i^{(n+1)} = (1-\omega)x_i^{(n)} + \frac{\omega}{a_{ii}}\left(-\sum_{j<i} a_{ij}x_j^{(n+1)} - \sum_{j>i} a_{ij}x_j^{(n)} + \mathbf{b}\right). \tag{5.99}$$

It can be shown that the successive over-relaxation method converges only for $0 < \omega < 2$.

A very important application is the Poisson equation

$$\nabla^2 f = -\rho, \tag{5.100}$$

which will be studied in detail in Chap. 15. Here 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.

5.6 Conjugate Gradients

At the minimum of the quadratic function

$$h(\mathbf{x}) = h_0 + \mathbf{b}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T A\mathbf{x}, \tag{5.101}$$

[3] This is also known as the method of successive over-relaxation (SOR).

the gradient

$$\mathbf{g}_r = \nabla h(\mathbf{x}) = A\mathbf{x} + \mathbf{b} \tag{5.102}$$

is zero and therefore the minimum of h is also a solution of the linear system of equations

$$A\mathbf{x} = -\mathbf{b}. \tag{5.103}$$

The stationary point can be found especially efficient with the method of conjugate gradients (page 68). The function h is minimized along the search direction

$$\mathbf{s}_{r+1} = -\mathbf{g}_{r+1} + \beta_{r+1}\mathbf{s}_r$$

by solving

$$\begin{aligned} 0 &= \frac{\partial}{\partial\lambda}\left(\mathbf{b}^T(\mathbf{x}_r + \lambda\mathbf{s}_r) + \frac{1}{2}(\mathbf{x}_r^T + \lambda\mathbf{s}_r^T)A(\mathbf{x}_r + \lambda\mathbf{s}_r)\right) \\ &= \mathbf{b}^T\mathbf{s}_r + \mathbf{x}_r^T A\mathbf{s}_r + \lambda\mathbf{s}_r^T A\mathbf{s}_r, \end{aligned} \tag{5.104}$$

$$\lambda_r = -\frac{\mathbf{b}^T\mathbf{s}_r + \mathbf{x}^T A\mathbf{s}_r}{\mathbf{s}_r^T A\mathbf{s}_r} = -\frac{\mathbf{g}_r\mathbf{s}_r}{\mathbf{s}_r^T A\mathbf{s_r}}. \tag{5.105}$$

The parameter β is chosen as

$$\beta_{r+1} = \frac{g_{r+1}^2}{g_r^2}. \tag{5.106}$$

The gradient of h is the residual vector and is iterated according to

$$\mathbf{g}_{r+1} = A(\mathbf{x}_r + \lambda_r\mathbf{s}_r) + \mathbf{b} = \mathbf{g}_r + \lambda_r A\mathbf{s}_r. \tag{5.107}$$

This method [17] 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.

Problems

Problem 5.1 Gaussian Elimination

In this computer experiment we solve the system of equations

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

Compare the results of Gaussian elimination without pivoting, Gaussian elimination with partial pivoting, and QR decomposition for the following systems of equations:

(a) a well-behaved matrix

$$A_{ii} = 1, \quad A_{i \neq j} = n$$

(b) an ill-conditioned Hilbert matrix

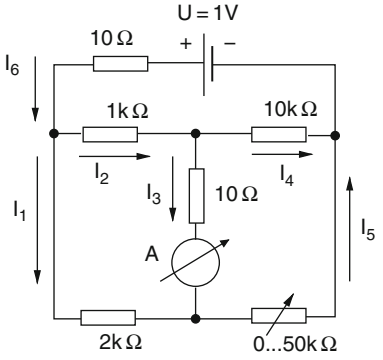
$$A_{ij} = \frac{1}{i + j - 1} \quad i, j = 1 \ldots n$$

(c) a random matrix

$$A_{ii} = 0.1 \quad A_{i \neq j} = \xi \in [0, 1]$$

The right-hand side is $\mathbf{b} = A \begin{pmatrix} 1 \\ 2 \\ \vdots \\ n \end{pmatrix}$, hence the exact solution is $\mathbf{x} = \begin{pmatrix} 1 \\ 2 \\ \vdots \\ n \end{pmatrix}$

(d) the system of linear equations for the currents of a Wheatstone bridge circuit



$$\begin{aligned} I_1 + I_2 - I_6 &= 0 \\ I_4 + I_5 - I_6 &= 0 \\ I_2 - I_3 - I_4 &= 0 \\ 10I_6 + 1000I_2 + 10000I_4 &= -1 \\ 2000I_1 - 10I_3 - 1000I_2 &= 0 \\ 10I_3 + R_x I_5 - 10000I_4 &= 0 \end{aligned}$$

Determine the current through the instrument as a function of the variable resistance R_x.

Chapter 6
Roots and Extremal Points

In physics very often roots, i.e., solutions of an equation like

$$f(x_1 \ldots x_N) = 0,$$

and extrema

$$\max f(x_1 \ldots x_N) \quad \min f(x_1 \ldots x_N)$$

have to be determined. Whereas global extrema are difficult to locate, stationary points can be found as the roots of the derivative:

$$\frac{\partial f(x_1 \ldots x_N)}{\partial x_i} = 0.$$

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 to solve

$$f(x) = 0 \tag{6.1}$$

uses the following algorithm (Fig. 6.1):

(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.

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_6,

(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.

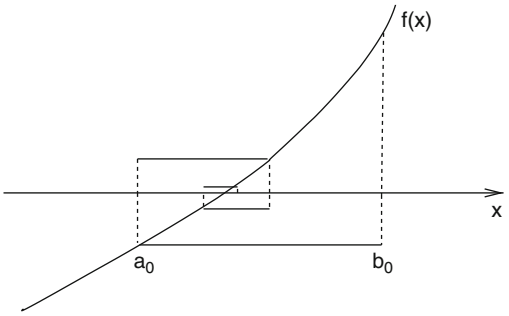


Fig. 6.1 Root finding by bisection

6.1.2 Regula Falsi Method

The regula falsi method (Fig. 6.2) is similar to the bisection method. 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

$$p(x) = f(x_r) + (x - x_r) \frac{f(a_r) - f(x_r)}{a_r - x_r} \tag{6.2}$$

is given by

$$\xi_r = x_r - f(x_r) \frac{a_r - x_r}{f(a_r) - f(x_r)} = \frac{a_r f(x_r) - x_r f(a_r)}{f(x_r) - f(a_r)}, \tag{6.3}$$

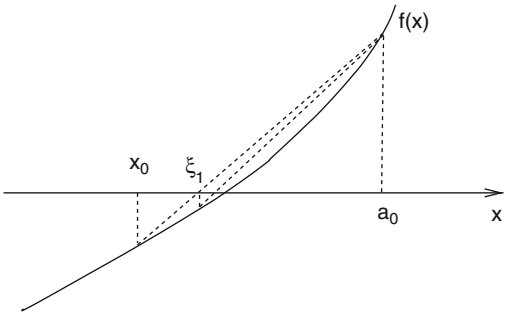


Fig. 6.2 Regula falsi method

which is inside the interval $[x_r, a_r]$. Choose the sub-interval which contains the sign change:

$$\begin{aligned} f(x_r)f(\xi_r) < 0 &\rightarrow [x_{r+1}, a_{r+1}] = [x_r, \xi_r], \\ f(x_r)f(\xi_r) > 0 &\rightarrow [x_{r+1}, a_{r+1}] = [\xi_r, a_r]. \end{aligned} \tag{6.4}$$

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

$$f(x) = f(x_0) + (x - x_0)f'(x_0) + \frac{1}{2}(x - x_0)^2 f''(x_0) + \cdots \tag{6.5}$$

gives for $x = \xi$

$$0 = f(x_0) + (\xi - x_0)f'(x_0) + \frac{1}{2}(\xi - x_0)^2 f''(x_0) + \cdots . \tag{6.6}$$

Truncation of the series and solving for ξ gives the first-order Newton–Raphson method

$$x^{(r+1)} = x^{(r)} - \frac{f(x^{(r)})}{f'(x^{(r)})} \tag{6.7}$$

and the second-order Newton–Raphson method (Fig. 6.3)

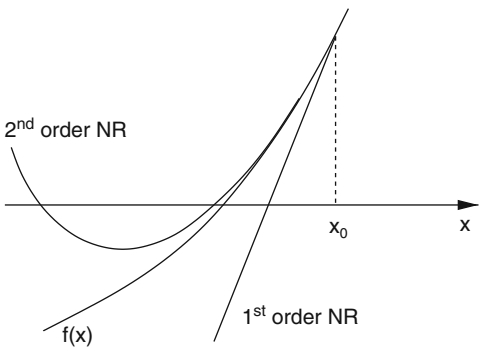


Fig. 6.3 Newton–Raphson method

$$x^{(r+1)} = x^{(r)} - \frac{f'(x^{(r)}) \pm \sqrt{f'(x^{(r)})^2 - 2f(x^{(r)})f''(x^{(r)})}}{f''(x^{(r)})}. \tag{6.8}$$

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 (Fig. 6.4)

$$x_{r+1} = x_r - f(x_r)\frac{x_r - x_{r-1}}{f(x_r) - f(x_{r-1})}. \tag{6.9}$$

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

$$x_1 = x_0 - f(x_0)\frac{2h}{f(x_0 + h) - f(x_0 - h)} \tag{6.10}$$

using a symmetrical difference quotient.

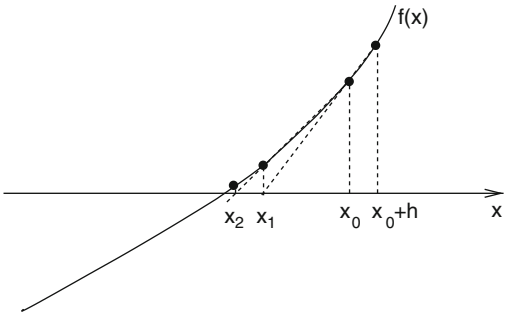


Fig. 6.4 Secant method

6.1.5 Roots of Vector Functions

The Newton–Raphson method can be easily generalized for functions of more than one variable. We search for the solution of

$$f(x) = \begin{pmatrix} f_1(x_1 \ldots x_n) \\ \vdots \\ f_n(x_1 \ldots x_n) \end{pmatrix} = 0. \tag{6.11}$$

The first-order Newton–Raphson method results from linearization of

$$0 = f(\xi) = f(x_0) + \mathrm{D}f(x_0)(\xi - x_0) + \cdots \tag{6.12}$$

with the Jacobian matrix

$$\mathrm{D}f = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}. \tag{6.13}$$

If the Jacobian matrix is not singular the equation

$$0 = f(x_1) + \mathrm{D}f(x_0)(x_1 - x_0) \tag{6.14}$$

can be solved and gives the iteration

$$x^{(r+1)} = x^{(r)} - (\mathrm{D}f(x^{(r)}))^{-1} f(x^{(r)}). \tag{6.15}$$

6.2 Optimization Without Constraints

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

$$\mathbf{g}^t(\mathbf{x}) = \left(\frac{\partial h}{\partial x_1}, \ldots, \frac{\partial h}{\partial x_n} \right) \tag{6.16}$$

and the matrix of second derivatives (Hessian) by

$$H = \left(\frac{\partial^2}{\partial x_i \partial x_j} h \right). \tag{6.17}$$

Starting from an initial guess $\mathbf{x}^{(0)}$ a direction $\mathbf{s}$ is determined, in which the function h decreases as well as a step length λ:

$$\mathbf{x}_{r+1} = \mathbf{x}_r + \lambda_r \mathbf{s}_r. \tag{6.18}$$

This is repeated until the norm of the gradient is small enough or no smaller values of h can be found (Fig. 6.5).

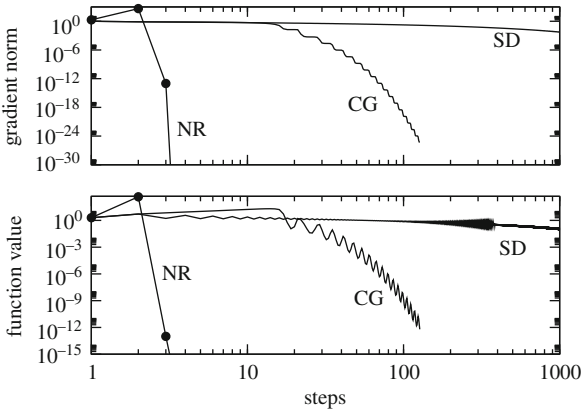


Fig. 6.5 Stationary points. The minimum of the Rosenbrock function $h(x, y) = 100(y - x^2)^2 + (1 - x)^2$ is determined with different methods. Conjugate gradients converge much faster than steepest descent. Newton–Raphson reaches the minimum at $x = y = 1$ within only four iterations to machine precision

6.2.1 Steepest Descent Method

The simplest choice is to go in the direction of the negative gradient

$$\mathbf{s}_r = -\mathbf{g}_r \tag{6.19}$$

and determine the step length by minimizing h along this direction:

$$0 = \frac{\partial}{\partial \lambda} h(\mathbf{x}_r - \lambda \mathbf{g}_r). \tag{6.20}$$

Obviously two consecutive steps are orthogonal to each other since

$$0 = \frac{\partial}{\partial \lambda} h(\mathbf{x}_{r+1} - \lambda \mathbf{g}_r)_{|\lambda=0} = -\mathbf{g}_{r+1}^t \mathbf{g}_r. \tag{6.21}$$

6.2.2 Conjugate Gradient Method

This method is similar to the steepest descent method but the search direction is iterated according to

$$\mathbf{s}_0 = -\mathbf{g}_0, \tag{6.22}$$

$$\mathbf{x}_{r+1} = \mathbf{x}_r + \lambda_r \mathbf{s}_r, \tag{6.23}$$

$$\mathbf{s}_{r+1} = -\mathbf{g}_{r+1} + \beta_{r+1} \mathbf{s}_r, \tag{6.24}$$

where λ_r is chosen to minimize $h(\mathbf{x}_{r+1})$ and the simplest choice for β is made by Fletcher and Rieves [18]

$$\beta_{r+1} = \frac{g_{r+1}^2}{g_r^2}. \tag{6.25}$$

6.2.3 Newton–Raphson Method

The first-order Newton–Raphson method uses the iteration

$$\mathbf{x}_{r+1} = \mathbf{x}_r - H(\mathbf{x}_r)^{-1}\mathbf{g}(\mathbf{x}_r). \tag{6.26}$$

The search direction is

$$\mathbf{s} = H^{-1}\mathbf{g}, \tag{6.27}$$

and the step length is $\lambda = 1$. This method converges fast if the starting point is close to the minimum. Calculation of the Hessian, however, can be very time consuming.

6.2.4 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

$$h(\mathbf{x}) = h_0 + \mathbf{b}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T H\mathbf{x} + \cdots, \tag{6.28}$$

we obtain the gradient

$$\mathbf{g}(\mathbf{x}_r) = \mathbf{b} + H\mathbf{x}_r + \cdots = \mathbf{g}(\mathbf{x}_{r-1}) + H(\mathbf{x}_r - \mathbf{x}_{r-1}) + \cdots. \tag{6.29}$$

Defining the differences

$$\mathbf{d}_r = \mathbf{x}_{r+1} - \mathbf{x}_r, \tag{6.30}$$

$$\mathbf{y}_r = \mathbf{g}_{r+1} - \mathbf{g}_r, \tag{6.31}$$

and neglecting higher order terms we obtain the so-called quasi-Newton condition

$$H\mathbf{d}_r = \mathbf{y}_r. \tag{6.32}$$

We attempt to construct a family of successive approximation matrices B_r so that, if H were a constant, the procedure would become consistent with the quasi-Newton condition. Then for the new update B_{r+1} we have

$$B_{r+1}\mathbf{d}_r = \mathbf{y}_r. \tag{6.33}$$

To specify B_{r+1} uniquely, additional conditions are required. For instance, it is reasonable to assume that B_{r+1} differs from B_r by a low-rank updating matrix that depends on $\mathbf{d}_r$, $\mathbf{y}_r$ and possibly B_r:

$$B_{r+1} = B_r + U_r(\mathbf{d}_k, \mathbf{y}_k, B_k). \tag{6.34}$$

For an update of rank one, written as

$$B_{r+1} = B_r + \mathbf{u}\mathbf{v}^T, \tag{6.35}$$

we obtain the condition

$$B_r\mathbf{d}_r + \mathbf{u}(\mathbf{v}^T\mathbf{d}_r) = \mathbf{y}_r; \tag{6.36}$$

hence

$$\mathbf{u} = \frac{1}{\mathbf{v}^T\mathbf{d}_r}(\mathbf{y}_r - B_r\mathbf{d}_r). \tag{6.37}$$

This gives the general rank one update formula as

$$B_{r+1} = B_r + \frac{1}{\mathbf{v}^T\mathbf{d}_r}(\mathbf{y}_r - B_r\mathbf{d}_r)\mathbf{v}^T. \tag{6.38}$$

Broyden's quasi-Newton method, for example, uses $\mathbf{v} = \mathbf{d}_r$.

One of the most successful and widely used update formulas is known as the BFGS (Broyden, Fletcher, Goldfarb, Shanno) method [19–22]. It is a rank two update with inherent positive definiteness:

$$B_{r+1} = B_r + U_r, \tag{6.39}$$

$$U_r = \frac{\mathbf{y_r}\mathbf{y}_r^T}{\mathbf{y}_r^T\mathbf{d}_r} - \frac{B_r^T\mathbf{d}_r^T\mathbf{d}_r B_r}{\mathbf{d}_r^T B_r\mathbf{d}_r}. \tag{6.40}$$

Problems

Problem 6.1 Bisection, Regula Falsi, and Newton–Raphson Methods

This computer experiment searches roots of several functions. You can vary the initial interval or starting value and compare the behavior of different methods.

Problem 6.2 Stationary Points

This computer experiment searches a local minimum of the Rosenbrock function[1]

$$h(x, y) = 100(y - x^2)^2 + (1 - x)^2$$

[1] A well-known test function for minimization algorithms.

(a) The method of steepest descent minimizes $h(x, y)$ along the search direction

$$s_x^{(n)} = -g_x^{(n)} = -400x(x_n^2 - y_n) - 2(x_n - 1)$$
$$s_y^{(n)} = -g_y^{(n)} = -200(y_n - x_n^2)$$

(b) Conjugate gradients make use of the search direction

$$s_x^{(n)} = -g_x^{(n)} + \beta_n s_x^{(n-1)}$$
$$s_y^{(n)} = -g_y^{(n)} + \beta_n s_y^{(n-1)}$$

(c) The Newton-Raphson method needs the inverse Hessian

$$H^{-1} = \frac{1}{\det(H)} \begin{pmatrix} h_{yy} & -h_{xy} \\ -h_{xy} & h_{xx} \end{pmatrix}$$
$$\det(H) = h_{xx}h_{yy} - h_{xy}^2$$
$$h_{xx} = 1200x^2 - 400y + 2 \quad h_{yy} = 200 \quad h_{xy} = -400x$$

and iterates according to

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} - H^{-1} \begin{pmatrix} g_x^n \\ q_y^n \end{pmatrix}$$

You can choose an initial point (x_0, y_0). The iteration stops if the gradient norm falls below 10^{-14} or if a maximum of 10,000 iterations is reached.

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. We use the symmetric definition of the Fourier transformation:

$$\widetilde{f}(\omega) = F[f](\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(t) \mathrm{e}^{-\mathrm{i}\omega t} \mathrm{d}t. \tag{7.1}$$

The inverse Fourier transformation

$$f(t) = F^{-1}[\widetilde{f}](t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \widetilde{f}(\omega) \mathrm{e}^{\mathrm{i}\omega t} \mathrm{d}\omega \tag{7.2}$$

decomposes $f(t)$ into a superposition of oscillations. The Fourier transform of a convolution integral

$$g(t) = f(t) \otimes h(t) = \int_{-\infty}^{\infty} f(t') h(t - t') \mathrm{d}t' \tag{7.3}$$

becomes a product of Fourier transforms:

$$\begin{aligned} \widetilde{g}(\omega) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \mathrm{d}t' f(t') \mathrm{e}^{-\mathrm{i}\omega t'} \int_{-\infty}^{\infty} h(t - t') \mathrm{e}^{-\mathrm{i}\omega(t-t')} \mathrm{d}(t - t') \\ &= \sqrt{2\pi}\, \widetilde{f}(\omega) \widetilde{h}(\omega). \end{aligned} \tag{7.4}$$

A periodic function with $f(t + T) = f(t)$[1] *is transformed into a Fourier series*

$$f(t) = \sum_{n=-\infty}^{\infty} \mathrm{e}^{\mathrm{i}\omega_n t} \hat{f}(\omega_n) \quad \textit{with } \omega_n = n\frac{2\pi}{T}, \quad \hat{f}(\omega_n) = \frac{1}{T} \int_0^T f(t) \mathrm{e}^{-\mathrm{i}\omega_n t} \mathrm{d}t \tag{7.5}$$

[1] This could also be the periodic continuation of a function which is only defined for $0 < t < T$.

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_7,

7.1 Discrete Fourier Transformation

We divide the time interval $0 \leq t < T$ by introducing a grid of N equidistant points

$$t_n = n\Delta t = n\frac{T}{N} \quad \text{with } n = 0, 1, \ldots, N-1. \tag{7.6}$$

The function values (samples)

$$f_n = f(t_n) \tag{7.7}$$

are arranged as components of a vector

$$\mathbf{f} = \begin{pmatrix} f_0 \\ \vdots \\ f_{N-1} \end{pmatrix}. \tag{7.8}$$

With respect to the orthonormal basis

$$\mathbf{e}_n = \begin{pmatrix} \delta_{0,n} \\ \vdots \\ \delta_{N-1,n} \end{pmatrix}, \quad n = 0, 1, \ldots, N-1. \tag{7.9}$$

$\mathbf{f}$ is expressed as a linear combination

$$\mathbf{f} = \sum_{n=0}^{N-1} f_n \mathbf{e}_n. \tag{7.10}$$

The discrete Fourier transformation is the transformation to an orthogonal base in frequency space

$$\mathbf{e}_{\omega_j} = \sum_{n=0}^{N-1} \mathrm{e}^{\mathrm{i}\omega_j t_n} \mathbf{e}_n = \begin{pmatrix} 1 \\ \mathrm{e}^{\mathrm{i}\frac{2\pi}{N}j} \\ \vdots \\ \mathrm{e}^{\mathrm{i}\frac{2\pi}{N}j(N-1)} \end{pmatrix}, \tag{7.11}$$

with

$$\omega_j = \frac{2\pi}{T} j. \tag{7.12}$$

These vectors are orthogonal:

$$\mathbf{e}_{\omega_j} \mathbf{e}^*_{\omega_{j'}} = \sum_{n=0}^{N-1} \mathrm{e}^{\mathrm{i}(j-j')\frac{2\pi}{N}n} = \begin{cases} \frac{1-\mathrm{e}^{\mathrm{i}(j-j')2\pi}}{1-\mathrm{e}^{\mathrm{i}(j-j')2\pi/N}} = 0 & \text{for} \quad j - j' \neq 0, \\ N & \text{for} \quad j - j' = 0 \end{cases} \tag{7.13}$$

$$\mathbf{e}_{\omega_j} \mathbf{e}^*_{\omega_{j'}} = N\delta_{j,j'}. \tag{7.14}$$

Alternatively a real-valued basis can be defined:

$$\begin{aligned} &\cos\left(\frac{2\pi}{N}jn\right) \quad j = 0, 1, \ldots, j_{\max} \\ &\sin\left(\frac{2\pi}{N}jn\right) \quad j = 1, 2, \ldots, j_{\max} \\ &j_{\max} = \frac{N}{2} \text{ (even N)} \quad j_{\max} = \frac{N-1}{2} \text{ (odd N).} \end{aligned} \tag{7.15}$$

The components of $\mathbf{f}$ in frequency space are given by the scalar product

$$\widetilde{f}_{\omega_j} = \mathbf{f}\mathbf{e}_{\omega_j} = \sum_{n=0}^{N-1} f_n \mathrm{e}^{-\mathrm{i}\omega_j t_n} = \sum_{n=0}^{N-1} f_n \mathrm{e}^{-\mathrm{i}j\frac{2\pi}{T}n\frac{T}{N}} = \sum_{n=0}^{N-1} f_n \mathrm{e}^{-\mathrm{i}\frac{2\pi}{N}jn}. \tag{7.16}$$

From

$$\sum_{j=0}^{N-1} \widetilde{f}_{\omega_j} \mathrm{e}^{\mathrm{i}\omega_j t_n} = \sum_{n'} \sum_{\omega_j} f_{n'} \mathrm{e}^{-\mathrm{i}\omega_j t_{n'}} \mathrm{e}^{\mathrm{i}\omega_j t_n} = N f_n, \tag{7.17}$$

we find the inverse transformation

$$f_n = \frac{1}{N} \sum_{j=0}^{N-1} \widetilde{f}_{\omega_j} \mathrm{e}^{\mathrm{i}\omega_j t_n} = \frac{1}{N} \sum_{j=0}^{N-1} \widetilde{f}_{\omega_j} \mathrm{e}^{\mathrm{i}\frac{2\pi}{N}nj}. \tag{7.18}$$

7.1.1 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:

$$f(t) = \frac{1}{N} \sum_{j=0}^{N-1} \widetilde{f}_{\omega_j} \left(\mathrm{e}^{\mathrm{i}\frac{2\pi}{T}t}\right)^j, \tag{7.19}$$

which is a polynomial of

$$q = \mathrm{e}^{\mathrm{i}\frac{2\pi}{T}t}. \tag{7.20}$$

Since

$$\mathrm{e}^{-\mathrm{i}\omega_j t_n} = \mathrm{e}^{-\mathrm{i}\frac{2\pi}{N}jn} = \mathrm{e}^{\mathrm{i}\frac{2\pi}{N}(N-j)n} = \mathrm{e}^{\mathrm{i}\omega_{N-j} t_n}, \tag{7.21}$$

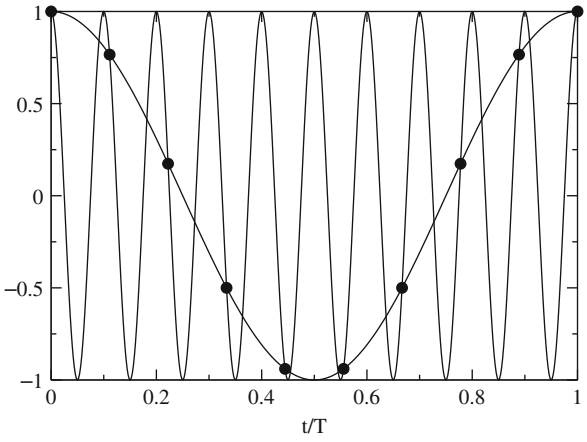


Fig. 7.1 Equivalence of ω_1 and ω_{N-1}. The two functions $\cos\omega t$ and $\cos(N-1)\omega t$ have the same values at the sample points t_n but are very different in between

the frequencies ω_j and ω_{N-j} are equivalent (Fig. 7.1):

$$\widetilde{f}_{\omega_{N-j}} = \sum_{n=0}^{N-1} f_n \mathrm{e}^{-\mathrm{i}\frac{2\pi}{N}(N-j)n} = \sum_{n=0}^{N-1} f_n \mathrm{e}^{\mathrm{i}\frac{2\pi}{N}jn} = \widetilde{f}_{\omega_{-j}}. \tag{7.22}$$

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):

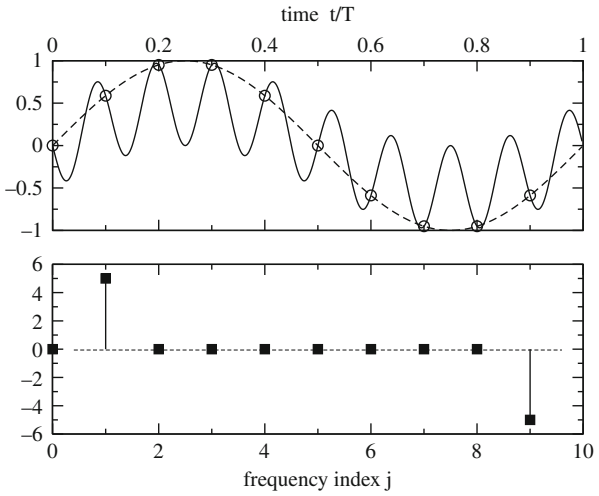


Fig. 7.2 Trigonometric interpolation. For trigonometric interpolation the high frequencies have to be replaced by the corresponding negative frequencies to provide meaningful results between the sampling points. The *circles* show sampling points which are fitted using only positive frequencies (*full curve*) or replacing the unphysical high frequency by its negative counterpart (*broken curve*). The *squares* show the calculated Fourier spectrum

$$
\begin{aligned}
-\frac{2\pi}{T}\frac{N-1}{2} \le \omega_j \le \frac{2\pi}{T}\frac{N-1}{2} \qquad & N \text{ odd}, \\
-\frac{2\pi}{T}\frac{N}{2} \le \omega_j \le \frac{2\pi}{T}\left(\frac{N}{2}-1\right) \qquad & N \text{ even}.
\end{aligned} \tag{7.23}
$$

The interpolating function (N even) is

$$
f(t) = \frac{1}{N}\sum_{j=-\frac{N}{2}}^{\frac{N}{2}-1} \widetilde{f}_{\omega_j} \mathrm{e}^{\mathrm{i}\omega_j t}. \tag{7.24}
$$

The maximum frequency is

$$
\omega_{\max} = \frac{2\pi}{T}\frac{N}{2}, \tag{7.25}
$$

and hence

$$
f_{\max} = \frac{1}{2\pi}\omega_{\max} = \frac{N}{2T} = \frac{f_s}{2}. \tag{7.26}
$$

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.

7.1.2 Real-Valued Functions

For a real-valued function

$$
f_n = f_n^*, \tag{7.27}
$$

and hence

$$
\widetilde{f}_{\omega_j}^* = \left(\sum_{n=0}^{N-1} f_n \mathrm{e}^{-\mathrm{i}\omega_j t_n}\right)^* = \sum_{n=0}^{N-1} f_n \mathrm{e}^{\mathrm{i}\omega_j t_n} = \widetilde{f}_{\omega_{-j}}. \tag{7.28}
$$

Here it is sufficient to calculate the sums for $j = 0, \ldots, N/2$.

7.1.3 Approximate Continuous Fourier Transformation

We continue the function $f(t)$ periodically by setting

$$
f_N = f_0 \tag{7.29}
$$

and write

$$\widetilde{f}_{\omega_j} = \sum_{n=0}^{N-1} f_n \mathrm{e}^{-\mathrm{i}\omega_j n} = \frac{1}{2} f_0 + \mathrm{e}^{-\mathrm{i}\omega_j} f_1 + \cdots + \mathrm{e}^{-\mathrm{i}\omega_j (N-1)} f_{N-1} + \frac{1}{2} f_N. \quad (7.30)$$

Comparing with the trapezoidal rule (4.9) for the integral

$$\int_0^T \mathrm{e}^{-\mathrm{i}\omega_j t} f(t)\mathrm{d}t \approx \frac{T}{N} \left(\frac{1}{2} \mathrm{e}^{-\mathrm{i}\omega_j 0} f(0) + \mathrm{e}^{-\mathrm{i}\omega_j \frac{T}{N}} f\left(\frac{T}{N}\right) + \cdots + \mathrm{e}^{-\mathrm{i}\omega_j \frac{T}{N}(N-1)} f\left(\frac{T}{N}(N-1)\right) + \frac{1}{2} f(T) \right), \quad (7.31)$$

we find

$$\hat{f}(\omega_j) = \frac{1}{T} \int_0^T \mathrm{e}^{-\mathrm{i}\omega_j t} f(t)\mathrm{d}t \approx \frac{1}{N} \widetilde{f}_{\omega_j}, \quad (7.32)$$

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

$$W(t) = \begin{cases} 1 & \text{for} \quad 0 < t < T \\ 0 & \text{else} \end{cases}. \quad (7.33)$$

The discrete Fourier transformation approximates the continuous Fourier transformation but windowing leads to a broadening of the spectrum. For practical purposes smoother windowing functions are used like a triangular window or one of the following [23]:

$$W(t_n) = \mathrm{e}^{-\frac{1}{2}\left(\frac{n-(N-1)/2}{\sigma(N-1)/2}\right)^2} \quad \sigma \leq 0.5 \quad \text{Gaussian window}$$

$$W(t_n) = 0.53836 - 0.46164 \cos\left(\frac{2\pi n}{N-1}\right) \quad \text{Hamming window}$$

$$W(t_n) = 0.5 \left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right) \quad \text{Hann(ing) window}$$

7.2 Algorithms

Straightforward evaluation of the sum

$$\widetilde{f}_{\omega_j} = \sum_{n=0}^{N-1} \cos\left(\frac{2\pi}{N} jn\right) f_n + \mathrm{i} \sin\left(\frac{2\pi}{N} jn\right) f_n \quad (7.34)$$

needs $O(N^2)$ additions, multiplications, and trigonometric functions.

7.2.1 Goertzel's Algorithm

Goertzel's method [24] is very useful if not the whole Fourier spectrum is needed but only a small number of 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

$$\sum_{n=0}^{N-1} f_n \mathrm{e}^{-\mathrm{i}\frac{2\pi}{N}jn} = f_0 + \mathrm{e}^{-\frac{2\pi\mathrm{i}}{N}j}(f_1 + \mathrm{e}^{-\frac{2\pi\mathrm{i}}{N}j} f_2 \cdots (f_{N-2} + \mathrm{e}^{-\frac{2\pi\mathrm{i}}{N}j} f_{N-1}) \cdots), \quad (7.35)$$

which can be evaluated recursively

$$\begin{aligned} y_{N-1} &= f_{N-1}, \\ y_n &= f_n + \mathrm{e}^{-\frac{2\pi\mathrm{i}}{N}j} y_{n+1} \quad n = N-2, \ldots, 0, \end{aligned} \quad (7.36)$$

to give the result

$$\hat{f}_{\omega_j} = y_0. \quad (7.37)$$

Equation (7.36) is a simple discrete filter function. Its transmission function is obtained by application of the z-transform [25]

$$u(z) = \sum_{n=0}^{\infty} u_n z^{-n} \quad (7.38)$$

(the discrete version of the Laplace transform), which yields

$$y(z) = \frac{f(z)}{1 - z\mathrm{e}^{-\frac{2\pi\mathrm{i}}{N}j}}. \quad (7.39)$$

One disadvantage of this method is that it uses complex numbers. This can be avoided by the following more complicated recursion:

$$\begin{aligned} u_{N+1} &= u_N = 0, \\ u_n &= f_n + 2u_{n+1}\cos\frac{2\pi}{N}k - u_{n+2} \quad \text{for } n = N-1, \ldots, 0, \end{aligned} \quad (7.40)$$

with the transmission function

$$\begin{aligned} \frac{u(z)}{f(z)} &= \frac{1}{1 - z\left(\mathrm{e}^{\frac{2\pi\mathrm{i}}{N}j} + \mathrm{e}^{-\frac{2\pi\mathrm{i}}{N}j}\right) + z^2} \\ &= \frac{1}{\left(1 - z\mathrm{e}^{-\frac{2\pi\mathrm{i}}{N}j}\right)\left(1 - z\mathrm{e}^{\frac{2\pi\mathrm{i}}{N}j}\right)}. \end{aligned} \quad (7.41)$$

A second filter removes one factor in the denominator

$$\frac{y(z)}{u(z)} = \left(1 - z\mathrm{e}^{\frac{2\pi\mathrm{i}}{N}j}\right), \tag{7.42}$$

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 Eq. (7.39).

$$\frac{y(z)}{f(z)} = \frac{1}{1 - z\mathrm{e}^{-\frac{2\pi\mathrm{i}}{N}j}}. \tag{7.43}$$

Hence the Fourier component of $\mathbf{f}$ is given by

$$\hat{f}_{\omega_j} = y_0 = u_0 - \mathrm{e}^{\frac{2\pi\mathrm{i}}{N}j} u_1. \tag{7.44}$$

The order of the iteration (7.35) can be reversed by writing

$$\hat{f}_{\omega_j} = f_0 \cdots \mathrm{e}^{\frac{2\pi\mathrm{i}}{N}(N-1)} f_{N-1} = \mathrm{e}^{-\frac{2\pi\mathrm{i}}{N}j(N-1)} (f_0 \mathrm{e}^{\frac{2\pi\mathrm{i}}{N}j(N-1)} \cdots f_{N-1}), \tag{7.45}$$

which is very useful for real-time filter applications.

7.2.2 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

$$\mathrm{e}^{-\mathrm{i}\frac{2\pi}{N}jm} = W_N^{jm} \tag{7.46}$$

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

$$F_N(f_0 \ldots f_{N-1}) = \sum_{n=0}^{N-1} f_n W_N^{jn}, \tag{7.47}$$

we separate even and odd powers of the unit root

[2] There exist several fast Fourier transformation algorithms [26, 27]. We consider only the simplest one here [28].

$$\begin{aligned} F_N(f_0 \dots f_{N-1}) &= \sum_{m=0}^{\frac{N}{2}-1} f_{2m} W_N^{j2m} + \sum_{m=0}^{\frac{N}{2}-1} f_{2m+1} W_N^{j(2m+1)} \\ &= \sum_{m=0}^{\frac{N}{2}-1} f_{2m} \mathrm{e}^{-\mathrm{i}\frac{2\pi}{N/2}jm} + W_N^j \sum_{m=0}^{\frac{N}{2}-1} f_{2m+1} \mathrm{e}^{-\mathrm{i}\frac{2\pi}{N/2}jm} \\ &= F_{N/2}(f_0, f_2, \dots, f_{N-2}) + W_N^j F_{N/2}(f_1, f_3, \dots, f_{N-1}). \end{aligned} \tag{7.48}$$

This division is repeated until only sums with one summand remain:

$$F_1(f_n) = f_n. \tag{7.49}$$

For example, consider the case $N = 8$:

$$\begin{gathered} F_8(f_0 \dots f_7) = F_4(f_0 f_2 f_4 f_6) + W_8^j F_4(f_1 f_3 f_5 f_7) \\ - - - \\ F_4(f_0 f_2 f_4 f_6) = F_2(f_0 f_4) + W_4^j F_2(f_2 f_6) \\ F_4(f_1 f_3 f_5 f_7) = F_2(f_1 f_5) + W_4^j F_2(f_3 f_7) \\ - - - \\ F_2(f_0 f_4) = f_0 + W_2^j f_4 \\ F_2(f_2 f_6) = f_2 + W_2^j f_6 \\ F_2(f_1 f_5) = f_1 + W_2^j f_5 \\ F_2(f_3 f_7) = f_3 + W_2^j f_7 \end{gathered} \tag{7.50}$$

Expansion gives

$$\begin{aligned} F_8 = {} & f_0 + W_2^j f_4 + W_4^j f_2 + W_4^j W_2^j f_6 \\ & + W_8^j f_1 + W_8^j W_2^j f_5 + W_8^j W_4^j f_3 + W_8^j W_4^j W_2^j f_7 . \end{aligned} \tag{7.51}$$

Generally a summand of the Fourier sum can be written using the binary representation of n

$$n = \sum l_i \quad l_i = 1, 2, 4, 8, \dots \tag{7.52}$$

in the following way:

$$f_n \mathrm{e}^{-\mathrm{i}\frac{2\pi}{N}jn} = f_n \mathrm{e}^{-\mathrm{i}\frac{2\pi}{N}(l_1+l_2+\cdots)j} = f_n W_{N/l_1}^j W_{N/l_2}^j \cdots . \tag{7.53}$$

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 $n = 000, 100, 010, \ldots$
(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

$$\begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \end{pmatrix} = \begin{pmatrix} f_0 \\ f_4 \\ f_2 \\ f_6 \\ f_1 \\ f_5 \\ f_3 \\ f_7 \end{pmatrix}. \tag{7.54}$$

Now calculate sums with two summands. Since W_2^j can take only two different values

$$W_2^j = \begin{cases} 1 & \text{for } j = 0, 2, 4, 6 \\ -1 & \text{for } j = 1, 3, 5, 7 \end{cases}, \tag{7.55}$$

a total of eight sums have to be calculated which can be stored again in the same workspace:

$$\begin{pmatrix} f_0 + f_4 \\ f_0 - f_4 \\ f_2 + f_6 \\ f_2 - f_6 \\ f_1 + f_5 \\ f_1 - f_5 \\ f_3 + f_7 \\ f_3 - f_7 \end{pmatrix} = \begin{pmatrix} s_0 + W_2^0 s_1 \\ s_0 + W_2^1 s_1 \\ s_2 + W_2^2 s_3 \\ s_2 + W_2^3 s_3 \\ s_4 + W_2^4 s_5 \\ s_4 + W_2^5 s_5 \\ s_6 + W_2^6 s_7 \\ s_6 + W_2^7 s_7 \end{pmatrix}. \tag{7.56}$$

Next calculate sums with four summands. W_4^j can take one of four values

$$W_4^j = \begin{cases} 1 & \text{for} \quad j = 0, 4 \\ -1 & \text{for} \quad j = 2, 6 \\ W_4 & \text{for} \quad j = 1, 5 \\ -W_4 & \text{for} \quad j = 3, 7 \end{cases}. \tag{7.57}$$

The following combinations are needed:

$$\begin{pmatrix} f_0+f_4+(f_2+f_6) \\ f_0+f_4-(f_2+f_6) \\ (f_0-f_4)+W_4(f_2-f_6) \\ (f_0-f_4)-W_4(f_2-f_6) \\ f_1+f_5+(f_3+f_7) \\ f_1+f_5-(f_3+f_7) \\ (f_1-f_5)\pm W_4(f_3-f_7) \\ (f_1-f_5)\pm W_4(f_3-f_7) \end{pmatrix} = \begin{pmatrix} s_0+W_4^0 s_2 \\ s_1+W_4^1 s_3 \\ s_0+W_4^2 s_2 \\ s_1+W_4^3 s_3 \\ s_4+W_4^4 s_6 \\ s_5+W_4^5 s_7 \\ s_4+W_4^6 s_6 \\ s_5+W_4^7 s_7 \end{pmatrix}. \tag{7.58}$$

The next step gives the sums with eight summands. With

$$W_8^j = \begin{cases} 1 & j=0 \\ W_8 & j=1 \\ W_8^2 & j=2 \\ W_8^3 & j=3 \\ -1 & j=4 \\ -W_8 & j=5 \\ -W_8^2 & j=6 \\ -W_8^3 & j=7 \end{cases}, \tag{7.59}$$

we calculate

$$\begin{pmatrix} f_0+f_4+(f_2+f_6)+(f_1+f_5+(f_3+f_7)) \\ f_0+f_4-(f_2+f_6)+W_8(f_1+f_5-(f_3+f_7)) \\ (f_0-f_4)+W_4(f_2-f_6)+W_8^2(f_1-f_5)\pm W_4(f_3-f_7) \\ (f_0-f_4)-W_4(f_2-f_6)+W_8^3((f_1-f_5)\pm W_4(f_3-f_7)) \\ f_0+f_4+(f_2+f_6)-(f_1+f_5+(f_3+f_7)) \\ f_0+f_4-(f_2+f_6)-W_8(f_1+f_5-(f_3+f_7)) \\ (f_0-f_4)+W_4(f_2-f_6)-W_8^2((f_1-f_5)\pm W_4(f_3-f_7)) \\ (f_0-f_4)-W_4(f_2-f_6)-W_8^3((f_1-f_5)\pm W_4(f_3-f_7)) \end{pmatrix} = \begin{pmatrix} s_0+W_8^0 s_4 \\ s_1+W_8^1 s_5 \\ s_2+W_8^2 s_6 \\ s_3+W_8^3 s_7 \\ s_0+W_8^4 s_4 \\ s_1+W_8^5 s_5 \\ s_2+W_8^6 s_6 \\ s_3+W_8^7 s_7 \end{pmatrix}, \tag{7.60}$$

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

$$f_n = f\left(n\frac{T}{N}\right) \quad n = 0 \ldots N-1$$

the Fourier coefficients

$$\widetilde{f}_{\omega_j} = \sum_{n=0}^{N-1} f_n e^{-i\omega_j t_n} \quad \omega_j = \frac{2\pi}{T} j, \quad j = 0 \ldots N-1$$

are calculated with Goertzel's method (Sect. 7.2.1).
The results from the inverse transformation

$$f_n = \frac{1}{N} \sum_{j=0}^{N-1} \widetilde{f}_{\omega_j} \mathrm{e}^{\mathrm{i}\frac{2\pi}{N}nj}$$

are compared with the original function values $f(t_n)$.
The Fourier sum is used for trigonometric interpolation with only positive frequencies

$$f(t) = \frac{1}{N} \sum_{j=0}^{N-1} \widetilde{f}_{\omega_j} \left(\mathrm{e}^{\mathrm{i}\frac{2\pi}{T}t}\right)^j$$

Finally the unphysical high frequencies are replaced by negative frequencies (7.23).
The results can be studied for several kinds of input data.

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.

Chapter 8
Random Numbers and Monte Carlo Methods

Many-body problems often involve the calculation of integrals of very high dimension which cannot be treated by standard methods. For the calculation of thermodynamical averages Monte Carlo methods [29–32] *are very useful which sample the integration volume at randomly chosen points.*

8.1 Some Basic Statistics

For more mathematical details see [33].

8.1.1 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 (Fig. 8.1).

The cumulative probability distribution is given by the function

$$F(x) = P\{\xi \leq x\} \tag{8.1}$$

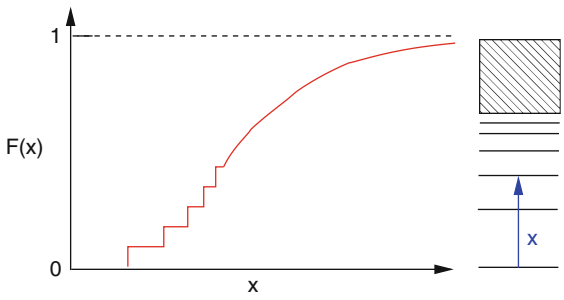


Fig. 8.1 Cumulative probability distribution of transition energies. The figure shows schematically the distribution of transition energies for an atom which has a discrete and a continuous part

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_8,

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

$$P(x_1 < \xi \leq x_2) = F(x_2) - F(x_1). \tag{8.2}$$

The height of a jump gives the probability of a discrete value

$$P(\xi = x_0) = F(x_0 + 0) - F(x_0 - 0). \tag{8.3}$$

In regions where F is continuous, the probability density can be defined as

$$f(x_0) = F'(x_0) = \lim_{\Delta x \to 0} \frac{1}{\Delta x} P(x_0 < \xi \leq x_0 + \Delta x). \tag{8.4}$$

8.1.2 Expectation Values and Moments

The expectation value of the random variable ξ is defined by

$$E(\xi) = \int_{-\infty}^{\infty} x \mathrm{d}F(x) = \lim_{a \to -\infty, b \to \infty} \int_a^b x \mathrm{d}F(x) \tag{8.5}$$

with the Stieltjes integral

$$\int_a^b x \mathrm{d}F(x) = \lim_{N \to \infty} \sum_{i=1}^{N} x_i (F(x_i) - F(x_{i-1}))|_{x_i = a + \frac{b-a}{N} i}. \tag{8.6}$$

Higher moments are defined as

$$E(\xi^k) = \int_{-\infty}^{\infty} x^k \mathrm{d}F(x) \tag{8.7}$$

if these integrals exist. Most important are the expectation value

$$\overline{x} = E(\xi) \tag{8.8}$$

and the standard deviation σ, which results from the first two moments

$$\sigma^2 = \int_{-\infty}^{\infty} (x - \overline{x})^2 \mathrm{d}F = \int x^2 \mathrm{d}F + \int \overline{x}^2 \mathrm{d}F - 2\overline{x} \int x \mathrm{d}F$$
$$= E(\xi^2) - (E(\xi))^2. \tag{8.9}$$

The expectation value of a function $\varphi(x)$ is defined by

$$E(\varphi(x)) = \int_{-\infty}^{\infty} \varphi(x) \mathrm{d}F(x) \tag{8.10}$$

For continuous $F(x)$ we have with $\mathrm{d}F(x) = f(x)\mathrm{d}x$, the ordinary integral,

$$E(\xi^k) = \int_{-\infty}^{\infty} x^k f(x) \mathrm{d}x \tag{8.11}$$

$$E(\varphi(x)) = \int_{-\infty}^{\infty} \varphi(x) f(x) \mathrm{d}x \tag{8.12}$$

whereas for a pure step function $F(x)$ (only discrete values x_i are observed with probabilities $p(x_i) = F(x_i + 0) - F(x_i - 0)$),

$$E(\xi^k) = \sum x_i^k p(x_i) \tag{8.13}$$

$$E(\varphi(x)) = \sum \varphi(x_i) p(x_i). \tag{8.14}$$

8.1.2.1 Ideal Dice

We consider an ideal dice. Each of its six faces appears with the same probability of 1/6. The cumulative probability distribution $F(x)$ is a pure step function (Fig. 8.2) and

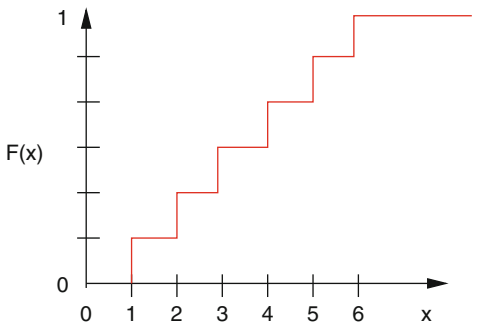


Fig. 8.2 Cumulative probability distribution of an ideal dice

$$\overline{x} = \int_{-\infty}^{\infty} x \mathrm{d}F = \sum_{i=1}^{6} x_i (F(x_i + 0) - F(x_i - 0)) = \frac{1}{6} \sum_{i=1}^{6} x_i = \frac{21}{6} = 3.5 \quad (8.15)$$

$$\overline{x^2} = \sum_{i=1}^{6} x_i^2 (F(x_i + 0) - F(x_i - 0)) = \frac{1}{6} \sum_{i=1}^{6} x_i^2 = \frac{91}{6} = 15.1666\cdots \quad (8.16)$$

$$\sigma = \sqrt{\overline{x^2} - \overline{x}^2} = 2.9. \quad (8.17)$$

8.1.2.2 Normal Distribution

The Gaussian normal distribution is defined by the cumulative probability distribution

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \mathrm{e}^{-t^2/2} \mathrm{d}t \quad (8.18)$$

and the probability density

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-x^2/2} \quad (8.19)$$

with the properties

$$\int_{-\infty}^{\infty} \varphi(x) \mathrm{d}x = \Phi(\infty) = 1 \quad (8.20)$$

$$\overline{x} = \int_{-\infty}^{\infty} x \varphi(x) \mathrm{d}x = 0 \quad (8.21)$$

$$\sigma^2 = \overline{x^2} = \int_{-\infty}^{\infty} x^2 \varphi(x) \mathrm{d}x = 1. \quad (8.22)$$

Since $\Phi(0) = \frac{1}{2}$ and with the definition

$$\Phi_0(x) = \frac{1}{\sqrt{2\pi}} \int_{0}^{x} \mathrm{e}^{-t^2/2} \mathrm{d}t \quad (8.23)$$

we have

$$\Phi(x) = \frac{1}{2} + \Phi_0(x) \quad (8.24)$$

which can be expressed in terms of the error function[1]

[1] $\mathrm{erf}(x)$ is an intrinsic function in FORTRAN or C.

$$\operatorname{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z \mathrm{e}^{-t^2} \mathrm{d}t = 2\Phi_0(\sqrt{2}z). \tag{8.25}$$

A general Gaussian distribution with mean value $\overline{x}$ and standard deviation σ has the cumulative distribution

$$F_{\overline{x},\sigma}(x) = \Phi\left(\frac{x-\overline{x}}{\sigma}\right) = \int_{-\infty}^{x} \mathrm{d}x' \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x'-\overline{x})^2}{2\sigma^2}\right). \tag{8.26}$$

8.1.2.3 Histogram

From an experiment $F(x)$ cannot be determined directly. Instead a finite number N of values x_i are measured. By

$$Z_N(x)$$

we denote the number of measurements with $x_i \leq x$. The cumulative probability distribution is the limit

$$F(x) = \lim_{N\to\infty} \frac{1}{N} Z_N(x). \tag{8.27}$$

A histogram counts the number of measured values which are in the interval $x_i < x \leq x_{i+1}$:

$$\frac{1}{N}(Z_N(x_{i+1}) - Z_N(x_i)) \approx F(x_{i+1}) - F(x_i) = P(x_i < \xi \leq x_{i+1}). \tag{8.28}$$

Contrary to $Z_N(x)$ itself, the histogram depends on the choice of the intervals (Fig. 8.3).

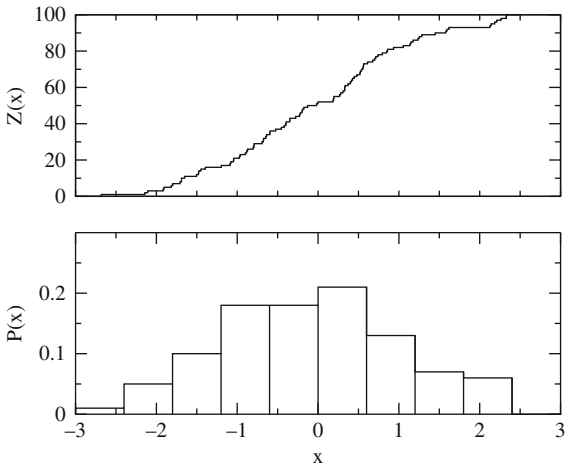


Fig. 8.3 Histogram. The cumulative distribution of 100 Gaussian random numbers is shown together with a histogram with bin width $\Delta x = 0.6$

8.1.3 Multivariate Distributions

Consider now two quantities which are measured simultaneously. ξ and η are the corresponding random variables. The cumulative distribution function is

$$F(x, y) = P\{\xi \leq x \quad \text{and} \quad \eta \leq y\}. \tag{8.29}$$

Expectation values are defined as

$$E(\varphi(x, y)) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \varphi(x, y) \mathrm{d}^2 F(x, y). \tag{8.30}$$

For continuous $F(x, y)$ the probability density is

$$f(x, y) = \frac{\partial^2 F}{\partial x \partial y} \tag{8.31}$$

and the expectation value is simply

$$E(\varphi(x, y)) = \int_{-\infty}^{\infty} \mathrm{d}x \int_{-\infty}^{\infty} \mathrm{d}y \varphi(x, y) f(x, y). \tag{8.32}$$

The moments of the distribution are the expectation values

$$M_{k,l} = E(\xi^k \eta^l). \tag{8.33}$$

Most important are the averages

$$\overline{x} = E(\xi), \quad \overline{y} = E(\eta), \tag{8.34}$$

and the covariance matrix

$$\begin{pmatrix} E((\xi - \overline{x})^2) & E((\xi - \overline{x})(\eta - \overline{y})) \\ E((\xi - \overline{x})(\eta - \overline{y})) & E((\eta - \overline{y})^2) \end{pmatrix} = \begin{pmatrix} \overline{x^2} - \overline{x}^2 & \overline{xy} - \overline{x}\,\overline{y} \\ \overline{xy} - \overline{x}\,\overline{y} & \overline{y^2} - \overline{y}^2 \end{pmatrix}. \tag{8.35}$$

The correlation coefficient is defined as

$$\rho = \frac{\overline{xy} - \overline{x}\,\overline{y}}{\sqrt{\left(\overline{x^2} - \overline{x}^2\right)\left(\overline{y^2} - \overline{y}^2\right)}}. \tag{8.36}$$

If there is no correlation then $\rho = 0$ and $F(x, y) = F_1(x) F_2(y)$.

8.1.4 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

$$\eta_N = \frac{\xi_1 + \xi_2 + \cdots + \xi_N}{\sqrt{N}} \tag{8.37}$$

with the cumulative distribution function $F_N(x)$. In the limit $N \to \infty$ this distribution approaches a cumulative normal distribution [34]

$$\lim_{N\to\infty} F_N(x) = \Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \mathrm{e}^{-t^2/2} \mathrm{d}t. \tag{8.38}$$

8.1.5 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 (Fig. 8.4)

$$\eta = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} \xi_i \tag{8.39}$$

can be derived from the binomial distribution

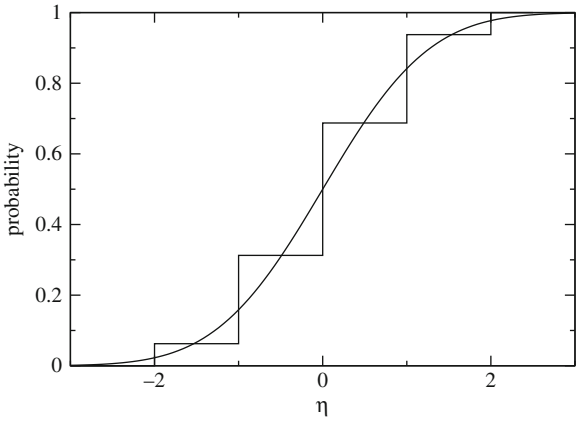


Fig. 8.4 Central limit theorem. The cumulative distribution function of η (8.39) is shown for $N = 4$ and compared to the normal distribution (8.18)

$$1 = \left[\frac{1}{2} + \left(-\frac{1}{2}\right)\right]^N = 2^{-N} \sum_{p=0}^{N} (-1)^{N-p} \binom{N}{N-p}, \tag{8.40}$$

where p counts the number of tosses with $\xi = +1$. Since

$$n = p \cdot 1 + (N - p) \cdot (-1) = 2p - N = -N \ldots N \tag{8.41}$$

the probability of finding $\eta = \frac{n}{\sqrt{N}}$ is given by the binomial coefficient

$$P\left(\eta = \frac{2p - N}{\sqrt{N}}\right) = 2^{-N} \binom{N}{N-p} \tag{8.42}$$

or

$$P\left(\eta = \frac{n}{\sqrt{N}}\right) = 2^{-N} \binom{N}{\frac{N-n}{2}} n \tag{8.43}$$

8.1.6 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 $f(X_i)$. The expectation value is the exact value $E(X_i) = \int \mathrm{d}X_i \, X_i \, f(X_i) = X$ and the standard deviation due to measurement uncertainties is $\sigma_X = \sqrt{E(X_i^2) - X^2}$. The new random variables

$$\xi_i = \frac{X_i - X}{\sigma_X} \tag{8.44}$$

have zero mean

$$E(\xi_i) = \frac{E(X_i) - X}{\sigma_X} = 0 \tag{8.45}$$

and unit standard deviation

$$\sigma_\xi^2 = E(\xi_i^2) - E(\xi_i)^2 = E\left(\frac{X_i^2 + X^2 - 2XX_i}{\sigma_X^2}\right) = \frac{E(X_i^2) - X^2}{\sigma_X^2} = 1. \tag{8.46}$$

Hence the quantity

$$\eta = \frac{\sum_1^N \xi_i}{\sqrt{N}} = \frac{\sum_1^N X_i - NX}{\sqrt{N}\sigma_X} = \frac{\sqrt{N}}{\sigma_X}(\overline{X} - X) \tag{8.47}$$

obeys a normal distribution

$$f(\eta) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\eta^2/2}. \tag{8.48}$$

From

$$f(\overline{X})\mathrm{d}\overline{X} = f(\eta)\mathrm{d}\eta = f(\eta(\overline{X}))\frac{\sqrt{N}}{\sigma_X}\mathrm{d}\overline{X} \tag{8.49}$$

we obtain

$$f(\overline{X}) = \frac{\sqrt{N}}{\sqrt{2\pi}\sigma_X} \exp\left\{-\frac{N}{2\sigma_X^2}(\overline{X} - X)^2\right\}. \tag{8.50}$$

The average of N measurements obeys a Gaussian distribution around the exact value X with a reduced standard deviation of

$$\sigma_{\overline{X}} = \frac{\sigma_X}{\sqrt{N}}. \tag{8.51}$$

8.2 Random Numbers

True random numbers of high quality can be generated using physical effects like thermal noise in a diode or from a light source. Special algorithms are available to generate pseudo-random numbers which have comparable statistical properties but are not unpredictable since they depend on some initial seed values. Often an iteration

$$r_{i+1} = f(r_i) \tag{8.52}$$

is used to calculate a series of pseudo-random numbers. Using 32-bit integers there are 2^{32} different numbers, hence the period cannot exceed 2^{32}. A simple algorithm is the linear congruent mapping

$$r_{i+1} = (ar_i + c) \bmod m \tag{8.53}$$

with maximum period m. A larger period is achieved if the random number depends on more than one predecessors. A function of the type

$$r_i = f(r_{i-1}, r_{i-2}, \ldots, r_{i-t}) \tag{8.54}$$

using 32-bit integers has a maximum period of 2^{32t}.

Example For $t = 2$ and generating 10^6 numbers per second the period is 584 942 years.

8.2.1 The Method by Marsaglia and Zamann

A high-quality random number generator can be obtained from the combination of two generators [35]. The first one

$$r_i = (r_{i-2} - r_{i-3} - c) \bmod (2^{32} - 18) \tag{8.55}$$

with

$$c = \begin{cases} 1 \text{ for } r_{n-2} - r_{n-3} < 0 \\ 0 \text{ else} \end{cases} \tag{8.56}$$

has a period of 2^{95}. The second one

$$r_i = (69069 r_{i-1} + 1013904243) \bmod 2^{32} \tag{8.57}$$

has a period of 2^{32}. The period of the combination is 2^{127}. Here is a short subroutine in C:

```
#define N 100000
typedef unsigned long int unlong /* 32 Bit */
unlong x=521288629, y=362436069, z=16163801 , c=1, n=1131199209;
unlong mzran()
{ unlong s;
   if (y>x+c) {s=y-(x+c)-18; c=0;}
   else {s=y-(x+c)-18;c=1;}
   x=y; y=z; z=s; n=69069*n+1013904243;
   return(z+n);
}
```

8.2.2 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

$$F_0(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad 0 \le x \le 1 \\ 1 & \text{for} \quad x > 1 \end{cases} . \tag{8.58}$$

Random numbers with cumulative distribution $F(x)$ can be obtained as follows:

choose a RN $r \in [0, 1]$ with $P(r \leq x) = F_0(x)$
let $\xi = F^{-1}(r)$.

$F(x)$ increases monotonously and therefore

$$P(\xi \leq x) = P(F(\xi) \leq F(x)) = P(r \leq F(x)) = F_0(F(x)) \tag{8.59}$$

but since $0 \leq F(x) \leq 1$ we have

$$P(\xi \leq x) = F(x). \tag{8.60}$$

This method of course is applicable only if F^{-1} can be expressed analytically.

8.2.3 Examples

8.2.3.1 Dice

Tossing a dice can be simulated as follows:

Choose a random number $r \in [0, 1]$ and

$$\text{let } \xi = F^{-1}(r) = \begin{cases} 1 & \text{for} \quad 0 \leq r < \frac{1}{6} \\ 2 & \text{for} \quad \frac{1}{6} \leq r < \frac{2}{6} \\ 3 & \text{for} \quad \frac{2}{6} \leq r < \frac{3}{6} \\ 4 & \text{for} \quad \frac{3}{6} \leq r < \frac{4}{6} \\ 5 & \text{for} \quad \frac{4}{6} \leq r < \frac{5}{6} \\ 6 & \text{for} \quad \frac{5}{6} \leq r < 1 \end{cases}.$$

8.2.3.2 Exponential Distribution

The cumulative distribution function

$$F(x) = 1 - \mathrm{e}^{-x/\lambda} \tag{8.61}$$

which corresponds to the exponential probability density

$$f(x) = \frac{1}{\lambda}\mathrm{e}^{-x/\lambda} \tag{8.62}$$

can be inverted by solving

$$r = 1 - \mathrm{e}^{-x/\lambda} \tag{8.63}$$

for x:

Choose a random number $r \in [0, 1]$
Let $x = F^{-1}(r) = -\lambda \ln(1 - r)$

8.2.3.3 Random Points on the Unit Sphere

We consider the surface element

$$\frac{1}{4\pi} R^2 \mathrm{d}\varphi \sin\theta \mathrm{d}\theta. \tag{8.64}$$

Our aim is to generate points on the unit sphere (θ, φ) with the probability density

$$f(\theta, \varphi)\mathrm{d}\varphi\mathrm{d}\theta = \frac{1}{4\pi}\mathrm{d}\varphi \sin\theta \mathrm{d}\theta = -\frac{1}{4\pi}\mathrm{d}\varphi \mathrm{d}\cos\theta. \tag{8.65}$$

The corresponding cumulative distribution is

$$F(\theta, \varphi) = -\frac{1}{4\pi}\int_1^{\cos\theta} \mathrm{d}\cos\theta \int_0^{\varphi} \mathrm{d}\varphi = \frac{\varphi}{2\pi}\frac{1 - \cos\theta}{2} = F_\varphi F_\theta. \tag{8.66}$$

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 $\theta = F_\theta^{-1}(r_2) = \arccos(1 - 2r_2)$

8.2.3.4 Gaussian Distribution (Box Muller)

For a Gaussian distribution the inverse F^{-1} has no simple analytical form. The famous Box Muller method [36] is based on a two-dimensional normal distribution with probability density

$$f(x, y) = \frac{1}{2\pi}\exp\left\{-\frac{x^2 + y^2}{2}\right\} \tag{8.67}$$

which reads in polar coordinates

$$f(x, y)\mathrm{d}x\mathrm{d}y = f_p(\rho, \varphi)\mathrm{d}\rho\mathrm{d}\varphi\frac{1}{2\pi}\mathrm{e}^{-\rho^2/2}\rho\mathrm{d}\rho\mathrm{d}\varphi. \tag{8.68}$$

Hence

$$f_p(\rho, \varphi) = \frac{1}{2\pi} \rho \mathrm{e}^{-\rho^2/2} \tag{8.69}$$

and the cumulative distribution factorizes

$$F_p(\rho, \varphi) = \frac{1}{2\pi} \varphi \int_0^\rho \rho' \mathrm{e}^{-\rho'^2/2} \mathrm{d}\rho' = \frac{\varphi}{2\pi} (1 - \mathrm{e}^{-\rho^2}) = F_\varphi(\varphi) F_\rho(\rho). \tag{8.70}$$

The inverse of F_ρ is

$$\rho = \sqrt{-\ln(1 - r)} \tag{8.71}$$

and the following algorithm generates Gaussian random numbers:

$r1 = RN \in [0, 1]$
$r2 = RN \in [0, 1]$
$\rho = \sqrt{-\ln(1 - r_1)}$
$\varphi = 2\pi r_2$
$x = \rho \cos \varphi$

8.3 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.

8.3.1 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 (Figs. 8.5 and 8.6):

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 $\frac{Z(r^2 \leq 1)}{N}$.

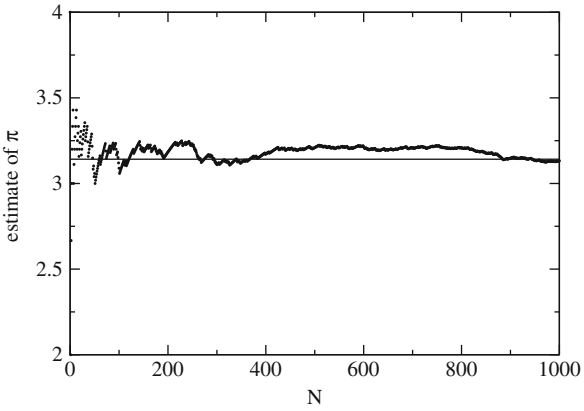


Fig. 8.5 Convergence of the numerical integration

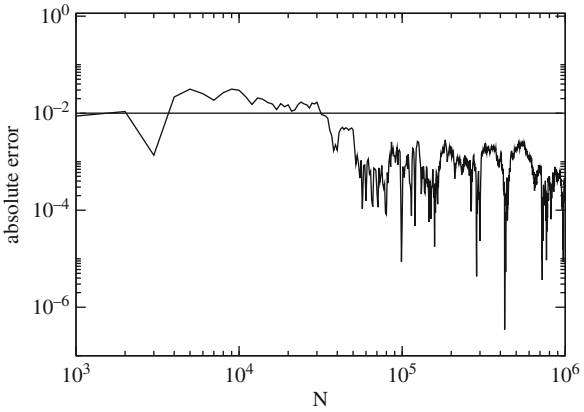


Fig. 8.6 Error of the numerical integration

8.3.2 Calculation of an Integral

Let ξ be a random variable in the interval $[a, b]$ with the distribution

$$P(x < \xi \leq x + \mathrm{d}x) = f(x)\mathrm{d}x = \begin{cases} \frac{1}{b-a} & \text{for} \quad x \in [a, b] \\ 0 & \text{else} \end{cases} . \tag{8.72}$$

The expectation value of a function $g(x)$ is

$$E(g(x)) = \int_{-\infty}^{\infty} g(x) f(x)\mathrm{d}x = \int_a^b g(x)\mathrm{d}x \tag{8.73}$$

hence the average of N randomly taken function values approximates the integral

$$\int_a^b g(x)\mathrm{d}x \approx \frac{1}{N}\sum_{i=1}^{N} g(\xi_i) = \overline{g(\xi)}. \tag{8.74}$$

To estimate the error we consider the new random variable

$$\sigma = \frac{1}{N}\sum_{i=1}^{N} g(\xi). \tag{8.75}$$

Its average is

$$\overline{\sigma} = E(\sigma) = \frac{1}{N}\sum_{i=1}^{N} E(g(x)) = E(g(x)) = \int_a^b g(x)\mathrm{d}x \tag{8.76}$$

and the standard deviation follows from

$$\Delta_\sigma = E\left((\sigma - \overline{\sigma})^2\right) = E\left(\left(\frac{1}{N}\sum g(\xi_i) - \overline{\sigma}\right)^2\right) = E\left(\left(\frac{1}{N}\sum [g(\xi_i) - \overline{\sigma}]\right)^2\right) \tag{8.77}$$

$$= \frac{1}{N^2} E\left(\sum [g(\xi_i) - \overline{\sigma}]^2\right) = \frac{1}{N}(\overline{g(\xi)^2} - \overline{g(\xi)}^2) = \frac{1}{N}\Delta_{g(\xi)}. \tag{8.78}$$

The width of the distribution and hence the uncertainty falls off as $1/\sqrt{N}$.

8.3.3 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

$$\frac{1}{N}\sum_{i=1}^{N} \frac{g(\xi_i)}{f(\xi_i)} = E\left(\frac{g(x)}{f(x)}\right) = \int_a^b \frac{g(x)}{f(x)} f(x)\mathrm{d}x = \int_a^b g(x)\mathrm{d}x. \tag{8.79}$$

The new random variable

$$\tau = \frac{1}{N}\sum_{i=1}^{N} \frac{g(\xi_i)}{f(\xi_i)} \tag{8.80}$$

has a standard deviation given by

$$\Delta_\tau = \frac{1}{N}\Delta\left(\frac{g(\xi)}{f(\xi)}\right) \tag{8.81}$$

which can be reduced by choosing f similar to g. Then preferentially ξ are generated in regions where the integrand is large (importance sampling).

8.4 Monte Carlo Method for Thermodynamic Averages

Consider the partition function of a classical N particle system

$$Z_{NVT} = \frac{1}{N!}\frac{1}{h^{3N}} \int \mathrm{d}p^{3N} \int \mathrm{d}q^{3N} \mathrm{e}^{-\beta H(p_1 \cdots p_N,\, q_1 \cdots q_N)} \tag{8.82}$$

with an energy function

$$H = \sum_{i=1}^{N} \frac{p_i^2}{2m_i} + V(q_1 \cdots q_N). \tag{8.83}$$

If the potential energy does not depend on the momenta and for equal masses the partition functions simplify to

$$\begin{aligned} Z_{NVT} &= \frac{1}{N!}\frac{1}{h^{3N}} \int \mathrm{d}p^{3N} \mathrm{e}^{-\beta \frac{p_i^2}{2m}} \int \mathrm{d}q^{3N} \mathrm{e}^{-\beta V(q)} \\ &= \frac{1}{N!} \left(\frac{2\pi mkT}{h^2}\right)^{3N/2} \int \mathrm{d}q^{3N} \mathrm{e}^{-\beta V(q)} \end{aligned} \tag{8.84}$$

and it remains the calculation of the configuration integral

$$Z_{NVT}^{\mathrm{conf}} = \int \mathrm{d}q^{3N} \mathrm{e}^{-\beta V(q)}. \tag{8.85}$$

In the following we do not need the partition function itself but only averages of some quantity $A(q)$ given by

$$\overline{A} = \frac{\int \mathrm{d}q^{3N} A(q) \mathrm{e}^{-\beta V(q)}}{\int \mathrm{d}q^{3N} \mathrm{e}^{-\beta V(q)}}. \tag{8.86}$$

8.4.1 Simple (Minded) Sampling

Let ξ be a random variable with probability distribution

$$P(\xi \in [q, q + \mathrm{d}q]) = f(q)\mathrm{d}q \tag{8.87}$$

$$\int f(q)\mathrm{d}q = 1. \tag{8.88}$$

We chose M random numbers ξ and calculated the expectation value of $A(\xi)$ from

$$E(A(\xi)) = \lim_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} A(\xi^{(m)}) = \int A(q) f(q) \mathrm{d}q. \tag{8.89}$$

Consider now the case of random numbers ξ equally distributed over the range $q_{\min} \cdots q_{\max}$:

$$f(q) = \begin{cases} \frac{1}{q_{\max} - q_{\min}} & q \in [q_{\min}, q_{\max}] \\ 0 \text{ else} \end{cases}. \tag{8.90}$$

Define a sample by choosing one random value ξ for each of the $3N$ coordinates. The average over a large number M of samples gives the expectation value

$$E\left(A(\xi_1 \cdots \xi_{3N}) \mathrm{e}^{-\beta V(\xi_1 \cdots \xi_{3N})}\right) = \lim_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} A(\xi_i^{(m)}) \mathrm{e}^{-\beta V(\xi_i^{(m)})} \tag{8.91}$$

as

$$\begin{aligned} &\int A(q_i) \mathrm{e}^{-\beta V(q_i)} f(q_1) \cdots f(q_{3N}) \mathrm{d}q_1 \cdots \mathrm{d}q_{3N} \\ &= \frac{1}{(q_{\max} - q_{\min})^{3N}} \int_{q_{\min}}^{q_{\max}} \cdots \int_{q_{\min}}^{q_{\max}} A(q_i) \mathrm{e}^{-\beta V(q_i)} \mathrm{d}q^{3N}. \end{aligned} \tag{8.92}$$

Hence

$$\frac{E\left(A(\xi_i) \mathrm{e}^{-\beta V(\xi_i)}\right)}{E\left(\mathrm{e}^{-\beta V(\xi_i)}\right)} = \frac{\int_{q_{\min}}^{q_{\max}} A(q_i) \mathrm{e}^{-\beta V(q_i)} \mathrm{d}q^{3N}}{\int_{q_{\min}}^{q_{\max}} \mathrm{e}^{-\beta V(q_i)} \mathrm{d}q^{3N}} \approx \overline{A} \tag{8.93}$$

is an approximation to the average of $A(q)$, if the range of the q_i is sufficiently large. However, many of the samples will have small weight and contribute only little.

8.4.2 Importance Sampling

Let us try to sample preferentially the most important configurations. Choose the distribution function as

$$f(q_1 \cdots q_{3N}) = \frac{\mathrm{e}^{-\beta V(q_1 \cdots q_{3N})}}{\int \mathrm{e}^{-\beta V(q_1 \cdots q_{3N})}}. \tag{8.94}$$

Now the expectation value of $A(q)$ approximates the thermal average

$$E\left(A(\xi_1 \cdots \xi_{3N})\right) = \lim_{M\to\infty} \frac{1}{M} \sum_{m=1}^{M} A(\xi_i^{(m)}) = \frac{\int A(q_i) \mathrm{e}^{-\beta V(q_i)} \mathrm{d}q^{3N}}{\int \mathrm{e}^{-\beta V(q_i)} \mathrm{d}q^{3N}} = \overline{A}. \quad (8.95)$$

The partition function is not needed explicitly.

8.4.3 Metropolis Algorithm

The algorithm by Metropolis [37] 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 (Fig. 8.7)

$$W_{i\to j} = P(\mathbf{q}_i \to \mathbf{q}_j) \quad (8.96)$$

are chosen to fulfill the condition of detailed balance

$$\frac{W_{i\to j}}{W_{j\to i}} = \mathrm{e}^{-\beta(V(\mathbf{q}_j)-V(\mathbf{q}_i))}. \quad (8.97)$$

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

$$\Delta N(\mathbf{q}_i) = N_{n+1}(\mathbf{q}_i) - N_n(\mathbf{q}_i) = \sum_{\mathbf{q}_j \in \text{conf.}} N_n(\mathbf{q}_j) W_{j\to i} - N_n(\mathbf{q}_i) W_{i\to j}. \quad (8.98)$$

In thermal equilibrium

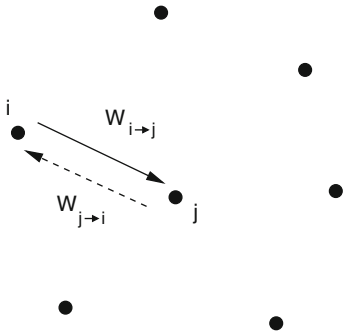


Fig. 8.7 Principle of detailed balance

$$N_{\mathrm{eq}}(\mathbf{q}_i) = N_0 \mathrm{e}^{-\beta V(\mathbf{q}_i)} \tag{8.99}$$

and the changes (8.98) vanish:

$$\begin{aligned}\Delta N(\mathbf{q}_i) &= N_0 \sum_{\mathbf{q}_j} \mathrm{e}^{-\beta V(\mathbf{q}_j)} W_{j\to i} - \mathrm{e}^{-\beta V(\mathbf{q}_i)} W_{i\to j} \\ &= N_0 \sum_{\mathbf{q}_j} \mathrm{e}^{-\beta V(\mathbf{q}_j)} W_{j\to i} - \mathrm{e}^{-\beta V(\mathbf{q}_i)} W_{j\to i} \mathrm{e}^{-\beta(V(\mathbf{q}_j)-V(\mathbf{q}_i))} \\ &= 0. \end{aligned} \tag{8.100}$$

A solution of

$$\Delta N(\mathbf{q}_i) = \sum_{\mathbf{q}_j \in \text{ conf.}} N_n(\mathbf{q}_j) W_{j\to i} - N_n(\mathbf{q}_i) W_{i\to j} = 0 \tag{8.101}$$

corresponds to a zero eigenvalue of the system of equations

$$\sum_{\mathbf{q}_j} N(\mathbf{q}_j) W_{j\to i} - N(\mathbf{q}_i) \sum_{\mathbf{q}_j} W_{i\to j} = \lambda N(\mathbf{q}_i). \tag{8.102}$$

One solution of this eigenvalue equation is given by

$$\frac{N_{\mathrm{eq}}(\mathbf{q}_j)}{N_{\mathrm{eq}}(\mathbf{q}_i)} = \mathrm{e}^{-\beta(V(\mathbf{q}_j)-V(\mathbf{q}_i))}. \tag{8.103}$$

There may be, however, other solutions. For instance, if not all configurations are connected by possible transitions and some isolated configurations are occupied initially.

The Metropolis algorithm consists of the following steps:

(a) choose a new configuration randomly (trial step) with probability $T(\mathbf{q}_i \to \mathbf{q}_{\mathrm{trial}}) = T(\mathbf{q}_{\mathrm{trial}} \to \mathbf{q}_i)$

(b) calculate $R = \frac{\mathrm{e}^{-\beta V(\mathbf{q}_{\mathrm{trial}})}}{\mathrm{e}^{-\beta V(\mathbf{q}_i)}}$

(c) if $R \geq 1$ the trial step is accepted $\mathbf{q}_{i+1} = \mathbf{q}_{\mathrm{trial}}$

(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

$$\mathbf{q}_{i+1} = \begin{cases} \mathbf{q}_{\mathrm{trial}} & \text{if} \quad \xi < R \\ \mathbf{q}_i & \text{if} \quad \xi \geq R \end{cases}.$$

The transition probability is the product

$$W_{i\to j} = T_{i\to j} A_{i\to j} \tag{8.104}$$

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

$$\begin{array}{c} \text{for } R \geq 1 \to A_{i\to j} = 1,\ A_{j\to i} = R^{-1} \\ \text{for } R < 1 \to A_{i\to j} = R,\ A_{j\to i} = 1 \end{array}. \tag{8.105}$$

Since $T_{i\to j} = T_{j\to i}$, in both cases

$$\frac{N_{\text{eq}}(\mathbf{q}_j)}{N_{\text{eq}}(\mathbf{q}_i)} = \frac{W_{i\to j}}{W_{j\to i}} = \frac{A_{i\to j}}{A_{j\to i}} = R = \mathrm{e}^{-\beta(V(\mathbf{q}_j)-V(\mathbf{q}_i))}. \tag{8.106}$$

Problems

Problem 8.1 Central Limit Theorem

This computer experiment draws a histogram for the random variable τ, which is calculated from N random numbers $\xi_1 \cdots \xi_N$:

$$\tau = \frac{\sum_{i=1}^{N} \xi_i}{\sqrt{N}}$$

The ξ_i are random numbers with zero mean and unit variance and can be chosen as

(a) $\xi_i = \pm 1$ (coin tossing)
(b) Gaussian random numbers

Investigate how a Gaussian distribution is approached for large N.

Problem 8.2 Nonlinear Optimization

MC methods can be used for nonlinear optimization (traveling salesman problem, structure optimization, etc.). Consider an energy function depending on a large number of coordinates

$$E(q_1, q_2, \cdots, q_N) \tag{8.107}$$

Introduce a fictitious temperature T and generate configurations with probabilities

$$P(q_1 \cdots q_N) = \frac{1}{Z}\mathrm{e}^{-E(q_1\cdots q_N)/T} \tag{8.108}$$

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 $N = 15$ given points. The fictitious Boltzmann factor for a path with total length L is

$$\mathrm{e}^{-L/T}$$

Starting from an initial path $S = (i_1, i_2, \ldots, i_N)$, $n < 5$ and p are chosen randomly and a new path $S' = (i_1, \ldots, i_{p-1}, i_{p+n}, \ldots, i_p, i_{p+n+1}, \ldots, i_N)$ 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 9
Eigenvalue Problems

Eigenvalue problems are very common in physics. In many cases[1] *they involve solution of a homogeneous system of linear equations*

$$\sum_{j=1}^{N} A_{ij} x_j = \lambda x_i \tag{9.1}$$

with a Hermitian (or symmetric, if real) matrix

$$A_{ji} = A_{ij}^{*}. \tag{9.2}$$

9.1 Direct Solution

For matrices of very small dimension (2, 3) the determinant

$$\det \left| A_{ij} - \lambda \delta_{ij} \right| = 0 \tag{9.3}$$

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

$$\sum_{j} (A_{ij} - \lambda \delta_{ij}) u_j = 0. \tag{9.4}$$

9.2 Jacobi Method

Any symmetric 2×2 matrix can be diagonalized by a rotation of the coordinate system. Rotation by the angle φ corresponds to an orthogonal transformation with the rotation matrix

[1] We do not consider more general eigenvalue problems here.

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_9,

$$R^{(12)} = \begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} \tag{9.5}$$

giving

$$\begin{aligned} &\begin{pmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{pmatrix} \begin{pmatrix} E_1 & V \\ V & E_2 \end{pmatrix} \begin{pmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{pmatrix} = \\ &= \begin{pmatrix} c^2E_1 + s^2E_2 - 2csV & cs(E_1 - E_2) + (c^2 - s^2)V \\ cs(E_1 - E_2) + (c^2 - s^2)V & s^2E_1 + c^2E_2 + 2csV \end{pmatrix} \end{aligned} \tag{9.6}$$

which is diagonal if

$$0 = cs(E_1 - E_2) + (c^2 - s^2)V = \frac{E_1 - E_2}{2}\sin(2\varphi) + V\cos(2\varphi). \tag{9.7}$$

Solving for φ we find

$$\tan(2\varphi) = \frac{2V}{E_2 - E_1} \tag{9.8}$$

or

$$\sin(2\varphi) = \frac{\frac{2V}{E_2 - E_1}}{\sqrt{1 + \frac{4V^2}{(E_2 - E_1)^2}}}. \tag{9.9}$$

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

$$R^{(ij)} = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & c & & s & & \\ & & & \ddots & & & \\ & & -s & & c & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{pmatrix} \tag{9.10}$$

(3) Repeat (1)–(2) until convergence (if possible).

The sequence of Jacobi rotations gives the overall transformation

$$RAR^{-1} = \cdots R_2 R_1 A R_1^{-1} R_2^{-1} \cdots = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_N \end{pmatrix}. \tag{9.11}$$

Hence

$$AR^{-1} = R^{-1} \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_N \end{pmatrix} \tag{9.12}$$

and the column vectors of $R^{-1} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N)$ are the eigenvectors of A:

$$A\,(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N) = (\lambda_1 \mathbf{v}_1, \lambda_2 \mathbf{v}_2, \ldots, \lambda_N \mathbf{v}_N) \tag{9.13}$$

9.3 Tridiagonal Matrices

The characteristic polynomial of a tridiagonal matrix

$$P_A(\lambda) = \det \begin{vmatrix} A_{11} - \lambda & A_{12} & & & \\ A_{21} & A_{22} - \lambda & & & \\ & & \ddots & A_{N-1N} & \\ & & A_{NN-1} & A_{NN} - \lambda & \end{vmatrix} \tag{9.14}$$

can be calculated recursively:

$$\begin{aligned} P_1(\lambda) &= A_{11} - \lambda \\ P_2(\lambda) &= (A_{22} - \lambda) P_1(\lambda) - A_{12}^2 \\ &\vdots \\ P_N(\lambda) &= (A_{NN} - \lambda) P_{N-1}(\lambda) - A_{NN-1}^2 P_{N-2}(\lambda). \end{aligned} \tag{9.15}$$

Eigenvalues and eigenvectors can be obtained, for instance, with the Newton–Raphson method.

9.4 Reduction to a Tridiagonal Matrix

Any symmetric matrix can be transformed to a tridiagonal matrix by a series of Householder transformations (Fig. 9.1) which have the form[2]

[2] $\mathbf{u}\mathbf{u}^t$ is the outer product or matrix product of two vectors.

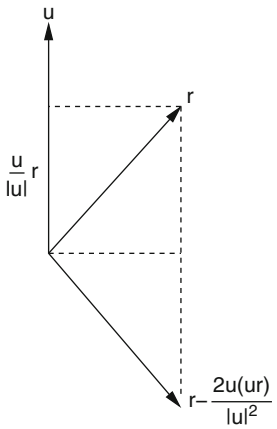


Fig. 9.1 Householder transformation. Geometrically the Householder transformation (9.16) is a mirror operation with respect to a plane with normal vector $\mathbf{u}/|\mathbf{u}|$

$$A' = PAP \quad \text{with} \quad P = P^T = 1 - 2\frac{\mathbf{u}\mathbf{u}^T}{|\mathbf{u}|^2}. \tag{9.16}$$

The following orthogonal transformation P_1 brings the first row and column to tridiagonal form. We divide the matrix A according to

$$A = \begin{pmatrix} A_{11} & \alpha^T \\ \alpha & A_{\text{rest}} \end{pmatrix} \tag{9.17}$$

with the $N-1$ dimensional vector

$$\boldsymbol{\alpha} = \begin{pmatrix} A_{12} \\ \vdots \\ A_{1n} \end{pmatrix}. \tag{9.18}$$

Now let

$$\mathbf{u} = \begin{pmatrix} 0 \\ A_{12} + \lambda \\ \vdots \\ A_{1N} \end{pmatrix} = \begin{pmatrix} 0 \\ \boldsymbol{\alpha} \end{pmatrix} + \lambda \mathbf{e}^{(2)} \quad \text{with} \quad \mathbf{e}^{(2)} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{9.19}$$

Then

$$|\mathbf{u}|^2 = |\boldsymbol{\alpha}|^2 + \lambda^2 + 2\lambda A_{12} \tag{9.20}$$

and

$$\mathbf{u}^T \begin{pmatrix} A_{11} \\ \boldsymbol{\alpha} \end{pmatrix} = |\alpha|^2 + \lambda A_{12}. \tag{9.21}$$

The first row of A is transformed by multiplication with P_1 according to

$$P_1 \begin{pmatrix} A_{11} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} A_{11} \\ \boldsymbol{\alpha} \end{pmatrix} - 2\frac{|\alpha|^2 + \lambda A_{12}}{|\boldsymbol{\alpha}|^2 + \lambda^2 + 2\lambda A_{12}} \left[\begin{pmatrix} 0 \\ \boldsymbol{\alpha} \end{pmatrix} + \lambda \mathbf{e}^{(2)} \right]. \tag{9.22}$$

The elements number $3 \ldots N$ are eliminated if we chose[3]

$$\lambda = \pm|\alpha| \tag{9.23}$$

because then

$$2\frac{|\alpha|^2 + \lambda A_{12}}{|\boldsymbol{\alpha}|^2 + \lambda^2 + 2\lambda A_{12}} = 2\frac{|\alpha|^2 \pm |\alpha| A_{12}}{|\alpha|^2 + |\alpha|^2 \pm 2|\alpha| A_{12}} = 1 \tag{9.24}$$

and

$$P_1 \begin{pmatrix} A_{11} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} A_{11} \\ \boldsymbol{\alpha} \end{pmatrix} - \begin{pmatrix} 0 \\ \boldsymbol{\alpha} \end{pmatrix} - \lambda \mathbf{e}^{(2)} = \begin{pmatrix} A_{11} \\ \mp|\alpha| \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{9.25}$$

Finally we have

$$A^{(2)} = P_1 A P_1 = \begin{pmatrix} A_{11} & A_{12}^{(2)} & 0 & \cdots & 0 \\ A_{12}^{(2)} & A_{22}^{(2)} & A_{23}^{(2)} & \cdots & A_{2N}^{(2)} \\ 0 & A_{23}^{(2)} & \ddots & & A_{3N}^{(2)} \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & A_{2N}^{(2)} & A_{3N}^{(2)} & \cdots & A_{NN}^{(2)} \end{pmatrix} \tag{9.26}$$

as desired.
For the next step we chose

$$\boldsymbol{\alpha} = \begin{pmatrix} A_{22}^{(2)} \\ \vdots \\ A_{2N}^{(2)} \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} 0 \\ 0 \\ \alpha \end{pmatrix} \pm |\alpha| \mathbf{e}^{(3)} \tag{9.27}$$

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

[3] To avoid numerical extinction we chose the sign to be that of A_{12}.

$$A^{(3)} = P_2 A^{(2)} P_2 = \begin{pmatrix} A_{11} & A_{12}^{(2)} & 0 & \cdots\cdots & 0 \\ A_{12}^{(2)} & A_{22}^{(2)} & A_{23}^{(3)} & 0 \cdots & 0 \\ 0 & A_{23}^{(3)} & A_{33}^{(3)} & \cdots\cdots & A_{3N}^{(3)} \\ \vdots & 0 & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & A_{3N}^{(3)} & \cdots\cdots & A_{NN}^{(3)} \end{pmatrix}. \tag{9.28}$$

After N − 1 transformations finally a tridiagonal matrix is obtained.

9.5 Large Matrices

Special algorithms are available for matrices of very large dimension to calculate a small number of eigenvalues and eigenvectors. The famous Lanczos method [38] diagonalizes the matrix in a subspace which is constructed from the vectors

$$\mathbf{x}_0, A\mathbf{x}_0, A^2\mathbf{x}_0, \ldots, A^N\mathbf{x}_0 \tag{9.29}$$

which, starting from an initial normalized guess vector $\mathbf{x}_0$, are orthonormalized to obtain a tridiagonal matrix:

$$\begin{aligned} \mathbf{x}_1 &= \frac{A\mathbf{x}_0 - (\mathbf{x}_0 A\mathbf{x}_0)\mathbf{x}_0}{|A\mathbf{x}_0 - (\mathbf{x}_0 A\mathbf{x}_0)\mathbf{x}_0|} = \frac{A\mathbf{x}_0 - a_0\mathbf{x}_0}{b_0} \\ \mathbf{x}_2 &= \frac{A\mathbf{x}_1 - b_0\mathbf{x}_0 - (\mathbf{x}_1 A\mathbf{x}_1)\mathbf{x}_1}{|A\mathbf{x}_1 - b_0\mathbf{x}_0 - (\mathbf{x}_1 A\mathbf{x}_1)\mathbf{x}_1|} = \frac{A\mathbf{x}_1 - b_0\mathbf{x_0} - a_1\mathbf{x}_1}{b_1} \\ &\vdots \\ \mathbf{x}_N &= \frac{A\mathbf{x}_{N-1} - b_{N-2}\mathbf{x}_{N-2} - (\mathbf{x}_{N-1} A\mathbf{x}_{N-1})\mathbf{x}_{N-1}}{|A\mathbf{x}_{N-1} - b_{N-2}\mathbf{x}_{N-2} - (\mathbf{x}_{N-1} A\mathbf{x}_{N-1})\mathbf{x}_{N-1}|} \\ &= \frac{A\mathbf{x}_{N-1} - b_{N-2}\mathbf{x}_{N-2} - a_{N-1}\mathbf{x}_{N-1}}{b_{N-1}} = \frac{\mathbf{r}_{N-1}}{b_{N-1}}. \end{aligned} \tag{9.30}$$

This series is truncated by setting

$$a_N = (\mathbf{x}_N A\mathbf{x}_N) \tag{9.31}$$

and neglecting

$$\mathbf{r}_N = A\mathbf{x}_N - b_{N-1}\mathbf{x}_{N-1} - a_N\mathbf{x}_N. \tag{9.32}$$

Within the subspace of the $\mathbf{x}_1 \cdots \mathbf{x}_N$ the matrix A is represented by the tridiagonal matrix

$$T = \begin{pmatrix} a_0 & b_0 & & & \\ b_0 & a_1 & b_1 & & \\ & \ddots & \ddots & & \\ & & \ddots & a_{N-1} & b_{N-1} \\ & & & b_{N-1} & a_N \end{pmatrix} \tag{9.33}$$

which can be diagonalized with standard methods. The whole method can be iterated using an eigenvector of T as the new starting vector and increasing N until the desired accuracy is achieved. The main advantage of the Lanczos method is that the matrix A will not be stored in memory. It is sufficient to calculate scalar products with A.

Problems

Problem 9.1 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

$$P(E) = \frac{1}{\Delta\sqrt{2\pi}} e^{-E^2/2\Delta^2}$$

The wave function of the system is given by a linear combination

$$\psi = \sum_{ij} C_{ij}\psi_{ij}$$

where on each particle (i, j) one basis function ψ_{ij} is located. The nonzero elements of the interaction matrix are given by

$$H(ij|ij) = E_{ij}$$
$$H(ij|i \pm 1, j) = H(ij|i, j \pm 1) = V$$

The matrix 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

$$\sum_{ij} |C_{ij}|^4$$

is evaluated. Explore the influence of coupling V and disorder Δ.

Chapter 10
Data Fitting

Often a set of data points have 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

$$(x_i, y_i) \quad i = 1 \ldots N \tag{10.1}$$

and a model function

$$f(x, a_1 \ldots a_m) \quad m \leq N \tag{10.2}$$

which depends on the variable x and $m \leq N$ additional parameters a_j. The errors of the fitting procedure are given by the residuals

$$r_i = y_i - f(x_i, a_1 \ldots a_m). \tag{10.3}$$

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]

$$S(a_1 \ldots a_m) = \frac{1}{N} \sum_{i=1}^{N} r_i^2. \tag{10.4}$$

10.1 Least Square Fit

A (local) minimum of (10.4) corresponds to a stationary point with zero gradient. For m model parameters there are m, generally nonlinear, equations which have to be solved. From the general condition

[1] Minimization of the sum of absolute errors $\sum |r_i|$ is much more complicated.

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_10,

$$\frac{\partial S}{\partial a_j} = 0 \quad j = 1 \ldots m \tag{10.5}$$

we find

$$\sum_{i=1}^{N} r_i \frac{\partial f(x_i, a_1 \ldots a_m)}{\partial a_j} = 0. \tag{10.6}$$

In principle, the methods discussed in (6.2) are applicable. For instance, the Newton–Raphson method starts from a suitable initial guess of parameters

$$(a_1^0 \ldots a_m^0) \tag{10.7}$$

and tries to improve the fit iteratively by making small changes to the parameters

$$a_j^{n+1} = a_j^n + \Delta a_j^n. \tag{10.8}$$

The changes Δa_j^n are determined approximately by expanding the model function

$$f(x_i, a_1^{n+1} \ldots a_m^{n+1}) = f(x_i, a_1^n \ldots a_m^n) + \sum_{j=1}^{m} \frac{\partial f(x_i, a_1^n \ldots a_m^n)}{\partial a_j} \Delta a_j^n + \ldots \tag{10.9}$$

to approximate the new residuals by

$$r_i^{n+1} = r_i^n - \sum_{j=1}^{m} \frac{\partial f(x_i, a_1^n \ldots a_m^n)}{\partial a_j} \Delta a_j^n \tag{10.10}$$

and the derivatives by

$$\frac{\partial r_i^n}{\partial a_j} = -\frac{\partial f(x_i, a_1^n \ldots a_m^n)}{\partial a_j}. \tag{10.11}$$

Equation (10.6) now becomes

$$\sum_{i=1}^{N} \left(r_i^n - \sum_{j=1}^{m} \frac{\partial f(x_i)}{\partial a_j} \Delta a_j^n \right) \frac{\partial f(x_i)}{\partial a_k} \tag{10.12}$$

which is a system of m linear equations for the Δa_j, the so-called normal equations:

$$\sum_{ij} \frac{\partial f(x_i)}{\partial a_j} \frac{\partial f(x_i)}{\partial a_k} \Delta a_j^n = \sum_{i=1}^{N} r_i^n \frac{\partial f(x_i)}{\partial a_k}. \tag{10.13}$$

With

$$A_{kj} = \frac{1}{n}\sum_{i=1}^{N} \frac{\partial f(x_i)}{\partial a_k}\frac{\partial f(x_i)}{\partial a_j} \tag{10.14}$$

$$b_k = \frac{1}{n}\sum_{i=1}^{N} y_i \frac{\partial f(x_i)}{\partial a_k} \tag{10.15}$$

the normal equations can be written as

$$\sum_{j=1}^{p} A_{kj}\Delta a_j = b_k. \tag{10.16}$$

10.1.1 Linear Least Square Fit

Especially important are model functions which depend linearly on the parameters

$$f(x, a_1 \ldots a_m) = \sum_{j=1}^{m} a_j f_j(x). \tag{10.17}$$

The derivatives are simply

$$\frac{\partial f(x_i)}{\partial a_j} = f_j(x_i). \tag{10.18}$$

The minimum of (10.4) is now determined by the normal equations

$$\frac{1}{n}\sum_{j=1}^{p}\sum_{i=1}^{n} f_k(x_i) f_j(x_i) a_j = \frac{1}{n}\sum y_i f_k(x_i) \tag{10.19}$$

which become

$$\sum_{j=1}^{p} A_{kj} a_j = b_k \tag{10.20}$$

with

$$A_{kj} = \frac{1}{n}\sum_{i=1}^{n} f_k(x_i) f_j(x_i) \tag{10.21}$$

$$b_k = \frac{1}{n}\sum_{i=1}^{n} y_i f_k(x_i). \tag{10.22}$$

Example: Linear Regression

For a linear fit function

$$f(x) = a_0 + a_1 x \tag{10.23}$$

we have

$$S = \frac{1}{n}\sum_{i=1}^{n}(y_i - a_0 - a_1 x_i)^2 \tag{10.24}$$

and we have to solve the equations

$$\begin{aligned} 0 = \frac{\partial S}{\partial a_0} &= \frac{1}{n}\sum_{i=1}^{n}(y_i - a_0 - a_1 x_i) = \bar{y} - a_0 - a_1\bar{x} \\ 0 = \frac{\partial S}{\partial a_1} &= \frac{1}{n}\sum_{i=1}^{n}(y_i - a_0 - a_1 x_i)x_i = \overline{yx} - a_0\bar{x} - a_1\overline{x^2} \end{aligned} \tag{10.25}$$

which can be done in this simple case using determinants:

$$a_0 = \frac{\begin{vmatrix} \bar{y} & \bar{x} \\ \overline{xy} & \overline{x^2} \end{vmatrix}}{\begin{vmatrix} 1 & \bar{x} \\ \bar{x} & \overline{x^2} \end{vmatrix}} = \frac{\bar{y}\overline{x^2} - \bar{x}\overline{xy}}{\overline{x^2} - \bar{x}^2} \tag{10.26}$$

$$a_1 = \frac{\begin{vmatrix} 1 & \bar{y} \\ \bar{x} & \overline{xy} \end{vmatrix}}{\begin{vmatrix} 1 & \bar{x} \\ \bar{x} & \overline{x^2} \end{vmatrix}} = \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2}. \tag{10.27}$$

10.1.2 Least Square Fit Using Orthogonalization

The problem to solve the linearized problem (10.12) can be formulated with the definitions

$$\mathbf{x} = \begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} \tag{10.28}$$

and the $N \times m$ matrix

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{Nm} \end{pmatrix} = \begin{pmatrix} \frac{\partial f(x_1)}{\partial a_1} & \cdots & \frac{\partial f(x_1)}{\partial a_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(x_N)}{\partial a_1} & \cdots & \frac{\partial f(x_N)}{\partial a_m} \end{pmatrix} \tag{10.29}$$

as the search for the minimum of

$$S = |A\mathbf{x} - \mathbf{b}| = \sqrt{(A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b})}. \tag{10.30}$$

In the last section we calculated

$$\frac{\partial S^2}{\partial \mathbf{x}} = A^T(A\mathbf{x} - \mathbf{b}) + (A\mathbf{x} - \mathbf{b})^T A = 2A^T A\mathbf{x} - 2A^T\mathbf{b} \tag{10.31}$$

and solved the system of linear equations[2]

$$A^T A\mathbf{x} = A^T\mathbf{b}. \tag{10.32}$$

This method can become numerically unstable. Alternatively we can use orthogonalization of the m column vectors $\mathbf{a}_k$ of A to have

$$A = (\mathbf{a}_1 \cdots \mathbf{a}_m) = (\mathbf{q}_1 \cdots \mathbf{q}_m) \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ & r_{22} & \cdots & r_{2m} \\ & & \ddots & \vdots \\ & & & r_{mm} \end{pmatrix}, \tag{10.33}$$

where $\mathbf{a}_k$ and $\mathbf{q}_k$ are now vectors of dimension N. Since the $\mathbf{q}_k$ are orthonormal $\mathbf{q}_i^t\mathbf{q}_k = \delta_{ik}$ we have

$$\begin{pmatrix} \mathbf{q}_1^T \\ \vdots \\ \mathbf{q}_m^T \end{pmatrix} A = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ & r_{22} & \cdots & r_{2m} \\ & & \ddots & \vdots \\ & & & r_{mm} \end{pmatrix}. \tag{10.34}$$

The $\mathbf{q}_k$ can be augmented by another $(N - m)$ vectors to provide an orthonormal basis of R^n. These will not be needed explicitly. They are orthogonal to the first m vectors and hence to the column vectors of A. All vectors $\mathbf{q}_k$ together form a unitary matrix

$$Q = \begin{pmatrix} \mathbf{q}_1 \cdots \mathbf{q}_m & \mathbf{q}_{m+1} \cdots \mathbf{q}_N \end{pmatrix} \tag{10.35}$$

and we can define the transformation of the matrix A:

$$\widetilde{A} = \begin{pmatrix} \mathbf{q}_1^T \\ \vdots \\ \mathbf{q}_m^T \\ \mathbf{q}_{m+1}^T \\ \vdots \\ \mathbf{q}_N^T \end{pmatrix} (\mathbf{a}_1 \cdots \mathbf{a}_m) = Q^H A = \begin{pmatrix} R \\ 0 \end{pmatrix} \quad R = \begin{pmatrix} r_{11} & \cdots & r_{1N} \\ & \ddots & \vdots \\ & & r_{NN} \end{pmatrix}. \tag{10.36}$$

[2] Also known as normal equations.

The vector $\mathbf{b}$ transforms as

$$\widetilde{\mathbf{b}} = Q^H \mathbf{b} = \begin{pmatrix} \mathbf{b}_u \\ \mathbf{b}_l \end{pmatrix} \quad \mathbf{b}_u = \begin{pmatrix} \mathbf{q}_1^T \\ \vdots \\ \mathbf{q}_m^T \end{pmatrix} \mathbf{b} \quad \mathbf{b}_l = \begin{pmatrix} \mathbf{q}_{m+1}^T \\ \vdots \\ \mathbf{q}_n^T \end{pmatrix} \mathbf{b}. \tag{10.37}$$

Since the norm of a vector is not changed by unitary transformations

$$|\mathbf{b} - A\mathbf{x}| = \sqrt{(\mathbf{b}_u - R\mathbf{x})^2 + \mathbf{b}_l^2} \tag{10.38}$$

which is minimized if

$$R\mathbf{x} = \mathbf{b}_u. \tag{10.39}$$

The error of the fit is given by

$$|\mathbf{b} - A\mathbf{x}| = |\mathbf{b}_l|. \tag{10.40}$$

Example: Linear Regression

Consider again the fit function

$$f(x) = a_0 + a_1 x \tag{10.41}$$

for the measured data (x_i, y_i). The fit problem is to determine

$$\left| \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_N \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} - \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} \right| = \min. \tag{10.42}$$

Orthogonalization of the column vectors

$$\mathbf{a}_1 = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \quad \mathbf{a}_2 = \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix} \tag{10.43}$$

with the Schmidt method gives

$$r_{11} = \sqrt{N} \tag{10.44}$$

$$\mathbf{q}_1 = \begin{pmatrix} \frac{1}{\sqrt{N}} \\ \vdots \\ \frac{1}{\sqrt{N}} \end{pmatrix} \tag{10.45}$$

$$r_{12} = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} x_i = \sqrt{N}\bar{x} \tag{10.46}$$

$$\mathbf{b}_2 = (x_i - \bar{x}) \tag{10.47}$$

$$r_{22} = \sqrt{\sum (x_i - \bar{x})^2} = \sqrt{N}\sigma_x \tag{10.48}$$

$$\mathbf{q}_2 = \left(\frac{x_i - \bar{x}}{\sqrt{N}\sigma_x} \right). \tag{10.49}$$

Transformation of the right-hand side gives

$$\begin{pmatrix} \mathbf{q}_1^t \\ \mathbf{q}_2^t \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} \sqrt{N}\bar{y} \\ \sqrt{N}\dfrac{\overline{yx} - \bar{x}\bar{y}}{\sigma_x} \end{pmatrix} \tag{10.50}$$

and we have to solve the system of linear equations

$$R\mathbf{x} = \begin{pmatrix} \sqrt{N} & \sqrt{N}\bar{x} \\ 0 & \sqrt{N}\sigma \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} = \begin{pmatrix} \sqrt{N}\bar{y} \\ \sqrt{N}\dfrac{\overline{yx} - \bar{x}\bar{y}}{\sigma_x} \end{pmatrix}. \tag{10.51}$$

The solution

$$a_1 = \frac{\overline{yx}/\bar{x}\bar{y}}{(x - \bar{x})^2} \tag{10.52}$$

$$a_0 = \bar{y} - \bar{x}a_1 = \frac{\bar{y}\overline{x^2} - \bar{x}\overline{xy}}{(x - \bar{x})^2} \tag{10.53}$$

coincides with the earlier results since

$$\overline{(x - \bar{x})^2} = \overline{x^2} - \bar{x}^2. \tag{10.54}$$

10.2 Singular Value Decomposition

Computational physics often has to deal with large amounts of data. The method of singular value decomposition is very useful to reduce redundancies and to extract the most important information from data. It has been used, for instance, for image compression [39], it is very useful to extract the essential dynamics from molecular dynamics simulations [40, 41], and it is an essential tool of bio-informatics [42]. The general idea is to approximate an $m \times n$ matrix of data of rank r ($m \geq n \geq r$) by a matrix with smaller rank $l < r$. This can be formally achieved by the decomposition

$$
\begin{aligned}
& X = U\,S\,V^{\mathrm{T}} \\
& \begin{pmatrix} x_{11} & \dots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \dots & x_{mn} \end{pmatrix} \\
& = \begin{pmatrix} u_{11} & \dots & u_{1n} \\ \vdots & \ddots & \vdots \\ u_{m1} & \dots & u_{mn} \end{pmatrix} \begin{pmatrix} s_1 & & \\ & \ddots & \\ & & s_n \end{pmatrix} \begin{pmatrix} v_{11} & \dots & v_{n1} \\ \vdots & \ddots & \vdots \\ v_{1n} & \dots & v_{nn} \end{pmatrix},
\end{aligned} \tag{10.55}
$$

where U is an $m \times n$ matrix, S is an $n \times n$ diagonal matrix, and V is another $n \times n$ matrix. The column vectors of U are called the left singular vectors and are orthonormal

$$\sum_{i=1}^{m} u_{i,r} u_{i,s} = \delta_{r,s} \tag{10.56}$$

as well as the column vectors of V which are called the right singular vectors

$$\sum_{i=1}^{n} v_{i,r} v_{i,s} = \delta_{r,s}. \tag{10.57}$$

The diagonal elements of S are called the singular values. For a square $n \times n$ matrix Eq. (10.55) is equivalent to diagonalization:

$$X = U S U^{\mathrm{T}}. \tag{10.58}$$

Component wise (10.55) reads[3]

$$x_{r,s} = \sum_{i=1}^{r} u_{r,i} s_i v_{s,i}. \tag{10.59}$$

Approximations to X of lower rank are obtained by reducing the sum to only the largest singular values. It can be shown that the matrix of rank $l \leq r$

$$x_{r,s}^{(l)} = \sum_{i=1}^{l} u_{r,i} s_i v_{s,i} \tag{10.60}$$

is the rank-l matrix which minimizes

[3] The singular values are ordered in descending order and the last $(n - r)$ singular values are zero.

$$\sum_{r,s} |x_{r,s} - x_{r,s}^{(l)}|^2. \tag{10.61}$$

One way to perform the singular value decomposition is to consider

$$X^{\mathrm{T}} X = (V S U^{\mathrm{T}})(U S V^{\mathrm{T}}) = V S^2 V^{\mathrm{T}}. \tag{10.62}$$

Hence V and the singular values can be obtained from diagonalization of the square $n \times n$ matrix:

$$X^{\mathrm{T}} X = V\, D\, V^{\mathrm{T}}, \tag{10.63}$$

where the (non-negative) eigenvalues d_i are ordered in descending order. The singular values are

$$S = D^{1/2} = \begin{pmatrix} \sqrt{d_1} & & \\ & \ddots & \\ & & \sqrt{d_n} \end{pmatrix}. \tag{10.64}$$

Now we have to determine a matrix U such that

$$X = U S V^{\mathrm{T}} \tag{10.65}$$

$$XV = US. \tag{10.66}$$

We have to be careful since some of the s_i might be zero. Therefore we consider only the nonzero singular values and retain from the equation

$$\begin{pmatrix} x_{11} & \dots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \dots & x_{mn} \end{pmatrix} \begin{pmatrix} v_{11} & \dots & v_{1n} \\ \vdots & \ddots & \vdots \\ v_{n1} & \dots & v_{nn} \end{pmatrix}$$

$$= \begin{pmatrix} u_{11} & \dots & u_{1n} \\ \vdots & \ddots & \vdots \\ u_{m1} & \dots & u_{mn} \end{pmatrix} \begin{pmatrix} s_1 & & & & & \\ & \ddots & & & & \\ & & s_r & & & \\ & & & 0_{r+1} & & \\ & & & & \ddots & \\ & & & & & 0_n \end{pmatrix} \tag{10.67}$$

only the first r columns of the matrix U

$$\begin{pmatrix} x v_{11} & \dots & x v_{1r} \\ \vdots & \ddots & \vdots \\ x v_{m1} & \dots & x v_{mr} \end{pmatrix} = \begin{pmatrix} u_{11} & \dots & u_{1r} \\ \vdots & \ddots & \vdots \\ u_{m1} & \dots & u_{mr} \end{pmatrix} \begin{pmatrix} s_1 & & \\ & \ddots & \\ & & s_r \end{pmatrix} \tag{10.68}$$

which can be solved by

$$\begin{pmatrix} u_{11} & \dots & u_{1r} \\ \vdots & \ddots & \vdots \\ u_{m1} & \dots & u_{mr} \end{pmatrix} = \begin{pmatrix} xv_{11} & \dots & xv_{1n} \\ \vdots & \ddots & \vdots \\ xv_{m1} & \dots & xv_{mn} \end{pmatrix} \begin{pmatrix} s_1^{-1} & & \\ & \ddots & \\ & & s_r^{-1} \end{pmatrix}. \tag{10.69}$$

The remaining 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.

Example:

Consider the data matrix

$$X^{\mathrm{T}} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2.1 & 3.05 & 3.9 & 4.8 \end{pmatrix}. \tag{10.70}$$

Diagonalization of

$$X^{\mathrm{T}}X = \begin{pmatrix} 55 & 53.95 \\ 53.95 & 52.9625 \end{pmatrix}$$

gives the eigenvalues

$$d_1 = 107.94 \quad d_2 = 0.0216 \tag{10.71}$$

and the eigenvectors

$$V = \begin{pmatrix} -0.714 & 0.7004 \\ -0.7004 & -0.714 \end{pmatrix}. \tag{10.72}$$

Since there are no zero singular values we find

$$\begin{aligned} U &= XVS^{-1} \\ &= \begin{pmatrix} -1.414 & -0.013 \\ -2.898 & -0.098 \\ -4.277 & -0.076 \\ -5.587 & 0.018 \\ -6.931 & 0.076 \end{pmatrix} \begin{pmatrix} 10.39 & \\ & 0.147 \end{pmatrix}^{-1} = \begin{pmatrix} -0.136 & -0.091 \\ -0.279 & -0.667 \\ -0.412 & -0.515 \\ -0.538 & 0.122 \\ -0.667 & 0.517 \end{pmatrix}. \end{aligned} \tag{10.73}$$

This gives the decomposition[4]

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

$$X = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 \end{pmatrix} \begin{pmatrix} s_1 & \\ & s_2 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^{\mathrm{T}} \\ \mathbf{v}_2^{\mathrm{T}} \end{pmatrix} = s_1 \mathbf{u}_1 \mathbf{v}_1^{\mathrm{T}} + s_2 \mathbf{u}_2 \mathbf{v}_2^{\mathrm{T}}$$
$$= \begin{pmatrix} 1.009 & 0.990 \\ 2.069 & 2.030 \\ 3.053 & 2.996 \\ 3.987 & 3.913 \\ 4.947 & 4.854 \end{pmatrix} + \begin{pmatrix} -0.009 & 0.0095 \\ -0.069 & 0.0700 \\ -0.053 & 0.0541 \\ 0.013 & -0.0128 \\ 0.053 & -0.0542 \end{pmatrix}. \tag{10.74}$$

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. If the column vectors of the data matrix are denoted as $\mathbf{x}$ and $\mathbf{y}$ they are approximated by

$$\mathbf{x} = s_1 v_{11} \mathbf{u}_1 \quad \mathbf{y} = s_1 v_{21} \mathbf{u}_1 \tag{10.75}$$

which is a linear relationship between $\mathbf{x}$ and $\mathbf{y}$ (Fig. 10.1)

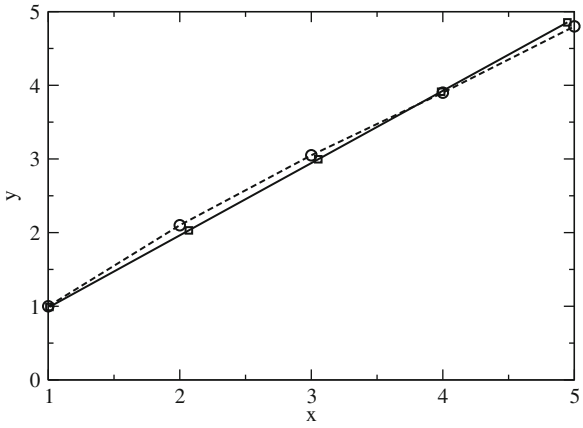


Fig. 10.1 Linear approximation by singular value decomposition. The data set (10.70) is shown as *circles*. The linear approximation which is obtained by retaining only the dominant singular value is shown by the *squares* and the *solid line*

Problems

Problem 10.1 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

$$C(T) = aT + bT^3$$

The computer experiment generates data with a random relative error

$$T_j = T_0 + j\,\Delta t$$
$$C_j = (a_0 T_j + b_0 T_j^3)(1 + \varepsilon_j)$$

and minimizes the sum of squares

$$S = \frac{1}{n} \sum_{j=1}^{n} (C_j - a T_i - b T_i^3)^2$$

Compare the "true values" a_0, b_0 with the fitted values a, b.

Chapter 11
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 differential equation

$$\frac{\mathrm{d}Y}{\mathrm{d}t} = f(Y(t), t), \tag{11.1}$$

where Y is a state vector which contains all information about the system.

11.1 State Vector of a Physical System

In the following we consider models for physical systems which all have in common that the state of the system can be described by specifying the position within a vector space (possibly of very high dimension). This state vector will be denoted by Y. For a classical N-particle system, for instance, the state is given by the position in phase space or equivalently by specifying position and velocity for all of the N particles

$$Y = (\mathbf{r}_1, \mathbf{v}_1, \ldots, \mathbf{r}_N, \mathbf{v}_N)\,. \tag{11.2}$$

The state of a quantum system has to be expanded with respect to some finite basis to become numerically tractable. The elements of the state vector then are the expansion coefficients of the wave function

$$|\Psi\rangle = \sum_{s=1}^{N} C_s |\Psi_s\rangle \tag{11.3}$$

$$Y = (C_1, \ldots, C_N)\,. \tag{11.4}$$

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

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_11,

$$\rho = \sum_{s=1}^{N}\sum_{s'=1}^{N} \rho_{ss'} |\Psi_s\rangle\langle\Psi_{s'}| = \sum_{s=1}^{N}\sum_{s'=1}^{N} \overline{C^*_{s'} C_s} |\Psi_s\rangle\langle\Psi_{s'}| \tag{11.5}$$

$$Y = (\rho_{11} \cdots \rho_{1N}, \rho_{21} \cdots \rho_{2N}, \ldots, \rho_{N1} \cdots \rho_{NN}) \,. \tag{11.6}$$

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 $\mathbf{x}$

$$Y = (c_1(\mathbf{x}), \ldots c_M(\mathbf{x})) \,. \tag{11.7}$$

A quantum particle moving in an external potential can be described by the amplitude of the wave function

$$Y = (\Psi(\mathbf{x})). \tag{11.8}$$

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 some discretization is necessary in the simplest case by introducing a grid of evenly spaced points

$$\mathbf{x}_{ijk} = (i\Delta x, j\Delta x, k\Delta x) \tag{11.9}$$

or in more sophisticated cases by expanding the continuous function with respect to a finite set of basic functions (so-called finite elements).

11.2 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

$$\frac{\mathrm{d}Y}{\mathrm{d}t} = f(Y, t) \tag{11.10}$$

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

$$\frac{\mathrm{d}Y}{\mathrm{d}t} = (\dot{\mathbf{r}}_1, \dot{\mathbf{v}}_1 \cdots) = (\mathbf{v}_1, \mathbf{a}_1 \cdots), \tag{11.11}$$

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 also be considered but are outside the scope of this book):

$$\mathbf{a}_i = \frac{\mathbf{F}_i(\mathbf{r}_1 \cdots \mathbf{r}_N, t)}{m_i}. \tag{11.12}$$

- Diffusion

Heat transport and other diffusive processes are described by the diffusion equation

$$\frac{\partial f}{\partial t} = D\Delta f + S(\mathbf{x}, t) \tag{11.13}$$

which in its simplest spatially discretized version for one-dimensional diffusion reads

$$\frac{\partial f(\mathbf{x}_i)}{\partial t} = \frac{D}{\Delta x^2}\left(f(\mathbf{x}_{i+1}) + f(\mathbf{x}_{i-1}) - 2f(\mathbf{x}_i)\right) + S(\mathbf{x}_i, t). \tag{11.14}$$

- Waves

Consider the simple one-dimensional wave equation

$$\frac{\partial^2 f}{\partial t^2} = c^2 \frac{\partial^2 f}{\partial x^2} \tag{11.15}$$

which by introducing the velocity $g(\mathbf{x}) = \dfrac{\partial}{\partial t} f(\mathbf{x})$ as an independent variable can be rewritten as

$$\frac{\partial}{\partial t}\left(f(\mathbf{x}), g(\mathbf{x})\right) = \left(g(\mathbf{x}), c^2 \frac{\partial^2}{\partial x^2} f(\mathbf{x})\right). \tag{11.16}$$

Discretization of space gives

$$\frac{\partial}{\partial t}\left(f(\mathbf{x}_i), g(\mathbf{x}_i)\right) = \left(g(\mathbf{x}_i), \frac{c^2}{\Delta x^2}\left(f(\mathbf{x}_{i+1}) + f(\mathbf{x}_{i-1}) - 2f(\mathbf{x}_i)\right)\right). \tag{11.17}$$

- Two-level quantum system (TLS)

The Schrödinger equation for a two-level system (for instance, a spin-1/2 particle in a magnetic field) reads

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} H_{11}(t) & H_{12}(t) \\ H_{21}(t) & H_{22}(t) \end{pmatrix}\begin{pmatrix} C_1 \\ C_2 \end{pmatrix}. \tag{11.18}$$

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]:

$$t_{n+1} = t_n + \Delta t. \tag{11.19}$$

Advancing time by one step involves the calculation of the integral

$$Y(t_{n+1}) - Y(t_n) = \int_{t_n}^{t_{n+1}} f(Y(t'), t')\mathrm{d}t' \tag{11.20}$$

which can be a formidable task since $f(Y(t), t)$ depends on time via the time dependence of all the elements of $Y(t)$.

11.3 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. 11.1):

$$Y(t_{n+1}) - Y(t_n) \approx f(Y(t_n), t_n)\Delta t. \tag{11.21}$$

The truncation error can be estimated from a Taylor series expansion

$$\begin{aligned} Y(t_{n+1}) - Y(t_n) &= \Delta t \frac{\mathrm{d}Y}{\mathrm{d}t}(t_n) + \frac{\Delta t^2}{2}\frac{\mathrm{d}^2 Y}{\mathrm{d}t^2}(t_n) + \cdots \\ &= \Delta t f(Y(t_n), t_n) + O(\Delta t^2). \end{aligned} \tag{11.22}$$

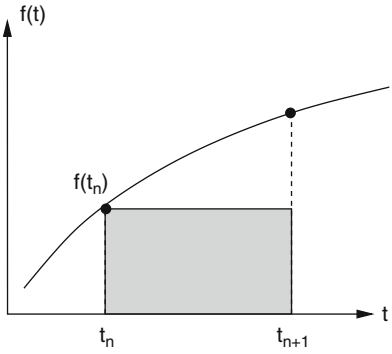


Fig. 11.1 Explicit Euler method

[1] Control of the step width will be discussed later.

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 overall 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):

$$\frac{\mathrm{d}z}{\mathrm{d}t} = \mathrm{i}\omega z. \tag{11.23}$$

The exact solution is obviously given by a circular orbit in the complex plane:

$$z = z_0 e^{\mathrm{i}\omega t} \tag{11.24}$$

$$|z| = |z_0| = \text{const.} \tag{11.25}$$

Application of the Euler method gives

$$z(t_{n+1}) = z(t_n) + \mathrm{i}\omega\Delta t\, z(t_n) = (1 + \mathrm{i}\omega\Delta t) z(t_n) \tag{11.26}$$

and you find immediately

$$|z(t_n)| = \sqrt{1 + \omega^2\Delta t^2}\, |z(t_{n-1})| = \left(1 + \omega^2\Delta t^2\right)^{n/2} |z(t_0)| \tag{11.27}$$

which shows that the radius increases continually even for the smallest time step possible (Fig. 11.2).

The same kind of error appears if you solve the Schrödinger 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

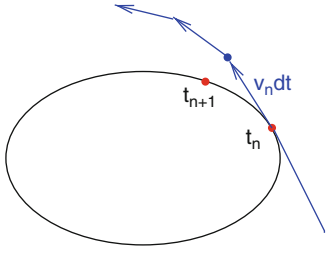


Fig. 11.2 Systematic errors of the Euler method

volume. This can introduce an additional sensitivity of the calculated results to the initial conditions. Consider a harmonic oscillator with the equation of motion

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} x(t) \\ v(t) \end{pmatrix} = \begin{pmatrix} v(t) \\ -\omega^2 x(t) \end{pmatrix}. \tag{11.28}$$

Application of the explicit Euler method gives

$$\begin{pmatrix} x(t+\Delta t) \\ v(t+\Delta t) \end{pmatrix} = \begin{pmatrix} x(t) \\ v(t) \end{pmatrix} + \begin{pmatrix} v(t) \\ -\omega^2 x(t) \end{pmatrix} \Delta t. \tag{11.29}$$

The change of the phase space volume is given by the Jacobi determinant

$$J = \left| \frac{\partial(x(t+\Delta t), v(t+\Delta t))}{\partial(x(t), v(t))} \right| = \begin{vmatrix} 1 & \Delta t \\ -\omega^2 \Delta t & 1 \end{vmatrix} = 1 + (\omega \Delta t)^2. \tag{11.30}$$

In this case the phase space volume increases continuously (Fig. 11.3).

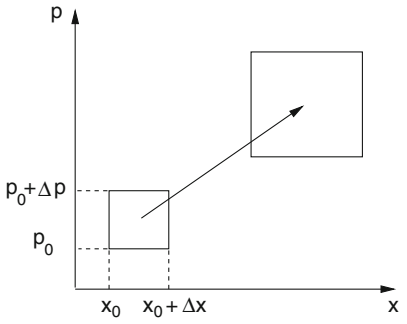


Fig. 11.3 Time evolution of the phase space volume

11.4 Implicit Backward Euler Method

Alternatively let us make a step backward in time (Fig. 11.4)

$$Y(t_n) - Y(t_{n+1}) \approx -f(Y(t_{n+1}), t_{n+1})\Delta t \tag{11.31}$$

which can be written as

$$Y(t_{n+1}) \approx Y(t_n) + f(Y(t_{n+1}), t_{n+1})\Delta t. \tag{11.32}$$

Taylor series expansion gives

$$Y(t_n) = Y(t_{n+1}) - \frac{\mathrm{d}}{\mathrm{d}t} Y(t_{n+1})\Delta t + \frac{\mathrm{d}^2}{\mathrm{d}t^2} Y(t_{n+1}) \frac{\Delta t^2}{2} + \cdots \tag{11.33}$$

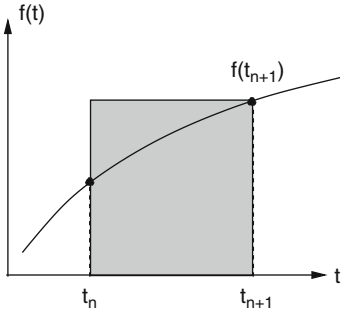


Fig. 11.4 Implicit backward Euler method

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. 11.3 it shows the opposite behavior. The radius of the circular orbit and the phase space volume decrease in time. The gradient at future time has to be estimated before an implicit step can be performed.

11.5 Improved Euler Methods

The quality of the approximation can be improved significantly by employing the midpoint rule (Fig. 11.5)

$$Y(t_{n+1}) - Y(t_n) \approx f\left(Y\left(t + \frac{\Delta t}{2}\right), t_n + \frac{\Delta t}{2}\right)\Delta t. \tag{11.34}$$

The error is smaller by one order of Δt :

$$\begin{aligned} & Y(t_n) + f\left(Y\left(t + \frac{\Delta t}{2}\right), t_n + \frac{\Delta t}{2}\right)\Delta t \\ & = Y(t_n) + \left(\frac{\mathrm{d}Y}{\mathrm{d}t}(t_n) + \frac{\Delta t}{2}\frac{\mathrm{d}^2 Y}{\mathrm{d}t^2}(t_n) + \cdots\right)\Delta t \\ & = Y(t_n + \Delta t) + O(\Delta t^3). \end{aligned} \tag{11.35}$$

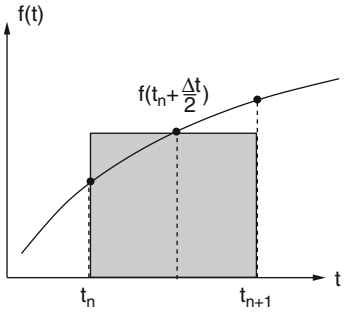


Fig. 11.5 Improved Euler method

The future value $Y\left(t+\frac{\Delta t}{2}\right)$ can be obtained by two different approaches:

- Predictor–corrector method

Since $f\left(Y\left(t+\frac{\Delta t}{2}\right), t_n+\frac{\Delta t}{2}\right)$ 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:

$$\begin{aligned} &\text{Predictor step:} && Y^{(p)} = Y(t_n) + \frac{\Delta t}{2} f(Y(t_n), t_n) \\ &\text{Corrector step:} && Y(t_n+\Delta t) = Y(t_n) + \Delta t\, f\left(Y^{(p)}, t_n+\frac{\Delta t}{2}\right) \end{aligned} \tag{11.36}$$

- Averaging (Heun method)

The average of $f(Y(t_n), t_n)$ and $f(Y(t_n+\Delta t), t+\Delta t)$ (Fig. 11.6) is another approximation to the midpoint value of comparable quality.

Expansion around $t_n + \Delta t/2$ gives

$$\begin{aligned} &\frac{1}{2}\left(f(Y(t_n), t_n) + f(Y(t_n+\Delta t), t+\Delta t)\right) \\ &= f\left(Y\left(t_n+\frac{\Delta t}{2}\right), t_n+\frac{\Delta t}{2}\right) + O(\Delta t^2). \end{aligned} \tag{11.37}$$

Inserting the average in Eq. (11.34) gives the following algorithm, which is also known as improved polygon method and corresponds to the trapezoidal rule for the integral (4.9) or to a combination of explicit and implicit Euler step:

$$Y(t_n+\Delta t) = Y(t_n) + \frac{\Delta t}{2}\left(f(Y(t_n), t_n) + f(Y(t_n+\Delta t), t+\Delta t)\right). \tag{11.38}$$

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

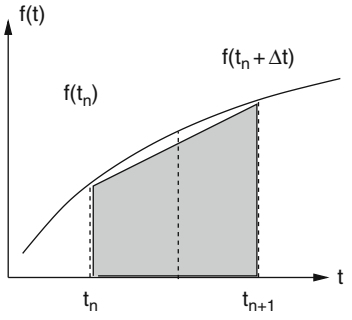


Fig. 11.6 Improved polygon (or Heun) method

$$Y(t_n + \Delta t) = \left(1 - \frac{\Delta t}{2} F\right)^{-1} \left(1 + \frac{\Delta t}{2} F\right) Y(t_n). \tag{11.39}$$

Numerically it is not necessary to perform the matrix inversion. Instead a linear system of equations is solved:

$$\left(1 - \frac{\Delta t}{2} F\right) Y(t_n + \Delta t) = \left(1 + \frac{\Delta t}{2} F\right) Y(t_n). \tag{11.40}$$

In certain cases the Heun method conserves the norm of the state vector, for instance, if F has only imaginary Eigenvalues (as for the one-dimensional Schrödinger equation, see page 280).

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 (11.38) can be evaluated:

$$Y^{(p)} = Y(t_n) + \Delta t \; f(Y(t_n), t_n). \tag{11.41}$$

11.6 Taylor Series Methods

If higher derivatives of f are easily available, then higher order methods can be obtained from the Taylor series expansion

$$Y(t_n + \Delta t) = Y(t_n) + \Delta t \; f(Y(t_n), t_n) + \frac{\Delta t^2}{2} \frac{\mathrm{d} f(Y(t_n), t_n)}{\mathrm{d}t} + \cdots . \tag{11.42}$$

The total time derivative can be expressed as

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\partial f}{\partial Y} \frac{\mathrm{d}Y}{\mathrm{d}t} + \frac{\partial f}{\partial t} = f' f + \dot{f}, \tag{11.43}$$

where the partial derivatives have been abbreviated in the usual way by $\dfrac{\partial f}{\partial t} = \dot{f}$ and $\dfrac{\partial f}{\partial Y} = f'$. Higher derivatives are given by

$$\frac{\mathrm{d}^2 f}{\mathrm{d}t^2} = f'' f^2 + f'^2 f + 2\dot{f}' f + \ddot{f} \tag{11.44}$$

$$\frac{\mathrm{d}^3 f}{\mathrm{d}t^3} = \frac{\partial^3 f}{\partial t^3} + f''' f^3 + 3\dot{f}'' f^2 + \ddot{f} f' + 3 f'' \dot{f} f + 3\dot{f}' + 4 f'' f' f^2 + 5\dot{f}' f' f + f'^3 f + f'^2 \dot{f}. \tag{11.45}$$

11.7 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 [43].

11.7.1 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

$$K_1 = f_n = f(Y(t_n), t_n) \tag{11.46}$$

we estimate the state vector at time $t_n + \Delta t$ as

$$Y(t_n + \Delta t) \approx \Delta t\, K_1. \tag{11.47}$$

The gradient at time $t_n + \Delta t$ is approximately

$$K_2 = f(Y(t_n) + \Delta t\, K_1, t_n + \Delta t) \tag{11.48}$$

which has the Taylor series expansion

$$K_2 = f_n + (\dot{f}_n + f_n' f_n)\Delta t + \cdots \tag{11.49}$$

and application of the trapezoidal rule (4.9) gives the second-order Runge–Kutta method

$$Y_{n+1} = Y_n + \frac{\Delta t}{2}(K_1 + K_2) \tag{11.50}$$

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:

$$\begin{aligned} Y_{n+1} &= Y_n + \frac{\Delta t}{2}(f_n + f_n + (\dot{f}_n + f_n' f_n)\Delta t + \cdots) \\ &= Y_n + f_n \Delta t + \frac{\mathrm{d} f_n}{\mathrm{d}t}\frac{\Delta t^2}{2} + \cdots . \end{aligned} \tag{11.51}$$

11.7.2 Third-Order Runge–Kutta Method

The accuracy can be further improved by calculating one additional function value at midtime. From Eq. (11.46) we estimate the gradient at midtime by

$$K_2 = f\left(Y(t) + \frac{\Delta t}{2}K_1, t + \frac{\Delta t}{2}\right)$$
$$= f_n + (\dot{f}_n + f'_n f_n)\frac{\Delta t}{2} + (\ddot{f}_n + f''_n f_n^2 + 2\dot{f}'_n f_n)\frac{\Delta t^2}{8} + \cdots . \tag{11.52}$$

The gradient at time $t_n + \Delta t$ is then estimated as

$$K_3 = f(Y(t_n) + \Delta t(2K_2 - K_1), t_n + \Delta t)$$
$$= f_n + \dot{f}_n \Delta t + f'_n(2K_2 - K_1)\Delta t + \ddot{f}_n \frac{\Delta t^2}{2}$$
$$+ f''_n \frac{(2K_2 - K_1)^2 \Delta t^2}{2} + 2\dot{f}'_n \frac{(2K_2 - K_1)\Delta t^2}{2} + \cdots . \tag{11.53}$$

Inserting the expansion (11.52) gives the leading terms

$$K_3 = f_n + (\dot{f}_n + f'_n f_n)\Delta t + (2{f'_n}^2 f_n + f''_n f_n^2 + \ddot{f}_n + 2f'_n \dot{f}_n + 2\dot{f}_n^2)\frac{\Delta t^2}{2} + \cdots . \tag{11.54}$$

Applying Simpson's rule (4.10) we combine the three gradients to get the third-order Runge–Kutta method

$$Y_{n+1} = Y(t_n) + \frac{\Delta t}{6}(K_1 + 4K_2 + K_3), \tag{11.55}$$

where the Taylor series

$$Y_{n+1} = Y(t_n) + \frac{\Delta t}{6}\big(6f_n + 3(\dot{f}_n + f_n f'_n)\Delta t$$
$$+ ({f'_n}^2 f_n + f''_n f_n^2 + 2\dot{f}'_n f_n + f_n + \dot{f}_n f'_n)\ddot{}\Delta t^2 + \cdots)$$
$$= Y(t_n + \Delta t) + O(\Delta t^4) \tag{11.56}$$

recovers the exact Taylor series (11.42) including terms of order $O(\Delta t^3)$.

11.7.3 Fourth-Order Runge–Kutta Method

The fourth-order Runge–Kutta method (RK4) is often used because of its robustness and accuracy. It uses two different approximations for the midpoint

$$
\begin{aligned}
K_1 &= f(Y(t_n), t_n) \\
K_2 &= f\left(Y(t_n) + \frac{K_1}{2}\Delta t, t_n + \frac{\Delta t}{2}\right) \\
K_3 &= f\left(Y(t_n) + \frac{K_2}{2}\Delta t, t_n + \frac{\Delta t}{2}\right) \\
K_4 &= f(Y(t_n) + K_3\Delta t, t_n + \Delta t)
\end{aligned} \tag{11.57}
$$

and Simpson's rule (4.10) to obtain

$$
Y_{n+1} = Y(t_n) + \frac{\Delta t}{6}(K_1 + 2K_2 + 2K_3 + K_4) = Y(t_n + \Delta t) + O(\Delta t^5). \tag{11.58}
$$

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.

11.8 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. 11.7). For the fourth order method we estimate the following errors:

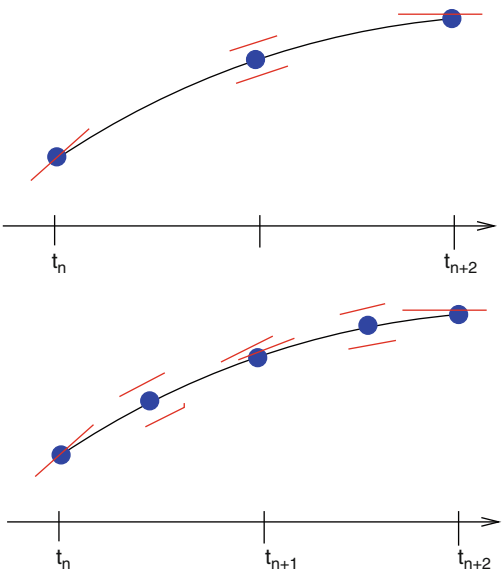


Fig. 11.7 Step doubling with the fourth-order Runge–Kutta method

$$\Delta\left(Y_{n+2}^{(\Delta t)}\right) = 2a\Delta t^5 \tag{11.59}$$

$$\Delta\left(Y_{n+2}^{(2\Delta t)}\right) = a(2\Delta t)^5. \tag{11.60}$$

The local error can be estimated from

$$|Y_{n+2}^{(\Delta t)} - Y_{n+2}^{(2\Delta t)}| = 30|a|\Delta t^5 \tag{11.61}$$

$$\Delta\left(Y_{n+1}^{(\Delta t)}\right) = a\Delta t^5 = \frac{|Y_{n+2}^{(\Delta t)} - Y_{n+2}^{(2\Delta t)}|}{30}. \tag{11.62}$$

The step size Δt can now be adjusted to keep the local error within the desired limits.

11.9 Extrapolation Methods

Application of the extrapolation method to calculate the integral $\int_{t_n}^{t_{n+1}} f(t)\mathrm{d}t$ 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:

$$h = \frac{\Delta t}{N}. \tag{11.63}$$

First a simple Euler step is performed:

$$\begin{aligned} u_0 &= Y(t_n) \\ u_1 &= u_0 + h\, f(u_0, t_n) \end{aligned} \tag{11.64}$$

and then the midpoint rule is applied repeatedly to obtain

$$u_{j+1} = u_{j-1} + 2h\, f(u_j, t_n + jh) \qquad j = 1, 2, \ldots, N-1. \tag{11.65}$$

Gragg [44] introduced a smoothing procedure to remove oscillations of the leading error term by defining

$$v_j = \frac{1}{4}u_{j-1} + \frac{1}{2}u_j + \frac{1}{4}u_{j+1}. \tag{11.66}$$

He showed that both approximations (11.65) and (11.66) 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:

$$
\begin{aligned}
u_0 &= Y(t_n) \\
u_1 &= u_0 + h\, f(u_0, t_n) \\
u_{j+1} &= u_{j-1} + 2h\, f(u_j, t_n + jh) \quad j = 1, 2, \ldots, N-1 \\
Y(t_n + \Delta t) &\approx \frac{1}{2}\left(u_N + u_{N-1} + h\, f(u_N, t_n + \Delta t)\right). \qquad (11.67)
\end{aligned}
$$

The number of sub-steps N is increased according to a sequence like

$$N = 2, 4, 6, 8, 12, 16, 24, 32, 48, 64 \cdots \quad N_j = 2N_{j-2} \quad \text{Bulirsch–Stoer sequence} \qquad (11.68)$$

or

$$N = 2, 4, 6, 8, 10, 12 \cdots \quad N_j = 2j \quad \text{Deuflhard sequence.} \qquad (11.69)$$

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 [45, 46].

11.10 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. These make use of the gradients from several steps and improve the accuracy by polynomial interpolation.

11.10.1 Explicit Multistep Methods

The explicit Adams–Bashforth method of order r uses the gradients from the last $r - 1$ steps to obtain the polynomial (Fig. 11.8)

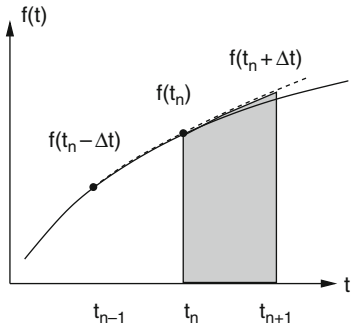


Fig. 11.8 Adams–Bashforth method

$$p(t_n) = f(Y_n, t_n), \cdots p(t_{n-r+1}) = f(Y_{n-r+1}, t_{n-r+1}) \tag{11.70}$$

and to calculate the approximation

$$Y_{n+1} - Y_n \approx \int_{t_n}^{t_{n+1}} p(t)\mathrm{d}t \tag{11.71}$$

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

$$\begin{aligned}
Y_{n+1} - Y_n &= \frac{\Delta t}{2}(3f_n - f_{n-1}) + O(\Delta t^3) \\
Y_{n+1} - Y_n &= \frac{\Delta t}{12}(23f_n - 16f_{n-1} + 5f_{m-2}) + O(\Delta t^4) \\
Y_{n+1} - Y_n &= \frac{\Delta t}{24}(55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) + O(\Delta t^5).
\end{aligned} \tag{11.72}$$

11.10.2 Implicit Multistep Methods

The implicit Adams–Moulton method also uses the yet not known value y_{n+1} to obtain the polynomial (Fig. 11.9)

$$p(t_{n+1}) = f_{n+1}, \ldots, p(t_{n-r+2}) = f_{n-r+2}. \tag{11.73}$$

The corresponding Adams–Moulton formulas of order 2–4 are

$$\begin{aligned}
Y_{n+1} - Y_n &= \frac{\Delta t}{2}(f_{n+1} + f_n) + O(\Delta t^3) \\
Y_{n+1} - Y_n &= \frac{\Delta t}{12}(5f_{n+1} + 8f_n - f_{n-1}) + O(\Delta t^4) \\
Y_{n+1} - Y_n &= \frac{\Delta t}{24}(9f_{n+1} + 19f_n - 5f_{n-1} + f_{n-2}) + O(\Delta t^5).
\end{aligned} \tag{11.74}$$

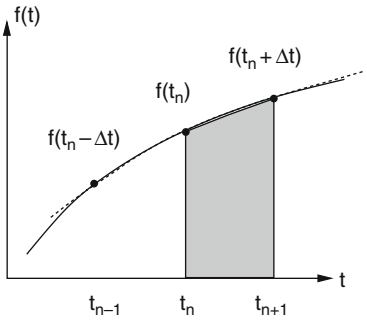


Fig. 11.9 Adams–Moulton method

11.10.3 Predictor–Corrector Methods

The Adams–Bashforth–Moulton method combines the explicit method as a predictor step to calculate an estimate y_{n+1}^{p} with a corrector step using the implicit method of same order.

Multistep methods need fewer function evaluations. They have to be combined with other methods (like Runge–Kutta) to start and end properly. A change of the step size is rather complicated.

11.11 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 [47–50]. The equations of motion of a classical interacting N-body system are

$$m_i \ddot{\mathbf{x}}_i = \mathbf{F}_i, \tag{11.75}$$

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

$$\begin{pmatrix} \dot{\mathbf{x}}_i \\ \dot{\mathbf{v}}_i \end{pmatrix} = \begin{pmatrix} \mathbf{v}_i \\ \mathbf{a}_i \end{pmatrix}, \tag{11.76}$$

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.

11.11.1 Liouville Equation

We rewrite (11.76) as

$$\begin{pmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{v}} \end{pmatrix} = \mathcal{L} \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix}, \tag{11.77}$$

where the Liouville operator $\mathcal{L}$ acts on the vector containing all coordinates and velocities:

$$\mathcal{L} \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} = \left(\mathbf{v} \frac{\partial}{\partial \mathbf{x}} + \mathbf{a} \frac{\partial}{\partial \mathbf{v}} \right) \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} \tag{11.78}$$

The Liouville equation (11.77) can be formally solved by

$$\begin{pmatrix} \mathbf{x}(t) \\ \mathbf{v}(t) \end{pmatrix} = e^{\mathcal{L}t} \begin{pmatrix} \mathbf{x}(0) \\ \mathbf{v}(0) \end{pmatrix}. \tag{11.79}$$

For a better understanding let us evaluate the first members of the Taylor series of the exponential:

$$\mathcal{L}\begin{pmatrix}\mathbf{x}\\ \mathbf{v}\end{pmatrix} = \left(\mathbf{v}\frac{\partial}{\partial \mathbf{x}} + \mathbf{a}\frac{\partial}{\partial \mathbf{v}}\right)\begin{pmatrix}\mathbf{x}\\ \mathbf{v}\end{pmatrix} = \begin{pmatrix}\mathbf{v}\\ \mathbf{a}\end{pmatrix} \tag{11.80}$$

$$\mathcal{L}^2\begin{pmatrix}\mathbf{x}\\ \mathbf{v}\end{pmatrix} = \left(\mathbf{v}\frac{\partial}{\partial \mathbf{x}} + \mathbf{a}\frac{\partial}{\partial \mathbf{v}}\right)\begin{pmatrix}\mathbf{v}\\ \mathbf{a}(\mathbf{x})\end{pmatrix} = \begin{pmatrix}\mathbf{a}\\ \mathbf{v}\dfrac{\partial}{\partial \mathbf{x}}\mathbf{a}\end{pmatrix} \tag{11.81}$$

$$\mathcal{L}^3\begin{pmatrix}\mathbf{x}\\ \mathbf{v}\end{pmatrix} = \left(\mathbf{v}\frac{\partial}{\partial \mathbf{x}} + \mathbf{a}\frac{\partial}{\partial \mathbf{v}}\right)\begin{pmatrix}\mathbf{a}\\ \mathbf{v}\frac{\partial}{\partial \mathbf{x}}\mathbf{a}\end{pmatrix} = \begin{pmatrix}\mathbf{v}\dfrac{\partial}{\partial \mathbf{x}}\mathbf{a}\\ \mathbf{a}\dfrac{\partial}{\partial \mathbf{x}}\mathbf{a} + \mathbf{v}\mathbf{v}\dfrac{\partial}{\partial \mathbf{x}}\dfrac{\partial}{\partial \mathbf{x}}\mathbf{a}\end{pmatrix}. \tag{11.82}$$

But since

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{a}(\mathbf{x}(t)) = \mathbf{v}\frac{\partial}{\partial \mathbf{x}}\mathbf{a} \tag{11.83}$$

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2}\mathbf{a}(\mathbf{x}(t)) = \frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{v}\frac{\partial}{\partial \mathbf{x}}\mathbf{a}\right) = \mathbf{a}\frac{\partial}{\partial \mathbf{x}}\mathbf{a} + \mathbf{v}\mathbf{v}\frac{\partial}{\partial \mathbf{x}}\frac{\partial}{\partial \mathbf{x}}\mathbf{a} \tag{11.84}$$

we recover

$$\left(1 + t\mathcal{L} + \frac{1}{2}t^2\mathcal{L}^2 + \frac{1}{6}t^3\mathcal{L}^3 + \cdots\right)\begin{pmatrix}\mathbf{x}\\ \mathbf{v}\end{pmatrix} = \begin{pmatrix}\mathbf{x} + \mathbf{v}t + \dfrac{1}{2}t^2\mathbf{a} + \dfrac{1}{6}t^3\dot{\mathbf{a}} + \cdots\\ \mathbf{v} + \mathbf{a}t + \dfrac{1}{2}t^2\dot{\mathbf{a}} + \dfrac{1}{6}t^3\ddot{\mathbf{a}} + \cdots\end{pmatrix}. \tag{11.85}$$

11.11.2 Split Operator Approximation

We introduce a small time step $\Delta t = t/N$ and write

$$e^{\mathcal{L}t} = \left(e^{\mathcal{L}\Delta t}\right)^N. \tag{11.86}$$

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

$$\mathcal{L}_A = \mathbf{v}\frac{\partial}{\partial \mathbf{x}} \qquad \mathcal{L}_B = \mathbf{a}\frac{\partial}{\partial \mathbf{v}} \tag{11.87}$$

and make the approximation

$$e^{\mathcal{L}\Delta t} = e^{\mathcal{L}_A\Delta t}e^{\mathcal{L}_B\Delta t} + \cdots. \tag{11.88}$$

Each of the two factors simply shifts positions or velocities

$$e^{\mathcal{L}_A \Delta t} \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \mathbf{x} + \mathbf{v}\Delta t \\ \mathbf{v} \end{pmatrix} \qquad e^{\mathcal{L}_B \Delta t} \begin{pmatrix} \mathbf{x} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \mathbf{x} \\ \mathbf{v} + \mathbf{a}\Delta t \end{pmatrix} \tag{11.89}$$

since these two steps correspond to motion with either constant velocities or constant coordinates and forces.

11.11.3 Position Verlet Method

Often the following approximation is used which is symmetrical in time (Fig. 11.10)

$$e^{\mathcal{L}\Delta t} = e^{\mathcal{L}_A \Delta t/2} e^{\mathcal{L}_B \Delta t} e^{\mathcal{L}_A \Delta t/2} + \cdots \tag{11.90}$$

The corresponding algorithm is the so-called position Verlet method:

$$\mathbf{x}_{n+1/2} = \mathbf{x}_n + \mathbf{v}_n \frac{\Delta t}{2} \tag{11.91}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_n + \mathbf{a}_{n+1/2}\Delta t = \mathbf{v}(t_n + \Delta t) + O(\Delta t^3) \tag{11.92}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_{n+1/2} + \mathbf{v}_{n+1}\frac{\Delta t}{2} = \mathbf{x}_n + \frac{\mathbf{v}_n + \mathbf{v}_{n+1}}{2}\Delta t = \mathbf{x}(t_n + \Delta t) + O(\Delta t^3). \tag{11.93}$$

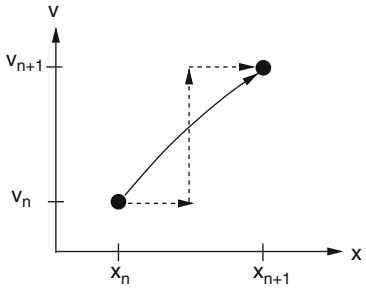


Fig. 11.10 Position Verlet method. The exact integration path is approximated by two half-steps with constant velocities and one step with constant coordinates

11.11.4 Velocity Verlet Method

If we exchange operators A and B we have (Fig. 11.11)

$$e^{\mathcal{L}\Delta t} = e^{\mathcal{L}_B \Delta t/2} e^{\mathcal{L}_A \Delta t} e^{\mathcal{L}_B \Delta t/2} + \cdots \tag{11.94}$$

which produces the velocity Verlet algorithm:

$$\mathbf{v}_{n+1/2} = \mathbf{v}_n + \mathbf{a}_n \frac{\Delta t}{2} \tag{11.95}$$

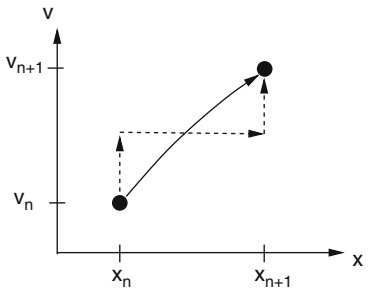


Fig. 11.11 Velocity Verlet method. The exact integration path is approximated by two half-steps with constant coordinates and one step with constant velocities

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{v}_{n+1/2}\Delta t = \mathbf{x}_n + \mathbf{v}_n \Delta t + \mathbf{a}_n \frac{\Delta t^2}{2} = \mathbf{x}(t_n + \Delta t) + O(\Delta t^3) \tag{11.96}$$

$$\mathbf{v}_{n+1} = \mathbf{v}_{n+1/2} + \mathbf{a}_{n+1}\frac{\Delta t}{2} = \mathbf{v}_n + \frac{\mathbf{a}_n + \mathbf{a}_{n+1}}{2}\Delta t = \mathbf{v}(t_n + \Delta t) + O(\Delta t^3). \tag{11.97}$$

11.11.5 Standard Verlet Method

The velocity Verlet method is equivalent to the standard Verlet method which is a two-step method given by

$$\mathbf{x}_{n+1} = 2\mathbf{x}_n - \mathbf{x}_{n-1} + \mathbf{a}_n \Delta t^2 \tag{11.98}$$

$$\mathbf{v}_n = \frac{\mathbf{x}_{n+1} - \mathbf{x}_{n-1}}{2\Delta t}. \tag{11.99}$$

To show the equivalence we add two consecutive position vectors

$$\mathbf{x}_{n+2} + \mathbf{x}_{n+1} = 2\mathbf{x}_{n+1} + 2\mathbf{x}_n - \mathbf{x}_n - \mathbf{x}_{n-1} + (\mathbf{a}_{n+1} + \mathbf{a}_n)\Delta t^2 \tag{11.100}$$

which simplifies to

$$\mathbf{x}_{n+2} - \mathbf{x}_n - (\mathbf{x}_{n+1} - \mathbf{x}_n) = (\mathbf{a}_{n+1} + \mathbf{a}_n)\Delta t^2. \tag{11.101}$$

This can be expressed as the difference of two consecutive velocities:

$$2(\mathbf{v}_{n+1} - \mathbf{v}_n) = (\mathbf{a}_{n+1} + \mathbf{a}_n)\Delta t. \tag{11.102}$$

Now we substitute

$$\mathbf{x}_{n-1} = \mathbf{x}_{n+1} - 2\mathbf{v}_n \Delta t \tag{11.103}$$

to get

$$\mathbf{x}_{n+1} = 2\mathbf{x}_n - \mathbf{x}_{n+1} + 2\mathbf{v_n}\Delta t + \mathbf{a}_n \Delta t^2 \tag{11.104}$$

which simplifies to

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{v}_n \Delta t + \frac{\mathbf{a}_n}{2}\Delta t^2. \tag{11.105}$$

Thus the equations of the velocity Verlet algorithm have been recovered. However, since the Verlet method is a two-step method, the results depend on the initial values. The standard Verlet method starts from two coordinate sets x_0, x_1. The first step is

$$\mathbf{x}_2 = 2\mathbf{x_1} - \mathbf{x_0} + a_1 \Delta t^2 \tag{11.106}$$

$$\mathbf{v}_1 = \frac{\mathbf{x}_2 - \mathbf{x}_0}{2\Delta t} = \frac{\mathbf{x}_1 - \mathbf{x}_0}{\Delta t} + \frac{\mathbf{a}_1}{2}\Delta t^2. \tag{11.107}$$

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

$$\mathbf{x}_2 = \mathbf{x}_1 + \mathbf{v}_1 \Delta t + \mathbf{a}_1 \frac{\Delta t^2}{2} \tag{11.108}$$

$$\mathbf{v}_2 = \mathbf{v}_1 + \frac{\mathbf{a}_1 + \mathbf{a}_2}{2}\Delta t. \tag{11.109}$$

The two methods give the same resulting trajectory if we choose

$$\mathbf{x}_0 = \mathbf{x}_1 - \mathbf{v}_1 \Delta t + \frac{\mathbf{a}_1}{2}\Delta t^2. \tag{11.110}$$

If, on the other hand, $\mathbf{x}_0$ is known with higher precision, the local error order of the standard Verlet changes as can be seen from addition of the two Taylor series

$$\mathbf{x}(t_n + \Delta t) = \mathbf{x}_n + \mathbf{v}_n \Delta t + \frac{\mathbf{a}_n}{2}\Delta t^2 + \frac{\dot{\mathbf{a}}_n}{6}\Delta t^3 + \cdots \tag{11.111}$$

$$\mathbf{x}(t_n - \Delta t) = \mathbf{x}_n - \mathbf{v}_n \Delta t + \frac{\mathbf{a}_n}{2}\Delta t^2 - \frac{\dot{\mathbf{a}}_n}{6}\Delta t^3 + \cdots \tag{11.112}$$

which gives

$$\mathbf{x}(t_n + \Delta t) = 2\mathbf{x}(t_n) - \mathbf{x}(t_n - \Delta t) + \mathbf{a}_n \Delta t^2 + O(\Delta t^4) \tag{11.113}$$

$$\frac{\mathbf{x}(t_n + \Delta t) - \mathbf{x}(t_n - \Delta t)}{2\Delta t} = \mathbf{v}_n + O(\Delta t^2). \tag{11.114}$$

11.11.6 Error Accumulation for the Standard Verlet Method

Equation (11.113) 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

$$\begin{aligned} \Delta x_3 &= 2\Delta x_2 - \Delta x_1 = 2\alpha \Delta t^4 \\ \Delta x_4 &= 2\Delta x_3 - \Delta x_2 = 3\alpha \Delta t^4 \\ &\vdots \\ \Delta x_{n+1} &= n\alpha \Delta t^4. \end{aligned} \tag{11.115}$$

This shows that the effective error order of the Verlet method is only $O(\Delta t^3)$ similar to the velocity Verlet method.

11.11.7 Leap Frog Method

Closely related to the Verlet methods is the so-called leap frog method. It uses the simple decomposition

$$e^{\mathcal{L}\Delta t} \approx e^{\mathcal{L}_A \Delta t} e^{\mathcal{L}_B \Delta t} \tag{11.116}$$

but introduces two different time grids for coordinates and velocities which are shifted by $\Delta t/2$ (Fig. 11.12).

The leap frog algorithm is given by

$$\mathbf{v}_{n+1/2} = \mathbf{v}_{n-1/2} + \mathbf{a}_n \Delta t \tag{11.117}$$

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{v}_{n+1/2} \Delta t. \tag{11.118}$$

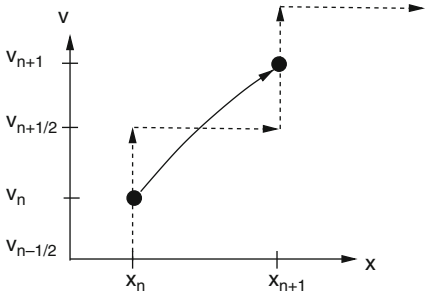


Fig. 11.12 Leap frog method. The exact integration path is approximated by one step with constant coordinates and one step with constant velocities. Two different grids are used for coordinates and velocities which are shifted by $\Delta t/2$

Due to the shifted arguments the order of the method is increased as can be seen from the Taylor series:

$$\mathbf{x}(t_n) + \left(\mathbf{v}(t_n) + \frac{\Delta t}{2}\mathbf{a}(t_n) + \cdots\right)\Delta t = \mathbf{x}(t_n + \Delta t) + O(\Delta t^3) \tag{11.119}$$

$$\mathbf{v}\left(t_n + \frac{\Delta t}{2}\right) - \mathbf{v}\left(t_n - \frac{\Delta t}{2}\right) = \mathbf{a}(t_n)\Delta t + O(\Delta t^3). \tag{11.120}$$

One disadvantage of the leap frog method is that some additional effort is necessary if the velocities are needed. The simple expression

$$\mathbf{v}(t_n) = \frac{1}{2}\left(\mathbf{v}\left(t_n - \frac{\Delta t}{2}\right) + \mathbf{v}\left(t_n + \frac{\Delta t}{2}\right)\right) + O(\Delta t^2) \tag{11.121}$$

is of lower error order than (11.120).

Problems

Problem 11.1 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:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{v}_x \\ \dot{v}_y \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\dfrac{1}{(x^2+y^2)^{3/2}} & 0 & 0 & 0 \\ 0 & -\dfrac{1}{(x^2+y^2)^{3/2}} & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix}$$

For initial values

$$\begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

the exact solution is given by

$$x = \cos t \quad y = \sin t.$$

The following methods are used to calculate the position $x(t)$, $y(t)$ and the energy

$$E_{\text{tot}} = E_{\text{kin}} + E_{\text{pot}} = \frac{1}{2}(v_x^2 + v_y^2) - \frac{1}{\sqrt{x^2+y^2}}$$

- The explicit Euler method (11.4):

$$\begin{aligned} x(t_{n+1}) &= x(t_n) + v_x(t_n)\Delta t \\ y(t_{n+1}) &= y(t_n) + v_y(t_n)\Delta t \\ v_x(t_{n+1}) &= v_x(t_n) - \frac{x(t_n)}{R(t_n)^3}\Delta t \\ v_y(t_{n+1}) &= v_y(t_n) - \frac{y(t_n)}{R(t_n)^3}\Delta t \end{aligned}$$

- The second-order Runge–Kutta method (11.7.1)

which consists of the predictor step

$$\begin{aligned} x(t_n + \Delta t/2) &= x(t_n) + \frac{\Delta t}{2} v_x(t_n) \\ y(t_n + \Delta t/2) &= y(t_n) + \frac{\Delta t}{2} v_y(t_n) \\ v_x(t_n + \Delta t/2) &= v_x(t_n) - \frac{\Delta t}{2}\frac{x(t_n)}{R(t_n)^3} \\ v_y(t_n + \Delta t/2) &= v_y(t_n) - \frac{\Delta t}{2}\frac{y(t_n)}{R(t_n)^3} \end{aligned}$$

and the corrector step

$$\begin{aligned} x(t_{n+1}) &= x(t_n) + \Delta t\ v_x(t_n + \Delta t/2) \\ y(t_{n+1}) &= y(t_n) + \Delta t\ v_y(t_n + \Delta t/2) \\ v_x(t_{n+1}) &= v_x(t_n) - \Delta t \frac{x(t_n + \Delta t/2)}{R^3(t_n + \Delta t/2)} \\ v_y(t_{n+1}) &= v_y(t_n) - \Delta t \frac{y(t_n + \Delta t/2)}{R^3(t_n + \Delta t/2)} \end{aligned}$$

- The fourth-order Runge–Kutta method (11.7.3)
- The Verlet method (11.11.5)

$$\begin{aligned} x(t_{n+1}) &= x(t_n) + (x(t_n) - x(t_{n-1})) - \Delta t \frac{x(t_n)}{R^3(t_n)} \\ y(t_{n+1}) &= y(t_n) + (y(t_n) - y(t_{n-1})) - \Delta t \frac{y(t_n)}{R^3(t_n)} \\ v_x(t_n) &= \frac{x(t_{n+1}) - x(t_{n-1})}{2\Delta t} = \frac{x(t_n) - x(t_{n-1})}{\Delta t} - \frac{\Delta t}{2}\frac{x(t_n)}{R^3(t_n)} \\ v_y(t_n) &= \frac{y(t_{n+1}) - y(t_{n-1})}{2\Delta t} = \frac{y(t_n) - y(t_{n-1})}{\Delta t} - \frac{\Delta t}{2}\frac{y(t_n)}{R^3(t_n)} \end{aligned}$$

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

$$x(t_{-1}) = x(t_0) - \Delta t\, v_x(t_0) - \frac{\Delta t^2}{2}\frac{x(t_0)}{R^3(t_0)}$$
$$y(t_{-1}) = y(t_0) - \Delta t\, v_y(t_0) - \frac{\Delta t^2}{2}\frac{y(t_0)}{R^3(t_0)}$$

- The leap frog method (11.11.7)

$$x(t_{n+1}) = x(t_n) + v_x(t_{n+1/2})\Delta t$$
$$y(t_{n+1}) = y(t_n) + v_y(t_{n+1/2})\Delta t$$
$$v_x(t_{n+1/2}) = v_x(t_{n-1/2}) - \frac{x(t_n)}{R(t_n)^3}\Delta t$$
$$v_y(t_{n+1/2}) = v_y(t_{n-1/2}) - \frac{y(t_n)}{R(t_n)^3}\Delta t$$

where the velocity at time t_n is calculated from

$$v_x(t_n) = v_x(t_{n+1/2}) - \frac{\Delta t}{2}\frac{x(t_{n+1})}{R^3(t_{n+1})}$$
$$v_y(t_n) = v_y(t_{n+1/2}) - \frac{\Delta t}{2}\frac{y(t_{n+1})}{R^3(t_{n+1})}$$

To start the leap frog method we need the velocity at time $t_{-1/2}$ which can be taken from the exact solution or from

$$v_x(t_{-1/2}) = v_x(t_0) - \frac{\Delta t}{2}\frac{x(t_0)}{R^3(t_0)}$$
$$v_y(t_{-1/2}) = v_y(t_0) - \frac{\Delta t}{2}\frac{y(t_0)}{R^3(t_0)}$$

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 leap frog and Verlet methods.

Problem 11.2 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 fourth-order Runge–Kutta method with quality control. The influence of the step size can be studied. The local integration error is estimated using the step doubling method.

Try to simulate a planet with a moon moving round a sun!

Problem 11.3 Adams–Bashforth Method

In this computer experiment we simulate a circular orbit with the Adams–Bashforth method of order $2 \cdots 7$. The absolute error at time T

$$\Delta(T) = |x(T) - \cos(T)| + |y(t) - \sin(T)| + |v_x(T) + \sin(T)| + |v_y(T) - \cos(T)|$$

is shown as a function of the time step Δt in a log–log plot. From the slope

$$s = \frac{\mathrm{d}(\log_{10}(\Delta))}{\mathrm{d}(\log_{10}(\Delta t))}$$

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 12
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 rotation matrix. In this chapter we discuss different parametrizations of the rotation matrix and methods to calculate the time evolution.

12.1 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) (Fig. 12.1).

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

$$\mathbf{R} = \frac{1}{\sum m_i} \sum m_i \mathbf{r}_i \tag{12.1}$$

and the position of m_i within the COM coordinate system $\mathrm{CS_c}$ is $\boldsymbol{\rho}_i$:

$$\mathbf{r}_i = \mathbf{R} + \boldsymbol{\rho}_i \tag{12.2}$$

Let us define a body fixed coordinate system $\mathrm{CS_{cb}}$, where the position $\boldsymbol{\rho}_{\mathbf{ib}}$ of m_i is time independent $\frac{d}{dt}\boldsymbol{\rho}_{\mathbf{ib}} = 0$. $\boldsymbol{\rho}_{\mathbf{i}}$ and $\boldsymbol{\rho}_{ib}$ are connected by a linear vector function

$$\boldsymbol{\rho}_{\mathbf{i}} = A\boldsymbol{\rho}_{ib}, \tag{12.3}$$

where A is a 3×3 matrix

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}. \tag{12.4}$$

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_12,

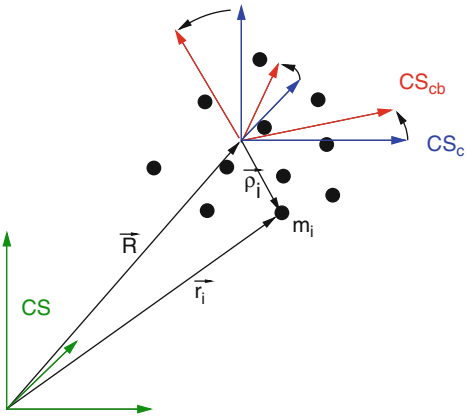


Fig. 12.1 Coordinate systems. Three coordinate systems will be used: The laboratory system CS, the center of mass system CS_c, and the body fixed system CS_{cb}

12.2 Properties of the Rotation Matrix

Rotation conserves the length of $\boldsymbol{\rho}$[1]:

$$\boldsymbol{\rho}^T \boldsymbol{\rho} = (A\boldsymbol{\rho})^T (A\boldsymbol{\rho}) = \boldsymbol{\rho}^T A^T A \boldsymbol{\rho}. \tag{12.5}$$

Consider the matrix

$$M = A^T A - 1 \tag{12.6}$$

for which

$$\boldsymbol{\rho}^T M \boldsymbol{\rho} = 0 \tag{12.7}$$

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

$$0 = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = M_{11}. \tag{12.8}$$

Similarly by choosing a unit vector in y- or z-direction we find $M_{22} = M_{33} = 0$.

Now choose $\boldsymbol{\rho} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$:

[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.

$$0 = \begin{pmatrix} 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} M_{11} + M_{12} \\ M_{21} + M_{22} \\ M_{31} + M_{32} \end{pmatrix} = M_{11} + M_{22} + M_{12} + M_{21}. \tag{12.9}$$

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 antisymmetric and has three independent components:

$$M = -M^T = \begin{pmatrix} 0 & M_{12} & M_{13} \\ -M_{12} & 0 & M_{23} \\ -M_{13} & -M_{23} & 0 \end{pmatrix}. \tag{12.10}$$

Inserting (12.6) we have

$$(A^T A - 1) = -(A^T A - 1)^T = -(A^T A - 1) \tag{12.11}$$

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

$$\mathbf{r_i} = \mathbf{R} + A\boldsymbol{\rho}_{\mathbf{ib}} \tag{12.12}$$

we calculate the velocity

$$\frac{\mathrm{d}\mathbf{r_i}}{\mathrm{d}t} = \frac{\mathrm{d}\mathbf{R}}{\mathrm{d}t} + \frac{\mathrm{d}A}{\mathrm{d}t}\boldsymbol{\rho}_{\mathbf{ib}} + A\frac{\mathrm{d}\boldsymbol{\rho}_{\mathbf{ib}}}{\mathrm{d}t} \tag{12.13}$$

but since $\boldsymbol{\rho}_{\mathbf{ib}}$ is constant by definition, the last summand vanishes

$$\dot{\mathbf{r}}_i = \dot{\mathbf{R}} + \dot{A}\boldsymbol{\rho}_{ib} = \dot{\mathbf{R}} + \dot{A}A^{-1}\boldsymbol{\rho}_i \tag{12.14}$$

and in the center of mass system we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\rho}_i = \dot{A}A^{-1}\boldsymbol{\rho}_i = W\boldsymbol{\rho}_i \tag{12.15}$$

with the matrix

$$W = \dot{A}A^{-1}. \tag{12.16}$$

12.3 Properties of W, Connection with the Vector of Angular Velocity

Since rotation does not change the length of $\boldsymbol{\rho}_i$, we have

$$0 = \frac{\mathrm{d}}{\mathrm{d}t}|\boldsymbol{\rho}_i|^2 \rightarrow 0 = \boldsymbol{\rho}_i \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\rho}_i = \boldsymbol{\rho}_i (W\boldsymbol{\rho}_i) \tag{12.17}$$

or in matrix notation

$$0 = \boldsymbol{\rho}_i^T W \boldsymbol{\rho}_i . \tag{12.18}$$

This holds for arbitrary $\boldsymbol{\rho}_i$. Hence W is antisymmetric and has three independent components

$$W = \begin{pmatrix} 0 & W_{12} & W_{13} \\ -W_{12} & 0 & W_{23} \\ -W_{13} & -W_{23} & 0 \end{pmatrix} . \tag{12.19}$$

Now consider infinitesimal rotation by the angle $\mathrm{d}\varphi$ (Fig. 12.2).
Then we have (the index i is suppressed)

$$\mathrm{d}\boldsymbol{\rho} = \frac{\mathrm{d}\boldsymbol{\rho}}{\mathrm{d}t}\mathrm{d}t = \begin{pmatrix} 0 & W_{12} & W_{13} \\ -W_{12} & 0 & W_{23} \\ -W_{13} & -W_{23} & 0 \end{pmatrix} \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{pmatrix} \mathrm{d}t = \begin{pmatrix} W_{12}\rho_2 + W_{13}\rho_3 \\ -W_{12}\rho_1 + W_{23}\rho_3 \\ -W_{13}\rho_1 - W_{23}\rho_2 \end{pmatrix} \mathrm{d}t \tag{12.20}$$

which can be written as a cross product:

$$\mathrm{d}\boldsymbol{\rho} = \mathrm{d}\boldsymbol{\varphi} \times \boldsymbol{\rho} \tag{12.21}$$

with

$$\mathrm{d}\boldsymbol{\varphi} = \begin{pmatrix} -W_{23}\mathrm{d}t \\ W_{13}\mathrm{d}t \\ -W_{12}\mathrm{d}t \end{pmatrix} . \tag{12.22}$$

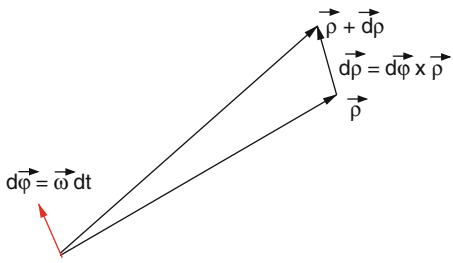


Fig. 12.2 Infinitesimal rotation

But this can be expressed in terms of the angular velocity $\boldsymbol{\omega}$ as

$$\mathrm{d}\boldsymbol{\varphi} = \boldsymbol{\omega}\mathrm{d}t \tag{12.23}$$

and finally we have

$$\mathrm{d}\boldsymbol{\varphi} = \boldsymbol{\omega}\mathrm{d}t = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \mathrm{d}t \quad W = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \tag{12.24}$$

and the more common form of the equation of motion

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\rho} = W\boldsymbol{\rho} = \boldsymbol{\omega} \times \boldsymbol{\rho}. \tag{12.25}$$

12.4 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

$$\boldsymbol{\rho_{1b}} = A^{-1}(\mathbf{r}_1 - \mathbf{R}) \tag{12.26}$$

and its velocity in the body fixed system is

$$\dot{\boldsymbol{\rho}}_{1b} = A^{-1}(\dot{\mathbf{r}}_1 - \dot{\mathbf{R}}) + \frac{\mathrm{d}A^{-1}}{\mathrm{d}t}(\mathbf{r}_1 - \mathbf{R}). \tag{12.27}$$

The time derivative of the inverse matrix follows from

$$0 = \frac{\mathrm{d}}{\mathrm{d}t}\left(A^{-1}A\right) = A^{-1}\dot{A} + \frac{\mathrm{d}A^{-1}}{\mathrm{d}t}A \tag{12.28}$$

$$\frac{\mathrm{d}A^{-1}}{\mathrm{d}t} = -A^{-1}\dot{A}A^{-1} = -A^{-1}W \tag{12.29}$$

and hence

$$\frac{\mathrm{d}A^{-1}}{\mathrm{d}t}\Big(\mathbf{r}_1 - \mathbf{R}\Big) = -A^{-1}W\Big(\mathbf{r}_1 - \mathbf{R}\Big). \tag{12.30}$$

Now we rewrite this using the angular velocity as observed in the body fixed system

$$-A^{-1}W\Big(\mathbf{r}_1 - \mathbf{R}\Big) = -W_bA^{-1}\Big(\mathbf{r_1} - \mathbf{R}\Big) = -W_b\boldsymbol{\rho}_{1b} = -\boldsymbol{\omega}_b \times \boldsymbol{\rho}_{1b}, \tag{12.31}$$

where W transforms as

$$W_b = A^{-1} W A. \tag{12.32}$$

W transforms like a second rank tensor. From that 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

$$\begin{aligned} W_b &= \begin{pmatrix} 0 & -\omega_{b3} & \omega_{b2} \\ \omega_{b3} & 0 & -\omega_{b1} \\ -\omega_{b2} & \omega_{b1} & 0 \end{pmatrix} = \\ &= \begin{pmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \begin{pmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} = \\ &= \begin{pmatrix} 0 & -\omega_3 & \cos\varphi\omega_2 - \sin\varphi\omega_1 \\ \omega_3 & 0 & -(\cos\varphi\omega_1 + \sin\varphi\omega_2) \\ -(\omega_2\cos\varphi - \sin\varphi\omega_1) & \cos\varphi\omega_1 + \sin\varphi\omega_2 & 0 \end{pmatrix} \end{aligned} \tag{12.33}$$

which shows that

$$\begin{pmatrix} \omega_{1b} \\ \omega_{2b} \\ \omega_{3b} \end{pmatrix} = \begin{pmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}, \tag{12.34}$$

i.e., $\boldsymbol{\omega}$ transforms like a vector under rotations. There is a subtle difference, however, considering general coordinate transformations involving reflections. For example, consider reflection at the xy-plane. Then transformation of W gives

$$\begin{aligned} W_b &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \\ &= \begin{pmatrix} 0 & -\omega_3 & -\omega_2 \\ \omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{pmatrix} \end{aligned} \tag{12.35}$$

$$\begin{pmatrix} \omega_{1b} \\ \omega_{2b} \\ \omega_{3b} \end{pmatrix} = - \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix}. \tag{12.36}$$

This shows that $\boldsymbol{\omega}$ is a so-called axial or pseudo-vector. Under a general coordinate transformation, it transforms as

$$\boldsymbol{\omega}_b = \det(A) A \boldsymbol{\omega}. \tag{12.37}$$

12.5 Momentum and Angular Momentum

The total momentum is

$$\mathbf{P} = \sum m_i \dot{\mathbf{r}}_i = \sum m_i \dot{\mathbf{R}} = M\dot{\mathbf{R}}, \tag{12.38}$$

since by definition we have $\sum m_i \boldsymbol{\rho}_i = 0$.
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:

$$\mathbf{L} = \sum m_i \mathbf{r}_i \times \dot{\mathbf{r}}_i = M\mathbf{R} \times \dot{\mathbf{R}} + \sum m_i \boldsymbol{\rho}_i \times \dot{\boldsymbol{\rho}}_i = \mathbf{L}_{\text{COM}} + \mathbf{L}_{\text{int}}. \tag{12.39}$$

The second contribution is

$$\mathbf{L}_{\text{int}} = \sum m_i \boldsymbol{\rho}_i \times (\boldsymbol{\omega} \times \boldsymbol{\rho}_i) = \sum m_i \left(\boldsymbol{\omega}\boldsymbol{\rho}_i^2 - \boldsymbol{\rho}_i (\boldsymbol{\rho}_i \boldsymbol{\omega}) \right). \tag{12.40}$$

This is a linear vector function of $\boldsymbol{\omega}$, which can be expressed simpler by introducing the tensor of inertia

$$I = \sum m_i \boldsymbol{\rho}_i^2 \, 1 - m_i \boldsymbol{\rho}_i \boldsymbol{\rho}_i^{\mathrm{T}} \tag{12.41}$$

or using components

$$I_{m,n} = \sum m_i \boldsymbol{\rho}_i^2 \delta_{m,n} - m_i \boldsymbol{\rho}_{i,m} \boldsymbol{\rho}_{i,n} \tag{12.42}$$

as

$$\mathbf{L}_{int} = I\boldsymbol{\omega}. \tag{12.43}$$

12.6 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

$$\frac{\mathrm{d}^2}{\mathrm{d}t^2} \sum m_i \mathbf{r}_i = M\ddot{\mathbf{R}} = \sum \mathbf{F}_i = \mathbf{F}_{\text{ext}}. \tag{12.44}$$

If there is no total external force $\mathbf{F}_{\text{ext}}$, the center of mass moves with constant velocity

$$\mathbf{R} = \mathbf{R}_0 + \mathbf{V}(t - t_0). \tag{12.45}$$

The time derivative of the angular momentum equals the total external torque

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{L} = \frac{\mathrm{d}}{\mathrm{d}t}\sum m_i \mathbf{r}_i \times \dot{\mathbf{r}}_i = \sum m_i \mathbf{r}_i \times \ddot{\mathbf{r}}_i = \sum \mathbf{r}_i \times \mathbf{F}_i = \sum \mathbf{N}_i = \mathbf{N}_{\text{ext}} \quad (12.46)$$

which can be decomposed into

$$\mathbf{N}_{\text{ext}} = \mathbf{R} \times \mathbf{F}_{\text{ext}} + \sum \boldsymbol{\rho}_i \times \mathbf{F}_i. \quad (12.47)$$

With the decomposition of the angular momentum

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{L} = \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{L}_{\text{COM}} + \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{L}_{\text{int}} \quad (12.48)$$

we have two separate equations for the two contributions:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{L}_{\text{COM}} = \frac{\mathrm{d}}{\mathrm{d}t} M\mathbf{R} \times \dot{\mathbf{R}} = M\mathbf{R} \times \ddot{\mathbf{R}} = \mathbf{R} \times \mathbf{F}_{\text{ext}} \quad (12.49)$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{L}_{\text{int}} = \sum \boldsymbol{\rho}_i \times \mathbf{F}_i = \mathbf{N}_{\text{ext}} - \mathbf{R} \times \mathbf{F}_{\text{ext}} = \mathbf{N}_{\text{int}}. \quad (12.50)$$

12.7 Moments of Inertia

The angular momentum (12.43) is

$$\mathbf{L}_{\text{Rot}} = I\boldsymbol{\omega} = AA^{-1}IAA^{-1}\boldsymbol{\omega} = AI_b\boldsymbol{\omega}_b, \quad (12.51)$$

where the tensor of inertia in the body fixed system is

$$\begin{aligned} I_b &= A^{-1}IA = A^{-1}\left(\sum m_i \boldsymbol{\rho}_i^{\mathrm{T}}\boldsymbol{\rho}_i - m_i \boldsymbol{\rho}_i \boldsymbol{\rho}_i^{\mathrm{T}}\right) A \\ &= \sum m_i A^{\mathrm{T}} \boldsymbol{\rho}_i^{\mathrm{T}} \boldsymbol{\rho}_i A - m_i A^{\mathrm{T}} \boldsymbol{\rho}_i \boldsymbol{\rho}_i^{\mathrm{T}} A \\ &= \sum m_i \boldsymbol{\rho}_{ib}^2 - m_i \boldsymbol{\rho}_{ib} \boldsymbol{\rho}_{ib}^{\mathrm{T}}. \end{aligned} \quad (12.52)$$

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

$$I_b = \begin{pmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{pmatrix} \quad (12.53)$$

with the principle moments of inertia $I_{1,2,3}$.

12.8 Equations of Motion for a Rotor

The following equations describe pure rotation of a rigid body:

$$\frac{\mathrm{d}}{\mathrm{d}t}A = WA = AW_b \tag{12.54}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{L}_{\mathrm{int}} = \mathbf{N}_{\mathrm{int}} \tag{12.55}$$

$$W = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \qquad W_{ij} = -\varepsilon_{ijk}\omega_k \tag{12.56}$$

$$\mathbf{L}_{\mathrm{int}} = A\mathbf{L}_{\mathrm{int},b} = I\boldsymbol{\omega} = AI_b\boldsymbol{\omega}_b \tag{12.57}$$

$$\boldsymbol{\omega}_b = I_b^{-1}\mathbf{L}_{\mathrm{int},b} = \begin{pmatrix} I_1^{-1} & 0 & 0 \\ 0 & I_2^{-1} & 0 \\ 0 & 0 & I_3^{-1} \end{pmatrix} \mathbf{L}_{\mathrm{int},b} \qquad \boldsymbol{\omega} = A\boldsymbol{\omega}_b \tag{12.58}$$

$$I_b = \text{const.} \tag{12.59}$$

12.9 Explicit Solutions

Equation (12.54) for the rotation matrix and (12.55) for the angular momentum have to be solved by a suitable algorithm. The simplest integrator is the explicit Euler method [51] (Fig. 12.3):

$$A(t + \mathrm{d}t) = A(t) + A(t)W_b(t)\mathrm{d}t + O(\mathrm{d}t^2) \tag{12.60}$$

$$\mathbf{L}_{\mathrm{int}}(t + \mathrm{d}t) = \mathbf{L}_{\mathrm{int}}(t) + \mathbf{N}_{\mathrm{int}}(t)\mathrm{d}t + O(\mathrm{d}t^2). \tag{12.61}$$

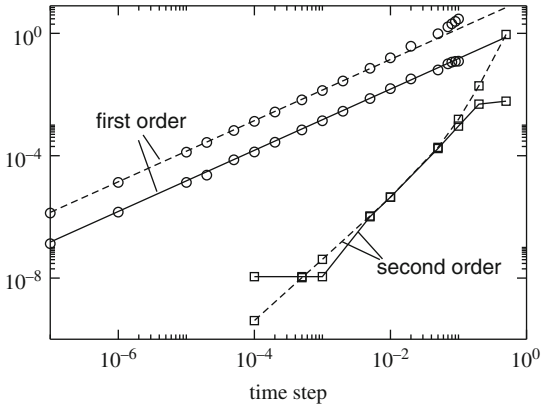


Fig. 12.3 Explicit methods for a free rotor. The equations of a free rotor (12.8) are solved using the explicit first- or second-order method. The deviations $|\det(A) - 1|$ (*dashed lines*) and $|E_{\mathrm{kin}} - E_{\mathrm{kin}}(0)|$ (*full lines*) at $t = 10$ are shown as a function of the time step Δt. The principal moments of inertia are 1, 2, 3 and the initial angular momentum is $\mathbf{L} = (1, 1, 1)$

Expanding the Taylor series of $A(t)$ to second order we have the second-order approximation

$$A(t+\mathrm{d}t) = A(t) + A(t)W_b(t)\mathrm{d}t + \frac{1}{2}\left(A(t)W_b^2(t) + A(t)\dot{W}_b(t)\right)\mathrm{d}t^2 + O(\mathrm{d}t^3). \tag{12.62}$$

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:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\omega}_b &= \frac{\mathrm{d}}{\mathrm{d}t}\left(I_b^{-1}A^{-1}\mathbf{L}_{\mathrm{int}}\right) = I_b^{-1}\left(\frac{\mathrm{d}}{\mathrm{d}t}A^{-1}\right)\mathbf{L}_{\mathrm{int}} + I_b^{-1}A^{-1}\mathbf{N}_{\mathrm{int}} = \\ &= I_b^{-1}\left(-A^{-1}W\right)\mathbf{L}_{\mathrm{int}} + I_b^{-1}A^{-1}\mathbf{N}_{\mathrm{int}} = -I_b^{-1}W_b\mathbf{L}_{\mathrm{int},b} + I_b^{-1}\mathbf{N}_{\mathrm{int},b}.\end{aligned} \tag{12.63}$$

Alternatively, in the laboratory system

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\omega} &= \frac{\mathrm{d}}{\mathrm{d}t}(A\boldsymbol{\omega}_b) = WA\boldsymbol{\omega}_b - AI_b^{-1}A^{-1}W\mathbf{L}_{\mathrm{int}} + AI_b^{-1}A^{-1}\mathbf{N}_{\mathrm{int}} \\ &= AI_b^{-1}A(\mathbf{N}_{\mathrm{int}} - W\mathbf{L}_{\mathrm{int}}),\end{aligned} \tag{12.64}$$

where the first summand vanishes due to

$$WA\boldsymbol{\omega}_b = AW_b\boldsymbol{\omega}_b = A\boldsymbol{\omega}_b \times \boldsymbol{\omega}_b = 0. \tag{12.65}$$

Substituting the angular momentum we have

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\omega}_b = I_b^{-1}\mathbf{N}_{\mathrm{int},b} - I_b^{-1}W_bI_b\boldsymbol{\omega}_b \tag{12.66}$$

which reads in components:

$$\begin{aligned}\begin{pmatrix}\dot{\omega}_{b1}\\ \dot{\omega}_{b2}\\ \dot{\omega}_{b3}\end{pmatrix} &= \begin{pmatrix}I_{b1}^{-1}N_{b1}\\ I_{b2}^{-1}N_{b2}\\ I_{b3}^{-1}N_{b3}\end{pmatrix} \\ &\quad - \begin{pmatrix}I_{b1}^{-1} & & \\ & I_{b2}^{-1} & \\ & & I_{b3}^{-1}\end{pmatrix}\begin{pmatrix}0 & -\omega_{b3} & \omega_{b2}\\ \omega_{b3} & 0 & -\omega_{b1}\\ -\omega_{b2} & \omega_{b1} & 0\end{pmatrix}\begin{pmatrix}I_{b1}\omega_{b1}\\ I_{b2}\omega_{b2}\\ I_{b3}\omega_{b3}\end{pmatrix}.\end{aligned} \tag{12.67}$$

Evaluation of the product gives a set of equations which are well known as Euler's equations:

$$\begin{aligned}\dot{\omega}_{b1} &= \tfrac{I_{b2}-I_{b3}}{I_{b1}}\omega_{b2}\omega_{b3} + \tfrac{N_{b1}}{I_{b1}}\\ \dot{\omega}_{b2} &= \tfrac{I_{b3}-I_{b1}}{I_{b2}}\omega_{b3}\omega_{b1} + \tfrac{N_{b2}}{I_{b2}}\\ \dot{\omega}_{b3} &= \tfrac{I_{b1}-I_{b2}}{I_{b3}}\omega_{b1}\omega_{b2} + \tfrac{N_{b3}}{I_{b3}}.\end{aligned} \tag{12.68}$$

12.10 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

$$\det(A+\mathrm{d}A) = \det(A+WA\mathrm{d}t) = \det A\,\det(1+W\mathrm{d}t) = \det A\,(1+\omega^2\mathrm{d}t^2). \tag{12.69}$$

The error is of order $\mathrm{d}t^2$, but the determinant will continuously increase, i.e., the rigid body will explode. For the second-order integrator we find

$$\begin{aligned}\det(A+\mathrm{d}A) &= \det\left(A + WA\mathrm{d}t + \frac{\mathrm{d}t^2}{2}(W^2A + \dot{W}A)\right)\\ &= \det A\,\det\left(1 + W\mathrm{d}t + \frac{\mathrm{d}t^2}{2}(W^2 + \dot{W})\right).\end{aligned} \tag{12.70}$$

This can be simplified to give

$$\det(A+\mathrm{d}A) = \det A\,(1 + \dot{\boldsymbol{\omega}}\boldsymbol{\omega}\mathrm{d}t^3 + \cdots). \tag{12.71}$$

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). Alternatively, the following method is very useful:

Consider correction of the rotation matrix by multiplication with a symmetric matrix S:

$$\widetilde{A} = AS, \tag{12.72}$$

where the resulting matrix is orthogonal

$$1 = \widetilde{A}^T\widetilde{A} = SA^TAS. \tag{12.73}$$

This equation can be formally solved:

$$S^{-2} = A^TA \tag{12.74}$$

$$S = (A^TA)^{-1/2}. \tag{12.75}$$

Since the deviation of A from orthogonality is small, we make the approximations

$$S = 1 + s \tag{12.76}$$

$$S^{-2} = 1 - 2s + \cdots = A^T A \tag{12.77}$$

$$S = 1 + \frac{1 - A^T A}{2} + \cdots \tag{12.78}$$

which can be easily evaluated.

12.11 Implicit Method

The quality of the method can be significantly improved by taking the time derivative at midstep (11.5):

$$A(t + \mathrm{d}t) = A(t) + A\left(t + \frac{\mathrm{d}t}{2}\right) W\left(t + \frac{\mathrm{d}t}{2}\right) \mathrm{d}t + \cdots \tag{12.79}$$

$$\mathbf{L}_{\mathrm{int}}(t + \mathrm{d}t) = \mathbf{L}_{\mathrm{int}}(t) + \mathbf{N}_{\mathrm{int}}\left(t + \frac{\mathrm{d}t}{2}\right) \mathrm{d}t + \cdots . \tag{12.80}$$

Taylor series expansion gives

$$A\left(t + \frac{\mathrm{d}t}{2}\right) W\left(t + \frac{\mathrm{d}t}{2}\right) \mathrm{d}t = A(t)W(t)\mathrm{d}t + \dot{A}(t)W(t)\frac{\mathrm{d}t^2}{2} + A(t)\dot{W}(t)\frac{\mathrm{d}t^2}{2} + O(\mathrm{d}t^3) \tag{12.81}$$

$$= A(t)W(t)\mathrm{d}t + (A(t)W^2(t) + A(t)\dot{W}(t))\frac{\mathrm{d}t^2}{2} + O(\mathrm{d}t^3) \tag{12.82}$$

which has the same error order as the explicit second-order method. The matrix $A\left(t + \frac{\mathrm{d}t}{2}\right)$ at midtime can be approximated by

$$\frac{1}{2}(A(t)+A(t+\mathrm{d}t)) = A\left(t + \frac{\mathrm{d}t}{2}\right)+\frac{\mathrm{d}t^2}{4}\ddot{A}\left(t + \frac{\mathrm{d}t}{2}\right)+\cdots = A\left(t + \frac{\mathrm{d}t}{2}\right)+O(\mathrm{d}t^2) \tag{12.83}$$

which does not change the error order of the implicit integrator which now becomes

$$A(t + \mathrm{d}t) = A(t) + \frac{1}{2}\left(A(t) + A(t + \mathrm{d}t)\right) W\left(t + \frac{\mathrm{d}t}{2}\right) \mathrm{d}t + O(\mathrm{d}t^3). \tag{12.84}$$

This equation can be formally solved by

$$A(t + \mathrm{d}t) = A(t)\left(1+\frac{\mathrm{d}t}{2}W\left(t+\frac{\mathrm{d}t}{2}\right)\right)\left(1 - \frac{\mathrm{d}t}{2}W\left(t + \frac{\mathrm{d}t}{2}\right)\right)^{-1} = A(t)T_b\left(\frac{\mathrm{d}t}{2}\right). \tag{12.85}$$

Alternatively, using angular velocities in the laboratory system we have the similar expression

$$A(t+\Delta t)=\left[1-\frac{\Delta t}{2}W\left(t+\frac{\Delta t}{2}\right)\right]^{-1}\left[1+\frac{\Delta t}{2}W\left(t+\frac{\Delta t}{2}\right)\right]A(t)=T\left(\frac{\Delta t}{2}\right)A(t). \tag{12.86}$$

The angular velocities at midtime can be calculated with sufficient accuracy from

$$W\left(t+\frac{\mathrm{d}t}{2}\right)=W(t)+\frac{\mathrm{d}t}{2}\dot{W}(t)+O(\mathrm{d}t^2). \tag{12.87}$$

With the help of an algebra program we easily prove that

$$\det\left(1+\frac{\Delta t}{2}W\right)=\det\left(1-\frac{\Delta t}{2}W\right)=1+\frac{\omega^2\Delta t^2}{4} \tag{12.88}$$

and therefore the determinant of the rotation matrix is conserved. The necessary matrix inversion can be easily done:

$$\begin{aligned}&\left[1-\frac{\Delta t}{2}W\right]^{-1}\\&=\begin{pmatrix}1+\frac{\omega_1^2\Delta t^2}{4} & -\omega_3\frac{\Delta t}{2}+\omega_1\omega_2\frac{\Delta t^2}{4} & \omega_2\frac{\Delta t}{2}+\omega_1\omega_3\frac{\Delta t^2}{4}\\ \omega_3\frac{\Delta t}{2}+\omega_1\omega_2\frac{\Delta t^2}{4} & 1+\frac{\omega_2^2\Delta t^2}{4} & -\omega_1\frac{\Delta t}{2}+\omega_2\omega_3\frac{\Delta t^2}{4}\\ -\omega_2\frac{\Delta t}{2}+\omega_1\omega_3\frac{\Delta t^2}{4} & \omega_1\frac{\Delta t}{2}+\omega_2\omega_3\frac{\Delta t^2}{4} & 1+\frac{\omega_3^2\Delta t^2}{4}\end{pmatrix}\frac{1}{1+\omega^2\frac{\Delta t^2}{4}}.\end{aligned} \tag{12.89}$$

The matrix product is explicitly

$$\begin{aligned}T_b&=\left[1+\frac{\Delta t}{2}W_b\right]\left[1-\frac{\Delta t}{2}W_b\right]^{-1}\\&=\begin{pmatrix}1+\frac{\omega_{b1}^2-\omega_{b2}^2-\omega_{b3}^2}{4}\Delta t^2 & -\omega_{b3}\Delta t+\omega_{b1}\omega_{b2}\frac{\Delta t^2}{2} & \omega_{b2}\Delta t+\omega_{b1}\omega_{b3}\frac{\Delta t^2}{2}\\ \omega_{b3}\Delta t+\omega_{b1}\omega_{b2}\frac{\Delta t^2}{2} & 1+\frac{-\omega_{b1}^2+\omega_{b2}^2-\omega_{b3}^2}{4}\Delta t^2 & -\omega_{b1}\Delta t+\omega_{b2}\omega_{b3}\frac{\Delta t^2}{2}\\ -\omega_{b2}\Delta t+\omega_{b1}\omega_{b3}\frac{\Delta t^2}{2} & \omega_{b1}\Delta t+\omega_{b2}\omega_{b3}\frac{\Delta t^2}{2} & 1+\frac{-\omega_{b1}^2-\omega_{b2}^2+\omega_{b3}^2}{4}\Delta t^2\end{pmatrix}\\&\quad\times\frac{1}{1+\omega_b^2\frac{\Delta t^2}{4}}.\end{aligned} \tag{12.90}$$

With the help of an algebra program it can be proved that this matrix is even orthogonal

$$T_b^T T_b=1 \tag{12.91}$$

and hence the orthonormality of A is conserved. The approximation for the angular momentum

$$\begin{aligned}&\mathbf{L}_{\text{int}}(t) + \mathbf{N}_{\text{int}}\left(t + \frac{\mathrm{d}t}{2}\right)\mathrm{d}t \\ &= \mathbf{L}_{\text{int}}(t) + \mathbf{N}_{\text{int}}(t)\mathrm{d}t + \dot{\mathbf{N}}_{\text{int}}(t)\frac{\mathrm{d}t^2}{2} + \cdots = \mathbf{L}_{\text{int}}(t + \mathrm{d}t) + O(\mathrm{d}t^3)\end{aligned} \tag{12.92}$$

can be used in an implicit way

$$\mathbf{L}_{\text{int}}(t + \mathrm{d}t) = \mathbf{L}_{\text{int}}(t) + \frac{\mathbf{N}_{\text{int}}(t + \mathrm{d}t) + \mathbf{N}_{\text{int}}(t)}{2}\mathrm{d}t + O(\mathrm{d}t^3). \tag{12.93}$$

Alternatively Euler's equations can be used in the form [52, 53]

$$\omega_{b1}\left(t + \frac{\Delta t}{2}\right) = \omega_{b1}\left(t - \frac{\Delta t}{2}\right) + \frac{I_{b2} - I_{b3}}{I_{b1}}\omega_{b2}(t)\omega_{b3}(t)\Delta t + \frac{N_{b1}}{I_{b1}}\Delta t, \text{ etc.}, \tag{12.94}$$

where the product $\omega_{b2}(t)\omega_{b3}(t)$ is approximated by

$$\omega_{b2}(t)\omega_{b3}(t) = \frac{1}{2}\left[\omega_{b2}\left(t - \frac{\Delta t}{2}\right)\omega_{b3}\left(t - \frac{\Delta t}{2}\right) + \omega_{b2}\left(t + \frac{\Delta t}{2}\right)\omega_{b3}\left(t + \frac{\Delta t}{2}\right)\right]. \tag{12.95}$$

$\omega_{b1}\left(t + \frac{\Delta t}{2}\right)$ is determined by iterative solution of the last two equations. Starting with $\omega_{b1}\left(t - \frac{\Delta t}{2}\right)$ convergence is achieved after few iterations.

12.12 Example: Free Symmetric Rotor

For the special case of a free symmetric rotor ($I_{b2} = I_{b3}$, $\mathbf{N}_{\text{int}} = 0$) Euler's equations simplify to (Fig. 12.4)

$$\dot{\omega}_{b1} = 0 \tag{12.96}$$

$$\dot{\omega}_{b2} = \frac{I_{b2(3)} - I_{b1}}{I_{b2(3)}}\omega_{b1}\omega_{b3} = \lambda\omega_{b3} \tag{12.97}$$

$$\dot{\omega}_{b3} = \frac{I_{b1} - I_{b2(3)}}{I_{b2(3)}}\omega_{b1}\omega_{b2} = -\lambda\omega_{b2} \tag{12.98}$$

$$\lambda = \frac{I_{b2(3)} - I_{b1}}{I_{b2(3)}}\omega_{b1}. \tag{12.99}$$

Coupled equations of this type appear often in physics. The solution can be found using a complex quantity

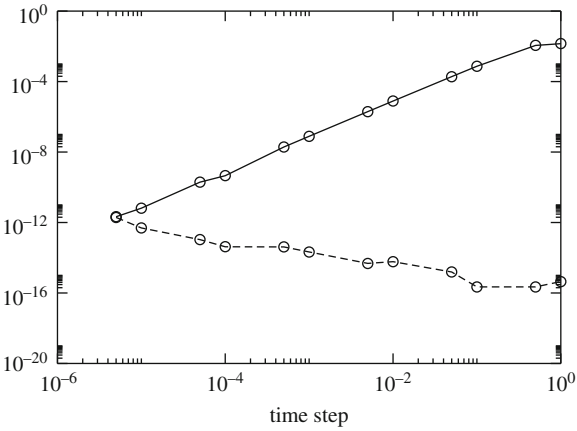


Fig. 12.4 Free rotation with the implicit method. The equations of a free rotor (12.8) are solved using the implicit method. The deviations $|\det(A) - 1|$ (*dashed line*) and $|E_{\text{kin}} - E_{\text{kin}}(0)|$ (*full line*) at $t = 10$ are shown as a function of the time step Δt. Initial conditions as in Fig. 12.3

$$\Omega = \omega_{b2} + \mathrm{i}\omega_{b3} \tag{12.100}$$

which obeys the simple differential equation

$$\dot{\Omega} = \dot{\omega}_{b2} + \mathrm{i}\dot{\omega}_{b3} = -\mathrm{i}(\mathrm{i}\lambda\omega_{b3} + \lambda\omega_{b2}) = -\mathrm{i}\lambda\Omega \tag{12.101}$$

with the solution

$$\Omega = \Omega_0 \mathrm{e}^{-\mathrm{i}\lambda t}. \tag{12.102}$$

Finally

$$\boldsymbol{\omega}_b = \begin{pmatrix} \omega_{b1}(0) \\ \Re(\Omega_0 e^{-\mathrm{i}\lambda t}) \\ \Im(\Omega_0 e^{-\mathrm{i}\lambda t}) \end{pmatrix} = \begin{pmatrix} \omega_{b1}(0) \\ \omega_{b2}(0)\cos(\lambda t) + \omega_{b3}(0)\sin(\lambda t) \\ \omega_{b3}(0)\cos(\lambda t) - \omega_{b2}(0)\sin(\lambda t) \end{pmatrix}, \tag{12.103}$$

i.e., $\boldsymbol{\omega}_b$ rotates around the 1-axis with frequency λ.

12.13 Kinetic Energy of a Rotor

The kinetic energy of the rotor is

$$\begin{aligned} E_{\text{kin}} &= \sum \frac{m_i}{2}\dot{r}_i^2 = \sum \frac{m_i}{2}(\dot{\mathbf{R}} + \dot{A}\boldsymbol{\rho}_{ib})^2 \\ &= \sum \frac{m_i}{2}(\dot{\mathbf{R}}^{\mathrm{T}} + \boldsymbol{\rho}_{ib}^{\mathrm{T}}\dot{A}^{\mathrm{T}})(\dot{\mathbf{R}} + \dot{A}\boldsymbol{\rho}_{ib}) = \frac{M}{2}\dot{\mathbf{R}}^2 + \sum \frac{m_i}{2}\boldsymbol{\rho}_{ib}^{\mathrm{T}}\dot{A}^{\mathrm{T}}\dot{A}\boldsymbol{\rho}_{ib}. \end{aligned} \tag{12.104}$$

The second part is the contribution of the rotational motion. It can be written as

$$E_{\text{rot}} = \sum \frac{m_i}{2} \boldsymbol{\rho}_{ib}^{\mathrm{T}} W_b^{\mathrm{T}} A^{\mathrm{T}} A W_b \boldsymbol{\rho}_{ib} = -\sum \frac{m_i}{2} \boldsymbol{\rho}_{ib}^{\mathrm{T}} W_b^2 \boldsymbol{\rho}_{ib} = \frac{1}{2} \boldsymbol{\omega}_b^{\mathrm{T}} I_b \boldsymbol{\omega}_b \quad (12.105)$$

since[2]

$$-W_b^2 = \begin{pmatrix} \omega_{b3}^2 + \omega_{b2}^2 & -\omega_{b1}\omega_{b2} & -\omega_{b1}\omega_{b3} \\ -\omega_{b1}\omega_{b2} & \omega_{b1}^2 + \omega_{b3}^2 & -\omega_{b2}\omega_{b3} \\ -\omega_{b1}\omega_{b3} & -\omega_{b2}\omega_{b3} & \omega_{b1}^2 + \omega_{b2}^2 \end{pmatrix} = \omega_b^2 - \boldsymbol{\omega}_b \omega_b^{\mathrm{T}}. \quad (12.106)$$

12.14 Parametrization by Euler Angles

So far we had to solve equations for all nine 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 nine 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 [54]

$$\begin{pmatrix} \cos\psi\cos\phi - \cos\theta\sin\phi\sin\psi & -\sin\psi\cos\phi - \cos\theta\sin\phi\cos\psi & \sin\theta\sin\phi \\ \cos\psi\sin\phi + \cos\theta\cos\phi\sin\psi & \sin\psi\sin\phi + \cos\theta\cos\phi\cos\psi & -\sin\theta\cos\phi \\ \sin\theta\sin\psi & \sin\theta\cos\psi & \cos\theta \end{pmatrix} \quad (12.107)$$

obeying the equations

$$\dot{\phi} = \omega_x \frac{\sin\phi\cos\theta}{\sin\theta} + \omega_y \frac{\cos\phi\cos\theta}{\sin\theta} + \omega_z \quad (12.108)$$

$$\dot{\theta} = \omega_x \cos\phi + \omega_y \sin\phi \quad (12.109)$$

$$\dot{\psi} = \omega_x \frac{\sin\phi}{\sin\theta} - \omega_y \frac{\cos\phi}{\sin\theta}. \quad (12.110)$$

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 somewhere in a denominator which causes numerical instabilities at the poles. One possible solution to this problem is to switch between two different coordinate systems.

12.15 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

[2] $\boldsymbol{\omega}_b \omega_b^{\mathrm{T}}$ denotes the outer or matrix product.

$$1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{12.111}$$

Any complex 2×2 matrix can be written as a linear combination

$$c_0 1 + \mathbf{c}\boldsymbol{\sigma}. \tag{12.112}$$

Accordingly any vector $\mathbf{x} \in R^3$ can be mapped onto a complex 2×2 matrix:

$$\mathbf{x} \;\rightarrow\; P = \begin{pmatrix} z & x - \mathrm{i}y \\ x + \mathrm{i}y & -z \end{pmatrix}. \tag{12.113}$$

Rotation of the coordinate system leads to the transformation

$$P' = QPQ^+, \tag{12.114}$$

where

$$Q = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \tag{12.115}$$

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., $Q^+ = Q^{-1}$ and its determinant must be 1. Inversion of this matrix is easily done:

$$Q^+ = \begin{pmatrix} \alpha^* & \gamma^* \\ \beta^* & \delta^* \end{pmatrix} \quad Q^{-1} = \frac{1}{\alpha\delta - \beta\gamma} \begin{pmatrix} \delta & -\beta \\ -\gamma & \alpha \end{pmatrix}. \tag{12.116}$$

Hence Q takes the form

$$Q = \begin{pmatrix} \alpha & \beta \\ -\beta^* & \alpha^* \end{pmatrix} \quad \text{with} \quad |\alpha|^2 + |\beta|^2 = 1. \tag{12.117}$$

Setting $x_\pm = x \pm iy$, the transformed matrix has the same form as P:

$$\begin{aligned} &QPQ^+ \\ &= \begin{pmatrix} \alpha^*\beta x_+ + \beta^*\alpha x_- + (|\alpha|^2 - |\beta|^2)z & -\beta^2 x_+ + \alpha^2 x_- - 2\alpha\beta z \\ \alpha^{*2} x_+ - \beta^{*2} x_- - 2\alpha^*\beta^* z & -\alpha^*\beta x_+ - \alpha\beta^* x_- - (|\alpha|^2 - |\beta|^2)z \end{pmatrix} \\ &= \begin{pmatrix} z' & x'_- \\ x'_+ & -z' \end{pmatrix}. \end{aligned} \tag{12.118}$$

From comparison we find the transformed vector components:

$$x' = \frac{1}{2}(x'_+ + x'_-) = \frac{1}{2}(\alpha^{*2} - \beta^2)x_+ + \frac{1}{2}(\alpha^2 - \beta^{*2})x_- - (\alpha\beta + \alpha^*\beta^*)z$$
$$= \frac{\alpha^{*2} + \alpha^2 - \beta^{*2} - \beta^2}{2}x + \frac{i(\alpha^{*2} - \alpha^2 + \beta^{*2} - \beta^2)}{2}y - (\alpha\beta + \alpha^*\beta^*)z \tag{12.119}$$

$$y' = \frac{1}{2i}(x'_+ - x'_-) = \frac{1}{2i}(\alpha^{*2} + \beta^2)x_+ + \frac{1}{2i}(-\beta^{*2} - \alpha^2)x_- + \frac{1}{i}(-\alpha^*\beta^* + \alpha\beta)z$$
$$= \frac{\alpha^{*2} - \alpha^2 - \beta^{*2} + \beta^2}{2i}x + \frac{\alpha^{*2} + \alpha^2 + \beta^{*2} + \beta^2}{2}y + i(\alpha^*\beta^* - \alpha\beta)z \tag{12.120}$$
$$z' = (\alpha^*\beta + \alpha\beta^*)x + i(\alpha^*\beta - \alpha\beta^*)y + (|\alpha|^2 - |\beta|^2)z.$$

This gives us the rotation matrix in terms of the Cayley–Klein parameters α and β:

$$A = \begin{pmatrix} \frac{\alpha^{*2}+\alpha^2-\beta^{*2}-\beta^2}{2} & \frac{i(\alpha^{*2}-\alpha^2+\beta^{*2}-\beta^2)}{2} & -(\alpha\beta + \alpha^*\beta^*) \\ \frac{\alpha^{*2}-\alpha^2-\beta^{*2}+\beta^2}{2i} & \frac{\alpha^{*2}+\alpha^2+\beta^{*2}+\beta^2}{2} & \frac{1}{i}(-\alpha^*\beta^* + \alpha\beta) \\ (\alpha^*\beta + \alpha\beta^*) & i(\alpha^*\beta - \alpha\beta^*) & (|\alpha|^2 - |\beta|^2) \end{pmatrix}. \tag{12.121}$$

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

$$\alpha = q_0 + \mathrm{i}q_3 \qquad \beta = q_2 + \mathrm{i}q_1. \tag{12.122}$$

Now the matrix Q

$$Q = \begin{pmatrix} q_0 + \mathrm{i}q_3 & q_2 + \mathrm{i}q_1 \\ -q_2 + \mathrm{i}q_1 & q_0 - \mathrm{i}q_3 \end{pmatrix} = q_0 1 + \mathrm{i}q_1\sigma_x + \mathrm{i}q_2\sigma_y + \mathrm{i}q_3\sigma_z \tag{12.123}$$

becomes a so-called quaternion which is a linear combination of the four matrices

$$U = 1 \qquad I = i\sigma_z \qquad J = \mathrm{i}\sigma_y \qquad K = \mathrm{i}\sigma_x \tag{12.124}$$

which obey the following multiplication rules:

$$\begin{aligned} I^2 = J^2 = K^2 &= -U \\ IJ &= -JI = K \\ JK &= -KJ = I \\ KI &= -IK = J. \end{aligned} \tag{12.125}$$

In terms of Euler parameters the rotation matrix reads

$$A = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \tag{12.126}$$

and from the equation $\dot{A} = WA$ we derive the equation of motion for the quaternion

$$\begin{pmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & \omega_1 & -\omega_2 & \omega_3 \\ -\omega_1 & 0 & -\omega_3 & -\omega_2 \\ \omega_2 & \omega_3 & 0 & -\omega_1 \\ -\omega_3 & \omega_2 & \omega_1 & 0 \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} \tag{12.127}$$

or from $\dot{A} = AW_b$ the alternative equation

$$\begin{pmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 & \omega_{1b} & \omega_{2b} & \omega_{3b} \\ -\omega_{1b} & 0 & \omega_{3b} & -\omega_{2b} \\ -\omega_{2b} & -\omega_{3b} & 0 & \omega_{1b} \\ -\omega_{3b} & \omega_{2b} & -\omega_{1b} & 0 \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix}. \tag{12.128}$$

Both of these equations can be written briefly in the form

$$\dot{\mathbf{q}} = \widetilde{W}\mathbf{q}. \tag{12.129}$$

Example: Rotation Around the z-Axis

Rotation around the z-axis corresponds to the quaternion with Euler parameters

$$\mathbf{q} = \begin{pmatrix} \cos\frac{\omega t}{2} \\ 0 \\ 0 \\ -\sin\frac{\omega t}{2} \end{pmatrix} \tag{12.130}$$

as can be seen from the rotation matrix

$$\begin{aligned} A &= \begin{pmatrix} \left(\cos\frac{\omega t}{2}\right)^2 - \left(\sin\frac{\omega t}{2}\right)^2 & -2\cos\frac{\omega t}{2}\sin\frac{\omega t}{2} & 0 \\ 2\cos\frac{\omega t}{2}\sin\frac{\omega t}{2} & \left(\cos\frac{\omega t}{2}\right)^2 - \left(\sin\frac{\omega t}{2}\right)^2 & 0 \\ 0 & 0 & \left(\cos\frac{\omega t}{2}\right)^2 + \left(\sin\frac{\omega t}{2}\right)^2 \end{pmatrix} \\ &= \begin{pmatrix} \cos\omega t & -\sin\omega t & 0 \\ \sin\omega t & \cos\omega t & 0 \\ 0 & 0 & 1 \end{pmatrix}. \end{aligned} \tag{12.131}$$

The time derivative of $\mathbf{q}$ obeys the equation

$$\dot{\mathbf{q}} = \frac{1}{2} \begin{pmatrix} 0 & 0 & 0 & \omega \\ 0 & 0 & -\omega & 0 \\ 0 & \omega & 0 & 0 \\ -\omega & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \cos\frac{\omega t}{2} \\ 0 \\ 0 \\ -\sin\frac{\omega t}{2} \end{pmatrix} = \begin{pmatrix} -\frac{\omega}{2}\sin\omega t \\ 0 \\ 0 \\ -\frac{\omega}{2}\cos\omega t \end{pmatrix}. \tag{12.132}$$

After a rotation by 2π the quaternion changes its sign, i.e., $\mathbf{q}$ and $-\mathbf{q}$ parametrize the same rotation matrix!

12.16 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

$$\mathbf{q}(t+\Delta t) = \mathbf{q}(t) + \widetilde{W}(t)\mathbf{q}(t)\Delta t + (\dot{\widetilde{W}}(t) + \widetilde{W}^2(t))\mathbf{q}(t)\frac{\Delta t^2}{2} + \cdots . \quad (12.133)$$

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 to much from unity.
It is also possible to use Omelyans [55] method:

$$\mathbf{q}(t+\Delta t) = \mathbf{q}(t) + \widetilde{W}\left(t+\frac{\Delta t}{2}\right)\frac{1}{2}(\mathbf{q}(t)+\mathbf{q}(t+\Delta t)) \quad (12.134)$$

gives

$$\mathbf{q}(t+\Delta t) = \left(1-\frac{\Delta t}{2}\widetilde{W}\right)^{-1}\left(1+\frac{\Delta t}{2}\widetilde{W}\right)\mathbf{q}(t), \quad (12.135)$$

where the inverse matrix is

$$\left(1-\frac{\Delta t}{2}\widetilde{W}\right)^{-1} = \frac{1}{1+\omega^2\frac{\Delta t^2}{16}}\left(1+\frac{\Delta t}{2}\widetilde{W}\right) \quad (12.136)$$

and the matrix product

$$\left(1-\frac{\Delta t}{2}\widetilde{W}\right)^{-1}\left(1+\frac{\Delta t}{2}\widetilde{W}\right) = \frac{1-\omega^2\frac{\Delta t^2}{16}}{1+\omega^2\frac{\Delta t^2}{16}} + \frac{\Delta t}{1+\omega^2\frac{\Delta t^2}{16}}\widetilde{W}. \quad (12.137)$$

This method conserves the norm of the quaternion and works quite well.

Problems

Problem 12.1 Rotor in a Field

In this computer experiment we simulate a molecule with a permanent dipole moment in a homogeneous electric field **E** (Fig. 12.5). 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 $\sum Q_i = 0$. The dipole moment is

$$\mathbf{p} = \sum Q_i \mathbf{r}_i \quad (12.138)$$

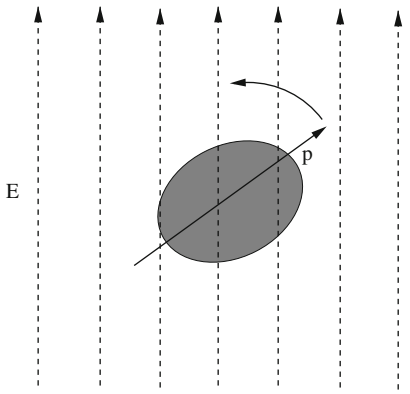


Fig. 12.5 Rotor in an electric field

The external force and torque are

$$\mathbf{F}_{\text{ext}} = \sum Q_i \mathbf{E} = 0 \tag{12.139}$$

$$\mathbf{N}_{\text{ext}} = \sum Q_i \mathbf{r}_i \times \mathbf{E} = \mathbf{p} \times \mathbf{E} \tag{12.140}$$

The angular momentum changes according to

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{L}_{\text{int}} = \mathbf{p} \times \mathbf{E} \tag{12.141}$$

where the dipole moment is constant in the body fixed system. We use the implicit integrator for the rotation matrix (12.85) and the equation

$$\dot{\boldsymbol{\omega}}_b(t) = -I_b^{-1} W_b(t) \mathbf{L}_{\text{int},b}(t) + I_b^{-1} A^{-1}(t)(\mathbf{p}(t) \times \mathbf{E}) \tag{12.142}$$

to solve the equations of motion numerically.
Obviously the component of the angular momentum parallel to the field is constant. The potential energy is

$$E_{\text{pot}} = \sum Q_i \mathbf{E} \mathbf{r}_i = \mathbf{p}\mathbf{E} \tag{12.143}$$

Problem 12.2 Molecular Collision

This computer experiment simulates the collision of two rigid methane molecules. The equations of motion are solved with the implicit quaternion method (12.134) and the velocity Verlet method (Sect. 11.11.4). The two molecules interact by a standard 6–12 Lennard Jones potential (13.3) [50]. 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

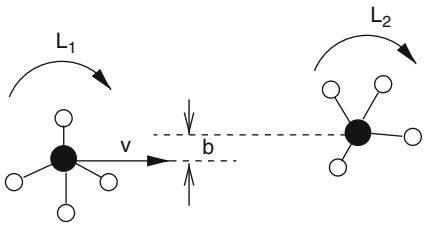


Fig. 12.6 Molecular collision

monitored and the decomposition of the total energy into translational, rotational, and potential energy is plotted as a function of time (Fig. 12.6).

Study the exchange of momentum and angular momentum and the transfer of energy between translational and rotational degrees of freedom.

Chapter 13
Simulation of 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 and the thermodynamic average is taken along one or more trajectories. In the second case we apply a Monte Carlo method to calculate the average over a set of random configurations.

13.1 Force Fields for Molecular Dynamics Simulations

Classical molecular dynamics calculations have become a very valuable tool for the investigation of molecular systems [56–60]. They consider a model system of mass points $m_i \quad i = 1 \cdots N$ with an interaction described by a suitable potential function (force field)

$$U(\mathbf{r}_1 \cdots \mathbf{r}_N) \tag{13.1}$$

and solve the classical equations of motion

$$\frac{\mathrm{d}^2\mathbf{x}_i}{\mathrm{d}t^2} = m_i\mathbf{F}_i = -m_i\frac{\partial U}{\partial \mathbf{x}_i} \tag{13.2}$$

numerically.

There exist a large number of different force fields in the literature. We discuss only the basic ingredients which are common to most of them.

13.1.1 Intramolecular Forces

Intramolecular degrees of freedom are often described by a simplified force field using internal coordinates which are composed of several terms including

- bond lengths $U^{\text{bond}} = \frac{k}{2}(r_{ij} - r_{ij}^0)^2$
- bond angles $U^{\text{angle}} = \frac{k}{2}(\phi - \phi^0)^2$

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_13,

- torsion angles $U^{\text{tors}} = -\frac{k}{2}\cos(m(\phi - \phi^0))$

13.1.2 Intermolecular Forces

Repulsion at short distances due to the Pauli principle and the weak attractive van der Waals forces are often modeled by a sum of pairwise Lennard-Jones potentials [50] (Fig. 13.1)

$$U^{\text{vdw}} = \sum_{A \neq B} \sum_{i \in A, j \in B} U_{i,j}^{\text{vdw}} = 4\varepsilon \sum \left(\frac{\sigma^{12}}{r_{ij}^{12}} - \frac{\sigma^6}{r_{ij}^6} \right). \tag{13.3}$$

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 Coulombic energy

$$U^{\text{Coul}} = \sum_{A \neq B} \sum_{i \in A, j \in B} \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}}. \tag{13.4}$$

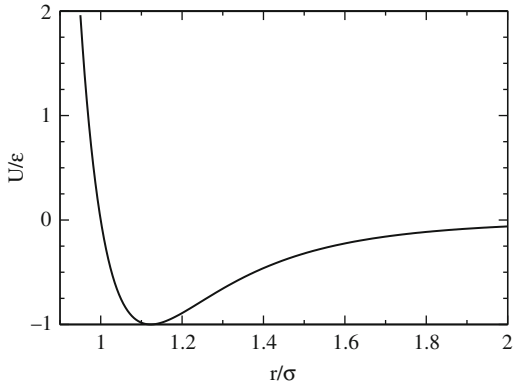


Fig. 13.1 Lennard-Jones potential. The 6–12 potential (13.3) has its minimum at $r_{\min} = \sqrt[6]{2}\sigma \approx 1.12\sigma$ with $U_{\min} = -\epsilon$

13.1.3 Approximate Separation of Rotation and Vibrations

If the amplitudes of internal vibrations are small the coordinates in the body fixed system can be divided into the equilibrium value and a small deviation

$$\boldsymbol{\rho}_{ib} = \boldsymbol{\rho}_{ib}^{(0)} + \xi_i. \tag{13.5}$$

The equation of motion is

$$m_i \ddot{\mathbf{r}}_i = m_i(\ddot{\mathbf{R}} + \ddot{\boldsymbol{\rho}}_i) = \mathbf{F}_i = \mathbf{F}_i^{\text{ bond}} + \mathbf{F}_i^{\text{ inter}}. \tag{13.6}$$

For the center of mass of molecule A we have

$$M_A \ddot{\mathbf{R}}_A = \sum_{i \in A} \mathbf{F}_i^{\text{ inter}} \tag{13.7}$$

and for the relative coordinates

$$m_i(\ddot{A}\boldsymbol{\rho}_{ib}^{(0)} + 2\dot{A}\dot{\xi}_i + A\ddot{\xi}_i) = \mathbf{F}_i^{\text{ bond}} + \mathbf{F}_i^{\text{ inter}} - \frac{m_i}{M}\sum \mathbf{F}_i^{\text{ inter}}. \tag{13.8}$$

If we neglect the effects of centrifugal and Coriolis forces and assume that the bonding interactions are much stronger than the intermolecular forces we have approximately

$$m_i \ddot{\xi}_i = A^{-1}\mathbf{F}_i^{\text{ bond}}, \tag{13.9}$$

where the bonding forces are derived from a force field for the intramolecular vibrations. For the rotational motion we have

$$m_i \mathbf{r}_i \times \ddot{\mathbf{r}}_i = M\mathbf{R} \times \ddot{\mathbf{R}} + m_i \boldsymbol{\rho}_i \times \ddot{\boldsymbol{\rho}}_i. \tag{13.10}$$

If we neglect the oscillations around the equilibrium positions (which are zero on the average) the rotational motion is approximated by a rigid rotor.

13.2 Simulation of a van der Waals System

In the following we describe a simple computer model of interacting particles without internal degrees of freedom (see Problems). The force on atom i is given by the gradient of the pairwise Lennard-Jones potential (13.3)

$$\mathbf{F}_i = \sum_{j \neq i} \mathbf{F}_{ij} = -4\varepsilon \sum_{j \neq i} \nabla_i \left(\frac{\sigma^{12}}{r_{ij}^{12}} - \frac{\sigma^6}{r_{ij}^6} \right) = 4\varepsilon \sum_{j \neq i} \left(\frac{12\sigma^{12}}{r_{ij}^{14}} - \frac{6\sigma^6}{r_{ij}^8} \right) (\mathbf{r}_i - \mathbf{r}_j). \tag{13.11}$$

13.2.1 Integration of the Equations of Motion

The equations of motion are integrated using the leapfrog algorithm (Sect. 11.11.7):

$$\mathbf{v}_i\left(t + \frac{\mathrm{d}t}{2}\right) = \mathbf{v}_i\left(t - \frac{\mathrm{d}t}{2}\right) + \frac{\mathbf{F}_i(t)}{m}\mathrm{d}t + O(\mathrm{d}t^3) \tag{13.12}$$

$$\mathbf{r}_i(t + \mathrm{d}t) = \mathbf{r}_i(t) + \mathbf{v}_i\left(t + \frac{\mathrm{d}t}{2}\right)\mathrm{d}t + O(\mathrm{d}t^3) \tag{13.13}$$

$$\mathbf{v}_i(t) = \frac{\mathbf{v}_i(t + \frac{\mathrm{d}t}{2}) + \mathbf{v}_i(t - \frac{\mathrm{d}t}{2})}{2} + O(\mathrm{d}t^2) \tag{13.14}$$

or with the Verlet algorithm (Sect. 11.11.5):

$$\mathbf{r}_i(t+\mathrm{d}t) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-\mathrm{d}t) + \frac{\mathbf{F}_i(t)}{m}\mathrm{d}t^2 + O(\mathrm{d}t^4) \tag{13.15}$$

$$\mathbf{v}_i(t+\mathrm{d}t) = \frac{\mathbf{r}_i(t+\mathrm{d}t) - \mathbf{r}_i(t)}{\mathrm{d}t} + O(\mathrm{d}t^2) \tag{13.16}$$

$$= \frac{\mathbf{r}_i(t) - \mathbf{r}_i(t-\mathrm{d}t)}{\mathrm{d}t} + \frac{\mathbf{F}_i(t)}{2m}\mathrm{d}t + O(\mathrm{d}t^2). \tag{13.17}$$

13.2.2 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 (Fig. 13.2).

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. 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

$$F_\perp = \left\langle \frac{\sum_{\text{refl.}} 2mv_\perp}{\mathrm{d}t} \right\rangle. \tag{13.18}$$

The pressure p is given by

$$p = \frac{1}{6L^2}\left\langle \frac{\sum_{\text{walls}}\sum_{\text{refl.}} 2mv_\perp}{\mathrm{d}t} \right\rangle. \tag{13.19}$$

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}.

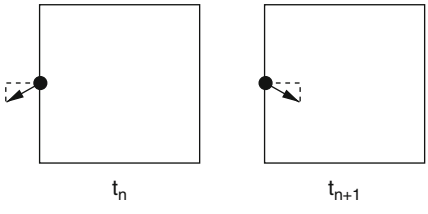


Fig. 13.2 Reflecting walls

13.2.3 Initial Conditions and Average Temperature

At the very beginning the N atoms are distributed over equally spaced lattice points within the cube. The velocities are randomly distributed following the Gaussian distribution

$$\sqrt{\frac{m}{2\pi kT}} \exp\left(-\frac{mv^2}{2kT}\right). \tag{13.20}$$

The effective temperature is calculated from the kinetic energy (assuming thermal equilibrium)

$$kT = \frac{2}{3N} E_{\text{kin}}. \tag{13.21}$$

The desired temperature is established by the rescaling procedure

$$\mathbf{v}_i \to \mathbf{v}_i \sqrt{\frac{kT_{\text{o}}}{kT_{\text{actual}}}} \tag{13.22}$$

which is applied several times during an equilibration run. A smoother method is the thermostat algorithm

$$\mathbf{v}_i \to \mathbf{v}_i \left(1 + \frac{1}{\tau_{\text{therm}}} \frac{kT_o - kT_{\text{actual}}}{kT_{\text{actual}}}\right), \tag{13.23}$$

where τ_{therm} is a suitable relaxation time (for instance, 20 time steps)

13.2.4 Analysis of the Results

13.2.4.1 Deviation from the Ideal Gas Behavior

A dilute gas is approximately ideal with

$$pV = NkT. \tag{13.24}$$

For a real gas the interaction between the particles has to be taken into account. From the equipartition theorem it can be found that[1]

$$pV = NkT + W \tag{13.25}$$

with the inner virial

[1] MD simulations with periodic boundary conditions use this equation to calculate the pressure.

$$W = \left\langle \frac{1}{3} \sum_i \mathbf{r}_i \mathbf{F}_i \right\rangle \tag{13.26}$$

which can be expanded as a power series of the density [61] to give

$$pV = NkT(1 + b(T)n + c(T)n^2 + \cdots). \tag{13.27}$$

The virial coefficient $b(T)$ can be calculated exactly for the Lennard-Jones gas [61]:

$$b(T) = \frac{2\pi}{3}\sigma^3 \sum_{j=0}^{\infty} \frac{2^{j-3/2}}{j!} \Gamma\left(\frac{2j-1}{4}\right) \left(\frac{\varepsilon}{kT}\right)^{(j/2+1/4)} \tag{13.28}$$

13.2.4.2 Degree of Order

The degree of order of the system can be analyzed in terms of the pair distance distribution function (Fig. 13.3)

$$g(R)\mathrm{d}R = P(R < r_{ij} < R + \mathrm{d}R). \tag{13.29}$$

In the condensed phase $g(R)$ shows maxima corresponding to the nearest neighbor distance, etc. In the gas phase this structure vanishes.

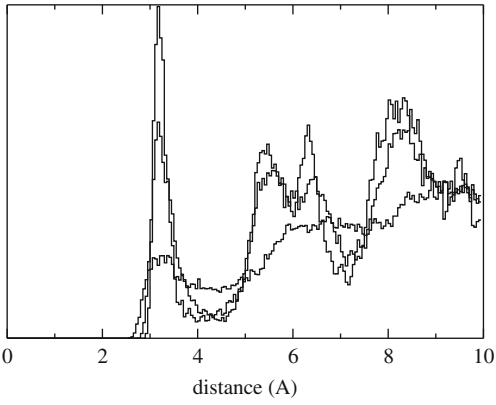


Fig. 13.3 Pair distance distribution. The pair distribution function is evaluated for $kT = 5\,K, 20\,K, 100\,K$ and a density of $\rho = 740$ amu A^{-3}

13.2.4.3 Ballistic and Diffusive Motion

The velocity auto-correlation function (Fig. 13.4)

$$\langle v(t - t_0)v(t_0)\rangle \tag{13.30}$$

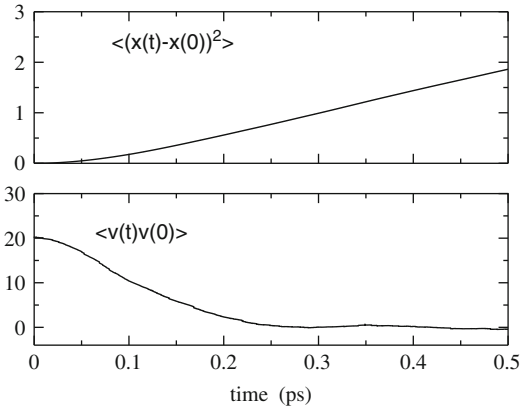


Fig. 13.4 Velocity auto-correlation and mean distance square. The van der Waals system is simulated for $kT = 100\,\mathrm{K}$ and $\rho = 740\,\mathrm{amu\,A^{-3}}$. On a time scale of 0.1 ps the velocity correlation decays and the transition from ballistic motion to diffusive motion occurs

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

$$\langle (x(t) - x(t_0))^2 \rangle. \tag{13.31}$$

Without 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 least square displacement grows linearly with time. The transition between this two types of motion can be analyzed within the model of Brownian motion [62] 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

$$\dot{v} + \gamma v = \Gamma(t) \tag{13.32}$$

with

$$\langle \Gamma(t) \rangle = 0 \tag{13.33}$$

$$\langle \Gamma(t)\Gamma(t') \rangle = \frac{2\gamma kT}{m} \delta(t - t'). \tag{13.34}$$

The velocity correlation decays exponentially

$$\langle v(t)v(t_0) \rangle = \frac{kT}{m} \mathrm{e}^{-\gamma |t - t_0|} \tag{13.35}$$

and the average velocity square is

$$\langle v^2 \rangle = \frac{kT}{m} = \frac{\langle E_{\mathrm{kin}} \rangle}{\frac{m}{2}}. \tag{13.36}$$

The average of x^2 is

$$\langle (x(t) - x(t_0))^2 \rangle = \frac{2kT}{m\gamma}(t - t_0) - \frac{2kT}{m\gamma^2}\left(1 - e^{-\gamma(t-t_0)}\right). \tag{13.37}$$

For small time differences $t - t_0$ the motion is ballistic with the thermal velocity

$$\langle (x(t) - x(t_0))^2 \rangle \approx \frac{kT}{m}t^2 = \langle v^2 \rangle t^2. \tag{13.38}$$

For large time differences a diffusive motion emerges with

$$\langle (x(t) - x(t_0)^2 \rangle \approx \frac{2kT}{m\gamma}t = 2Dt \tag{13.39}$$

and the diffusion constant is given by the Einstein relation

$$D = \frac{kT}{m\gamma}. \tag{13.40}$$

In three dimensions the contributions of the three independent squares have to be summed up.

13.3 Monte Carlo Simulation

The basic principles of Monte Carlo simulations are discussed in Chap. 8. Here we will apply the Metropolis algorithm to simulate the Ising model in one or two dimensions. The Ising model [63, 64] is primarily a model for the phase transition of a ferromagnetic system. It has, however, further applications, for instance, for a polymer under the influence of an external force or protonation equilibria in proteins.

13.3.1 One-Dimensional Ising Model

We consider a chain consisting of N spins which can be either up ($S_i = 1$) or down ($S_i = -1$) (Fig. 13.5). The total energy in a magnetic field is

$$H = -MB = -B\sum_{i=1}^{N} \mu S_i \tag{13.41}$$

and the average magnetic moment of one spin is

$$< M >= \mu \frac{e^{\mu B/kT} - e^{-\mu B/kT}}{e^{\mu B/kT} + e^{-\mu B/kT}} = \mu \tanh\left(\frac{\mu B}{kT}\right). \tag{13.42}$$

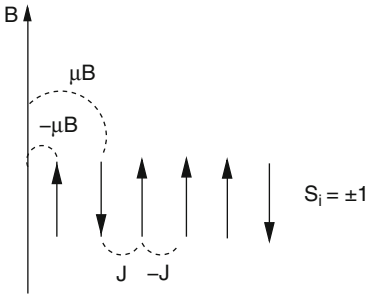


Fig. 13.5 Ising model. N spins can be up or down. The interaction with the magnetic field is $-\mu B S_i$, the interaction between nearest neighbors is $-J S_i S_j$

If interaction between neighboring spins is included the energy of a configuration $(S_1 \cdots S_N)$ becomes

$$H = -\mu B \sum_{i=1}^{N} S_i - J \sum_{i=1}^{N-1} S_i S_{i+1}. \tag{13.43}$$

The one-dimensional model can be solved analytically [61]. In the limit $N \to \infty$ the magnetization is

$$< M >= \mu \frac{\sinh(\frac{\mu B}{kT})}{\sqrt{\sinh^2(\frac{\mu B}{kT}) + e^{4J/kT}}}. \tag{13.44}$$

The numerical simulation (Fig. 13.6) starts either with the ordered state $S_i = 1$ or with a random configuration. New configurations are generated with the Metropolis method as follows:

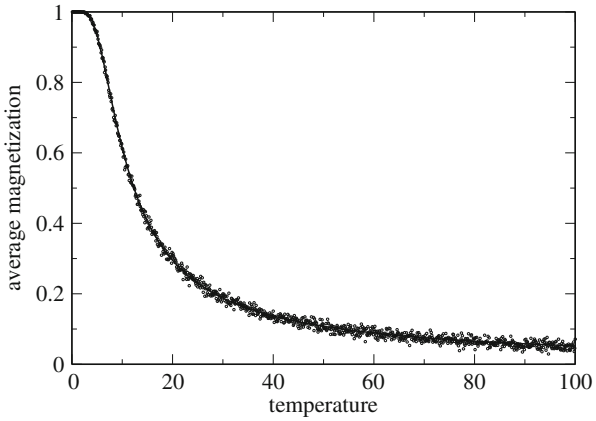


Fig. 13.6 Numerical simulation of the one-dimensional Ising model. The average magnetization per spin is calculated from a MC simulation (*circles*) and compared to the exact solution (13.44). Parameters are $\mu B = -5$ and $J = -2$

- flip one randomly chosen spin S_i[2] and calculate the energy change due to the change $\Delta S_i = (-S_i) - S_i = -2S_i$

$$\Delta E = -\mu B \Delta S_i - J \Delta S_i (S_{i+1} + S_{i-1}) = 2\mu B S_i + 2J S_i \, (S_{i+1} + S_{i-1}) \quad (13.45)$$

- if $\Delta E < 0$ then accept the flip, otherwise accept it with a probability of $P = \mathrm{e}^{-\Delta E/kT}$

As a simple example consider $N = 3$ spins which have eight possible configurations. The probabilities of the trial step $T_{i \to j}$ are shown in Table 13.1.

The Table 13.1 is symmetric and all configurations are connected

Table 13.1 Transition probabilities for a three-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

13.3.2 Two-Dimensional Ising Model

For dimension $d > 1$ the Ising model behaves qualitatively different as a phase transition appears. For $B = 0$ the two-dimensional Ising model (Fig. 13.7) with four

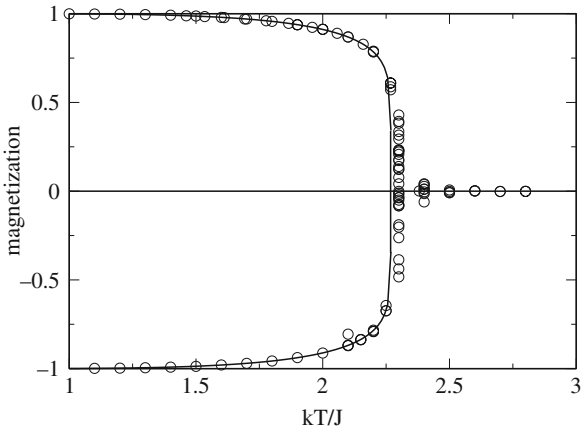


Fig. 13.7 Numerical simulation of the two-dimensional Ising model. The average magnetization per spin is calculated for $B = 0$ from a MC simulation (*circles*) and compared to Eq. (13.47)

[2] Or try one spin after the other.

nearest neighbors can be solved analytically [65, 66]. The magnetization disappears above the critical temperature T_c, which is given by

$$\frac{J}{kT_c} = -\frac{1}{2}\ln\left(\sqrt{2}-1\right) \approx \frac{1}{2.27}. \tag{13.46}$$

Below T_c the average magnetization is given by

$$< M >= \left(1 - \frac{1}{\sinh^4\left(\frac{2J}{kT}\right)}\right)^{\frac{1}{8}}. \tag{13.47}$$

Problems

Problem 13.1 van der Waals System

In this computer experiment a van der Waals System is simulated. The pressure is calculated from the average transfer of momentum (13.19) and compared with expression (13.25). In our example we use the van der Waals parameters for oxygen [50]. In fact there exists only one universal Lennard-Jones system which can be mapped onto arbitrary potential parameters by a rescaling procedure.

- 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_{mol} 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 13.2 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

$$M = \sum_{i=1}^{N} S_i$$

is calculated. It is averaged over 500 such cycles and then compared graphically with the analytical solution for the infinite chain (13.44). Temperature and magnetic field can be varied.

Problem 13.3 Two-State Model for a Polymer

Consider a polymer consisting of N units which can be in two states $S_i = +1$ or $S_i = -1$ with corresponding lengths l_+ and l_- (Fig. 13.8). 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

$$E = -\kappa \sum_i l(S_i) + \sum_i w(S_i, S_{i+1})$$

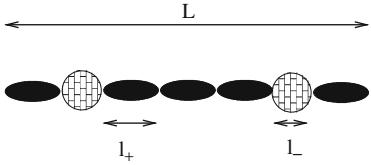


Fig. 13.8 Two-state model for a polymer

This model is isomorphic to the one-dimensional Ising model:

$$\begin{aligned} E &= -\kappa N \frac{l_- + l_+}{2} - \kappa \frac{l_+ - l_-}{2} \sum S_i \\ &\quad + \sum \left(w_{+-} + \frac{w_{++} - w_{+-}}{2} S_i + \frac{w_{+-} - w_{--}}{2} S_{i+1} \right. \\ &\qquad \left. + \frac{w_{++} + w_{--} - 2w_{+-}}{2} S_i S_{i+1} \right) \\ &= \kappa N \frac{l_- + l_+}{2} + N w_{+-} \\ &\quad - \kappa \frac{l_+ - l_-}{2} M + \frac{w_{++} - w_{--}}{2} M \\ &\quad + \frac{w_{++} + w_{--} - 2w_{+-}}{2} \sum S_i S_{i+1} \end{aligned}$$

Comparison with (13.43) shows the correspondence

$$\begin{aligned} -J &= \frac{w_{++} + w_{--} - 2w_{+-}}{2} \\ -\mu B &= -\kappa \frac{l_+ - l_-}{2} + \frac{w_{++} - w_{--}}{2} \\ L &= \sum l(S_i) = N \frac{l_+ + l_-}{2} + \frac{l_+ - l_-}{2} M \end{aligned}$$

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 (13.44). Temperature and magnetic field can be varied as well as the coupling J. For negative J the antiferromagnetic state becomes stable at low-magnetic field strengths.

Problem 13.4 Two-Dimensional Ising Model

In this computer experiment a 200×200 square lattice with periodic boundaries and interaction with the four 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 (13.47).

Chapter 14
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 [67] *is a sequence of successive random steps. In this chapter we study Markovian* [68, 69][1] *discrete time*[2] *models. The time evolution of a system is described in terms of a 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. 14.1):

$$\mathbf{r}(t_{n+1}) = \mathbf{r}(t_n) + \Delta\mathbf{r}_n, \tag{14.1}$$

where the steps are distributed according to the probability distribution[3]

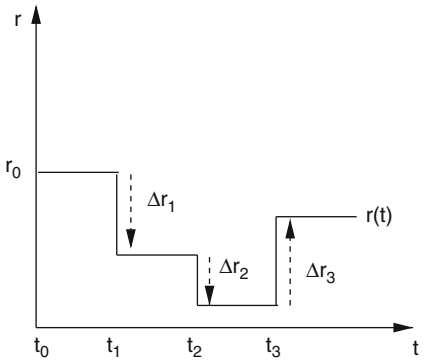


Fig. 14.1 Discrete time random walk

[1] Different steps are independent.

[2] A special case of the more general continuous time random walk with a waiting time distribution of $P(\tau) = \delta(\tau - \Delta t)$.

[3] General random walk processes are characterized by a distribution function $P(\mathbf{R}, \mathbf{R}')$. Here we consider only correlated processes for which $P(\mathbf{R}, \mathbf{R}') = P(\mathbf{R}' - \mathbf{R})$.

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_14,

$$P(\Delta \mathbf{r}_n = \mathbf{b}) = f(\mathbf{b}). \tag{14.2}$$

The probability of reaching the position $\mathbf{R}$ *after* $n+1$ *steps obeys the equation*

$$\begin{aligned} P_{n+1}(\mathbf{R}) &= P\left(\mathbf{r}(t_{n+1}) = \mathbf{R}\right) \\ &= \int \mathrm{d}^N \mathbf{b}\, P_n(\mathbf{R} - \mathbf{b})\, f(\mathbf{b}). \end{aligned} \tag{14.3}$$

14.1 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

$$\overline{b} = \int \mathrm{d}b\, b\, f(b) \quad \overline{b^2} = \int \mathrm{d}b\, b^2\, f(b) \quad \sigma_b = \sqrt{\overline{b^2} - \overline{b}^2}. \tag{14.4}$$

Hence the normalized quantity

$$\xi_i = \frac{\Delta x_i - \overline{b}}{\sigma_b} \tag{14.5}$$

is a random variable with zero average and unit standard deviation. The distribution function of the new random variable

$$\eta_n = \frac{\xi_1 + \xi_2 + \cdots + \xi_n}{\sqrt{n}} = \frac{r_n - n\overline{b}}{\sigma_b \sqrt{n}} \tag{14.6}$$

approaches a normal distribution for large n

$$f(\eta_n) \to \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-\eta_n^2/2} \tag{14.7}$$

and finally from

$$f(r_n)\mathrm{d}r_n = f(\eta_n)\mathrm{d}\eta_n = f(\eta_n)\frac{\mathrm{d}r_n}{\sigma_b \sqrt{n}}$$

we have

$$f(r_n) = \frac{1}{\sqrt{2\pi n}\sigma_b} \exp\left\{ -\frac{(r_n - n\overline{b})^2}{2n\sigma_b^2} \right\}. \tag{14.8}$$

The position of the walker after n steps obeys approximately a Gaussian distribution centered at $\overline{r}_n = n\overline{b}$ with a standard deviation of (Fig. 14.2)

$$\sigma_{r_n} = \sqrt{n}\sigma_b. \tag{14.9}$$

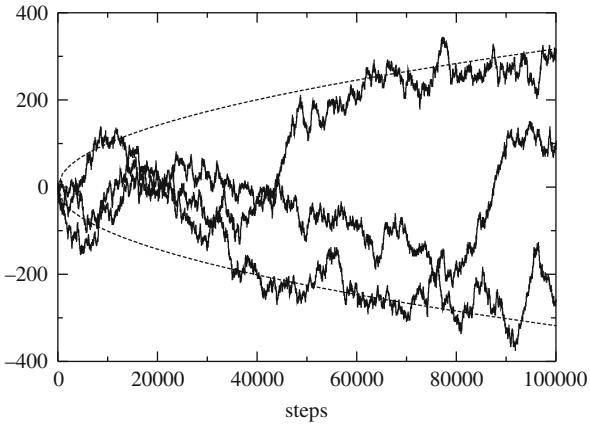


Fig. 14.2 Random walk with constant step size. The figure shows the position r_n for three different one-dimensional random walks with step size $\Delta x = \pm 1$. The *dashed curves* show the width $\pm\sigma = \pm\sqrt{n}$ of the Gaussian approximation (14.8)

14.1.1 Random Walk with Constant Step Size

In the following we consider the classical example of a one-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 $q = 1 - p$ (Fig. 14.3). The corresponding step size distribution function is

$$f(b) = p\delta(b + \Delta x) + q\delta(b - \Delta x) \tag{14.10}$$

with the first two moments

$$\overline{b} = (q - p)\Delta x \quad \overline{b^2} = \Delta x^2. \tag{14.11}$$

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

$$P_{n+1}(m) = p\,P_n(m + 1) + q\,P_n(m - 1) \tag{14.12}$$

which obviously leads to a binomial distribution. From the expansion of

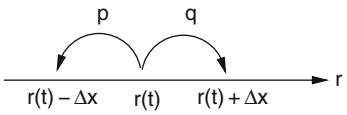


Fig. 14.3 Random walk with constant step size

$$(p+q)^n = \sum \binom{n}{m} p^m q^{n-m} \tag{14.13}$$

we see that

$$P_n(n-2m) = \binom{n}{m} p^m q^{n-m} \tag{14.14}$$

or after substitution $m' = n - 2m = -n, -n+2, \ldots n-2, n$:

$$P_n(m') = \binom{n}{(n-m')/2} p^{(n-m')/2} q^{(n+m')/2}. \tag{14.15}$$

Since the steps are uncorrelated we easily find the first two moments

$$\overline{r}_n = \sum_{i=1}^{n} \overline{\Delta x_i} = n\overline{b} = n\Delta x(q-p) \tag{14.16}$$

and

$$\overline{r_n^2} = \overline{\left(\sum_{i=1}^{n} \Delta x_i\right)^2} = \overline{\sum_{i,j=1}^{n} \Delta x_i \Delta x_j} = \sum_{i=1}^{n} \overline{(\Delta x_i)^2} = n\overline{b^2} = n\Delta x^2. \tag{14.17}$$

14.2 The Freely Jointed Chain

We consider a simple statistical model for the conformation of a biopolymer like DNA or a protein (Figs. 14.4, 14.5).

The polymer is modeled by a three-dimensional chain consisting of M units with constant bond length. The relative orientation of the segments is arbitrary. 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 three-dimensional random walk with constant step size

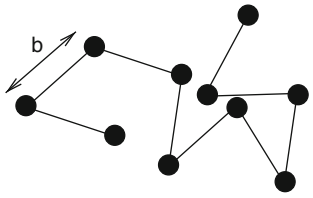


Fig. 14.4 Freely jointed chain with constant bond length b

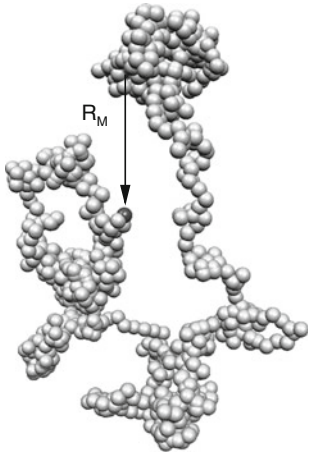


Fig. 14.5 Freely jointed chain. The figure shows a random three-dimensional structure with 1000 segments visualized as *balls* (Molekel graphics [70])

$$(\mathbf{r}_0, \mathbf{r}_1 \cdots \mathbf{r}_M) \quad \mathbf{r}_M = \mathbf{r}_0 + \sum_{i=1}^{M} \mathbf{b}_i . \tag{14.18}$$

14.2.1 Basic Statistic Properties

The M bond vectors

$$\mathbf{b}_i = \mathbf{r}_i - \mathbf{r}_{i-1} \tag{14.19}$$

have a fixed length $|\mathbf{b}_i| = b$ and are oriented randomly. The first two moments are

$$\overline{\mathbf{b}_i} = 0 \quad \overline{\mathbf{b}_i^2} = b^2. \tag{14.20}$$

Since different units are independent

$$\overline{\mathbf{b}_i \mathbf{b}_j} = \delta_{i,j} b^2. \tag{14.21}$$

Obviously the relative position of segment j

$$\mathbf{R}_j = \mathbf{r}_j - \mathbf{r}_0 = \sum_{i=1}^{j} \mathbf{b}_i \tag{14.22}$$

has zero mean

$$\overline{\mathbf{R}_j} = \sum_{i=1}^{j} \overline{\mathbf{b}_i} = 0 \tag{14.23}$$

and its second moment is

$$\overline{R_j^2} = \overline{\left(\sum_{i=1}^{j} \mathbf{b}_i \sum_{k=1}^{j} \mathbf{b}_k\right)} = \sum_{i,k=1}^{j} \overline{\mathbf{b}_i \mathbf{b}_k} = jb^2. \tag{14.24}$$

For the end to end distance

$$\mathbf{R}_M = \mathbf{r}_M - \mathbf{r}_0 = \sum_{i=1}^{M} \mathbf{b}_i \tag{14.25}$$

this gives

$$\overline{\mathbf{R}}_M = 0, \quad \overline{R_M^2} = Mb^2. \tag{14.26}$$

Let us apply the central limit theorem for large M. For the x-coordinate of the end to end vector we have

$$X = \sum_{i=1}^{M} \mathbf{b}_i \mathbf{e}_x = b \sum_i \cos\theta_i. \tag{14.27}$$

With the help of the averages[4]

$$\overline{\cos\theta_i} = \frac{1}{4\pi} \int_0^{2\pi} \mathrm{d}\phi \int_0^{\pi} \cos\theta \, \sin\theta \mathrm{d}\theta = 0 \tag{14.28}$$

$$\overline{(\cos\theta_i)^2} = \frac{1}{4\pi} \int_0^{2\pi} \mathrm{d}\phi \int_0^{\pi} \cos^2\theta \, \sin\theta \mathrm{d}\theta = \frac{1}{3} \tag{14.29}$$

we find that the scaled difference

$$\xi_i = \sqrt{3} \cos\theta_i \tag{14.30}$$

has zero mean and unit variance and therefore the sum

$$\widetilde{X} = \frac{\sqrt{3}}{b\sqrt{M}} X = \sqrt{\frac{3}{M}} \sum_{i=1}^{M} \cos\theta_i \tag{14.31}$$

converges to a normal distribution:

[4] For a one-dimensional polymer $\overline{\cos\theta_i} = 0$ and $\overline{(\cos\theta_i)^2} = 1$. In two dimensions $\overline{\cos\theta_i} = \frac{1}{\pi}\int_0^{\pi} \cos\theta \, \mathrm{d}\theta = 0$ and $\overline{(\cos\theta_i)^2} = \frac{1}{\pi}\int_0^{\pi} \cos^2\theta \, \mathrm{d}\theta = \frac{1}{2}$. To include these cases the factor 3 in the exponent of (14.34) should be replaced by the dimension d.

$$P(\widetilde{X}) = \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{\widetilde{X}^2}{2}\right\}. \tag{14.32}$$

Hence

$$P(X) = \frac{1}{\sqrt{2\pi}} \frac{\sqrt{3}}{b\sqrt{M}} \exp\left\{-\frac{3}{2Mb^2} X^2\right\} \tag{14.33}$$

and finally in three dimensions

$$\begin{aligned} P(\mathbf{R}_M) &= P(X)\, P(Y)\, P(Z) \\ &= \frac{\sqrt{27}}{b^3\sqrt{(2\pi M)^3}} \exp\left\{-\frac{3}{2Mb^2}\mathbf{R}_M^2\right\}. \end{aligned} \tag{14.34}$$

14.2.2 Gyration Tensor

For the center of mass

$$\mathbf{R}_\mathrm{c} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{R}_i \tag{14.35}$$

we find

$$\overline{\mathbf{R}_\mathrm{c}} = 0 \quad \overline{R_c^2} = \frac{1}{M^2}\sum_{i,j} \overline{\mathbf{R}_i \mathbf{R}_j} \tag{14.36}$$

and since

$$\overline{\mathbf{R}_i \mathbf{R}_j} = \min(i,\, j)\, b^2 \tag{14.37}$$

we have

$$\overline{R_\mathrm{c}^2} = \frac{b^2}{M^2}\left(2\sum_{i=1}^{M} i(M-i+1) - \sum_{i=1}^{M} i\right) = \frac{b^2}{M^2}\left(\frac{M^3}{3} + \frac{M^2}{2} + \frac{M}{6}\right) \approx \frac{Mb^2}{3}. \tag{14.38}$$

The gyration radius [71] is generally defined by

$$\begin{aligned} R_\mathrm{g}^2 &= \frac{1}{M}\sum_{i=1}^{M} \overline{(\mathbf{R}_i - \mathbf{R}_\mathrm{c})^2} \\ &= \frac{1}{M}\sum_{i=1}^{M}\left(\overline{R_i^2} + \overline{R_c^2} - 2\frac{1}{M}\sum_{j=1}^{M}\overline{\mathbf{R}_i \mathbf{R}_j}\right) = \frac{1}{M}\sum_{i}(\overline{R_i^2}) - \overline{R_\mathrm{c}^2} \\ &= b^2\frac{M+1}{2} - \frac{b^2}{M^2}\left(\frac{M^3}{3} + \frac{M^2}{2} + \frac{M}{6}\right) = b^2\left(\frac{M}{6} - \frac{1}{6M}\right) \approx \frac{Mb^2}{6}. \end{aligned} \tag{14.39}$$

R_g can be also written as

$$R_g^2 = \left(\frac{1}{M} \sum_i \overline{R_i^2} - \frac{1}{M^2} \sum_{ij} \overline{\mathbf{R}_i \mathbf{R}_j} \right) = \frac{1}{2M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \overline{(\mathbf{R}_i - \mathbf{R}_j)^2} \tag{14.40}$$

and can be experimentally measured with the help of scattering phenomena. It is related to the gyration tensor which is defined as

$$\Omega_g = \frac{1}{M} \sum_i \overline{(\mathbf{R}_i - \mathbf{R}_c)(\mathbf{R}_i - \mathbf{R}_c)^t}. \tag{14.41}$$

Its trace is

$$\mathrm{tr}(\Omega_g) = R_g^2 \tag{14.42}$$

and its eigenvalues give us information about the shape of the polymer (Fig. 14.6).

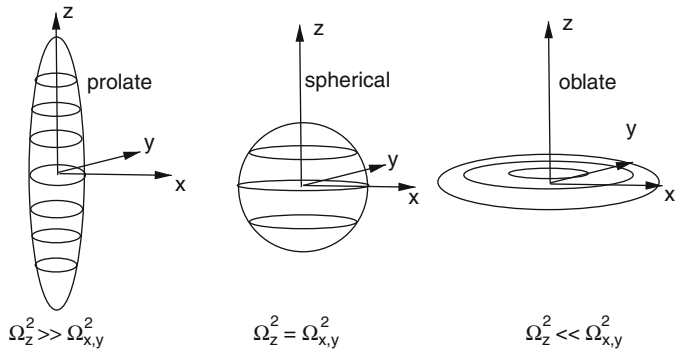


Fig. 14.6 Gyration tensor. The eigenvalues of the gyration tensor give information on the shape of the polymer. If the extension is larger (*smaller*) along one direction than in the perpendicular plane, one eigenvalue is larger (*smaller*) than the two other

14.2.3 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 simulation of a model which treats the segments as Hookean springs. In the limit of a large force constant the two models give equivalent results (Fig. 14.7).

We assume that the segments are independent (self-crossing is not avoided). Then for one segment the energy contribution is

$$E_i = \frac{f}{2} \left(|\mathbf{b}_i| - b \right)^2 . \tag{14.43}$$

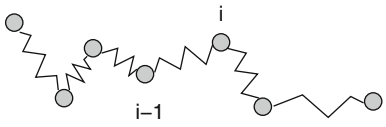


Fig. 14.7 Polymer model with Hookean springs

If the fluctuations are small (Fig. 14.8)

$$\overline{||\mathbf{b}_i| - b|} \ll b \tag{14.44}$$

then

$$\overline{|\mathbf{b}_i|} \approx b \quad \overline{\mathbf{b}_i^2} \approx b^2 \tag{14.45}$$

and the freely jointed chain model (14.34) gives the entropy as a function of the end to end vector

$$S = -k_\mathrm{B}\ \ln\left(P(\mathbf{R}_M)\right) = -k_\mathrm{B}\ \ln\left(\frac{\sqrt{27}}{b^3\sqrt{(2\pi M)^3}}\right) + \frac{3k_\mathrm{B}}{2Mb^2}{\mathbf{R}_M}^2. \tag{14.46}$$

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

$$F = TS - \boldsymbol{\kappa}\mathbf{R}_M = \frac{3k_\mathrm{B}T}{2Mb^2}\mathbf{R}_M^2 - \boldsymbol{\kappa}\mathbf{R}_M + \text{const.} \tag{14.47}$$

In thermodynamic equilibrium the free energy is minimal, hence the average extension is

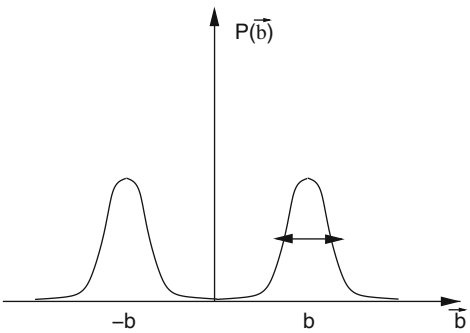


Fig. 14.8 Distribution of bond vectors. The bond vector distribution for a one-dimensional chain of springs has maxima at $\pm b$. For large force constants the width of the two peaks becomes small and the chain of springs resembles a freely jointed chain with constant bond length

$$\overline{\mathbf{R}_M} = \frac{Mb^2}{3k_\mathrm{B}T}\boldsymbol{\kappa}. \tag{14.48}$$

This linear behavior is similar to a Hookean spring with an effective force constant

$$f_\mathrm{eff} = \frac{Mb^2}{3k_\mathrm{B}T} \tag{14.49}$$

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 $R_M \to M(b + \kappa/f)$.

14.3 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 [72–74]. 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

$$\mathbf{F}_\mathrm{fr} = -\gamma\mathbf{v} \tag{14.50}$$

and a randomly fluctuating force with zero mean and infinitely short correlation time

$$\overline{\mathbf{F}_\mathrm{rand}(t)} = 0 \quad \overline{\mathbf{F}_\mathrm{rand}(t)\mathbf{F}_\mathrm{rand}(t')} = \overline{\mathbf{F}^2_\mathrm{rand}}\delta(t - t'). \tag{14.51}$$

The equations of motion for the heavy particle are

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x} &= \mathbf{v} \\ \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{v} &= -\gamma\mathbf{v} + \frac{1}{m}\mathbf{F}_{fr}(t) - \frac{1}{m}\nabla U(\mathbf{x}) \end{aligned} \tag{14.52}$$

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

$$\mathbf{F}_\mathrm{rand}(t) = \mathbf{F}_n \quad t_n \le t < t_{n+1} \tag{14.53}$$

and that the values at different intervals are uncorrelated

$$\overline{\mathbf{F}_n\mathbf{F}_m} = \delta_{m,n}\overline{\mathbf{F}_n^2}. \tag{14.54}$$

The auto-correlation function then is given by

$$\overline{\mathbf{F}_{\text{rand}}(t)\mathbf{F}_{\text{rand}}(t')} = \begin{cases} 0 & \text{different intervals} \\ \overline{\mathbf{F}_n^2} & \text{same interval} \end{cases}. \tag{14.55}$$

Division by Δt gives a sequence of functions which converges to a delta function in the limit $\Delta t \to 0$:

$$\frac{1}{\Delta t}\overline{\mathbf{F}_{\text{rand}}(t)\mathbf{F}_{\text{rand}}(t')} \to \overline{\mathbf{F}_n^2}\,\delta(t-t'). \tag{14.56}$$

Hence we find

$$\overline{\mathbf{F}_n^2} = \frac{1}{\Delta t}\overline{\mathbf{F}_{\text{rand}}^2}. \tag{14.57}$$

Within a short time interval $\Delta t \to 0$ the velocity changes by

$$\mathbf{v}(t_n+\Delta t) = \mathbf{v} - \gamma\mathbf{v}\Delta t - \frac{1}{m}\nabla U(\mathbf{x})\Delta t + \frac{1}{m}\mathbf{F}_n\Delta t + \cdots \tag{14.58}$$

and taking the square gives

$$\mathbf{v}^2(t_n+\Delta t) = \mathbf{v}^2 - 2\gamma\mathbf{v}^2\Delta t - \frac{2}{m}\mathbf{v}\nabla U(\mathbf{x})\Delta t + \frac{2}{m}\mathbf{v}\mathbf{F}_n\Delta t + \frac{\mathbf{F}_n^2}{m^2}(\Delta t)^2 + \cdots. \tag{14.59}$$

Hence for the total energy

$$\begin{aligned} E(t_n+\Delta t) &= \frac{m}{2}\mathbf{v}^2(t_n+\Delta t) + U(\mathbf{x}(t_n+\Delta t)) \\ &= \frac{m}{2}\mathbf{v}^2(t_n+\Delta t) + U(\mathbf{x}) + \mathbf{v}\nabla U(\mathbf{x})\Delta t + \cdots \end{aligned} \tag{14.60}$$

we have

$$E(t_n+\Delta t) = E(t_n) - m\gamma\mathbf{v}^2\Delta t + \mathbf{v}\mathbf{F}_n\Delta t + \frac{\mathbf{F}_n^2}{2m}(\Delta t)^2 + \cdots. \tag{14.61}$$

On the average the total energy $\overline{E}$ should be constant and furthermore in d dimensions

$$\frac{m}{2}\overline{\mathbf{v}^2} = \frac{d}{2}k_{\text{B}}T. \tag{14.62}$$

Therefore we conclude

$$m\gamma\overline{\mathbf{v}^2} = \frac{\Delta t}{2m}\overline{\mathbf{F}_n^2} = \frac{1}{2m}\overline{\mathbf{F}_{\text{rand}}^2} \tag{14.63}$$

from which we obtain finally

$$\overline{\mathbf{F}_n^2} = \frac{2m\gamma d}{\Delta t} k_\mathrm{B} T. \tag{14.64}$$

Problems

Problem 14.1 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. 14.2.

Problem 14.2 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 a large number of random structures. The gyration tensor (Sect. 14.2.2) is diagonalized and the ordered eigenvalues are averaged.

Problem 14.3 Brownian Motion in a Harmonic Potential

The program simulates a particle in a one-dimensional harmonic potential

$$U(\mathbf{x}) = \frac{f}{2} x^2 - \kappa x$$

where κ is an external force. We use the improved Euler method (11.34). First the coordinate and the velocity at midtime are estimated

$$\mathbf{x}\left(t_n + \frac{\mathrm{d}t}{2}\right) = \mathbf{x}(t_n) + \mathbf{v}(t_n)\frac{\mathrm{d}t}{2} \tag{14.65}$$

$$\mathbf{v}\left(t_n + \frac{\mathrm{d}t}{2}\right) = \mathbf{v}(t_n) - \gamma\mathbf{v}(t_n)\frac{\mathrm{d}t}{2} + \frac{\mathbf{F}_n}{m}\frac{\mathrm{d}t}{2} - \frac{f}{m}\mathbf{x}(t_n)\frac{\mathrm{d}t}{2} \tag{14.66}$$

where $\mathbf{F}_n$ is a random number obeying (14.64)

Now the values at t_{n+1} are calculated as

$$\begin{aligned}
\mathbf{x}(t_n + \mathrm{d}t) &= \mathbf{x}(t_n) + \mathbf{v}\left(t_n + \frac{\mathrm{d}t}{2}\right)\mathrm{d}t \\
\mathbf{v}(t_n + \mathrm{d}t) &= \mathbf{v}(t_n) - \gamma\mathbf{v}\left(t_n + \frac{\mathrm{d}t}{2}\right)\mathrm{d}t + \frac{\mathbf{F}_n}{m}\mathrm{d}t - \frac{f}{m}\mathbf{x}\left(t_n + \frac{\mathrm{d}t}{2}\right)\mathrm{d}t
\end{aligned} \tag{14.67}$$

Problem 14.4 Force Extension Relation

The program simulates a chain of springs (Sect. 14.2.3) with potential energy

$$U = \frac{f}{2} \sum (|\mathbf{b}_i| - b)^2 - \boldsymbol{\kappa} \mathbf{R}_M \tag{14.68}$$

The force can be varied and the extension along the force direction is averaged over a large number of time steps.

Chapter 15
Electrostatics

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^{2+}, $Cl^- \cdots$). From a combination of the basic equations of electrostatics

$$div\, D(\mathbf{r}) = \rho(\mathbf{r}) \tag{15.1}$$
$$D(\mathbf{r}) = \varepsilon(\mathbf{r}) E(\mathbf{r}) \tag{15.2}$$
$$E(\mathbf{r}) = -grad\, \Phi(\mathbf{r}) \tag{15.3}$$

the generalized Poisson equation is obtained

$$div(\varepsilon(\mathbf{r})\, grad\, \Phi(\mathbf{r})) = -\rho(\mathbf{r}) = -\rho_{fix}(\mathbf{r}) - \rho_{mobile}(\mathbf{r}), \tag{15.4}$$

where the charge density is formally divided into a fixed and a mobile part

$$\rho(\mathbf{r}) = \rho_{fix}(\mathbf{r}) + \rho_{mobile}(\mathbf{r}). \tag{15.5}$$

Our goal is to calculate the potential $\Phi(\mathbf{r})$ together with the density of mobile charges in a self-consistent way.

15.1 Poisson Equation

We start with the simple case of a dielectric medium without mobile charges and solve (15.5) numerically.

15.1.1 Homogeneous Dielectric Medium

If ε is constant (15.5) simplifies to the Poisson equation

$$\Delta\Phi = -\frac{\rho}{\varepsilon}. \tag{15.6}$$

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_15,

We make use of the discretized Laplace operator

$$\Delta f = \frac{1}{h^2}\{f(x+h,y,z) + f(x-h,y,z) + f(x,y+h,z) + f(x,y-h,z) + \\ + f(x,y,z+h) + f(x,y,z-h) - 6f(x,y,z)\} + O(h^2). \tag{15.7}$$

The integration volume is divided into small cubes which are centered at the grid points

$$\mathbf{r}_{ijk} = (hi, hj, hk). \tag{15.8}$$

The discretized Poisson equation averages over the six neighboring cells ($d\mathbf{r}_1 = (h,0,0)$, etc.):

$$\frac{1}{h^2}\sum_{s=1}^{6}(\Phi(\mathbf{r}_{ijk} + d\mathbf{r}_s) - \Phi(\mathbf{r}_{ijk})) = -\frac{Q_{ijk}}{\varepsilon h^3}, \tag{15.9}$$

where $Q_{ijk} = \rho(\mathbf{r}_{ijk})h^3$ is the total charge in a cell. Equation (15.9) 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 $10^6 \times 10^6$!). We use the iterative method (Sect. 5.5)

$$\Phi^{\text{new}}(\mathbf{r}_{ijk}) = \frac{1}{6}\left(\sum \Phi^{\text{old}}(\mathbf{r}_{ijk} + d\mathbf{r}_s) + \frac{Q_{ijk}}{\varepsilon h}\right). \tag{15.10}$$

The Jacobi method ((5.88) on page 57) makes all the changes in one step whereas the Gauss–Seidel method ((5.91) on page 58) 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, (5.95) on page 59) using a mixture of old and new values

$$\Phi^{\text{new}}(\mathbf{r}_{ijk}) = (1-\omega)\Phi^{\text{old}}(\mathbf{r}_{ijk}) + \omega\frac{1}{6}\left(\sum \Phi^{\text{old}}(\mathbf{r}_{ijk} + d\mathbf{r}_s) + \frac{Q_{ijk}}{\varepsilon h}\right) \tag{15.11}$$

with the relaxation parameter ω. For $1 < \omega < 2$ convergence is faster than for $\omega = 1$. For a square grid with $N = M^2$ points the optimum value of the relaxation parameter is

$$\omega_{\text{opt}} \approx \frac{2}{1 + \frac{\pi}{\sqrt{M}}}. \tag{15.12}$$

Convergence can be further improved by multigrid methods [75, 76]. Error components with short wavelengths are strongly damped during 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. A small number of iterations give a first approximation f_1 with the finite residual

$$r_1 = \nabla^2 f_1 + \rho. \tag{15.13}$$

Then more iterations on a coarser grid are made to solve the equation

$$\nabla^2 f = -r_1 = -\rho - \nabla^2 f_1 \tag{15.14}$$

approximately. The residual of the approximation f_2 is

$$r_2 = \nabla^2 f_2 + r_1 \tag{15.15}$$

and the sum $f_1 + f_2$ gives an improved approximation to the solution since

$$\nabla^2(f_1 + f_2) = -\rho + r_1 + (r_2 - r_1) = -\rho + r_2. \tag{15.16}$$

This method can be extended to a hierarchy of many grids.

15.1.2 Charged Sphere

As a simple example we consider a sphere with a homogeneous charge density (Fig. 15.1)

$$\rho = e\,\frac{3}{4\pi R^3}. \tag{15.17}$$

The potential is given by

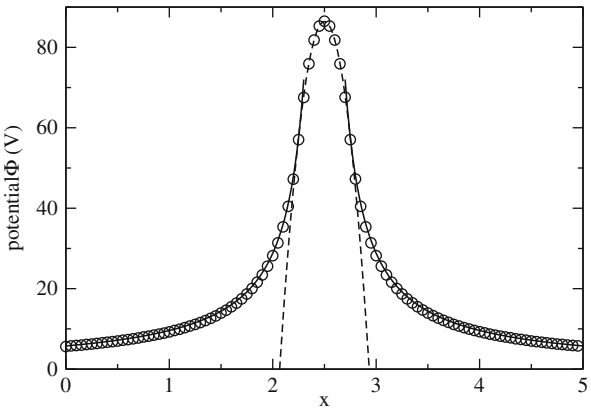


Fig. 15.1 Electrostatic potential of a charged sphere. A charged sphere is simulated with radius $R = 0.25\,A$ and a homogeneous charge density $\rho = e\ 3/4\pi R^3$ embedded in a dielectric medium. The grid consists of 100^3 points with a spacing of $h = 0.05\,A$. The calculated potential (*circles*) is compared to the exact solution ((15.18), *curves*)

$$\Phi(r) = \frac{e}{4\pi\varepsilon_0 R} + \frac{e}{8\pi\varepsilon_0 R}\left(1 - \frac{r^2}{R^2}\right) \quad \text{for } r < R$$
$$\Phi(r) = \frac{e}{4\pi\epsilon_0 r} \quad \text{for } r > R. \tag{15.18}$$

Initial values as well as boundary values are taken from the potential of a point charge which is modified to take into account the finite size of the grid cells

$$\Phi_0(r) = \frac{e}{4\pi\varepsilon_0(r+h)}. \tag{15.19}$$

The interaction energy is (Sect. 15.5) (Fig. 15.2)

$$E_{\text{int}} = \frac{1}{2}\int \varrho(\mathbf{r})\Phi(\mathbf{r})\mathrm{d}^3 r = \frac{3}{20}\frac{e^2}{\pi\varepsilon_0 R} \tag{15.20}$$

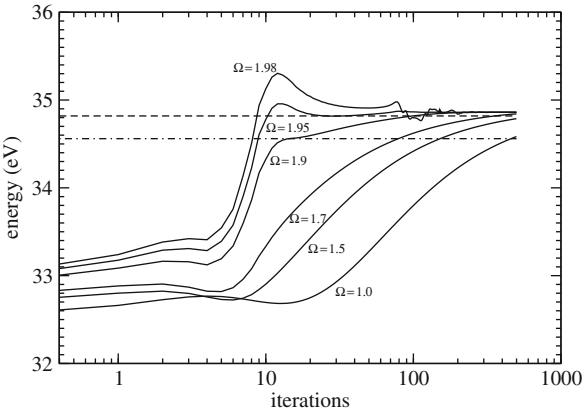


Fig. 15.2 Influence of the relaxation parameter. The convergence of the interaction energy ((15.20), which has a value of 34.56 eV for this example) is studied as a function of the relaxation parameter ω. The optimum value is around $\omega \approx 1.9$. For $\omega > 2$ there is no convergence. The *dash-dotted line* shows the exact value. The *dashed line* shows the exact value for a corrected radius which is derived from the occupied volume (15.35)

15.1.3 Variable ε

For variable ε we use the theorem of Gauss for a vector field $\mathbf{F}$

$$\int \mathrm{d}V\ \mathrm{div}\mathbf{F} = \oint \mathrm{d}A\ \mathbf{F}. \tag{15.21}$$

We choose $\mathbf{F} = \varepsilon\ \mathrm{grad}\,\Phi$ and integrate over one cell

$$\int \mathrm{d}V\ \mathrm{div}(\varepsilon\ \mathrm{grad}\,\Phi) = \int \mathrm{d}V(-\rho) = -Q_{ijk} \tag{15.22}$$

$$\oint \mathrm{d}A\, \varepsilon\, \mathrm{grad}\, \Phi = \sum_{\text{faces}} h^2 \varepsilon\, \mathrm{grad}\, \Phi. \tag{15.23}$$

We approximate grad Φ and ε on the cell face in direction $d\mathbf{r}_s$ by

$$\mathrm{grad}\, \Phi = \frac{1}{h}(\Phi(\mathbf{r}_{ijk} + d\mathbf{r}_s) - \Phi(\mathbf{r}_{ijk})) \tag{15.24}$$

$$\varepsilon = \frac{1}{2}(\varepsilon(\mathbf{r}_{ijk} + d\mathbf{r}_s) + \varepsilon(\mathbf{r}_{ijk})) \tag{15.25}$$

and obtain the discrete equation

$$-Q_{ijk} = h \sum_{s=1}^{6} \frac{\varepsilon(\mathbf{r}_{ijk} + \mathrm{d}\mathbf{r}_s) + \varepsilon(\mathbf{r}_{ijk})}{2} (\Phi(\mathbf{r}_{ijk} + \mathrm{d}\mathbf{r}_s) - \Phi(\mathbf{r}_{ijk})) \tag{15.26}$$

and finally the iteration

$$\Phi^{\mathrm{new}}(\mathbf{r}_{ijk}) = \frac{\sum \frac{\varepsilon(\mathbf{r}_{ijk} + d\mathbf{r}_s) + \varepsilon(\mathbf{r}_{ijk})}{2} \Phi^{\mathrm{old}}(\mathbf{r}_{ijk} + d\mathbf{r}_s) + \frac{Q_{ijk}}{h}}{\sum \frac{\varepsilon(\mathbf{r}_{ijk} + d\mathbf{r}_s) + \varepsilon(\mathbf{r}_{ijk})}{2}}. \tag{15.27}$$

15.1.4 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. Here $\varepsilon = \varepsilon_0$ within the molecule and $\varepsilon = \varepsilon_0\varepsilon_1$ within the medium. At the boundary ε is discontinuous.

Equation (15.27) replaces the discontinuity by the average value $\varepsilon = \varepsilon_0(1+\varepsilon_1)/2$ which can be understood as the discretization of a linear transition between the two values.

15.1.5 Solvation Energy of a Charged Sphere

We consider again a charged sphere, which is now embedded in a dielectric medium with relative dielectric constant ε_1 (Fig. 15.3).

For a spherically symmetrical problem (15.4) can be solved by application of Gauss's theorem

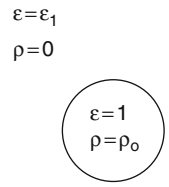


Fig. 15.3 Solvation of a charged sphere in a dielectric medium

$$4\pi r^2 \varepsilon(r) \frac{\mathrm{d}\Phi}{\mathrm{d}r} = -4\pi \int \rho(r) r^2 \mathrm{d}r = -q(r) \tag{15.28}$$

$$\Phi(r) = -\int_0^r \frac{q(r)}{4\pi r^2 \varepsilon(r)} + \Phi(0). \tag{15.29}$$

For the charged sphere we find

$$q(r) = \begin{cases} Qr^3/R^3 \text{ for } r < R \\ Q \text{ for } r > R \end{cases} \tag{15.30}$$

$$\Phi(r) = -\frac{Q}{4\pi\varepsilon_0 R^3}\frac{r^2}{2} + \Phi(0) \quad \text{for } r < R \tag{15.31}$$

$$\Phi(r) = -\frac{Q}{8\pi\varepsilon_0 R} + \Phi(0) + \frac{Q}{4\pi\varepsilon_0\varepsilon_1}\left(\frac{1}{r} - \frac{1}{R}\right) \quad \text{for } r > R. \tag{15.32}$$

The constant $\Phi(0)$ is chosen to give vanishing potential at infinity

$$\Phi(0) = \frac{Q}{4\pi\varepsilon_0\varepsilon_1 R} + \frac{Q}{8\pi\varepsilon_0 R}. \tag{15.33}$$

15.1.5.1 Numerical Results

The numerical results show systematic errors in the center of the sphere. These are mainly due to the discretization of the sphere (Fig. 15.4). The charge is distributed over a finite number N_C of grid cells and therefore the volume deviates from $4\pi R^3/3$. Defining an effective radius by

$$\frac{4\pi}{3} R_\mathrm{eff}^3 = N_\mathrm{C} h^3 \tag{15.34}$$

the deviation of the potential is

$$\Delta\Phi(0) = \frac{Q}{4\pi\varepsilon_0}\left(\frac{1}{\varepsilon_1} + \frac{1}{2}\right)\left(\frac{1}{R_\mathrm{eff}} - \frac{1}{R}\right) \approx \frac{Q}{4\pi\varepsilon_0 R}\left(\frac{1}{\varepsilon_1} + \frac{1}{2}\right)\frac{R - R_\mathrm{eff}}{R} \tag{15.35}$$

which for our numerical experiment amounts to 0.26 V.

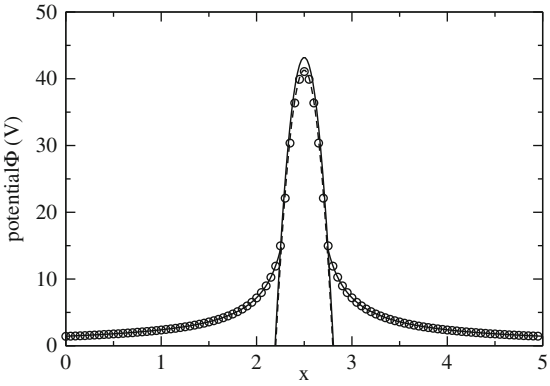


Fig. 15.4 Charged sphere in a dielectric medium. Numerical results for $\varepsilon_1 = 4$ outside the sphere (*circles*) are compared to the exact solution (15.31) and (15.32), *solid curves*). The *dashed line* shows the analytical result corrected for the error which is induced by the continuous transition of ε_1 (15.1.4)

15.1.6 The Shifted Grid Method

The error (Sect. 15.1.4) can be reduced by the following method. Consider a thin box with the normal vector **A** parallel to the gradient of ε. Application of Gauss's theorem gives (Fig. 15.5)

$$\varepsilon_+ \mathbf{A} \operatorname{grad} \Phi_+ = \varepsilon_- \mathbf{A} \operatorname{grad} \Phi_- \tag{15.36}$$

The normal component of grad Φ changes by a factor of $\varepsilon_+/\varepsilon_-$. The discontinuity is located at the surface of a grid cell. Therefore it is of advantage to use a different grid for ε which is shifted by $h/2$ in all directions [77] (Figs. 15.6, 15.7, 15.8):

$$\varepsilon_{ijk} = \varepsilon\left(\left(i+\frac{1}{2}\right)h, \left(j+\frac{1}{2}\right)h, \left(k+\frac{1}{2}\right)h\right). \tag{15.37}$$

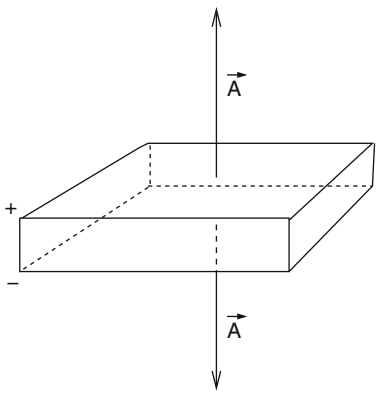


Fig. 15.5 Discontinuity in three dimensions

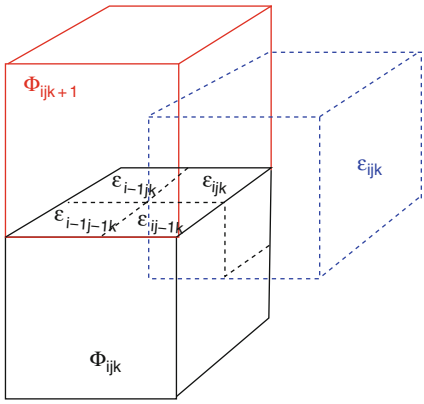


Fig. 15.6 Shifted grid method. A different grid is used for the discretization of ε which is shifted by $h/2$ in all directions

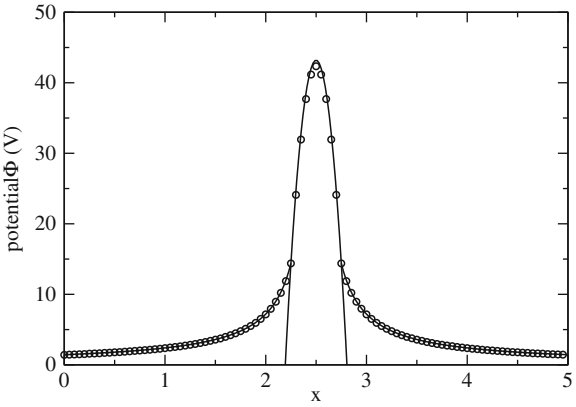


Fig. 15.7 Charged sphere with the shifted grid method. The numerically calculated potential for $\varepsilon_1 = 4$ outside the sphere (*circles*) is compared to the exact solution ((15.31) and (15.32), *solid curves*)

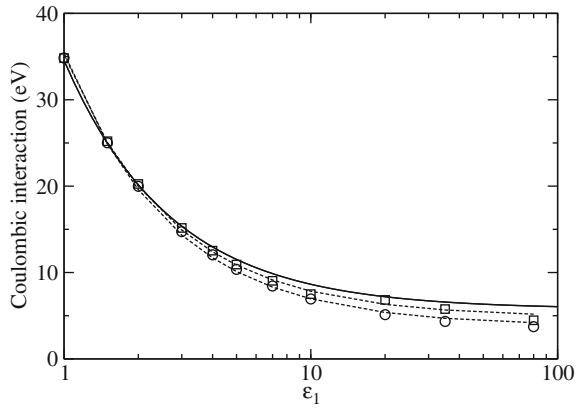


Fig. 15.8 Comparison of numerical errors. The Coulombic interaction of a charged sphere is calculated according to ((15.27), *circles*) and ((15.38), *squares*) and compared to the analytical solution (*solid curve*)

ε has to be averaged over four neighboring cells to give

$$(\varepsilon\,\mathbf{A}\mathrm{grad}\,\phi)_{ijk} = \frac{\phi_{i,j,k+1} - \phi_{i,j,k}}{h}\,\frac{\varepsilon_{ijk} + \varepsilon_{i,j-1,k} + \varepsilon_{i-1,j,k} + \varepsilon_{i-1,j-1,k}}{4} + \cdots . \tag{15.38}$$

15.2 Poisson Boltzmann Equation for an Electrolyte

Let us consider additional mobile charges (for instance, ions in an electrolyte). N_i denotes the average number of ions of type i with charge Q_i. The system is neutral if

$$\sum_i N_i Q_i = 0. \tag{15.39}$$

The interaction energy of a charge Q_i in the electrostatic potential Φ is

$$\Phi Q_i . \tag{15.40}$$

This interaction changes the ion numbers according to the Boltzmann factor:

$$N_i' = N_i \mathrm{e}^{-Q_i\Phi/k_\mathrm{B}T}. \tag{15.41}$$

The charge density of the free ions is

$$\rho_\mathrm{Ion} = \sum_i N_i' Q_i = \sum N_i Q_i \mathrm{e}^{-Q_i\Phi/k_\mathrm{B}T} \tag{15.42}$$

which has to be taken into account in the Poisson equation. Combination gives the Poisson–Boltzmann equation [78–80]

$$\mathrm{div}(\varepsilon\ \mathrm{grad}\ \Phi) = -\sum_i N_i Q_i \mathrm{e}^{-Q_i\Phi/kT} - \rho_\mathrm{fix}. \tag{15.43}$$

For small ion concentrations the exponential can be expanded

$$\mathrm{e}^{-Q_i\Phi/kT} \approx 1 - \frac{Q_i\Phi}{kT} + \frac{1}{2}\left(\frac{Q_i\Phi}{kT}\right)^2 + \cdots \tag{15.44}$$

and the linearized Poisson–Boltzmann equation is obtained:

$$\mathrm{div}(\varepsilon\ \mathrm{grad}\ \Phi) = -\rho_\mathrm{fix} + \sum_i \frac{N_i Q_i^2}{kT}\Phi. \tag{15.45}$$

With

$$\varepsilon = \varepsilon_0\varepsilon_r \tag{15.46}$$

and the definition

$$\kappa^2 = \frac{1}{\varepsilon_0 \varepsilon_r kT} \sum N_i Q_i^2 = \frac{e^2}{\varepsilon_0 \varepsilon_r kT} \sum N_i Z_i^2 \tag{15.47}$$

we have finally

$$\operatorname{div}(\varepsilon_r \operatorname{grad} \Phi) - \varepsilon_r \kappa^2 \Phi = -\frac{1}{\varepsilon_0} \rho. \tag{15.48}$$

For a charged sphere with radius a embedded in a homogeneous medium the solution of (15.48) is given by

$$\Phi = \frac{A}{r} \mathrm{e}^{-\kappa r} \quad A = \frac{e}{4\pi \varepsilon_0 \varepsilon_r} \frac{\mathrm{e}^{\kappa a}}{1 + \kappa a}. \tag{15.49}$$

The potential is shielded by the ions. Its range is of the order $\lambda_{\text{Debye}} = 1/\kappa$ (the so-called Debye length).

15.2.1 Discretization of the Linearized Poisson–Boltzmann Equation

To solve (15.48) the discrete equation (15.26) is generalized to [81]

$$\sum \frac{\varepsilon_r(\mathbf{r}_{ijk} + d\mathbf{r}_s) + \varepsilon_r(\mathbf{r}_{ijk})}{2} \left(\Phi(\mathbf{r}_{ijk} + d\mathbf{r}_s) - \Phi(\mathbf{r}_{ijk})\right) - \varepsilon_r(\mathbf{r}_{ijk}) \kappa^2(\mathbf{r}_{ijk}) h^2 \Phi(\mathbf{r}_{ijk}) = -\frac{Q_{ijk}}{h\varepsilon_0}. \tag{15.50}$$

If ε is constant then we have to iterate

$$\Phi^{\text{new}}(\mathbf{r}_{ijk}) = \frac{\frac{Q_{ijk}}{h\varepsilon_0\varepsilon_r} + \sum \Phi^{\text{old}}(\mathbf{r}_{ijk} + d\mathbf{r}_s)}{6 + h^2 \kappa^2(\mathbf{r}_{ijk})}. \tag{15.51}$$

15.3 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 $\varepsilon = \varepsilon_0$. 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.

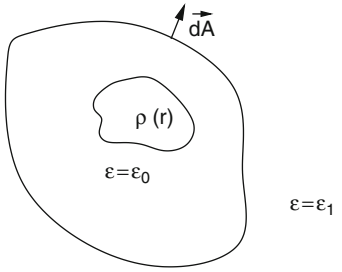


Fig. 15.9 Cavity in a dielectric medium

In the following we consider model systems which are composed of two spatial regions (Fig. 15.9):

- 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 $\varepsilon = \varepsilon_0$.

15.3.1 Integral Equations for the Potential

Starting from the Poisson equation

$$\operatorname{div}(\varepsilon(\mathbf{r})\operatorname{grad}\Phi(\mathbf{r})) = -\rho(\mathbf{r}) \tag{15.52}$$

we will derive some useful integral equations in the following. First we apply Gauss's theorem to the expression [82]

$$\begin{aligned}&\operatorname{div}\left[G(\mathbf{r}-\mathbf{r}')\varepsilon(\mathbf{r})\operatorname{grad}(\Phi(\mathbf{r})) - \Phi(\mathbf{r})\varepsilon(\mathbf{r})\operatorname{grad}(G(\mathbf{r}-\mathbf{r}'))\right]\\ &= -\rho(\mathbf{r})G(\mathbf{r}-\mathbf{r}') - \Phi(\mathbf{r})\varepsilon(\mathbf{r})\operatorname{div}\operatorname{grad}(G(\mathbf{r}-\mathbf{r}')) - \Phi(\mathbf{r})\operatorname{grad}(\varepsilon(\mathbf{r}))\operatorname{grad}(G(\mathbf{r}-\mathbf{r}'))\end{aligned} \tag{15.53}$$

with the yet undetermined function $G(\mathbf{r}-\mathbf{r}')$. Integration over a volume V gives

$$\begin{aligned}&-\int_V \mathrm{d}V\ \big(\rho(\mathbf{r})G(\mathbf{r}-\mathbf{r}') + \Phi(\mathbf{r})\varepsilon(\mathbf{r})\operatorname{div}\operatorname{grad}(G(\mathbf{r}-\mathbf{r}'))\\ &\qquad + \Phi(\mathbf{r})\operatorname{grad}(\varepsilon(\mathbf{r}))\operatorname{grad}(G(\mathbf{r}-\mathbf{r}'))\big)\\ &= \oint_{(V)} \mathrm{d}A\left(G(\mathbf{r}-\mathbf{r}')\varepsilon(\mathbf{r})\frac{\partial}{\partial n}(\Phi(\mathbf{r})) - \Phi(\mathbf{r})\varepsilon(\mathbf{r})\frac{\partial}{\partial n}(G(\mathbf{r}-\mathbf{r}'))\right).\end{aligned} \tag{15.54}$$

Now chose G as the fundamental solution of the Poisson equation

$$G_0(\mathbf{r}-\mathbf{r}') = -\frac{1}{4\pi|\mathbf{r}-\mathbf{r}'|} \tag{15.55}$$

which obeys

$$\text{div grad } G_0 = \delta(\mathbf{r} - \mathbf{r}') \tag{15.56}$$

to obtain the following integral equation for the potential:

$$\Phi(\mathbf{r}')\varepsilon(\mathbf{r}) = \int \mathrm{d}V \frac{\rho(\mathbf{r})}{4\pi|\mathbf{r}-\mathbf{r}'|} + \frac{1}{4\pi}\int \mathrm{d}V\,\Phi(\mathbf{r})\text{grad}(\varepsilon(\mathbf{r}))\text{grad}\left(\frac{1}{|\mathbf{r}-\mathbf{r}'|}\right)$$
$$-\frac{1}{4\pi}\oint_{(V)} \mathrm{d}A\left(\frac{1}{|\mathbf{r}-\mathbf{r}'|}\varepsilon(\mathbf{r})\frac{\partial}{\partial n}(\Phi(\mathbf{r})) + \Phi(\mathbf{r})\varepsilon(\mathbf{r})\frac{\partial}{\partial n}\left(\frac{1}{|\mathbf{r}-\mathbf{r}'|}\right)\right) \tag{15.57}$$

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) [82] (Fig. 15.10).

The gradient of $\varepsilon(\mathbf{r})$ is nonzero only on the boundary surface of the cavity and with the limiting procedure ($d \to 0$)

$$\text{grad}(\varepsilon(\mathbf{r}))\mathrm{d}V = \mathbf{n}\frac{\epsilon_1 - 1}{d}\epsilon_0 \mathrm{d}V = \mathrm{d}A\,\mathbf{n}(\epsilon_1 - 1)\epsilon_0 \tag{15.58}$$

we obtain

$$\Phi(\mathbf{r}') = \frac{1}{\varepsilon(\mathbf{r}')}\int_{\text{cav}} \mathrm{d}V \frac{\rho(\mathbf{r})}{4\pi|\mathbf{r}-\mathbf{r}'|} + \frac{(\epsilon_1 - 1)\epsilon_0}{4\pi\varepsilon(\mathbf{r}')}\oint_S \mathrm{d}A\,\Phi(\mathbf{r})\frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}'|}. \tag{15.59}$$

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 (15.57) to the cavity volume (where $\varepsilon = \epsilon_0$) and obtain

$$\Phi_{\text{in}}(\mathbf{r}') = \int_V \mathrm{d}V\,\frac{\rho(\mathbf{r})}{4\pi|\mathbf{r}-\mathbf{r}'|\epsilon_0}$$
$$-\frac{1}{4\pi}\oint_{(V)} \mathrm{d}A\left(\Phi_{\text{in}}(\mathbf{r})\frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}'|} - \frac{1}{|\mathbf{r}-\mathbf{r}'|}\frac{\partial}{\partial n}\Phi_{\text{in}}(\mathbf{r})\right). \tag{15.60}$$

From comparison with (15.59) we have

$$\oint \mathrm{d}A\frac{1}{|\mathbf{r}-\mathbf{r}'|}\frac{\partial}{\partial n}\Phi_{\text{in}}(\mathbf{r}) = \epsilon_1 \oint \mathrm{d}A\,\Phi_{\text{in}}(\mathbf{r})\frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}'|} \tag{15.61}$$

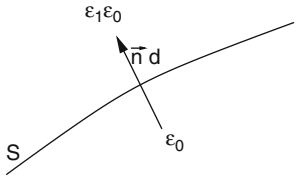


Fig. 15.10 Discontinuity at the cavity boundary

and the potential can be alternatively calculated from the values of its normal gradient at the boundary

$$\Phi(\mathbf{r}') = \frac{1}{\varepsilon(\mathbf{r}')}\int_{\text{cav}} \mathrm{d}V \frac{\rho(\mathbf{r})}{4\pi|\mathbf{r}-\mathbf{r}'|} + \frac{(1-\frac{1}{\epsilon_1})\epsilon_0}{4\pi\varepsilon(\mathbf{r}')}\oint_S \mathrm{d}A \frac{1}{|\mathbf{r}-\mathbf{r}'|}\frac{\partial}{\partial n}\Phi_{\text{in}}(\mathbf{r}). \quad (15.62)$$

This equation can be interpreted as the potential generated by the charge density ρ plus an additional surface charge density

$$\sigma(\mathbf{r}) = \left(1-\frac{1}{\epsilon_1}\right)\epsilon_0 \frac{\partial}{\partial n}\Phi_{\text{in}}(\mathbf{r}). \quad (15.63)$$

Integration over the volume outside the cavity (where $\varepsilon = \epsilon_1\epsilon_0$) gives the following expression for the potential:

$$\Phi_{\text{out}}(\mathbf{r}') = \frac{1}{4\pi}\oint_{(V)} \mathrm{d}A \left(\Phi_{\text{out}}(\mathbf{r})\frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}'|} - \frac{1}{|\mathbf{r}-\mathbf{r}'|}\frac{\partial}{\partial n}\Phi_{\text{out}}(\mathbf{r})\right). \quad (15.64)$$

At the boundary the potential is continuous

$$\Phi_{\text{out}}(\mathbf{r}) = \Phi_{\text{in}}(\mathbf{r}) \quad \mathbf{r}\in A, \quad (15.65)$$

whereas the normal derivative (hence the normal component of the electric field) has a discontinuity

$$\epsilon_1 \frac{\partial\Phi_{\text{out}}}{\partial n} = \frac{\partial\Phi_{\text{in}}}{\partial n}. \quad (15.66)$$

15.3.2 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 $\mathbf{n}_i$. (We assume planar elements in the following, the curvature leads to higher order corrections) (Fig. 15.11).

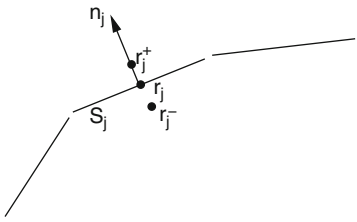


Fig. 15.11 Representation of the boundary by surface elements

The corresponding values of the potential and its normal derivative are denoted as $\Phi_i = \Phi(\mathbf{r}_i)$ and $\frac{\partial \Phi_i}{\partial n} = \mathbf{n}_i \operatorname{grad} \Phi(\mathbf{r}_i)$. At a point $\mathbf{r}_j^{\pm}$ close to the element S_j we obtain the following approximate equations:

$$\Phi_{\text{in}}(\mathbf{r}_j^-) = \int_V \mathrm{d}V \, \frac{\rho(\mathbf{r})}{4\pi\,|\mathbf{r}-\mathbf{r}_j^-|\epsilon_0} - \frac{1}{4\pi}\sum_i \Phi_i \oint_{S_i} \mathrm{d}A \, \frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}_j^-|} + \frac{1}{4\pi}\sum_i \frac{\partial \Phi_{i,\text{in}}}{\partial n} \oint_{S_i} \mathrm{d}A \, \frac{1}{|\mathbf{r}-\mathbf{r}_j^-|} \tag{15.67}$$

$$\Phi_{\text{out}}(\mathbf{r}_j^+) = \frac{1}{4\pi}\sum_i \Phi_i \oint_{S_i} \mathrm{d}A \, \frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}_j^+|} - \frac{1}{4\pi}\sum_i \frac{\partial \Phi_{i,\text{out}}}{\partial n} \oint_{S_i} \mathrm{d}A \, \frac{1}{|\mathbf{r}-\mathbf{r}_j^+|}. \tag{15.68}$$

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

$$\begin{aligned}
\oint_{S_i} \mathrm{d}A \, \frac{1}{|\mathbf{r}-\mathbf{r}_j^+|} &= \oint_{S_i} \mathrm{d}A \, \frac{1}{|\mathbf{r}-\mathbf{r}_j^-|} \\
\oint_{S_i} \mathrm{d}A \, \frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}_i^+|} &= -\oint_{S_i} \mathrm{d}A \, \frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}_i^-|} \\
\oint_{S_i} \mathrm{d}A \, \frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}_j^+|} &= \oint_{S_i} \mathrm{d}A \, \frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}_j^-|} \quad j \neq i
\end{aligned} \tag{15.69}$$

and we find

$$(1+\epsilon_1)\Phi_j = \int_V \mathrm{d}V \, \frac{\rho(\mathbf{r})}{4\pi\epsilon_0|\mathbf{r}-\mathbf{r}_j|} - \frac{1}{4\pi}\sum_{i\neq j}(1-\epsilon_1)\Phi_i \oint_{S_i} \mathrm{d}A \, \frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}_j^-|} - \frac{1}{4\pi}(1+\epsilon_1)\Phi_j \oint_{S_j} \mathrm{d}A \, \frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}_j^-|}. \tag{15.70}$$

The integrals for $i \neq j$ can be approximated by

$$\oint_{S_i} \mathrm{d}A \, \frac{\partial}{\partial n}\frac{1}{|\mathbf{r}-\mathbf{r}_j^-|} = A_i \, \mathbf{n}_i \operatorname{grad}_i \frac{1}{|\mathbf{r}_i-\mathbf{r}_j|}. \tag{15.71}$$

The second integral has a simple geometrical interpretation (Fig. 15.12).

Since $\operatorname{grad}\frac{1}{|r-r'|} = -\frac{1}{|r-r'|^2}\frac{r-r'}{|r-r'|}$ the area element $\mathrm{d}A$ is projected onto a sphere with unit radius. The integral $\oint_{S_j} \mathrm{d}A \operatorname{grad}_{r-}\frac{1}{|\mathbf{r_j}-\mathbf{r}_j^-|}$ 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

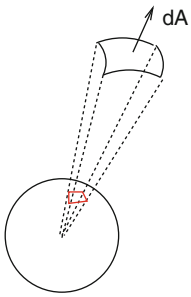


Fig. 15.12 Projection of the surface element

$$(1+\epsilon_1)\Phi_j = \int_V \mathrm{d}V\ \frac{\rho(\mathbf{r})}{4\pi|\mathbf{r}-\mathbf{r}_j|\epsilon_0} - \frac{1}{4\pi}\sum_{i\neq j}(1-\epsilon_1)\Phi_i A_i \frac{\partial}{\partial n_i}\frac{1}{|\mathbf{r}_i-\mathbf{r}_j|} + \frac{1}{2}(1+\epsilon_1)\Phi_j \tag{15.72}$$

or

$$\Phi_j = \frac{2}{1+\epsilon_1}\int_V \mathrm{d}V\ \frac{\rho(\mathbf{r})}{4\pi\epsilon_0|\mathbf{r}-\mathbf{r}_j|} + \frac{1}{2\pi}\sum_{i\neq j}\frac{\epsilon_1-1}{\epsilon_1+1}\Phi_i A_i \frac{\partial}{\partial n_i}\frac{1}{|\mathbf{r}_i-\mathbf{r}_j|}. \tag{15.73}$$

This system of equations can be used to calculate the potential on the boundary. The potential inside the cavity is then given by (15.59). Numerical stability is improved by a related method which considers the potential gradient along the boundary. Taking the normal derivative

$$\frac{\partial}{\partial n_j} = \mathbf{n}_j \mathrm{grad}_{r_j\pm} \tag{15.74}$$

of (15.67, 15.68) gives

$$\frac{\partial}{\partial n_j}\Phi_{\mathrm{in}}(\mathbf{r}_j^-) = \frac{\partial}{\partial n_j}\int_V \mathrm{d}V\ \frac{\rho(\mathbf{r})}{4\pi|\mathbf{r}-\mathbf{r}_j^-|\epsilon_0} - \frac{1}{4\pi}\sum_i \Phi_i \oint_{S_i}\mathrm{d}A\ \frac{\partial^2}{\partial n\partial n_j}\frac{1}{|\mathbf{r}-\mathbf{r}_j^-|} + \frac{1}{4\pi}\sum_i \frac{\partial\Phi_{i,\mathrm{in}}}{\partial n}\oint_{S_i}\mathrm{d}A\ \frac{\partial}{\partial n_j}\frac{1}{|\mathbf{r}-\mathbf{r}_j^-|} \tag{15.75}$$

$$\frac{\partial}{\partial n_j}\Phi_{\mathrm{out}}(\mathbf{r}_j^+) = \frac{1}{4\pi}\sum_i \Phi_i \oint_{S_i}\mathrm{d}A\ \frac{\partial^2}{\partial n\partial n_j}\frac{1}{|\mathbf{r}-\mathbf{r}_j^+|} - \frac{1}{4\pi}\sum_i \frac{\partial\Phi_{i,\mathrm{out}}}{\partial n}\oint_{S_i}\mathrm{d}A\ \frac{\partial}{\partial n_j}\frac{1}{|\mathbf{r}-\mathbf{r}_j^+|}. \tag{15.76}$$

In addition to (15.69) we have now

$$\oint_{S_i}\mathrm{d}A\ \frac{\partial^2}{\partial n\partial n_j}\frac{1}{|\mathbf{r}-\mathbf{r}_j^-|} = \oint_{S_i}\mathrm{d}A\ \frac{\partial^2}{\partial n\partial n_j}\frac{1}{|\mathbf{r}-\mathbf{r}_j^+|} \tag{15.77}$$

and the sum of the two equations gives

$$\left(1+\frac{1}{\epsilon_1}\right)\frac{\partial}{\partial n_j}\Phi_{\text{in},j} = \frac{\partial}{\partial n_j}\left(\int_V \mathrm{d}V\,\frac{\rho(\mathbf{r})}{4\pi\epsilon_0|\mathbf{r}-\mathbf{r}_j|}+\frac{1-\frac{1}{\epsilon_1}}{4\pi}\sum_{i\neq j}A_i\frac{\partial\Phi_{i,\text{in}}}{\partial n}\frac{1}{|\mathbf{r}_i-\mathbf{r}_j|}\right) + \frac{1+\frac{1}{\epsilon_1}}{2\pi}\frac{\partial\Phi_{j,\text{in}}}{\partial n} \tag{15.78}$$

or finally

$$\frac{\partial}{\partial n_j}\Phi_{\text{in},j} = \frac{2\epsilon_1}{\epsilon_1+1}\frac{\partial}{\partial n_j}\int_V \mathrm{d}V\,\frac{\rho(\mathbf{r})}{4\pi\epsilon_0|\mathbf{r}-\mathbf{r}_j|} + 2\frac{\epsilon_1-1}{\epsilon_1+1}\sum_{i\neq j}A_i\frac{\partial\Phi_{i,\text{in}}}{\partial n}\frac{\partial}{\partial n_j}\frac{1}{|\mathbf{r}_i-\mathbf{r}_j|}. \tag{15.79}$$

In terms of the surface charge density this reads:

$$\sigma'_j = 2\varepsilon_0\frac{(1-\varepsilon_1)}{(1+\varepsilon_1)}\left(-\mathbf{n}_j\,\text{grad}\int \mathrm{d}V\,\frac{\rho(\mathbf{r})}{4\pi\epsilon_0|\mathbf{r}-\mathbf{r}'|}+\frac{1}{4\pi\varepsilon_0}\sum_{i\neq j}\sigma'_i A_i\frac{\mathbf{n}_j(\mathbf{r}_j-\mathbf{r}_i)}{|\mathbf{r}_i-\mathbf{r}_j|^3}\right). \tag{15.80}$$

This system of linear equations can be solved directly or iteratively (a simple damping scheme $\sigma'_m \to \omega\sigma'_m+(1-\omega)\sigma'_{m,\text{old}}$ 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 (15.62).

15.4 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 (15.48)

$$G_\kappa(\mathbf{r}-\mathbf{r}') = -\frac{e^{-\kappa|\mathbf{r}-\mathbf{r}'|}}{4\pi|\mathbf{r}-\mathbf{r}'|} \tag{15.81}$$

obeys

$$\text{div grad}\,G_\kappa(\mathbf{r}-\mathbf{r}') - \kappa^2 G_\kappa(\mathbf{r}-\mathbf{r}') = \delta(\mathbf{r}-\mathbf{r}'). \tag{15.82}$$

Inserting into Green's theorem (15.54) we obtain the potential outside the cavity

$$\Phi_{\text{out}}(\mathbf{r}') = -\oint_{(V)} \mathrm{d}A \left(\Phi_{\text{out}}(\mathbf{r}) \frac{\partial}{\partial n} G_\kappa(\mathbf{r}-\mathbf{r}') - G_\kappa(\mathbf{r}-\mathbf{r}') \frac{\partial}{\partial n} \Phi_{\text{out}}(\mathbf{r}) \right) \quad (15.83)$$

which can be combined with (15.60, 15.66) to give the following equations [83]

$$(1+\epsilon_1)\Phi(\mathbf{r}') = \oint \mathrm{d}A \left[\Phi(\mathbf{r}) \frac{\partial}{\partial n}(G_0 - \epsilon_1 G_\kappa) - (G_0 - G_\kappa) \frac{\partial}{\partial n} \Phi_{\text{in}}(\mathbf{r}) \right] + \int \frac{\rho(\mathbf{r})}{4\pi\epsilon_0 |\mathbf{r}-\mathbf{r}'|} \mathrm{d}V \quad (15.84)$$

$$(1+\epsilon_1) \frac{\partial}{\partial n'} \Phi_{\text{in}}(\mathbf{r}') = \oint \mathrm{d}A \Phi(\mathbf{r}) \frac{\partial^2}{\partial n \partial n'}(G_0 - G_\kappa) - \oint \mathrm{d}A \frac{\partial}{\partial n} \Phi_{\text{in}}(\mathbf{r}) \frac{\partial}{\partial n'} \left(G_0 - \frac{1}{\epsilon_1} G_k \right) + \frac{\partial}{\partial n'} \int \frac{\rho(\mathbf{r})}{4\pi\epsilon |\mathbf{r}-\mathbf{r}'|} \mathrm{d}V. \quad (15.85)$$

For a set of discrete boundary elements the following equations determine the values of the potential and its normal derivative at the boundary:

$$\frac{1+\epsilon_1}{2} \Phi_j = \sum_{i \neq j} \Phi_i \oint \mathrm{d}A \frac{\partial}{\partial n}(G_0 - \epsilon_1 G_\kappa) - \sum_{i \neq j} \frac{\partial}{\partial n} \Phi_{i,\text{in}} \oint \mathrm{d}A (G_0 - G_\kappa) + \int \frac{\rho(\mathbf{r})}{4\pi\epsilon_0 |\mathbf{r}-\mathbf{r}_i|} \mathrm{d}V \quad (15.86)$$

$$\frac{1+\epsilon_1}{2} \frac{\partial}{\partial n'} \Phi_{i,\text{in}} = \sum_{i \neq j} \Phi_i \oint \mathrm{d}A \frac{\partial^2}{\partial n \partial n'}(G_0 - G_\kappa) - \sum_{i \neq j} \frac{\partial}{\partial n} \Phi_{i,\text{in}} \oint \mathrm{d}A \frac{\partial}{\partial n'} \left(G_0 - \frac{1}{\epsilon_1} G_k \right) + \frac{\partial}{\partial n'} \int \frac{\rho(\mathbf{r})}{4\pi\epsilon |\mathbf{r}-\mathbf{r}_i|} \mathrm{d}V. \quad (15.87)$$

The situation is much more involved than for the simpler Poisson equation (with $\kappa = 0$) since the calculation of a large number of integrals including such with singularities is necessary [83, 84].

15.5 Electrostatic Interaction Energy (Onsager Model)

A very important quantity in molecular physics is the electrostatic interaction of a molecule and the surrounding solvent [85, 86]. 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 $(\lambda + \mathrm{d}\lambda)\rho(r)$ with $0 \leq \lambda \leq 1$. The corresponding energy change is

$$\begin{aligned} \mathrm{d}E &= \int \mathrm{d}\lambda\, \rho(r)\, \Phi_\lambda(r)\mathrm{d}V \\ &= \int \mathrm{d}\lambda\, \rho(r) \left(\sum_n \frac{\sigma_n(\lambda) A_n}{4\pi\varepsilon_0 |\mathbf{r} - \mathbf{r}_n|} + \int \frac{\lambda\rho(r')}{4\pi\varepsilon_0 |\mathbf{r} - r'|} \mathrm{d}V' \right) \mathrm{d}V. \end{aligned} \tag{15.88}$$

Multiplication of (15.80) 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

$$\mathrm{d}E = \lambda \mathrm{d}\lambda \int \rho(r) \left(\sum_n \frac{\sigma_n A_n}{4\pi\varepsilon_0 |\mathbf{r} - \mathbf{r}_n|} + \frac{\rho(r')}{4\pi\varepsilon_0 |\mathbf{r} - r'|} \mathrm{d}V' \right). \tag{15.89}$$

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

$$\begin{aligned} E_{\mathrm{int}} &= \int \mathrm{d}E = \int_0^1 \lambda \mathrm{d}\lambda \int \rho(r) \sum_n \frac{\sigma_n A_n}{4\pi\varepsilon_0 |\mathbf{r} - \mathbf{r}_n|} \mathrm{d}V \\ &= \sum_n \sigma_n A_n \int \frac{\rho(r)}{8\pi\varepsilon_0 |\mathbf{r} - \mathbf{r}_n|} \mathrm{d}V. \end{aligned} \tag{15.90}$$

For the special case of a spherical cavity with radius a an analytical solution by multipole expansion is available [87]

$$E_{\mathrm{int}} = -\frac{1}{8\pi\varepsilon_0} \sum_l \sum_{m=-l}^{l} \frac{(l+1)(\varepsilon_1 - 1)}{[l + \varepsilon_1(l+1)]\, a^{2l+1}} M_l^m M_l^m \tag{15.91}$$

with the multipole moments

$$M_l^m = \int \rho(r, \theta, \varphi) \sqrt{\frac{4\pi}{2l+1}} r^l Y_l^m(\theta, \varphi) \mathrm{d}V. \tag{15.92}$$

The first two terms of this series are

$$E_{\mathrm{int}}^{(0)} = -\frac{1}{8\pi\varepsilon_0} \frac{\varepsilon_1 - 1}{\varepsilon_1 a} M_0^0 M_0^0 = -\frac{1}{8\pi\varepsilon_0} \left(1 - \frac{1}{\varepsilon_1}\right) \frac{Q^2}{a} \tag{15.93}$$

$$\begin{aligned} E_{\mathrm{int}}^{(1)} &= -\frac{1}{8\pi\varepsilon_0} \frac{2(\varepsilon_1 - 1)}{(1 + 2\varepsilon_1) a^3} (M_1^{-1} M_1^{-1} + M_1^0 M_1^0 + M_1^1 M_1^1) \\ &= -\frac{1}{8\pi\varepsilon_0} \frac{2(\varepsilon_1 - 1)}{1 + 2\varepsilon_1} \frac{\mu^2}{a^3}. \end{aligned} \tag{15.94}$$

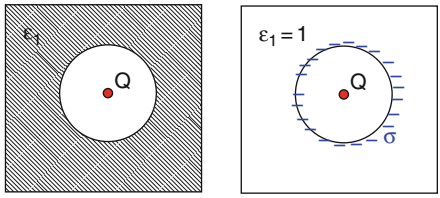


Fig. 15.13 Surface charges

15.5.1 Example: Point Charge in a Spherical Cavity

Consider a point charge Q in the center of a spherical cavity of radius R. The dielectric constant is given by

$$\varepsilon = \begin{cases} \varepsilon_0 & r < R \\ \varepsilon_1 \varepsilon_0 & r > R \end{cases} . \tag{15.95}$$

Electric field and potential are inside the cavity

$$E = \frac{Q}{4\pi\varepsilon_0 r^2} \quad \Phi = \frac{Q}{4\pi\varepsilon_0 r} + \frac{Q}{4\pi\varepsilon_0 R}\left(\frac{1}{\varepsilon_1} - 1\right) \tag{15.96}$$

and outside

$$E = \frac{Q}{4\pi\varepsilon_1\varepsilon_0 r^2} \quad \Phi = \frac{Q}{4\pi\varepsilon_1\varepsilon_0 r} \quad r > R \tag{15.97}$$

which in terms of the surface charge density σ is

$$E = \frac{Q + 4\pi R^2\sigma}{4\pi\varepsilon_0 r^2} \quad r > R \tag{15.98}$$

with the total surface charge

$$4\pi R^2\sigma = Q\left(\frac{1}{\varepsilon_1} - 1\right). \tag{15.99}$$

The solvation energy (15.90) is given by

$$E_{\text{int}} = \frac{Q^2}{8\pi\varepsilon_0}\left(\frac{1}{\varepsilon_1} - 1\right) \tag{15.100}$$

which is the first term (15.93) of the multipole expansion.

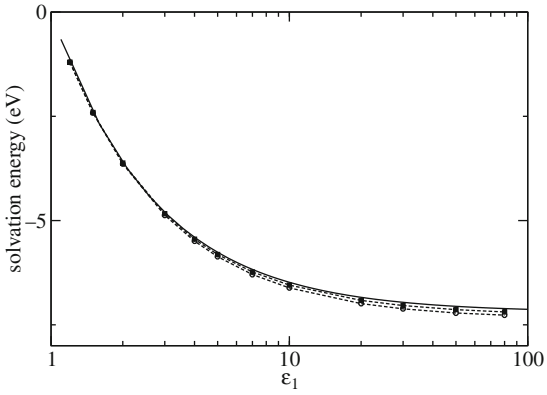


Fig. 15.14 Solvation energy with the boundary element method. A spherical cavity is simulated with a radius $a = 1$ Å which contains a point charge in its center. The solvation energy is calculated with 25×25 (*circles*) and 50×50 (*squares*) surface elements of equal size. The exact expression (15.93) is shown by the *solid curve*

Problems

Problem 15.1 Linearized Poisson–Boltzmann Equation

This computer experiment simulates a homogeneously charged sphere in a dielectric medium. The electrostatic potential is calculated from the linearized Poisson–Boltzmann equation (15.50) 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

$$|\Phi^{(n+1)}(\mathbf{r}) - \Phi^{(n)}(\mathbf{r})|$$

as a measure of convergence (Fig. 15.15).

Explore the dependence of convergence on

- the initial values which can be chosen either $\Phi(\mathbf{r}) = 0$ or from the analytical solution

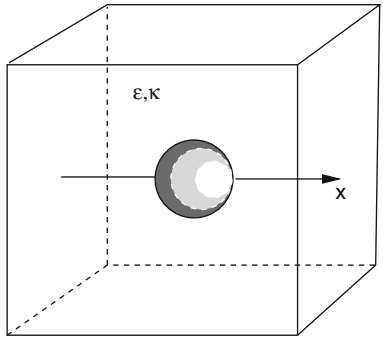


Fig. 15.15 Charged sphere in a dielectric medium

$$\Phi(\mathbf{r}) = \begin{cases} \frac{Q}{8\pi\epsilon\epsilon_0 a}\frac{2+\epsilon(1+\kappa a)}{1+\kappa a} - \frac{Q}{8\pi\epsilon_0 a^3}r^2 & \text{for } r < a \\ \frac{Qe^{-\kappa(r-a)}}{4\pi\epsilon_0\epsilon(\kappa a+1)r} & \text{for } r > a \end{cases}$$

- the relaxation parameter ω for different combinations of ϵ and κ
- the resolution of the grid

Problem 15.2 Boundary Element Method

In this computer element the solvation energy of a point charge within a spherical cavity is calculated with the boundary element method (15.80) (Fig. 15.16).

The calculated solvation energy is compared to the analytical value from (15.91)

$$E_{\text{solv}} = \frac{Q^2}{8\pi\epsilon_0 R}\sum_{n=1}^{\infty}\frac{s^{2n}}{R^{2n}}\frac{(\epsilon_1-\epsilon_2)(n+1)}{n\epsilon_1+(n+1)\epsilon_2} \tag{15.101}$$

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 $\mathrm{d}\phi\mathrm{d}\theta$ or $\mathrm{d}\phi\mathrm{d}\cos\theta$ (equal areas)
- the position of the charge.

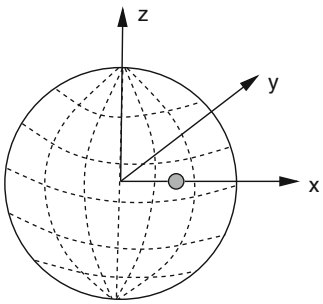


Fig. 15.16 Point charge inside a spherical cavity

Chapter 16
Waves

In this chapter we simulate waves and analyze the numerical stability of simple integration algorithms. We perform computer experiments to study reflection at a boundary or at the border between two media with different refractive indices and we observe the effect of dispersion.

16.1 One-Dimensional Waves

We consider a simple model for one-dimensional longitudinal waves from solid state physics [88] (Fig. 16.1).

The equilibrium position of mass point j is $x_j = j\Delta x$; its elongation from the equilibrium is ξ_j. The potential energy of a spring between mass points j and $j+1$ is

$$\frac{K}{2}\left[\left(\Delta x + \xi_{j+1} - \xi_j\right) - \Delta x\right]^2 = \frac{K}{2}(\xi_{j+1} - \xi_j)^2 \tag{16.1}$$

and the total potential energy is

$$U = \sum \frac{K}{2}(\xi_{j+1} - \xi_j)^2. \tag{16.2}$$

The equation of motion is

$$m\ddot{\xi}_j = -K(\xi_j - \xi_{j-1}) - K(\xi_j - \xi_{j+1}) \tag{16.3}$$

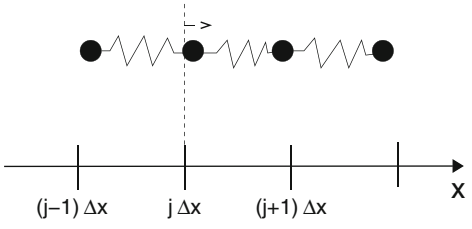


Fig. 16.1 One-dimensional longitudinal waves

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_16,

or

$$\ddot{\xi}_j = \frac{K}{m}(\xi_{j+1} + \xi_{j-1} - 2\xi_j). \tag{16.4}$$

The elongations are described by a continuous function $f(t, x)$ with

$$\xi_j(t) = f(t, j\Delta x). \tag{16.5}$$

The function $f(t, x)$ obeys the equation

$$\ddot{f}(t, j\Delta x) = \frac{K}{m}\left(f(t, j\Delta x - \Delta x) + f(t, j\Delta x + \Delta x) - 2f(t, j\Delta x)\right). \tag{16.6}$$

With the help of the shift operator

$$\mathrm{e}^{\Delta x \frac{\partial}{\partial x}} = \sum_{n=0}^{\infty} \frac{(\Delta x)^n}{n!} \frac{\partial^n}{\partial x^n} \tag{16.7}$$

we have

$$\ddot{f}(t, x) = \frac{K}{m}\left(\mathrm{e}^{\Delta x \frac{\partial}{\partial x}} + \mathrm{e}^{-\Delta x \frac{\partial}{\partial x}} - 2\right) f(t, x) = 2\frac{K}{m}\left(\cosh\left(\Delta x \frac{\partial}{\partial x}\right) - 1\right) \tag{16.8}$$

which is now valid on the whole interval $[0, N\Delta x]$.

For small enough Δx the Taylor series expansion

$$\ddot{f}(t, x) = \frac{K}{m}\left((\Delta x)^2 f''(t, x) + \frac{1}{2}(\Delta x)^4 f^{IV}(t, x) + \ldots\right) \tag{16.9}$$

gives in lowest order the one-dimensional wave equation

$$\frac{\partial^2}{\partial t^2} f = c^2 \frac{\partial^2}{\partial x^2} f, \tag{16.10}$$

where

$$c = \Delta x \sqrt{\frac{K}{m}} \tag{16.11}$$

is the velocity. The general solution of (16.10) according to d'Alembert has the form of waves traveling to the right or to the left with constant envelope and velocity c:

$$f(x, t) = f_+(x - ct) + f_-(x + ct). \tag{16.12}$$

A special solution of this kind is the plane wave solution

$$f(x,t) = \mathrm{e}^{i\omega t \pm ikx} \tag{16.13}$$

with the dispersion relation

$$\omega = ck. \tag{16.14}$$

If higher derivatives are taken into account, the dispersion relation becomes more complicated and (16.12) no longer gives a solution.

16.2 Discretization of the Wave Equation

Using the simplest discretization of the second derivatives we have from (16.10)

$$\begin{aligned} &\frac{f(t+\Delta t,x)+f(t-\Delta t,x)-2f(t,x)}{\Delta t^2} \\ &= c^2\,\frac{f(t,x+\Delta x)+f(t,x-\Delta x)-2f(t,x)}{\Delta x^2}. \end{aligned} \tag{16.15}$$

For a plane wave solution

$$f = \mathrm{e}^{i(\omega t - kx)} \tag{16.16}$$

we find

$$\mathrm{e}^{i\omega\Delta t} + \mathrm{e}^{-i\omega\Delta t} - 2 = c^2\,\frac{\Delta t^2}{\Delta x^2}\left(\mathrm{e}^{ik\Delta x} + \mathrm{e}^{-ik\Delta x} - 2\right) \tag{16.17}$$

which can be written as

$$\sin\frac{\omega\Delta t}{2} = \alpha \sin\frac{k\Delta x}{2} \tag{16.18}$$

with the so-called Courant number [89]

$$\alpha = c\,\frac{\Delta t}{\Delta x}. \tag{16.19}$$

From (16.18) 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. 16.2).

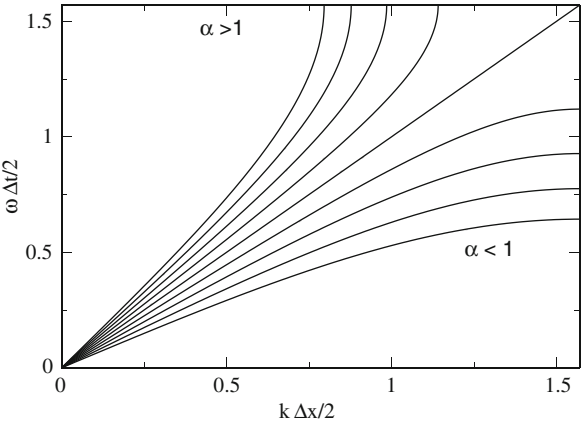


Fig. 16.2 Dispersion of the discrete wave equation. Only for small values of $k\Delta x$ and $\omega\Delta t$ is the dispersion approximately linear. For $\alpha < 1$ only frequencies $\omega < \omega_{\max} = 2\arcsin(\alpha)/\Delta t$ are allowed whereas for $\alpha > 1$ the range of k-values is bounded by $k_{\max} = 2\arcsin(1/\alpha)/\Delta x$

16.3 Boundary Values

The following boundary values can be used for the simulation of waves on a finite grid $x_1 = \Delta x, \ldots, x_N = N\Delta x$:

- fixed boundaries $f(x_0) = 0$ and $f(x_{N+1}) = 0$ (two extra points added)

$$\begin{aligned} \Delta f(x_1) &= \frac{c^2}{\Delta x^2}(f(x_2) - 2f(x_1)) \\ \Delta f(x_N) &= \frac{c^2}{\Delta x^2}(f(x_{N-1}) - 2f(x_N)) \end{aligned} \tag{16.20}$$

- periodic boundary conditions $x_0 \equiv x_N$, $x_{N+1} \equiv x_1$,

$$\begin{aligned} \Delta f(x_1) &= \frac{c^2}{\Delta x^2}(f(x_2) + f(x_N) - 2f(x_1)) \\ \Delta f(x_N) &= \frac{c^2}{\Delta x^2}(f(x_{N-1}) + f(x_1) - 2f(x_N)) \end{aligned} \tag{16.21}$$

- open boundaries

$$\begin{aligned} \Delta f(x_1) &= \frac{c^2}{\Delta x^2}(f(x_2) - f(x_1)) \\ \Delta f(x_N) &= \frac{c^2}{\Delta x^2}(f(x_{N-1}) - f(x_N)) \end{aligned} \tag{16.22}$$

- moving boundaries with given $f(x_0, t) = \xi_0(t)$ or $f(x_{N+1}, t) = \xi_{N+1}(t)$

$$\Delta f(x_1) = \frac{c^2}{\Delta x^2}\left(f(x_2) - 2f(x_1) + \xi_0(t)\right)$$
$$\Delta f(x_N) = \frac{c^2}{\Delta x^2}\left(f(x_{N-1}) - 2f(x_N) + \xi_{N+1}(t)\right) \tag{16.23}$$

16.4 The Wave Equation as an Eigenvalue Problem

16.4.1 Eigenfunction Expansion

We write the general linear wave equation in operator form

$$\frac{\partial^2}{\partial t^2} f = Df \tag{16.24}$$

where for the continuous equation (16.10) the operator D is given by

$$Df = c^2 \nabla^2 f. \tag{16.25}$$

From the eigenvalue problem

$$Df = \lambda f \tag{16.26}$$

we obtain the eigenvalues λ and eigenfunctions f_λ which provide the particular solutions:

$$f = \mathrm{e}^{\pm t\sqrt{\lambda}}\, f_\lambda \tag{16.27}$$
$$\frac{\partial^2}{\partial t^2}(\mathrm{e}^{\pm t\sqrt{\lambda}}\, f_\lambda) = \lambda(\mathrm{e}^{\pm t\sqrt{\lambda}}\, f_\lambda) = D(\mathrm{e}^{\pm t\sqrt{\lambda}}\, f_\lambda). \tag{16.28}$$

These can be used to expand the general solution

$$f(t, x) = \sum_\lambda \left(C_{\lambda+}\mathrm{e}^{t\sqrt{\lambda}} + C_{\lambda-}\mathrm{e}^{-t\sqrt{\lambda}}\right) f_\lambda(x). \tag{16.29}$$

The coefficients $C_{\lambda\pm}$ follow from the initial values by solving the linear equations

$$f(t=0) = \sum_\lambda (C_{\lambda+} + C_{\lambda-}) f_\lambda(x)$$
$$\frac{\partial f}{\partial t}(t=0) = \sum_\lambda \sqrt{\lambda}(C_{\lambda+} - C_{\lambda-}) f_\lambda(x). \tag{16.30}$$

16.4.2 Application to the Discrete One-Dimensional Wave Equation[1]

We consider the discretized second derivative

$$Df = \frac{c^2}{\Delta x^2}\left(f(x+\Delta x) + f(x-\Delta x) - 2f(x)\right). \tag{16.31}$$

x is one of the grid points $x_n = n\Delta x$ with $n = 1, 2, \ldots, N$. The function values are arranged as a column vector:

$$f = \begin{pmatrix} f(\Delta x) \\ \vdots \\ f(N\Delta x) \end{pmatrix}. \tag{16.32}$$

The operator D is represented by the matrix

$$D = \begin{pmatrix} -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{pmatrix} \frac{c^2}{\Delta x^2} \tag{16.33}$$

which can be easily diagonalized since it is tridiagonal. The solutions of the eigenvalue problem

$$Df = \lambda f \tag{16.34}$$

have the form

$$f(n\Delta x) = \sin(nk\Delta x). \tag{16.35}$$

This can be seen by inserting (16.35) into the nth line of (16.34)

$$\begin{aligned}(Df)_n &= \left(\sin\left((n-1)k\Delta x\right) + \sin\left((n+1)k\Delta x\right) - 2\sin(nk\Delta x)\right)\frac{c^2}{\Delta x^2} = \\ &= \left(\sin(nk\Delta x)\cos(k\Delta x) - \cos(nk\Delta x)\sin(k\Delta x) + \sin(nk\Delta x)\cos(k\Delta x)\right. \\ &\quad \left. + \cos(nk\Delta x)\sin(k\Delta x) - 2\sin(nk\Delta x)\right)\frac{c^2}{\Delta x^2} = \\ &= 2\sin(nk\Delta x)\left(\cos(k\Delta x) - 1\right)\frac{c^2}{\Delta x^2} = \lambda\,(f)_n \end{aligned} \tag{16.36}$$

[1] We consider only fixed boundaries here.

with the eigenvalue

$$\lambda = 2\frac{c^2}{\Delta x^2}\left(\cos(k\Delta x) - 1\right). \tag{16.37}$$

The first line of the eigenvalue equation (16.34) gives

$$\begin{aligned}(Df)_1 &= \left(-2\sin(k\Delta x) + \sin(2k\Delta x)\right)\frac{c^2}{\Delta x^2} \\ &= 2\sin(k\Delta x)(\cos(k\Delta x) - 1)\frac{c^2}{\Delta x^2} = \lambda(f)_1\end{aligned} \tag{16.38}$$

and from the last line we have

$$\begin{aligned}(Df)_N &= \left(-2\sin(Nk\Delta x) + \sin((N-1)k\Delta x)\right)\frac{c^2}{\Delta x^2} \\ &= \lambda(f)_N = 2\frac{c^2}{\Delta x^2}(\cos(k\Delta x) - 1)\sin(Nk\Delta x)\end{aligned} \tag{16.39}$$

which holds if

$$\sin((N-1)k\Delta x) = 2\sin(Nk\Delta x)\cos(k\Delta x). \tag{16.40}$$

This simplifies to

$$\begin{aligned}&\sin(Nk\Delta x)\cos(k\Delta x) - \cos(Nk\Delta x)\sin(k\Delta x) = 2\sin(Nk\Delta x)\cos(k\Delta x) \\ &\sin(Nk\Delta x)\cos(k\Delta x) + \cos(Nk\Delta x)\sin(k\Delta x) = 0 \\ &\sin((N+1)k\Delta x) = 0.\end{aligned} \tag{16.41}$$

Hence the possible values of k are

$$k = \frac{\pi}{(N+1)\Delta x}l \text{ with } l = 1, 2, \ldots, N. \tag{16.42}$$

The two boundary points $f(0) = 0$ and $f((N+1)\Delta x) = 0$ can be added without any changes. For other kinds of boundary conditions the following derivations become more complicated.

The eigenvalue can be written as

$$\lambda = 2\frac{c^2}{\Delta x^2}(\cos(k\Delta x) - 1) = -\frac{4c^2}{\Delta x^2}\sin^2\left(\frac{k\Delta x}{2}\right) = (i\omega_k)^2 \tag{16.43}$$

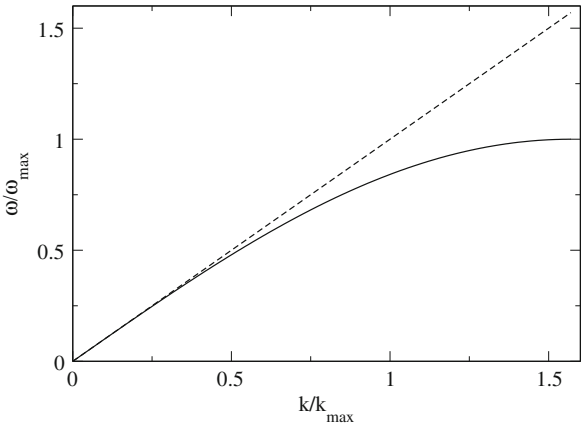


Fig. 16.3 Dispersion of the discrete wave equation

with the frequencies[2] (Fig. 16.3)

$$\omega_k = \frac{2c}{\Delta x} \sin\left(\frac{k\Delta x}{2}\right). \tag{16.44}$$

The general solution has the form

$$f(t, n\Delta x) = \sum_{l=1}^{N} \left(C_{l+} \mathrm{e}^{i\omega_l t} + C_{l-} \mathrm{e}^{-i\omega_l t}\right) \sin\left(n \frac{\pi l}{(N+1)}\right). \tag{16.45}$$

The initial amplitudes and velocities are

$$\begin{aligned} f(t=0, n\Delta x) &= \sum_{l=1}^{N} (C_{l+} + C_{l-}) \sin\left(n \frac{\pi l}{(N+1)}\right) = F_n \\ \dot{f}(t=0, n\Delta x) &= \sum_{l=1}^{N} i\omega_l \ (C_{l+} - C_{l-}) \sin\left(n \frac{\pi l}{(N+1)}\right) = G_n \end{aligned} \tag{16.46}$$

with F_n and G_n given. Different eigenfunctions of a tridiagonal matrix are mutual orthogonal

$$\sum_{n=1}^{N} \sin\left(n \frac{\pi l}{N+1}\right) \sin\left(n \frac{\pi l'}{N+1}\right) = \frac{N}{2} \delta_{l,l'} \tag{16.47}$$

[2] Only for small enough $k\Delta x \ll 1$ the dispersion relation of the continuous wave equation $\omega_k = ck$ follows.

and the coefficients $C_{l\pm}$ follow from a discrete Fourier transformation:

$$\begin{aligned}\widetilde{F}_l &= \frac{1}{N}\sum_{n=1}^{N}\sin\left(n\frac{\pi l}{N+1}\right)F_n \\ &= \frac{1}{N}\sum_{n=1}^{N}\sum_{l'=1}^{N}(C_{l'+}+C_{l'-})\sin\left(n\frac{\pi l'}{N+1}\right)\sin\left(n\frac{\pi l}{N+1}\right) = \frac{1}{2}(C_{l+}+C_{l-})\end{aligned} \tag{16.48}$$

$$\begin{aligned}\widetilde{G}_l &= \frac{1}{N}\sum_{n=1}^{N}\sin\left(n\frac{\pi l}{N+1}\right)G_n \\ &= \frac{1}{N}\sum_{n=1}^{N}\sum_{l'=1}^{N}i\omega_l\,(C_{l+}-C_{l-})\sin\left(n\frac{\pi l'}{N+1}\right)\sin\left(n\frac{\pi l}{N+1}\right) = \frac{1}{2}i\omega_l(C_{l+}-C_{l-})\end{aligned} \tag{16.49}$$

$$\begin{aligned}C_{l+} &= \widetilde{F}_l + \frac{1}{i\omega_l}\widetilde{G}_l \\ C_{l-} &= \widetilde{F}_l - \frac{1}{i\omega_l}\widetilde{G}_l.\end{aligned} \tag{16.50}$$

Finally the explicit solution of the wave equation is

$$f(t,n\Delta x) = \sum_{l=1}^{N}2\left(\widetilde{F}_l\cos(\omega_l\,t)+\frac{\widetilde{G}_l}{\omega_l}\sin(\omega_l\,t)\right)\sin\left(n\frac{\pi l}{N+1}\right). \tag{16.51}$$

16.5 Numerical Integration of the Wave Equation

16.5.1 Simple Algorithm

We solve the discrete wave equation (16.15) with fixed boundaries for $f(t+\Delta t,x)$:

$$f(t+\Delta t,x) = 2f(t,x)(1-\alpha^2)+\alpha^2(f(t,x+\Delta x)+f(t,x-\Delta x))-f(t-\Delta t,x). \tag{16.52}$$

Using the discrete values $x_m = m\Delta x$ and $t_n = n\Delta t$ we have the iteration

$$f(t_{n+1},x_m) = 2(1-\alpha^2)f(t_n,x_m)+\alpha^2 f(t_n,x_{m+1})+\alpha^2 f(t_n,x_{m-1})-f(t_{n-1},x_m). \tag{16.53}$$

This is a two-step method which can be rewritten as a one-step method of double dimension

$$\begin{pmatrix} f_{n+1} \\ f_n \end{pmatrix} = T \begin{pmatrix} f_n \\ f_{n-1} \end{pmatrix} = \begin{pmatrix} 2+\alpha^2 M & -1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} f_n \\ f_{n-1} \end{pmatrix} \tag{16.54}$$

with the column vector

$$f_n = \begin{pmatrix} f(\Delta x) \\ \vdots \\ f(M\Delta x) \end{pmatrix} \tag{16.55}$$

and the tridiagonal matrix

$$M = \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix}. \tag{16.56}$$

The latter has the eigenvalues

$$\lambda = 2\cos(k\Delta x) - 2 = -4\sin^2\left(\frac{k\Delta x}{2}\right). \tag{16.57}$$

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

$$f(t_{n+1}, x_1) = 2(1-\alpha^2) f(t_n, x_1) + \alpha^2 f(t_n, x_2) + \alpha^2 \xi_0(t_n) - f(t_{n-1}, x_1). \tag{16.58}$$

16.5.2 Stability Analysis

Repeated iteration gives the series of function values

$$\begin{pmatrix} f_1 \\ f_0 \end{pmatrix}, \begin{pmatrix} f_2 \\ f_1 \end{pmatrix} = T \begin{pmatrix} f_1 \\ f_0 \end{pmatrix}, \begin{pmatrix} f_3 \\ f_2 \end{pmatrix} = T^2 \begin{pmatrix} f_1 \\ f_0 \end{pmatrix}, \cdots \tag{16.59}$$

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

$$\begin{pmatrix} 2+\alpha^2 M - \sigma & -1 \\ 1 & -\sigma \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{16.60}$$

We substitute the solution of the second equation

$$u = \sigma v \tag{16.61}$$

into the first equation and use the known eigenfunctions of M to have

$$(2 + \alpha^2\lambda - \sigma)\sigma v - v = 0. \tag{16.62}$$

Hence we have to solve

$$\sigma^2 - \sigma(\alpha^2\lambda + 2) + 1 = 0 \tag{16.63}$$

which gives

$$\sigma = 1 + \frac{\alpha^2\lambda}{2} \pm \sqrt{\left(\frac{\alpha^2\lambda}{2} + 1\right)^2 - 1}. \tag{16.64}$$

From

$$-4 < \lambda < 0 \tag{16.65}$$

we have

$$1 - 2\alpha^2 < \frac{\alpha^2\lambda}{2} + 1 < 1 \tag{16.66}$$

and the square root in (16.64) is imaginary if

$$-1 < \frac{\alpha^2\lambda}{2} + 1 < 1 \tag{16.67}$$

which is the case for

$$\sin^2\left(\frac{k\Delta x}{2}\right)\alpha^2 < 1. \tag{16.68}$$

This holds for all k only if

$$|\alpha| < 1. \tag{16.69}$$

But then

$$|\mu|^2 = \left(1 + \frac{\alpha^2\lambda}{2}\right)^2 + \left(1 - \left(\frac{\alpha^2\lambda}{2} + 1\right)^2\right) = 1 \tag{16.70}$$

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

$$1 + \frac{\alpha^2 \lambda}{2} < -1 \tag{16.71}$$

and finally

$$1 + \frac{\alpha^2 \lambda}{2} - \sqrt{\left(1 + \frac{\alpha^2 \lambda}{2}\right)^2 - 1} < -1 \tag{16.72}$$

which shows that instabilities are possible in this case.

16.5.3 Alternative Algorithm with Explicit Velocities

Now let us use a leap frog-like algorithm (page 149):

$$\begin{aligned} f(t_{n+1}, x_m) &= f(t_n, x_m) + v(t_n, x_m)\Delta t + Df(t_n, x_m)\frac{\Delta t^2}{2} \\ &= f(t_n, x_m) + v\left(t_n + \frac{\Delta t}{2}, x_m\right)\Delta t \\ v\left(t_n + \frac{\Delta t}{2}, x_m\right) &= v\left(t_n - \frac{\Delta t}{2}, x_m\right) + Df(t_n, x_m)\Delta t. \end{aligned} \tag{16.73}$$

Since the velocity appears explicitly we can easily add a velocity-dependent damping like

$$-\gamma v(t_n, x_m) \tag{16.74}$$

which we approximate by

$$-\gamma v\left(t_n - \frac{\Delta t}{2}, x_m\right). \tag{16.75}$$

We assume weak damping with

$$\gamma \Delta t \ll 1. \tag{16.76}$$

16.5.4 Stability Analysis

The algorithm can be written in matrix form as

$$\begin{pmatrix} f_{n+1} \\ v_{n+1} \end{pmatrix} = \begin{pmatrix} 1 + \alpha^2 M & \Delta t(1 - \gamma \Delta t) \\ \dfrac{\alpha^2}{\Delta t} M & 1 - \gamma \Delta t \end{pmatrix} \begin{pmatrix} f_n \\ v_n \end{pmatrix}. \tag{16.77}$$

Using the eigenvalues of M

$$\lambda = -4\sin^2\left(\frac{k\Delta x}{2}\right) \tag{16.78}$$

we find the following equation for the eigenvalues σ:

$$\begin{aligned}(1+\alpha^2\lambda-\sigma)u + \Delta t(1-\gamma\Delta t)v = 0\\ \alpha^2\lambda u + \Delta t(1-\gamma\Delta t-\sigma)v = 0.\end{aligned} \tag{16.79}$$

Solving the second equation for u and substituting into the first equation we have

$$\left[(1+\alpha^2\lambda-\sigma)\frac{\Delta t}{-\alpha^2\lambda}(1-\gamma\Delta t-\sigma)+\Delta t(1-\gamma\Delta t)\right]=0 \tag{16.80}$$

hence

$$\begin{aligned}(1+\alpha^2\lambda-\sigma)(1-\gamma\Delta t-\sigma)-\alpha^2\lambda(1-\gamma\Delta t)=0\\ \sigma^2-\sigma(2-\gamma\Delta t+\alpha^2\lambda)+(1-\gamma\Delta t)=0\\ \sigma = 1-\frac{\gamma\Delta t}{2}+\frac{\alpha^2\lambda}{2}\pm\sqrt{\left(1-\frac{\gamma\Delta t}{2}+\frac{\alpha^2\lambda}{2}\right)^2-(1-\gamma\Delta t)}\ .\end{aligned} \tag{16.81}$$

Instabilities are possible if the square root is real and $\sigma < -1$. ($\sigma > 1$ is not possible) (Fig. 16.4). This is the case for

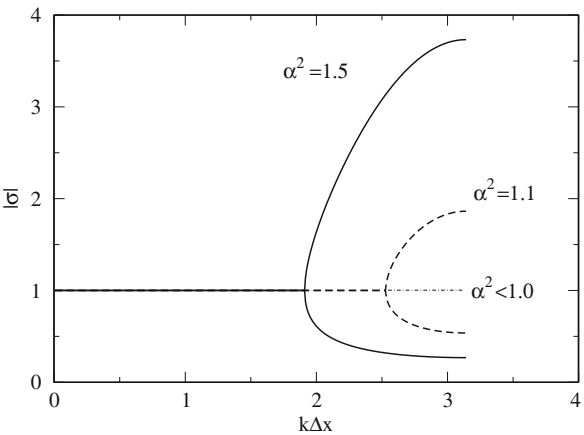


Fig. 16.4 Region of instability. Instabilities appear for $|\alpha| > 1$. One of the two eigenvalues σ becomes unstable ($|\sigma| > 1$) for waves with large k-values

$$-1+\frac{\gamma\Delta t}{2}\approx-\sqrt{1-\gamma\Delta t}<1-\frac{\gamma\Delta t}{2}+\frac{\alpha^2\lambda}{2}<\sqrt{1-\gamma\Delta t}\approx1-\frac{\gamma\Delta t}{2} \tag{16.82}$$

$$-2+\gamma\Delta t<\frac{\alpha^2\lambda}{2}<0. \tag{16.83}$$

The right inequality is satisfied, hence it remains

$$\alpha^2\sin^2\left(\frac{k\Delta x}{2}\right)<1-\frac{\gamma\Delta t}{2}. \tag{16.84}$$

This holds for all k-values if it holds for the maximum of the sine function

$$\alpha^2<1-\frac{\gamma\Delta t}{2}. \tag{16.85}$$

This shows that inclusion of the damping term even favors instabilities.

Problems

Problem 16.1 Waves on a Damped String

In this computer experiment we simulate waves on a string with a moving boundary with the method from Sect. 16.5.3.

- 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 for $x > 0$ can be changed by a factor n (refractive index). Observe reflection at $x = 0$

Problem 16.2 Waves with the Fourier Transform Method

In this computer experiment we use the method from Sect. 16.4.2 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 (16.44).

Chapter 17
Diffusion

Diffusion is one of the simplest non-equilibrium processes. It describes the transport of heat [90, 91] and the time evolution of differences in substance concentrations [92].

In this chapter we consider the diffusion equation

$$\frac{\partial f}{\partial t} = \mathrm{div}(D\ \mathrm{grad}\ f) + S, \tag{17.1}$$

where D is the diffusion constant (which may depend on position) and S is a source term.

17.1 Basic Physics of Diffusion

Let f denote the concentration of a particle species or the temperature. **J** is the corresponding flux of particles. Consider a small cube $\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z$ (Fig. 17.1).

The change of the number of particles within this volume is given by the sum of all incoming and outgoing fluxes

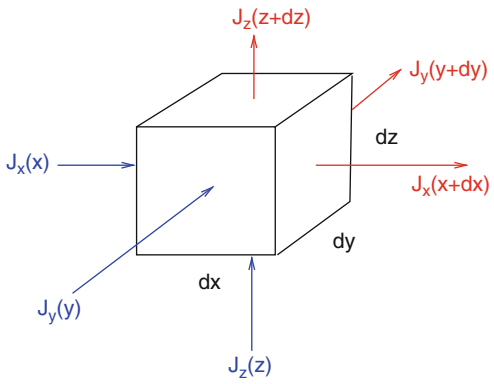


Fig. 17.1 Flux through a volume element $\mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z$

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_17,

$$\begin{aligned}&\frac{\partial f}{\partial t}\mathrm{d}x\mathrm{d}y\mathrm{d}z\\&=(J_x(x,y,z)-J_x(x+\mathrm{d}x,y,z))\mathrm{d}y\mathrm{d}z\\&+(J_y(x,y,z)-J_y(x,y+\mathrm{d}y,z))\mathrm{d}x\mathrm{d}z\\&+(J_z(x,y,z)-J_z(x,y,z+\mathrm{d}z))\mathrm{d}x\mathrm{d}y\end{aligned}\tag{17.2}$$

from which the continuity equation follows:

$$\frac{\partial f}{\partial t}=-\frac{\partial J_x}{\partial x}-\frac{\partial J_y}{\partial y}-\frac{\partial J_z}{\partial z}=-\mathrm{div}\,\mathbf{J}.\tag{17.3}$$

Within the framework of linear response theory the flux is proportional to the gradient of f,

$$\mathbf{J}=-D\ \mathrm{grad}\ f.\tag{17.4}$$

Together we have

$$\mathrm{div}(D\ \mathrm{grad}\ f)=-\mathrm{div}\,\mathbf{J}=\frac{\partial f}{\partial t}.\tag{17.5}$$

Addition of a source (or sink) term completes the diffusion equation. In the special case of constant D it simplifies to

$$\frac{\partial f}{\partial t}=D\Delta f+S.\tag{17.6}$$

17.2 Boundary Conditions

The following choices of boundary conditions are important:

- Dirichlet b.c.: $f(t,x_{\mathrm{bound}})$ given. Can be realized by adding additional points x_{-1} and x_N with given $f(t,x_{-1})$ and $f(t,x_N)$.

- Neumann b.c.: The flux through the boundary is given. Can be realized by adding additional points x_{-1} and x_N with given $f(t,x_{-1})=f(t,x_0)+D^{-1}\Delta x\ j_1(t)$ and $f(t,x_N)=f(t,x_{N-1})-D^{-1}\Delta x\ j_{N-1}(t)$.
- No-flow b.c.: no flux through the boundary. Can be realized by a reflection at the boundary. Additional points x_{-1} and x_N are added with $f(t,x_{-1})=f(t,x_1)$ and $f(t,x_N)=f(t,x_{N-2})$ which compensates the flux through the boundary (Fig. 17.2).

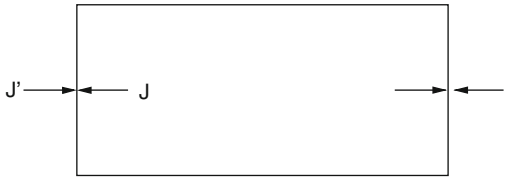


Fig. 17.2 No-flow boundary conditions

17.3 Numerical Integration of the Diffusion Equation

We use discrete values of time and space (one dimensional for now) $t_n = n\Delta t$, $x_m = m\Delta x$, $m = 0, 1, \ldots N-1$ and the discretized derivatives

$$\frac{\partial f}{\partial t} = \frac{f(t_{n+1}, x_m) - f(t_n, x_m)}{\Delta t} \tag{17.7}$$

$$\Delta f = \frac{f(t_n, x_{m+1}) + f(t_n, x_{m-1}) - 2f(t_n, x_m)}{\Delta x^2}. \tag{17.8}$$

17.3.1 Forward Euler or Explicit Richardson Method

A simple Euler step (11.3) is given by

$$f(t_{n+1}, x_m) = f(t_n, x_m) + D\frac{\Delta t}{\Delta x^2}\left(f(t_n, x_{m+1}) + f(t_n, x_{m-1}) - 2f(t_n, x_m)\right) + S(t_n, x_m)\Delta t. \tag{17.9}$$

17.3.2 Stability Analysis

In matrix notation the one-dimensional algorithm with boundary condition $f = 0$ is given by

$$\begin{pmatrix} f(t_{n+1}, x_1) \\ \vdots \\ f(t_{n+1}, x_M) \end{pmatrix} = A \begin{pmatrix} f(t_n, x_1) \\ \vdots \\ f(t_n, x_M) \end{pmatrix} + \begin{pmatrix} S(t_n, x_1)\Delta t \\ \vdots \\ S(t_n, x_M)\Delta t \end{pmatrix} \tag{17.10}$$

with the tridiagonal matrix

$$A = \begin{pmatrix} 1-2D\frac{\Delta t}{\Delta x^2} & D\frac{\Delta t}{\Delta x^2} & & & \\ D\frac{\Delta t}{\Delta x^2} & 1-2D\frac{\Delta t}{\Delta x^2} & & & \\ & \ddots & \ddots & \ddots & \\ & & D\frac{\Delta t}{\Delta x^2} & 1-2D\frac{\Delta t}{\Delta x^2} & D\frac{\Delta t}{\Delta x^2} \\ & & & D\frac{\Delta t}{\Delta x^2} & 1-2D\frac{\Delta t}{\Delta x^2} \end{pmatrix}. \tag{17.11}$$

We use the abbreviation

$$r = D\frac{\Delta t}{\Delta x^2} \tag{17.12}$$

and write A as

$$A = 1 + rM \tag{17.13}$$

with the tridiagonal matrix

$$M = \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix}. \tag{17.14}$$

The eigenvalues of M are (compare (16.43))

$$\lambda = -4\sin^2\left(\frac{k}{2}\right) \text{ with } k = \frac{\pi}{N+1}, \frac{2\pi}{N+1}, \dots \frac{N\pi}{N+1} \tag{17.15}$$

and hence the eigenvalues of A are given by

$$1 + r\lambda = 1 - 4r\sin^2\frac{k}{2}. \tag{17.16}$$

For stability we need

$$|1 + r\lambda| < 1 \quad \text{for all } \lambda \tag{17.17}$$

which holds if

$$-1 < 1 - 4r\sin^2\frac{k}{2} < 1. \tag{17.18}$$

The maximum of the sine function is $\sin\left(\frac{N\pi}{2(N+1)}\right) \approx 1$. Hence the right-hand inequation is fulfilled and from the left one we have

$$-1 < 1 - 4r \tag{17.19}$$

and finally stability for[1]

$$r = D\frac{\Delta t}{\Delta x^2} < \frac{1}{2}. \tag{17.20}$$

17.3.3 Implicit Backward Euler Algorithm

Consider now the implicit method

$$\begin{aligned} f(t_n, x_m) &= f(t_{n+1}, x_m) \\ &- D\frac{\Delta t}{\Delta x^2}(f(t_{n+1}, x_{m+1}) + f(t_{n+1}, x_{m-1}) - 2f(t_{n+1}, x_m)) - S(t_{n+1}, x_m)\Delta t \end{aligned} \tag{17.21}$$

or in matrix notation

$$f(t_n) = Af(t_{n+1}) - S(t_{n+1})\Delta t \quad \text{with } A = 1 - rM \tag{17.22}$$

which can be solved formally by

$$f(t_{n+1}) = A^{-1}f(t_n) + A^{-1}S(t_{n+1})\Delta t. \tag{17.23}$$

The eigenvalues of A are

$$\lambda = 1 + 4r\sin^2\frac{k}{2} > 1 \tag{17.24}$$

and the eigenvalues of A^{-1} are

$$\lambda^{-1} = \frac{1}{1 + r\sin^2\frac{k}{2}}. \tag{17.25}$$

The implicit method is stable since

$$|\lambda^{-1}| < 1. \tag{17.26}$$

[1] $m = \frac{\Delta t}{\Delta x^2}$ is the Courant number [89] for the diffusion equation.

17.3.4 Crank–Nicolson Method

Combination of implicit and explicit method gives the Crank–Nicolson method [93] which is often used for diffusion problems:

$$\begin{aligned}&\frac{f(t_{n+1},x_n)-f(t_n,x_n)}{\Delta t}\\&=D\frac{f(t_n,x_{m+1})+f(t_n,x_{m-1})-2f(t_n,x_m)}{2\Delta x^2}\\&\quad+D\frac{f(t_{n+1},x_{m+1})+f(t_{n+1},x_{m-1})-2f(t_{n+1},x_m)}{2\Delta x^2}\\&\quad+\frac{S(t_n,x_m)+S(t_{n+1},x_m)}{2}\Delta t\end{aligned}\tag{17.27}$$

or in matrix notation

$$\left(1-\frac{r}{2}M\right)f(t_{n+1})=\left(1+\frac{r}{2}M\right)f(t)+\frac{S(t_n)+S(t_{n+1})}{2}\Delta t.\tag{17.28}$$

This can be solved for $f(t_{n+1})$:

$$f(t_{n+1})=\left(1-\frac{r}{2}M\right)^{-1}\left(1+\frac{r}{2}M\right)f(t_n)+\left(1-\frac{r}{2}M\right)^{-1}\frac{S(t_n)+S(t_{n+1})}{2}\Delta t.\tag{17.29}$$

The eigenvalues are now

$$\lambda=\frac{1+\frac{r}{2}\mu}{1-\frac{r}{2}\mu}\text{ with }\mu=-4\sin^2\frac{k}{2}=-4\cdots 0.\tag{17.30}$$

Since $r\mu<0$ it follows

$$1+\frac{r}{2}\mu<1-\frac{r}{2}\mu\tag{17.31}$$

and hence

$$\lambda<1.\tag{17.32}$$

On the other hand we have

$$1>-1\tag{17.33}$$

$$1+\frac{r}{2}\mu>-1+\frac{r}{2}\mu\tag{17.34}$$

$$\lambda>-1.\tag{17.35}$$

which shows that the Crank–Nicolson method is stable [94].

17.3.5 Error Order Analysis

Taylor series expansion of $f(t+\Delta t,x)-f(t,x)$ gives for the explicit method

$$\begin{aligned} &f(t+\Delta t,x)-f(t,x)=rMf(t,x)+S(t,x)\Delta t \\ &=D\frac{\Delta t}{\Delta x^2}\left(f(t,x+\Delta x)+f(t,x-\Delta x)-2f(t,x)\right)+S(t,x)\Delta t. \end{aligned} \tag{17.36}$$

Making use of the diffusion equation we have

$$\begin{aligned} &D\frac{\Delta t}{\Delta x^2}\left(\Delta x^2 f''(t,x)+\frac{\Delta x^4}{12}\frac{\partial^4}{\partial x^4}f(t,x)+\cdots\right)+S(t,x)\Delta t \\ &=\Delta t\dot{f}(t,x)+D\Delta t\frac{\Delta x^2}{12}\frac{\partial^4}{\partial x^4}f(t,x)+\cdots. \end{aligned} \tag{17.37}$$

For the implicit method we find

$$\begin{aligned} &f(t+\Delta t,x)-f(t,x)=rMf(t+\Delta t,x)+S(t+\Delta t,x)\Delta t \\ &=D\frac{\Delta t}{\Delta x^2}\left(f(t+\Delta t,x+\Delta x)+f(t+\Delta t,x-\Delta x)-2f(t+\Delta t,x)\right) \\ &\quad+S(t+\Delta t,x)\Delta t \\ &=D\frac{\Delta t}{\Delta x^2}\left(\Delta x^2 f''(t,x)+\frac{\Delta x^4}{12}f^{(4)}(t,x)+\cdots\right) \\ &\quad+S(t,x)\Delta t+D\frac{\Delta t^2}{\Delta x^2}\left(\Delta x^2\dot{f}''(t,x)+\frac{\Delta x^4}{12}\dot{f}^{(4)}(t,x)+\cdots\right)+\dot{S}(t,x)\Delta t^2 \\ &=\Delta t\dot{f}(t,x)+\Delta t^2\ddot{f}(t,x)+D\Delta t\frac{\Delta x^2}{12}(f^{(4)}(t,x)+\Delta t\dot{f}^{(4)}(t,x))+\cdots. \end{aligned} \tag{17.38}$$

We compare with the exact Taylor series

$$f_{\text{exact}}(t+\Delta t,x)-f(t,x)=\Delta t\dot{f}(t,x)+\frac{\Delta t^2}{2}\ddot{f}(t,x)+\frac{\Delta t^3}{6}\frac{\partial^3}{\partial t^3}f(t,x)\cdots \tag{17.39}$$

and have for the explicit method

$$\begin{aligned} &f_{\text{expl}}(t+\Delta t,x)-f(t,x)=\Delta t\dot{f}(t,x)+\frac{D\Delta x^2\Delta t}{12}f^{(4)}(t,x)+\cdots \\ &=f_{\text{exact}}(t+\Delta t,x)-f(t,x)+O(\Delta t^2,\Delta x^2\Delta t) \end{aligned} \tag{17.40}$$

and for the implicit method

$$f_{\text{impl}}(t+\Delta t,x)-f(t,x)=\Delta t\dot{f}(t,x)+D\Delta t^2\ddot{f}(t,x)+\cdots$$
$$=f_{\text{exact}}(t+\Delta t,x)-f(t,x)+O(\Delta t^2,\Delta x^2\Delta t). \tag{17.41}$$

The error order of the Crank–Nicolson method is higher in Δt:

$$f_{\text{CN}}(t+\Delta t,x)-f(t,x)=\frac{f_{\text{expl}}(t+\Delta t,x)-f(t,x)}{2}+\frac{f_{\text{impl}}(t+\Delta t,x)-f(t,x)}{2}$$
$$=\Delta t\dot{f}(t,x)+\frac{\Delta t^2}{2}\ddot{f}(t,x)+\cdots=f_{\text{exact}}(t+\Delta t,x)-f(t,x)+O(\Delta t^3,\Delta x^2\Delta t). \tag{17.42}$$

17.3.6 Practical Considerations

For the implicit (17.22) and the Crank–Nicolson (17.29) method formally a tridiagonal matrix has to be inverted. However, it is numerically much more efficient to solve the tridiagonal systems of equations:

$$(1-rM)f(t_{n+1})=f(t_n)+S(t_{n+1})\Delta t \tag{17.43}$$
$$\left(1-\frac{r}{2}M\right)f(t_{n+1})=\left(1+\frac{r}{2}M\right)f(t_n)+\frac{S(t_n)+S(t_{n+1})}{2}\Delta t$$

which can be done with the methods discussed in Part I on page 53.

17.3.7 Split Operator Method for $d > 1$ Dimensions

The simplest discretization of the Laplace operator in three dimensions is given by

$$\Delta f=\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}+\frac{\partial^2}{\partial z^2}\right)$$
$$=\frac{1}{\Delta x^2}(\mathrm{d}_x^2+\mathrm{d}_y^2+\mathrm{d}_z^2)f, \tag{17.44}$$

where

$$\frac{1}{\Delta x^2}\mathrm{d}_x^2 f=\frac{f(x+\Delta x,y,z)+f(x-\Delta x,y,z)-2f(x,y,z)}{\Delta x^2}, \tag{17.45}$$

etc., denote the discretized second derivatives. Generalization of the Crank–Nicolson method for the three-dimensional problem gives

$$f(t_{n+1})=\left(1-\frac{r}{2}\mathrm{d}_x^2-\frac{r}{2}\mathrm{d}_y^2-\frac{r}{2}\mathrm{d}_z^2\right)^{-1}\left(1+\frac{r}{2}\mathrm{d}_x^2+\frac{r}{2}\mathrm{d}_y^2+\frac{r}{2}\mathrm{d}_z^2\right)f(t). \tag{17.46}$$

But now the matrices $M_{x,y,z}$ representing the operators $d^2_{x,y,z}$ are not tridiagonal. To keep the advantages of tridiagonal matrices we use the approximations

$$\left(1+\frac{r}{2}\mathrm{d}_x^2+\frac{r}{2}\mathrm{d}_y^2+\frac{r}{2}\mathrm{d}_z^2\right)\approx\left(1+\frac{r}{2}\mathrm{d}_x^2\right)\left(1+\frac{r}{2}\mathrm{d}_y^2\right)\left(1+\frac{r}{2}\mathrm{d}_z^2\right) \tag{17.47}$$

$$\left(1-\frac{r}{2}\mathrm{d}_x^2-\frac{r}{2}\mathrm{d}_y^2-\frac{r}{2}\mathrm{d}_z^2\right)\approx\left(1-\frac{r}{2}\mathrm{d}_x^2\right)\left(1-\frac{r}{2}\mathrm{d}_y^2\right)\left(1-\frac{r}{2}\mathrm{d}_z^2\right) \tag{17.48}$$

and rearrange the factors to obtain

$$\begin{aligned} f(t_{n+1}) = &\left(1-\frac{r}{2}\mathrm{d}_x^2\right)^{-1}\left(1+\frac{r}{2}\mathrm{d}_x^2\right)\left(1-\frac{r}{2}\mathrm{d}_y^2\right)^{-1}\left(1+\frac{r}{2}\mathrm{d}_y^2\right)\left(1-\frac{r}{2}d_z^2\right)^{-1}\\ &\left(1+\frac{r}{2}\mathrm{d}_z^2\right)f(t_n) \end{aligned} \tag{17.49}$$

which represents successive application of the one-dimensional method for the three directions separately. The last step was possible since operators d_i^2 and d_j^2 for different directions $i \neq j$ commute. For instance,

$$\begin{aligned} \mathrm{d}_x^2\mathrm{d}_y^2 f &= \mathrm{d}_x^2(f(x,y+\Delta x)+f(x,y-\Delta x)-2f(x,y))\\ &= f(x+\Delta x,y+\Delta y)+f(x-\Delta x,y+\Delta x)\\ &\quad -2f(x,y+\Delta x)+f(x+\Delta x,y-\Delta x)\\ &\quad +f(x-\Delta x,y-\Delta x)-2f(x,y-\Delta x)\\ &\quad -2f(x+\Delta x,y)-2f(x-\Delta x,y)+4f(x,y)\\ &= \mathrm{d}_y^2\mathrm{d}_x^2 f. \end{aligned} \tag{17.50}$$

The Taylor series of (17.46) and (17.49) coincides up to second order with respect to $r\mathrm{d}^2_{x,y,z}$:

$$\begin{aligned} &\left(1-\frac{r}{2}\mathrm{d}_x^2-\frac{r}{2}\mathrm{d}_y^2-\frac{r}{2}\mathrm{d}_z^2\right)^{-1}\left(1+\frac{r}{2}\mathrm{d}_x^2+\frac{r}{2}\mathrm{d}_y^2+\frac{r}{2}\mathrm{d}_z^2\right)\\ &= 1+r(\mathrm{d}_x^2+\mathrm{d}_y^2+\mathrm{d}_z^2)+\frac{r^2}{2}(\mathrm{d}_x^2+\mathrm{d}_y^2+\mathrm{d}_z^2)^2+O(r^3) \end{aligned} \tag{17.51}$$

$$\begin{aligned} &\left(1-\frac{r}{2}\mathrm{d}_x^2\right)^{-1}\left(1+\frac{r}{2}\mathrm{d}_x^2\right)\left(1-\frac{r}{2}\mathrm{d}_y^2\right)^{-1}\left(1+\frac{r}{2}\mathrm{d}_y^2\right)\left(1-\frac{r}{2}\mathrm{d}_z^2\right)^{-1}\left(1+\frac{r}{2}\mathrm{d}_z^2\right)\\ &= \left(1+r\mathrm{d}_x^2+\frac{r^2\mathrm{d}_x^4}{2}\right)\left(1+r\mathrm{d}_y^2+\frac{r^2\mathrm{d}_y^4}{2}\right)\left(1+r\mathrm{d}_z^2+\frac{r^2\mathrm{d}_z^4}{2}\right)+O(r^3)\\ &= 1+r(\mathrm{d}_x^2+\mathrm{d}_y^2+\mathrm{d}_z^2)+\frac{r^2}{2}(\mathrm{d}_x^2+\mathrm{d}_y^2+\mathrm{d}_z^2)^2+O(r^3). \end{aligned} \tag{17.52}$$

Hence we have

$$
\begin{aligned}
f_{n+1} &= \left(1 + D\Delta t\left(\Delta + \frac{\Delta x^2}{12}\Delta^2 + \cdots\right) + \frac{D^2\Delta t^2}{2}(\Delta^2 + \cdots)\right) f_n \\
&\quad + \left(1 + \frac{D\Delta t}{2}\Delta + \cdots\right)\frac{S_{n+1} + S_n}{2}\Delta t \\
&= f_n + \Delta t(D\Delta f_n + S_n) + \frac{\Delta t^2}{2}(D^2\Delta^2 + D\Delta S_n + \dot{S}_n) + O(\Delta t\Delta x^2, \Delta t^3).
\end{aligned}
\tag{17.53}
$$

and the error order is conserved by the split operator method.

Problems

Problem 17.1 Diffusion in Two 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 methods.

Chapter 18
Nonlinear Systems

Nonlinear problems [95, 96] 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 [97]

$$x_{n+1} = f(x_n) \tag{18.1}$$

systems of ordinary differential equations like population dynamics models [98–100]

$$\begin{aligned} \dot{x}(t) &= f(x, y) \\ \dot{y}(t) &= g(x, y) \end{aligned} \tag{18.2}$$

or partial differential equations like the reaction diffusion equation [99–102]

$$\frac{\partial}{\partial t} c(x, t) = D \frac{\partial^2}{\partial x^2} c(x, t) + f(c), \tag{18.3}$$

where f and g are nonlinear in the mathematical sense that means they satisfy both the following properties

$$\begin{array}{ll} \textit{additivity} & f(x + y) = f(x) + f(y) \\ \textit{homogeneity} & f(\alpha x) = \alpha f(x). \end{array} \tag{18.4}$$

18.1 Iterated Functions

Starting from an initial value x_0 a function f is iterated repeatedly

$$\begin{aligned} x_1 &= f(x_0) \\ x_2 &= f(x_1) \\ &\vdots \\ x_{i+1} &= f(x_i). \end{aligned} \tag{18.5}$$

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_18,

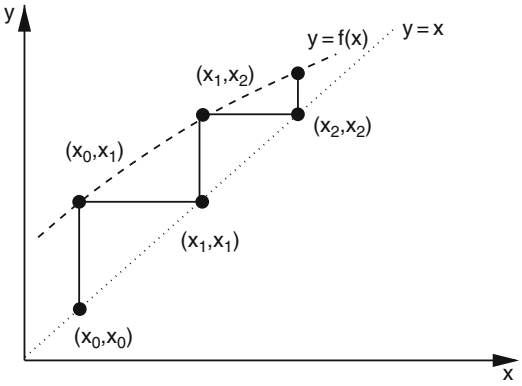


Fig. 18.1 Orbit of an iterated function. The sequence of points (x_i, x_{i+1}), (x_{i+1}, x_{i+1}) is plotted together with the curves $y = f(x)$ (*dashed*) and $y = x$ (*dotted*)

The sequence of function values $x_0, x_1 \cdots$ is called the orbit of x_0. It can be visualized in a two-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. 18.1).

18.1.1 Fixed Points and Stability

If the equation

$$x^* = f(x^*) \tag{18.6}$$

has solutions x^*, then these are called fixed points. Consider a point in the vicinity of a fixed point

$$x = x^* + \varepsilon_0 \tag{18.7}$$

and make a Taylor series expansion

$$f(x) = f(x^* + \varepsilon_0) = f(x^*) + \varepsilon_0 f'(x^*) + \cdots = x^* + \varepsilon_1 + \cdots \tag{18.8}$$

with the notation

$$\varepsilon_1 = \varepsilon_0 f'(x^*). \tag{18.9}$$

Repeated iteration gives[1]

[1] Here and in the following $f^{(n)}$ denotes an iterated function, not a derivative.

$$f^{(2)}(x) = f(f(x)) = f(x^* + \varepsilon_1) + \cdots = x^* + \varepsilon_1 f'(x^*) = x^* + \varepsilon_2$$
$$\vdots$$
$$f^{(n)}(x^*) = x^* + \varepsilon_n \tag{18.10}$$

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. 18.2).

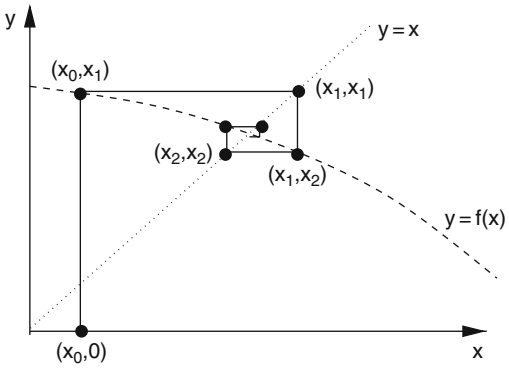


Fig. 18.2 Attractive fixed point. The orbit of an attractive fixed point converges to the intersection of the curves $y = x$ and $y = f(x)$

Higher order fixed points are defined by iterating $f(x)$ several times. A nth order fixed point solves

$$\begin{aligned} f(x^*) &\neq x^* \\ f^{(2)}(x^*) &\neq x^* \\ f^{(n-1)}(x^*) &\neq x^* \\ f^{(n)}(x^*) &= x^*. \end{aligned} \tag{18.11}$$

The iterated function values cycle periodically through (Fig. 18.3)

$$x^* \to f(x^*) \to f^{(2)}(x^*) \cdots f^{(n-1)}(x^*).$$

This period is attractive if

$$|f'(x^*)\, f'(f(x^*))\, f'(f^{(2)}(x^*)) \cdots f'(f^{(n-1)}(x^*))| < 1. \tag{18.12}$$

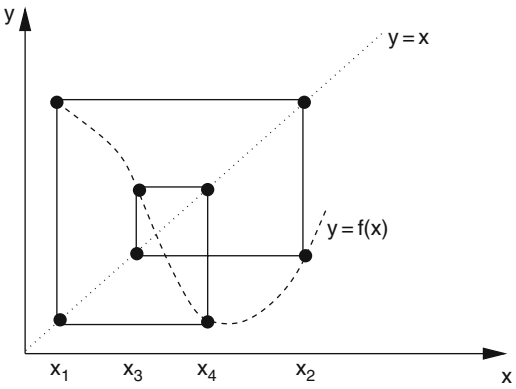


Fig. 18.3 Periodic orbit. The orbit of an attractive fourth-order fixed point cycles through the values $x_1 = f(x_4), x_2 = f(x_1), x_3 = f(x_2), x_4 = f(x_3)$

18.1.2 The Ljapunow Exponent

Consider two neighboring orbits with initial values x_0 and $x_0+\varepsilon_0$. After n iterations the distance is

$$|f(f(\cdots f(x_0))) - f(f(\cdots f(x_0+\varepsilon_0)))| = |\varepsilon_0| e^{\lambda n} \tag{18.13}$$

with the so-called Ljapunow exponent [103] λ which is useful to characterize the orbit. The Ljapunow exponent can be determined from

$$\lambda = \lim_{n\to\infty} \frac{1}{n} ln \left(\frac{|f^{(n)}(x_0+\varepsilon_0) - f^{(n)}(x_0)|}{|\varepsilon_0|} \right) \tag{18.14}$$

or numerically easier with the approximation

$$\begin{aligned} |f(x_0+\varepsilon_0) - f(x_0)| &= |\varepsilon_0||f'(x_0)| \\ |f(f(x_0+\varepsilon_0)) - f(f(x_0))| &= |(f(x_0+\varepsilon_0) - f(x_0))||f'(x_0+\varepsilon_0)| \\ &= |\varepsilon_0||f'(x_0)||f'(x_0+\varepsilon_0)| \qquad (18.15) \\ |f^{(n)}(x_0+\varepsilon_0) - f^{(n)}(x_0)| &= |\varepsilon_0||f'(x_0)||f'(x_1)|\cdots|f'(x_{n-1})| \qquad (18.16) \end{aligned}$$

from

$$\lambda = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)|. \tag{18.17}$$

For a stable fixed point

$$\lambda \to \ln |f'(x^*)| < 0 \tag{18.18}$$

and for an attractive period

$$\lambda \to \ln |f'(x^*)\, f'(f(x^*)\, \cdots f'(f^{(n-1)}(x^*))| < 0. \tag{18.19}$$

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.

18.1.3 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

$$N_{n+1} = r\, N_n, \tag{18.20}$$

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 [104]. He assumed that the reproduction rate r depends on the population density N in a simple way (Fig. 18.4)

$$r = r_0 \left(1 - \frac{N}{K}\right). \tag{18.21}$$

The Verhulst model (18.21) leads to the iterated nonlinear function

$$N_{n+1} = r_0 N_n - \frac{r_0}{K} N_n^2 \tag{18.22}$$

with $r_0 > 0$, $K > 0$. We denote the quotient of population density and carrying capacity by the new variable

$$x_n = \frac{1}{K} N_n \tag{18.23}$$

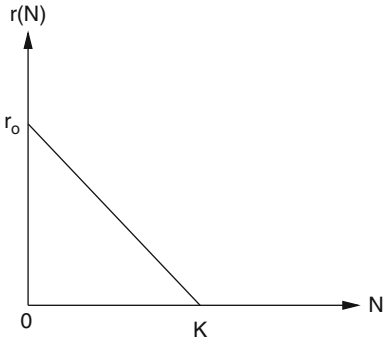


Fig. 18.4 Reproduction rate of the logistic model. At low densities the growth rate has its maximum value r_0. At larger densities the growth rate declines and reaches $r = 0$ for $N = K$. The parameter K is called carrying capacity

and obtain an equation with only one parameter, the so-called logistic mapping

$$x_{n+1} = \frac{1}{K} N_{n+1} = \frac{1}{K} r_0 N_n \left(1 - \frac{N_n}{K}\right) = r_0\, x_n\, (1 - x_n). \tag{18.24}$$

18.1.4 Fixed Points of the Logistic Map

Consider an initial point in the interval

$$0 < x_0 < 1. \tag{18.25}$$

We want to find conditions on r to keep the orbit in this interval. The maximum value of x_{n+1} is found from

$$\frac{\mathrm{d}x_{n+1}}{\mathrm{d}x_n} = r(1 - 2x_n) = 0 \tag{18.26}$$

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

$$\frac{|x_{n+1}|}{|x_n|} = |r|(1 + |x_n|) > 1. \tag{18.27}$$

The fixed point equation

$$x^* = rx^* - rx^{*2} \tag{18.28}$$

always has the trivial solution

$$x^* = 0 \tag{18.29}$$

and a further solution

$$x^* = 1 - \frac{1}{r} \tag{18.30}$$

which is only physically reasonable for $r > 1$, since x should be a positive quantity. For the logistic mapping the derivative is

$$f'(x) = r - 2rx \tag{18.31}$$

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 $r < 1$

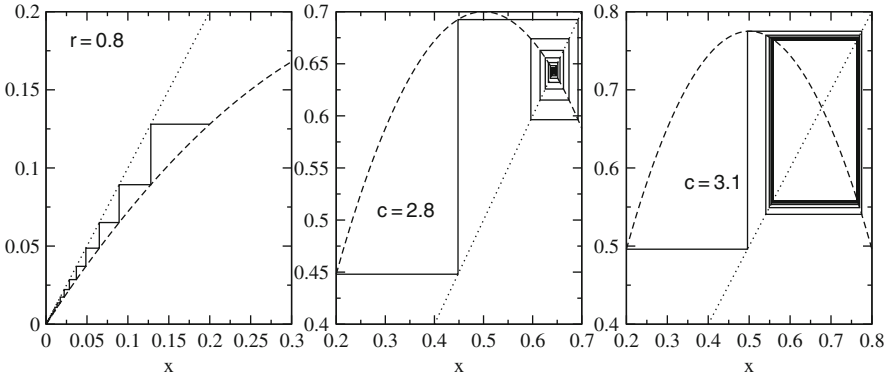


Fig. 18.5 Orbits of the logistic map. *left:* For $0 < r < 1$ the logistic map has the attractive fixed point $x^* = 0$. *middle:* In the region $1 < r < 3$ this fixed point becomes unstable and another stable fixed point is at $x^* = 1 - 1/r$. *right:* For $3 < r < 1 + \sqrt{6}$ the second-order fixed point (18.33) is stable. For larger values of r more and more bifurcations appear

no such fixed point exists. For $r_1 = 3$ the first bifurcation appears and higher order fixed points become stable (Fig. 18.5).

Consider the fixed point of the double iteration

$$x^* = r(r(x^* - x^{*^2}) - r^2(x^* - x^{*2})^2). \tag{18.32}$$

All roots of this fourth-order equation can be found since we already know two of them. The remaining roots are

$$x^*_{1,2} = \frac{\frac{r+1}{2} \pm \sqrt{r^2 - 2r - 3}}{r} \tag{18.33}$$

They are real valued if

$$(r-1)^2 - 4 > 0 \rightarrow r > 3 \quad (\text{or} \quad r < -1). \tag{18.34}$$

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.

18.1.5 Bifurcation Diagram

The bifurcation diagram visualizes the appearance of period doubling and chaotic behavior as a function of the control parameter r (Fig. 18.6).

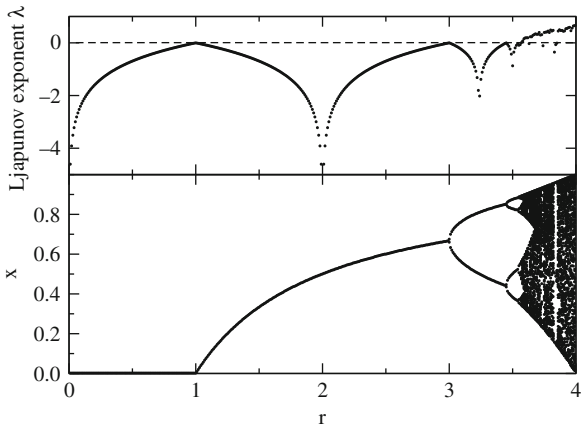


Fig. 18.6 Bifurcation diagram of the logistic map. For different values of r the function is iterated 1100 times. The first 1000 iterations are dropped to allow the trajectory to approach stable fixed points or periods. The iterated function values $x_{1000} \cdots x_{1100}$ are plotted in a r–x diagram together with the estimate (18.17) of the Ljapunow exponent. The first period doublings appear at $r = 3$ and $r = 1 + \sqrt{6}$. For larger values chaotic behavior is observed and the estimated Ljapunow exponent becomes positive. In some regions motion is regular again with negative Ljapunow exponent

18.2 Population Dynamics

If time is treated as a continuous variable, the iterated function has to be replaced by a differential equation

$$\frac{\mathrm{d}N}{\mathrm{d}t} = f(N) \tag{18.35}$$

or more generally by a system of equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} N_1 \\ N_2 \\ \vdots \\ N_n \end{pmatrix} = \begin{pmatrix} f_1(N_1 \cdots N_n) \\ f_2(N_1 \cdots N_n) \\ \\ f_n(N_1 \cdots N_n) \end{pmatrix}. \tag{18.36}$$

18.2.1 Equilibria and Stability

The role of the fixed points is now taken over by equilibria, which are solutions of

$$0 = \frac{\mathrm{d}N}{\mathrm{d}t} = f(N_{\mathrm{eq}}) \tag{18.37}$$

which means roots of $f(N)$. Let us investigate small deviations from equilibrium with the help of a Taylor series expansion. Inserting

$$N = N_{\mathrm{eq}} + \xi \tag{18.38}$$

we obtain

$$\frac{\mathrm{d}\xi}{\mathrm{d}t} = f(N_{\mathrm{eq}}) + f'(N_{\mathrm{eq}})\xi + \cdots \tag{18.39}$$

but since $f(N_{\mathrm{eq}}) = 0$, we have approximately

$$\frac{\mathrm{d}\xi}{\mathrm{d}t} = f'(N_{\mathrm{eq}})\xi \tag{18.40}$$

with the solution

$$\xi(t) = \xi_0 \exp\left\{f'(N_{\mathrm{eq}})t\right\}. \tag{18.41}$$

The equilibrium is only stable if $\mathfrak{Re}\, f'(N_{\mathrm{eq}}) < 0$, since then small deviations disappear exponentially. For $\mathfrak{Re}\, f'(N_{\mathrm{eq}}) > 0$ deviations will increase, but the exponential behavior holds only for not too large deviations and saturation may appear. If the derivative $f'(N_{\mathrm{eq}})$ has a nonzero imaginary part then oscillations will be superimposed. For a system of equations the equilibrium is defined by

$$\begin{pmatrix} f_1(N_1^{\mathrm{eq}} \cdots N_n^{\mathrm{eq}}) \\ f_2(N_1^{\mathrm{eq}} \cdots N_n^{\mathrm{eq}}) \\ \\ f_N(N_1^{\mathrm{eq}} \cdots N_n^{\mathrm{eq}}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{18.42}$$

and if such an equilibrium exists, linearization gives

$$\begin{pmatrix} N_1 \\ N_2 \\ \vdots \\ N_n \end{pmatrix} = \begin{pmatrix} N_1^{\mathrm{eq}} \\ N_2^{\mathrm{eq}} \\ \vdots \\ N_n^{\mathrm{eq}} \end{pmatrix} + \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{pmatrix} \tag{18.43}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_N \end{pmatrix} = \begin{pmatrix} \dfrac{\partial f_1}{\partial N_1} & \dfrac{\partial f_1}{\partial N_2} & \cdots & \dfrac{\partial f_1}{\partial N_n} \\ \dfrac{\partial f_2}{\partial N_1} & \dfrac{\partial f_2}{\partial N_2} & \cdots & \dfrac{\partial f_2}{\partial N_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial f_n}{\partial N_1} & \dfrac{\partial f_n}{\partial N_2} & \cdots & \dfrac{\partial f_n}{\partial N_n} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{pmatrix}. \tag{18.44}$$

The equilibrium is stable if all eigenvalues λ_i of the derivative matrix have a negative real part.

18.2.2 The Continuous Logistic Model

The continuous logistic model describes the evolution by the differential equation

$$\frac{\mathrm{d}x}{\mathrm{d}t} = r_0 x(1-x). \tag{18.45}$$

To find possible equilibria we have to solve (Fig. 18.7)

$$x_{\mathrm{eq}}(1-x_{\mathrm{eq}}) = 0 \tag{18.46}$$

which has the two roots $x_{\mathrm{eq}} = 0$ and $x_{\mathrm{eq}} = 1$.

The derivative f' is

$$f'(x) = \frac{\mathrm{d}}{\mathrm{d}x}\left(r_0 x(1-x)\right) = r_0(1-2x). \tag{18.47}$$

Since $f'(0) = r_0 > 0$ and $f'(1) = -r_0 < 0$ only the second equilibrium is stable.

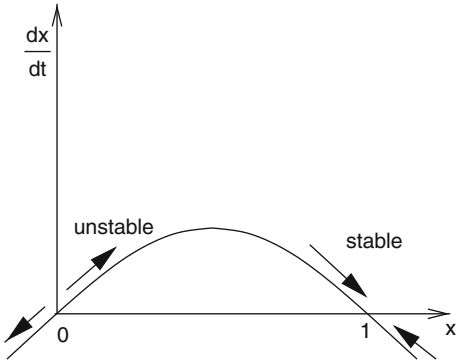


Fig. 18.7 Equilibria of the logistic model. The equilibrium $x_{\mathrm{eq}} = 0$ is unstable since an infinitesimal deviation grows exponentially in time. The equilibrium $x_{\mathrm{eq}} = 1$ is stable since initial deviations disappear exponentially

18.3 Lotka–Volterra model

The model by Lotka [105] and Volterra [106] 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

$$\frac{\mathrm{d}N}{\mathrm{d}t} = (r-m)N \tag{18.48}$$

both of which 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

$$m_{\mathrm{H}} = aP \quad r_P = bH, \tag{18.49}$$

where a is the predation rate coefficient and b is the reproduction rate of predators per one prey eaten. Together we end up with a system of two coupled nonlinear differential equations

$$\begin{aligned} \frac{\mathrm{d}H}{\mathrm{d}t} &= f(H, P) = r_{\mathrm{H}} H - aHP \\ \frac{\mathrm{d}P}{\mathrm{d}t} &= g(H, P) = bHP - m_{\mathrm{P}} P, \end{aligned} \tag{18.50}$$

where r_{H} is the intrinsic rate of prey population increase and m_{P} the predator mortality rate.

18.3.1 Stability Analysis

To find equilibria we have to solve the system of equations

$$\begin{aligned} f(H, P) &= r_H H - aHP = 0 \\ g(H, P) &= bHP - m_P P = 0. \end{aligned} \tag{18.51}$$

The first equation is solved by $H_{\mathrm{eq}} = 0$ or by $P_{\mathrm{eq}} = r_{\mathrm{H}}/a$. The second equation is solved by $P_{\mathrm{eq}} = 0$ or by $H_{\mathrm{eq}} = m_{\mathrm{P}}/b$. Hence there are two equilibria, the trivial one

$$P_{\mathrm{eq}} = H_{\mathrm{eq}} = 0 \tag{18.52}$$

and a nontrivial one

$$P_{\mathrm{eq}} = \frac{r_{\mathrm{H}}}{a} \quad H_{\mathrm{eq}} = \frac{m_{\mathrm{P}}}{b}. \tag{18.53}$$

Linearization around the zero equilibrium gives

$$\frac{\mathrm{d}H}{\mathrm{d}t} = r_{\mathrm{H}} H + \cdots \quad \frac{\mathrm{d}P}{\mathrm{d}t} = -m_{\mathrm{P}} P + \cdots \tag{18.54}$$

This equilibrium is unstable since a small prey population will increase exponentially. Now expand around the nontrivial equilibrium:

$$P = P_{\mathrm{eq}} + \xi, \quad H = H_{\mathrm{eq}} + \eta \tag{18.55}$$

$$\frac{\mathrm{d}\eta}{\mathrm{d}t} = \frac{\partial f}{\partial H}\eta + \frac{\partial f}{\partial P}\xi = (r_{\mathrm{H}} - aP_{\mathrm{eq}})\eta - aH_{\mathrm{eq}}\xi = -\frac{am_{\mathrm{P}}}{b}\xi \tag{18.56}$$

$$\frac{\mathrm{d}\xi}{\mathrm{d}t} = \frac{\partial g}{\partial H}\eta + \frac{\partial g}{\partial P}\xi = bP_{\mathrm{eq}}\eta + (bH_{\mathrm{eq}} - m_{\mathrm{P}})\xi = \frac{br_{\mathrm{H}}}{a}\eta \tag{18.57}$$

or in matrix notation

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\eta\\ \xi\end{pmatrix} = \begin{pmatrix}0 & -\frac{am_{\mathrm{P}}}{b}\\ \frac{br_{\mathrm{H}}}{a} & 0\end{pmatrix}\begin{pmatrix}\eta\\ \xi\end{pmatrix}. \tag{18.58}$$

The eigenvalues are purely imaginary

$$\lambda = \pm\mathrm{i}\sqrt{m_{\mathrm{H}}r_{\mathrm{P}}} = \pm\mathrm{i}\omega \tag{18.59}$$

and the corresponding eigenvectors are

$$\begin{pmatrix}\mathrm{i}\sqrt{m_{\mathrm{H}}r_p}\\ br_{\mathrm{H}}/a\end{pmatrix}, \begin{pmatrix}am_{\mathrm{P}}/b\\ \mathrm{i}\sqrt{m_{\mathrm{H}}r_{\mathrm{P}}}\end{pmatrix}. \tag{18.60}$$

The solution of the linearized equations is then given by

$$\begin{aligned}\xi(t) &= \xi_0\cos\omega t + \frac{b}{a}\sqrt{\frac{r_{\mathrm{P}}}{m_{\mathrm{H}}}}\eta_0\sin\omega t\\ \eta(t) &= \eta_0\cos\omega t - \frac{a}{b}\sqrt{\frac{m_{\mathrm{H}}}{r_{\mathrm{P}}}}\xi_0\sin\omega t\end{aligned} \tag{18.61}$$

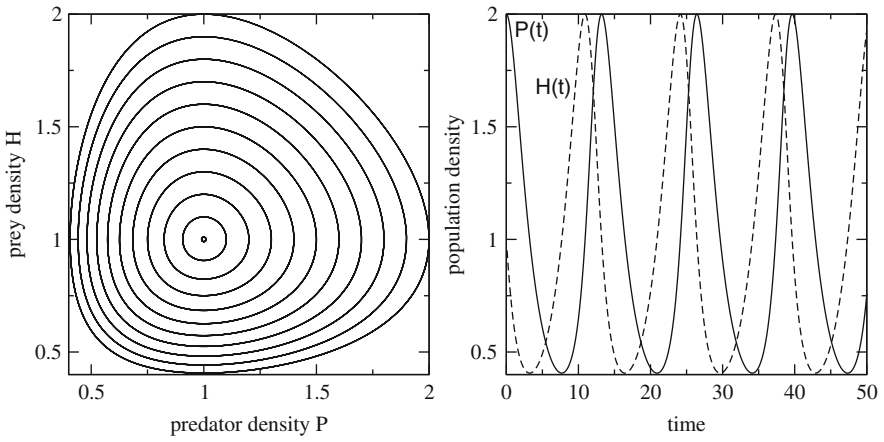


Fig. 18.8 Lotka–Volterra model. The predator and prey population densities show periodic oscillations (*right*). In the H–P plane the system moves on a *closed curve*, which becomes an ellipse for small deviations from equilibrium (*left*)

which describes an ellipse in the $\xi - \eta$ plane (Fig. 18.8). The nonlinear equations (18.51) have a first integral

$$r_{\mathrm{H}} \ln P(t) - a\, P(t) - b\, H(t) + m_{\mathrm{P}} \ln H(t) = C \tag{18.62}$$

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.

18.4 Functional Response

Holling [107, 108] 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

$$T = T_{\mathrm{search}} + T_{\mathrm{handling}}. \tag{18.63}$$

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 $\alpha T_{\mathrm{search}}$ and captured $H_{\mathrm{T}} = H\alpha T_{\mathrm{search}}$ prey. Hence the predation rate is

$$a = \frac{H_{\mathrm{T}}}{HT} = \alpha \frac{T_{\mathrm{search}}}{T} = \alpha \frac{1}{1 + T_{\mathrm{handling}}/T_{\mathrm{search}}}. \tag{18.64}$$

The handling time is assumed to be proportional to the number of prey captured

$$T_{\mathrm{handling}} = T_{\mathrm{h}} H \alpha T_{\mathrm{search}}, \tag{18.65}$$

where T_{h} is the handling time spent per one prey. The predation rate then is given by (Fig. 18.9)

$$a = \frac{\alpha}{1 + \alpha H T_{\mathrm{h}}}. \tag{18.66}$$

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 $a_\infty = \frac{1}{HT_{\mathrm{h}}}$.

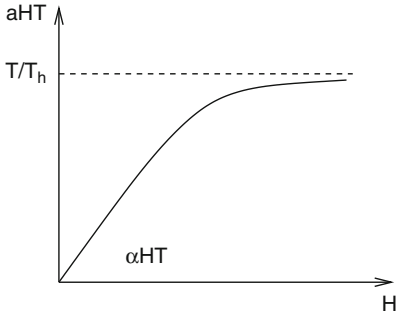


Fig. 18.9 Functional response of Holling's model

18.4.1 Holling–Tanner Model

We combine the logistic model with Holling's model for the predation rate [107, 108, 110]

$$\begin{aligned}\frac{\mathrm{d}H}{\mathrm{d}t} &= r_\mathrm{H} H\left(1-\frac{H}{K_\mathrm{H}}\right) - aHP \\ &= r_\mathrm{H} H\left(1-\frac{H}{K_\mathrm{H}}\right) - \frac{\alpha}{1+\alpha H T_\mathrm{h}} HP = f(H,P)\end{aligned} \tag{18.67}$$

and assume that the carrying capacity of the predator is proportional to the density of prey (Fig. 18.11)

$$\frac{\mathrm{d}P}{\mathrm{d}t} = r_\mathrm{P} P\left(1-\frac{P}{K_\mathrm{P}}\right) = r_\mathrm{P} P\left(1-\frac{P}{kH}\right) = g(H,P). \tag{18.68}$$

Obviously there is a trivial equilibrium with $P_\mathrm{eq} = H_\mathrm{eq} = 0$. Linearization gives

$$\frac{\mathrm{d}H}{\mathrm{d}t} = r_\mathrm{H} H + \cdots \quad \frac{\mathrm{d}P}{\mathrm{d}t} = r_\mathrm{P} P + \cdots \tag{18.69}$$

which shows that this equilibrium is unstable. There is another trivial equilibrium with $P_\mathrm{eq} = 0, \quad H_\mathrm{eq} = K_\mathrm{H}$. Here we find

$$\begin{aligned}&\frac{\mathrm{d}H}{\mathrm{d}t} = r_\mathrm{H}(K_\mathrm{H}+h)(1-\frac{K_\mathrm{H}+h}{K_\mathrm{H}}) - \frac{\alpha}{1+\alpha H T_\mathrm{h}} K_\mathrm{H} p = r_\mathrm{H} h - \frac{\alpha}{1+\alpha H T_\mathrm{h}} K_\mathrm{H} p \\ &\frac{\mathrm{d}P}{\mathrm{d}t} = r_\mathrm{P} P \\ &\begin{pmatrix}\dot{h}\\ \dot{p}\end{pmatrix} = \begin{pmatrix} r_H & -\frac{\alpha}{1+\alpha H T_\mathrm{h}} K_\mathrm{H} \\ 0 & r_\mathrm{P}\end{pmatrix}\begin{pmatrix}h\\ p\end{pmatrix} \\ &\lambda = \frac{r_\mathrm{H}+r_\mathrm{P}}{2} \pm \frac{1}{2}\sqrt{(r_\mathrm{H}-r_\mathrm{P})^2} = r_\mathrm{H}, r_\mathrm{P}.\end{aligned} \tag{18.70}$$

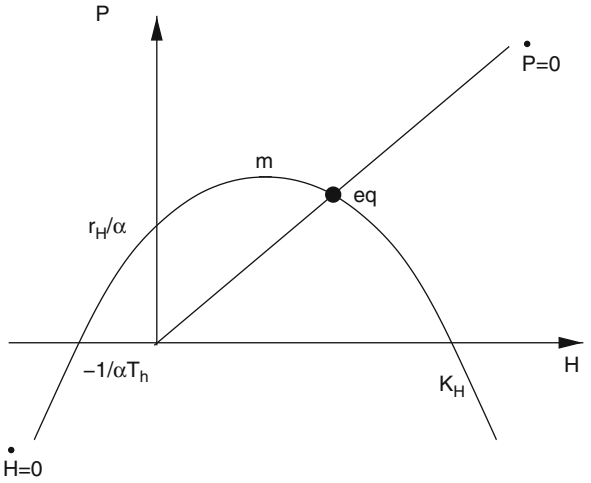


Fig. 18.10 Nullclines of the predator–prey model

Let us now look for nontrivial equilibria. The nullclines are the curves defined by $\frac{\mathrm{d}H}{\mathrm{d}t} = 0$ and $\frac{\mathrm{d}P}{\mathrm{d}t} = 0$, hence by (Fig. 18.10)

$$P = \frac{r_\mathrm{H}}{\alpha}\left(1 - \frac{H}{K_\mathrm{H}}\right)(1 + \alpha H T_\mathrm{h}) \tag{18.71}$$

$$P = kH. \tag{18.72}$$

The H-nullcline is a parabola at

$$H_\mathrm{m} = \frac{\alpha T_\mathrm{h} - K_\mathrm{H}^{-1}}{2\alpha T_\mathrm{h} K_\mathrm{H}^{-1}} \quad P_\mathrm{m} = \frac{(\alpha T_\mathrm{h} + K_\mathrm{H}^{-1})^2}{4\alpha T_\mathrm{h} K_\mathrm{H}^{-1}} > 0. \tag{18.73}$$

It intersects the H-axis at $H = K_\mathrm{H}$ and $H = -1/\alpha T_\mathrm{h}$ and the P-axis at $P = r_\mathrm{H}/\alpha$. 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

$$r_\mathrm{H} a T_\mathrm{h} H_\mathrm{eq}^2 + \left(r_\mathrm{H} + a K_\mathrm{p} K_\mathrm{H} - r_\mathrm{H} K_\mathrm{H} a T_\mathrm{h}\right) H_\mathrm{eq} - r_\mathrm{H} K_\mathrm{H} = 0. \tag{18.74}$$

It is explicitly given by

$$H_\mathrm{eq} = -\frac{r_\mathrm{H} + a K_\mathrm{p} K_\mathrm{H} - r_\mathrm{H} K_\mathrm{H} a T_\mathrm{h}}{2 r_\mathrm{H} a T_\mathrm{h}} + \frac{\sqrt{\left(r_\mathrm{H} + a K_\mathrm{p} K_\mathrm{H} - r_\mathrm{H} K_\mathrm{H} a T_\mathrm{h}\right)^2 + 4 r_\mathrm{H} a T_\mathrm{h} r_\mathrm{H} K_\mathrm{H}}}{2 r_\mathrm{H} a T_\mathrm{h}}. \tag{18.75}$$

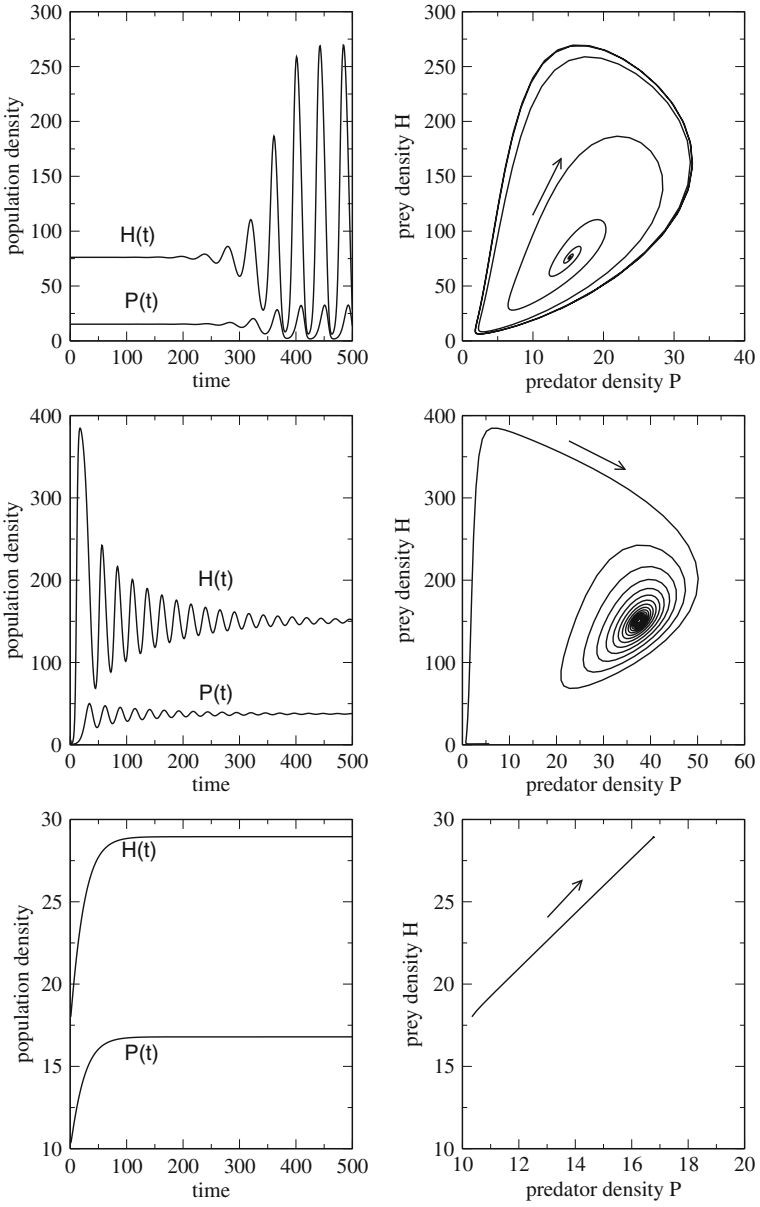


Fig. 18.11 Holling–Tanner model. *top:* evolution from an unstable equilibrium to a limit cycle, *middle:* a stable equilibrium is approached with oscillations, *bottom:* stable equilibrium without oscillations

The prey density then follows from

$$P_{\mathrm{eq}} = H_{\mathrm{eq}} K_{\mathrm{P}}. \tag{18.76}$$

The matrix of derivatives has the elements

$$\begin{aligned}
m_{\mathrm{hp}} &= \frac{\partial f}{\partial P} = -\frac{a H_{\mathrm{eq}}}{1 + a T_{\mathrm{h}} H_{\mathrm{eq}}} \\
m_{\mathrm{hh}} &= \frac{\partial f}{\partial H} = r_{\mathrm{H}} \left(1 - 2\frac{H_{\mathrm{eq}}}{K_{\mathrm{h}}}\right) - \frac{a K_{\mathrm{p}} H_{\mathrm{eq}}}{1 + a T_{\mathrm{h}} H} + \frac{a^2 H_{\mathrm{eq}}^2 K_{\mathrm{p}} T_{\mathrm{h}}}{(1 + a T_{\mathrm{h}} H_{\mathrm{eq}})^2} \\
m_{\mathrm{pp}} &= \frac{\partial g}{\partial P} = = -r_{\mathrm{P}} \\
m_{\mathrm{ph}} &= \frac{\partial g}{\partial H} = r_{\mathrm{P}} K_{\mathrm{P}}
\end{aligned} \tag{18.77}$$

from which the eigenvalues are calculated as

$$\lambda = \frac{m_{\mathrm{hh}} + m_{\mathrm{pp}}}{2} \pm \sqrt{\frac{(m_{\mathrm{hh}} + m_{\mathrm{pp}})^2}{4} - \left(m_{\mathrm{hh}} m_{\mathrm{pp}} - m_{\mathrm{hp}} m_{\mathrm{ph}}\right)}. \tag{18.78}$$

18.5 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.

18.5.1 General Properties of Reaction–Diffusion Systems

Reaction–diffusion systems are described by a diffusion equation[2] where the source term depends nonlinearly on the concentrations

$$\frac{\partial}{\partial t} \begin{pmatrix} c_1 \\ \vdots \\ c_N \end{pmatrix} = \begin{pmatrix} D_1 & & \\ & \ddots & \\ & & D_N \end{pmatrix} \triangle \begin{pmatrix} c_1 \\ \vdots \\ c_N \end{pmatrix} + \begin{pmatrix} F_1(\{c\}) \\ \vdots \\ F_N(\{c\}) \end{pmatrix}. \tag{18.79}$$

[2] We consider only the case, that different species diffuse independently and that the diffusion constants do not depend on direction.

18.5.2 Chemical Reactions

Consider a number of chemical reactions which are described by stoichiometric equations

$$\sum_i \nu_i A_i = 0. \tag{18.80}$$

The concentration of agent A_i is

$$c_i = c_{i,0} + \nu_i x \tag{18.81}$$

with the reaction variable

$$x = \frac{c_i - c_{i,0}}{\nu_i} \tag{18.82}$$

and the reaction rate

$$r = \frac{\mathrm{d}x}{\mathrm{d}t} = \frac{1}{\nu_i}\frac{\mathrm{d}c_i}{\mathrm{d}t} \tag{18.83}$$

which, in general is a nonlinear function of all concentrations. The total concentration change due to diffusion and reactions is given by

$$\frac{\partial}{\partial t}c_k = D_k \triangle c_k + \sum_j \nu_{kj} r_j = D_k \triangle c_k + F_k(\{c_i\}). \tag{18.84}$$

18.5.3 Diffusive Population Dynamics

Combination of population dynamics (18.2) and diffusive motion gives a similar set of coupled equations for the population densities

$$\frac{\partial}{\partial t}N_k = D_k \triangle N_k + f_k(N_1, N_2, \ldots, N_n). \tag{18.85}$$

18.5.4 Stability Analysis

Since a solution of the nonlinear equations is not generally possible we discuss small deviations from an equilibrium solution N_k^{eq}[3] with

[3] We assume tacitly that such a solution exists.

$$\frac{\partial}{\partial t} N_k = \triangle N_k = 0. \tag{18.86}$$

Obviously the equilibrium obeys

$$f_k(N_1 \cdots N_n) = 0 \quad k = 1, 2, \ldots n. \tag{18.87}$$

We linearize the equations by setting

$$N_k = N_k^{\text{eq}} + \xi_k \tag{18.88}$$

and expand around the equilibrium

$$\frac{\partial}{\partial t} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{pmatrix} = \begin{pmatrix} D_1 & & \\ & \ddots & \\ & & D_N \end{pmatrix} \begin{pmatrix} \triangle\xi_1 \\ \triangle\xi_2 \\ \vdots \\ \triangle\xi_n \end{pmatrix} + \begin{pmatrix} \frac{\partial f_1}{\partial N_1} & \frac{\partial f_1}{\partial N_2} & \cdots & \frac{\partial f_1}{\partial N_n} \\ \frac{\partial f_2}{\partial N_1} & \frac{\partial f_2}{\partial N_2} & \cdots & \frac{\partial f_2}{\partial N_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial N_1} & \frac{\partial f_n}{\partial N_2} & \cdots & \frac{\partial f_n}{\partial N_n} \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{pmatrix} + \cdots . \tag{18.89}$$

Plane waves are solutions of the linearized problem.[4] Using the ansatz

$$\xi_j = \xi_{j,0} \mathrm{e}^{\mathrm{i}(\omega t - \mathbf{kx})} \tag{18.90}$$

we obtain

$$i\omega \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{pmatrix} = -k^2 D \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{pmatrix} + M_0 \begin{pmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_n \end{pmatrix}, \tag{18.91}$$

where M_0 denotes the matrix of derivatives and D the matrix of diffusion constants. For a stable plane wave solution $\lambda = \mathrm{i}\omega$ is an Eigenvalue of

$$M_k = M_0 - k^2 D \tag{18.92}$$

with

$$\Re(\lambda) \leq 0. \tag{18.93}$$

[4] Strictly this is true only for an infinite or periodic system.

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 [110].

18.5.5 Lotka–Volterra Model with Diffusion

As a simple example we consider again the Lotka–Volterra model. Adding diffusive terms we obtain the equations

$$\frac{\partial}{\partial t}\begin{pmatrix} H \\ P \end{pmatrix} = \begin{pmatrix} r_{\mathrm{H}} H - aHP \\ bHP - m_{\mathrm{P}} P \end{pmatrix} + \begin{pmatrix} D_{\mathrm{H}} & \\ & D_{\mathrm{P}} \end{pmatrix} \Delta \begin{pmatrix} H \\ P \end{pmatrix}. \tag{18.94}$$

There are two equilibria

$$H_{\mathrm{eq}} = P_{\mathrm{eq}} = 0 \tag{18.95}$$

and

$$P_{\mathrm{eq}} = \frac{r_{\mathrm{H}}}{a} \quad H_{\mathrm{eq}} = \frac{m_{\mathrm{P}}}{b}. \tag{18.96}$$

The Jacobian matrix is

$$M_0 = \frac{\partial}{\partial C} F(C_0) = \begin{pmatrix} r_{\mathrm{H}} - a P_{\mathrm{eq}} & -a H_{\mathrm{eq}} \\ b P_{\mathrm{eq}} & b H_{\mathrm{eq}} - m_{\mathrm{P}} \end{pmatrix} \tag{18.97}$$

which gives for the trivial equilibrium

$$M_k = \begin{pmatrix} r_{\mathrm{H}} - D_{\mathrm{H}} k^2 & 0 \\ 0 & -m_{\mathrm{P}} - D_{\mathrm{P}} k^2 \end{pmatrix}. \tag{18.98}$$

One of the eigenvalue $\lambda_1 = -m_{\mathrm{P}} - D_P k^2$ is negative whereas the second $\lambda_2 = r_{\mathrm{H}} - D_{\mathrm{H}} k^2$ is positive for $k^2 < r_{\mathrm{H}}/D_{\mathrm{H}}$. Hence this equilibrium is unstable against fluctuations with long wavelengths. For the second equilibrium we find

$$M_k = \begin{pmatrix} -D_{\mathrm{H}} k^2 & -\frac{\mathrm{am_P}}{b} \\ \frac{\mathrm{br_H}}{a} & -D_{\mathrm{P}} k^2 \end{pmatrix} \tag{18.99}$$

$$\mathrm{tr}\,(M_k) = -(D_{\mathrm{H}} + D_{\mathrm{P}}) k^2$$

$$\det(M_K) = m_{\mathrm{P}} r_{\mathrm{H}} + D_{\mathrm{H}} D_{\mathrm{P}} k^4$$

$$\lambda = -\frac{D_{\mathrm{H}} + D_{\mathrm{P}}}{2} k^2 \pm \frac{1}{2}\sqrt{(D_{\mathrm{H}} - D_{\mathrm{P}})^2 k^4 - 4 m_{\mathrm{P}} r_{\mathrm{H}}}. \tag{18.100}$$

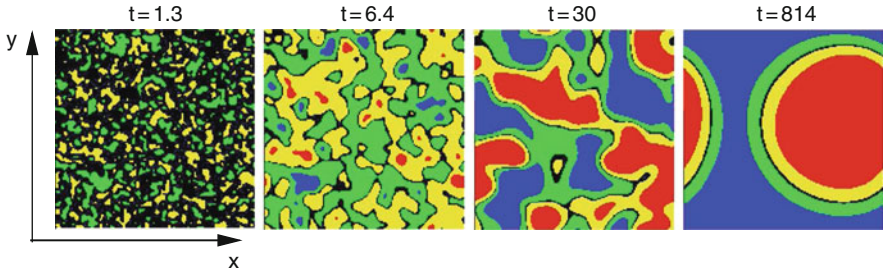
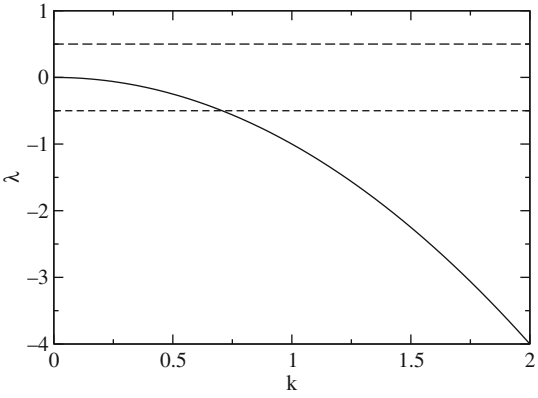


Fig. 18.12 Lotka–Volterra model with diffusion. The time evolution is calculated for initial random fluctuations. Colors indicate the deviation of the predator concentration $P(x, y, t)$ from its average value (*blue*: $\Delta P < -0.1$, *green*: $-0.1 < \Delta P < -0.01$, *black*: $-0.01 < \Delta P < 0.01$, *yellow*: $0.01 < \Delta P < 0.1$, *red*: $\Delta P > 0.1$. Parameters as in Fig. 18.13

Fig. 18.13 Dispersion of the diffusive Lotka–Volterra model. Real (*full curve*) and imaginary part (*broken line*) of the eigenvalue λ (18.100) are shown as a function of k. Parameters are $D_H = D_P = 1$, $m_P = r_H = a = b = 0.5$



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. 18.12–18.14).

Problems

Problem 18.1: Orbits of the Iterated Logistic Map

This computer example draws orbits (Fig. 18.5) of the logistic map

$$x_{n+1} = r_0\, x_n\, (1 - x_n).$$

You can select the initial value x_0 and the variable r.

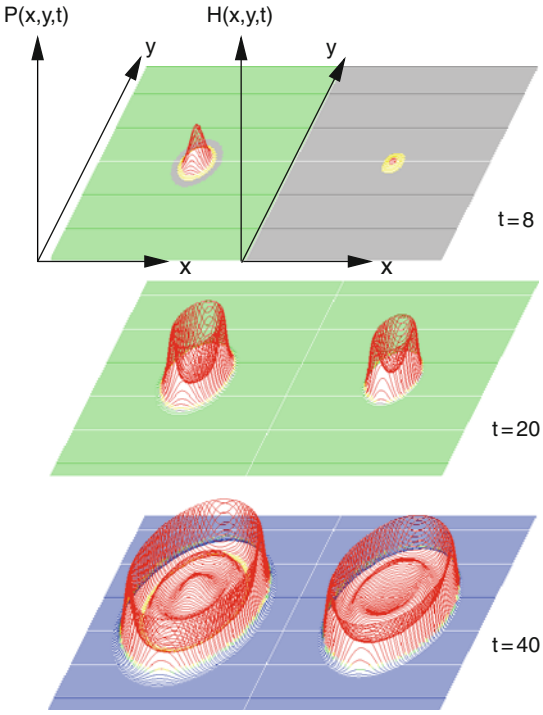


Fig. 18.14 Traveling waves in the diffusive Lotka–Volterra model. Initially $P(x, y) = P_{\text{eq}}$ and $H(x, y)$ is peaked in the center. This leads to oscillations and a sharp wavefront moving away from the excitation. Color code and parameters as in Fig. 18.12

Problem 18.2: Bifurcation Diagram of the Logistic Map

This computer example generates a bifurcation diagram of the logistic map (Fig. 18.6). You can select the range of r.

Problem 18.3: Lotka–Volterra Model

Equations (18.50) are solved with the improved Euler method (Fig. 18.8). The predictor step uses an explicit Euler step to calculate the values at $t + \Delta t/2$

$$H_{\text{pr}}\left(t + \frac{\Delta t}{2}\right) = H(t) + (r_{\text{H}} H(t) - a H(t) P(t)) \frac{\Delta t}{2}$$
$$P_{\text{pr}}\left(t + \frac{\Delta t}{2}\right) = P(t) + \left(b H(t) P(t) - m_{\text{p}} P(t)\right) \frac{\Delta t}{2}$$

and the corrector step advances time by Δt

$$H(t + \Delta t) = H(t) + \left(r_{\text{H}} H_{\text{pr}}\left(t + \frac{\Delta t}{2}\right) - a H_{\text{pr}}\left(t + \frac{\Delta t}{2}\right) P_{\text{pr}}\left(t + \frac{\Delta t}{2}\right)\right) \Delta t$$

$$P(t+\Delta t) = P(t) + \left(bH_{\mathrm{pr}}\left(t+\frac{\Delta t}{2}\right) P_{\mathrm{pr}}\left(t+\frac{\Delta t}{2}\right) - m_{\mathrm{p}} P_{\mathrm{pr}}\left(t+\frac{\Delta t}{2}\right)\right)\Delta t$$

Problem 18.4: Holling–Tanner Model

The equations of the Holling–Tanner model (18.67, 18.68) are solved with the improved Euler method (see Fig. 18.11). The pedictor step uses an explicit Euler step to calculate the values at $t + \Delta t/2$:

$$H_{\mathrm{pr}}\left(t+\frac{\Delta t}{2}\right) = H(t) + f(H(t), P(t))\frac{\Delta t}{2}$$
$$P_{\mathrm{pr}}\left(t+\frac{\Delta t}{2}\right) = P(t) + g(H(t), P(t))\frac{\Delta t}{2}$$

and the corrector step advances time by Δt:

$$H(t+\Delta t) = H(t) + f\left(H_{\mathrm{pr}}\left(t+\frac{\Delta t}{2}\right), P_{\mathrm{pr}}\left(t+\frac{\Delta t}{2}\right)\right)\Delta t$$
$$P(t+\Delta t) = P(t) + g\left(H_{\mathrm{pr}}\left(t+\frac{\Delta t}{2}\right), P_{\mathrm{pr}}\left(t+\frac{\Delta t}{2}\right)\right)\Delta t$$

Problem 18.5: Diffusive Lotka–Volterra Model

The Lotka–Volterra model with diffusion (18.94) is solved in two dimensions with an implicit method (17.3.3) for the diffusive motion (Figs. 18.12 and 18.14). The split operator approximation (17.3.7) is used to treat diffusion in x- and y-direction independently. The equations

$$\begin{pmatrix} H(t+\Delta t) \\ P(t+\Delta t) \end{pmatrix} = \begin{pmatrix} A^{-1}H(t) \\ A^{-1}P(t) \end{pmatrix} + \begin{pmatrix} A^{-1}f(H(t), P(t))\Delta t \\ A^{-1}g(H(t), P(t))\Delta t \end{pmatrix}$$
$$\approx \begin{pmatrix} A_x^{-1}A_y^{-1}\left[H(t) + f(H(t), P(t))\Delta t\right] \\ A_x^{-1}A_y^{-1}\left[P(t) + g(H(t), P(t))\Delta t\right] \end{pmatrix}$$

are equivalent to the following systems of linear equations with tridiagonal matrix (5.3):

$$A_y U = H(t) + f(H(t), P(t))\Delta t$$
$$U = A_x H(t+\Delta t)$$
$$A_y V = P(t) + g(H(t), P(t))\Delta t$$
$$V = A_x P(t+\Delta t)$$

Periodic boundary conditions are implemented with the method described in Sect. 5.4.

Chapter 19
Simple Quantum Systems

The time evolution of a quantum system is governed by the time-dependent Schrödinger equation [111]

$$\mathrm{i}\hbar\frac{\partial}{\partial t}|\psi\rangle = H|\psi\rangle \tag{19.1}$$

for the wavefunction ψ. The brackets indicate that $|\psi\rangle$ is a vector in abstract Hilbert space [112]. Vectors can be added

$$|\psi\rangle = |\psi_1\rangle + |\psi_2\rangle = |\psi_1 + \psi_2\rangle \tag{19.2}$$

and can be multiplied with a complex number

$$|\psi\rangle = \lambda|\psi_1\rangle = |\lambda\psi_1\rangle. \tag{19.3}$$

Finally a complex valued scalar product of two vectors is defined[1]

$$C = \langle\psi_1|\psi_2\rangle \tag{19.4}$$

which has the properties

$$\begin{aligned}
&\langle\psi_1|\psi_2\rangle = \langle\psi_2|\psi_1\rangle^* \\
&\langle\psi_1|\lambda\psi_2\rangle = \lambda\langle\psi_1|\psi_2\rangle = \langle\lambda^*\psi_1|\psi_2\rangle \\
&\langle\psi|\psi_1 + \psi_2\rangle = \langle\psi|\psi_1\rangle + \langle\psi|\psi_2\rangle \\
&\langle\psi_1 + \psi_2|\psi\rangle = \langle\psi_1|\psi\rangle + \langle\psi_2|\psi\rangle.
\end{aligned} \tag{19.5}$$

[1] If, for instance, the wavefunction depends on the coordinates of N particles, the scalar product is defined by $\langle\psi_n|\psi_{n'}\rangle = \int \mathrm{d}^3r_1 \cdots \mathrm{d}^3r_N \psi_n^*(r_1 \cdots r_N)\psi_{n'}(r_1 \cdots r_N)$.

P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1_19,

In this chapter we study simple quantum systems like a particle in a one-dimensional potential well $V(x)$ which is described by the partial differential equation [113]

$$\mathrm{i}\hbar\frac{\partial}{\partial t}\psi(x) = H\psi(x) = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}\psi(x) + V(x)\psi(x), \tag{19.6}$$

or systems which can be approximately described with a finite set of basis states ψ_n, $n = 1 \cdots n_{\max}$. 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.[2] It is described by a two-component vector

$$|\psi\rangle = \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \tag{19.7}$$

and two coupled ordinary differential equations for the amplitudes $C_{1,2}$ of the two states

$$\mathrm{i}\hbar\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix}\begin{pmatrix} C_1 \\ C_2 \end{pmatrix}. \tag{19.8}$$

The two-state model also represents a Qubit, a basic element of a future quantum computer [114]. Whereas a classical bit is in either one of its two states (0 or 1), the wavefunction of a Qubit is generally a superposition of the two states

$$|\psi\rangle = C_0|\psi_0\rangle + C_1|\psi_1\rangle \tag{19.9}$$

and the coefficients $C_{0,1}$ obey an equation similar to (19.8).

19.1 Quantum Particle in a Potential Well

A quantum mechanical particle in a finite[3] potential well, i.e., a potential $V(\mathbf{r})$ which has no upper bound outside a finite interval $a < r < b$ (Fig. 19.1)

$$V(\mathbf{r}) = \infty \quad \text{for} \quad r < a \quad \text{or} \quad r > b \tag{19.10}$$

is described by a complex valued wavefunction

$$\psi(\mathbf{r}) \quad \text{with} \quad \psi(r) = 0 \quad \text{for} \quad r < a \quad \text{or}\, r > b. \tag{19.11}$$

[2] For instance, collisions or the electromagnetic radiation field.

[3] Numerically we can treat only finite systems.

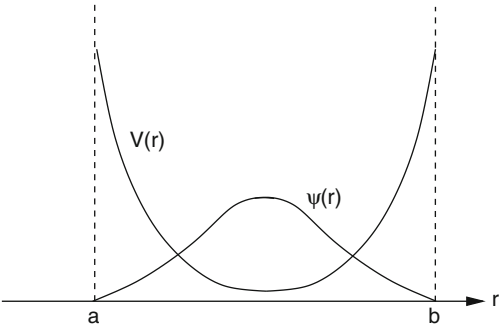


Fig. 19.1 Finite potential well

All observables (quantities which can be measured) of the particle are expectation values with respect to the wavefunction, for instance, its average position is

$$\langle \mathbf{r} \rangle = \langle \psi(\mathbf{r})\, \mathbf{r}\, \psi(\mathbf{r}) \rangle \int \mathrm{d}^3 r\, \psi^*(\mathbf{r}) \mathbf{r} \psi(\mathbf{r}). \tag{19.12}$$

The probability of finding the particle at the position $\mathbf{r_0}$ is given by

$$P(\mathbf{r} = \mathbf{r}_0) = |\psi(\mathbf{r})|^2 . \tag{19.13}$$

In the following we consider a particle in a one-dimensional potential $V(x)$. The Schrödinger equation

$$\mathrm{i}\hbar\dot{\psi} = H\psi = \left(-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x) \right) \psi \tag{19.14}$$

is very similar to a diffusion equation with imaginary diffusion constant. Consider a simple explicit Euler step

$$\psi_{n+1} = (1 - \frac{\mathrm{i}\Delta t}{\hbar} H)\psi_n. \tag{19.15}$$

From the real eigenvalues E of the Hamiltonian we find the eigenvalues of the explicit method

$$\lambda = 1 - \frac{\mathrm{i}\Delta t}{\hbar} E \tag{19.16}$$

which all have absolute values

$$|\lambda| = \sqrt{1 + \frac{\Delta t^2 E^2}{\hbar^2}} > 1. \tag{19.17}$$

Hence the explicit method is not stable. The implicit method

$$\psi_{n+1} = \psi_n - \frac{\mathrm{i}\Delta t}{\hbar} H \psi_{n+1} \tag{19.18}$$

can be rearranged as

$$\psi_{n+1} = \left(1 + \frac{\mathrm{i}\Delta t}{\hbar} H\right)^{-1} \psi_n. \tag{19.19}$$

Here all eigenvalues have absolute values < 1. This method is stable but the norm of the wave function is not conserved. Again combination of implicit and explicit method gives a superior method

$$\psi_{n+1} - \psi_n = -\frac{\mathrm{i}\Delta t}{\hbar} H \left(\frac{\psi_{n+1}}{2} + \frac{\psi_n}{2}\right). \tag{19.20}$$

This equation can be solved for the new value of the wavefunction

$$\psi_{n+1} = \left(1 + \mathrm{i}\frac{\Delta t}{2\hbar} H\right)^{-1} \left(1 - \mathrm{i}\frac{\Delta t}{2\hbar} H\right) \psi_n. \tag{19.21}$$

The eigenvalues of (19.21) all have an absolute value of

$$|\lambda| = \left| \left(1 + \mathrm{i}\frac{E\Delta t}{2\hbar}\right)^{-1} \left(1 - \mathrm{i}\frac{E\Delta t}{2\hbar}\right) \right| = \frac{\sqrt{1 + \frac{E^2\Delta t^2}{4\hbar^2}}}{\sqrt{1 + \frac{E^2\Delta t^2}{4\hbar^2}}} = 1. \tag{19.22}$$

Hence the operator

$$\left(1 + \mathrm{i}\frac{\Delta t}{2\hbar} H\right)^{-1} \left(1 - \mathrm{i}\frac{\Delta t}{2\hbar} H\right) \tag{19.23}$$

is unitary and conserves the norm of the wavefunction. From the Taylor series we find the error order

$$\begin{aligned} &\left(1 + \mathrm{i}\frac{\Delta t}{2\hbar} H\right)^{-1} \left(1 - \mathrm{i}\frac{\Delta t}{2\hbar} H\right) = \left(1 - \mathrm{i}\frac{\Delta t}{2\hbar} H - \frac{\Delta t^2}{4\hbar^2} H^2 + \cdots\right) \left(1 - \mathrm{i}\frac{\Delta t}{2\hbar} H\right) \\ &= 1 - \frac{\mathrm{i}\Delta t}{\hbar} H - \frac{\Delta t^2}{2\hbar^2} H^2 + \cdots = \exp\left(-\frac{i\Delta t}{\hbar} H\right) + O(\Delta t^3). \end{aligned} \tag{19.24}$$

For practical application we rewrite [115]

$$\left(1+\mathrm{i}\frac{\Delta t}{2\hbar}H\right)^{-1}\left(1-\mathrm{i}\frac{\Delta t}{2\hbar}H\right)$$
$$=\left(1+\mathrm{i}\frac{\Delta t}{2\hbar}H\right)^{-1}\left(-1-\mathrm{i}\frac{\Delta t}{2\hbar}H+2\right)=-1+2\left(1+\mathrm{i}\frac{\Delta t}{2\hbar}H\right)^{-1} \tag{19.25}$$

hence

$$\psi_{n+1}=2\left(1+\mathrm{i}\frac{\Delta t}{2\hbar}H\right)^{-1}\psi_n-\psi_n=2\chi-\psi_n. \tag{19.26}$$

ψ_{n+1} is obtained in two steps. First we have to solve

$$\left(1+\mathrm{i}\frac{\Delta t}{2\hbar}H\right)\chi=\psi_n. \tag{19.27}$$

Then ψ_{n+1} is given by

$$\psi_{n+1}=2\chi-\psi_n. \tag{19.28}$$

We introduce a coordinate grid

$$x_j=j\Delta x \qquad j=0\cdots j_{\max} \tag{19.29}$$

and approximate the second derivative by

$$\frac{\partial^2}{\partial x^2}\psi(x_j)=\frac{\psi(x_{j+1})+\psi(x_{j-1})-2\psi(x_j)}{\Delta x^2}. \tag{19.30}$$

Equation (19.27) becomes a system of linear equations

$$A\begin{bmatrix}\chi(x_0)\\ \chi(x_1)\\ \chi(x_2)\\ \vdots\end{bmatrix}=\begin{bmatrix}\psi_n(x_0)\\ \psi_n(x_1)\\ \psi_n(x_2)\\ \vdots\end{bmatrix} \tag{19.31}$$

with a tridiagonal matrix

$$A = 1 + \mathrm{i}\frac{\Delta t}{2\hbar\Delta x^2}\begin{pmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & \ddots \\ & & \ddots & \ddots \end{pmatrix} \tag{19.32}$$

$$+\mathrm{i}\frac{\Delta t}{2\hbar}\begin{pmatrix} V(0) & & & \\ & V(\Delta x) & & \\ & & V(2\Delta x) & \\ & & & \ddots \end{pmatrix}. \tag{19.33}$$

The second step (19.28) becomes

$$\begin{bmatrix} \psi_{n+1}(x_0) \\ \psi_{n+1}(x_1) \\ \psi_{n+1}(x_2) \\ \vdots \end{bmatrix} = 2\begin{bmatrix} \chi(x_0) \\ \chi(x_1) \\ \chi(x_2) \\ \vdots \end{bmatrix} - \begin{bmatrix} \psi_n(x_0) \\ \psi_n(x_1) \\ \psi_n(x_2) \\ \vdots \end{bmatrix}. \tag{19.34}$$

19.2 Expansion in a Finite Basis

We consider a quantum system which is described by the wavefunction[4]

$$|\psi(t)\rangle. \tag{19.35}$$

The time-dependent Schrödinger equation is

$$\mathrm{i}\hbar|\dot{\psi}(t)\rangle = H|\psi(t)\rangle. \tag{19.36}$$

The eigenvalues of

$$H|\psi_n\rangle = E_n|\psi_n\rangle \tag{19.37}$$

are the energy values E_n of the stationary states $|\psi_n\rangle$ which are assumed to form an orthonormal basis[5]

[4] In general the wavefunction depends on a large number of variables, for instance, the coordinates and spin variables of N particles.

[5] We assume that the system has a discrete and finite spectrum of eigenvalues, for instance, if the system is bounded by a finite box.

$$\langle \psi_n | \psi_{n'} \rangle = \delta_{n,n'}. \tag{19.38}$$

The general solution of the time-dependent Schrödinger equation can be constructed as a linear combination of the stationary states [113]

$$|\psi(t)\rangle = \sum_n C_n \exp\left\{\frac{E_n}{\mathrm{i}\hbar} t\right\} |\psi_n\rangle. \tag{19.39}$$

The coefficients C_n are determined by the initial values of the wavefunction

$$|\psi(t=0)\rangle = \sum_n C_n |\psi_n> \tag{19.40}$$

and can be obtained from the scalar product

$$\langle \psi_m | \psi(t=0)\rangle = \sum_n C_n \langle \psi_m | \psi_n \rangle = C_m. \tag{19.41}$$

In the following we discuss simple models which approximate the sum over a very large number of eigenstates by the sum over a small number of important states, for instance, an initial and a final state which are coupled by some resonant interaction. Formally we introduce an (incomplete) set of orthonormal states[6]

$$\begin{aligned} &|\phi_1\rangle \cdots |\phi_M\rangle \\ &\langle \phi_i | \phi_j \rangle = \delta_{ij} \end{aligned} \tag{19.42}$$

and approximate the wave function by a linear combination

$$|\psi(t)\rangle \approx \sum_{j=1}^{M} C_j(t) |\phi_j\rangle. \tag{19.43}$$

Inserting into the time-dependent Schrödinger (19.36) equation gives

$$\mathrm{i}\hbar \sum_j \dot{C}_j(t) |\phi_j\rangle = \sum_j C_j(t) H |\phi_j\rangle \tag{19.44}$$

and after taking the scalar product with $|\phi_i\rangle$ we arrive at the system of ordinary differential equations

$$\mathrm{i}\hbar \dot{C}_i = \sum_{j=1}^{M} H_{i,j} C_j(t) \tag{19.45}$$

[6] In general these are linear combinations of the eigenstates.

with the matrix elements of the Hamiltonian

$$H_{i,j} = \langle \phi_i | H | \phi_j \rangle. \tag{19.46}$$

In matrix form (19.45) reads

$$\mathrm{i}\hbar \begin{pmatrix} \dot{C}_1(t) \\ \vdots \\ \dot{C}_M(t) \end{pmatrix} = \begin{pmatrix} H_{1,1} & \cdots & H_{1,M} \\ \vdots & \ddots & \vdots \\ H_{M,1} & \cdots & H_{M,M} \end{pmatrix} \begin{pmatrix} C_1(t) \\ \vdots \\ C_M(t) \end{pmatrix} \tag{19.47}$$

or more symbolically

$$\mathrm{i}\hbar \dot{\mathbf{C}}(t) = \mathbf{H}\mathbf{C}(t). \tag{19.48}$$

19.3 Time-Independent Problems

If the Hamilton operator does not depend explicitly on time ($H = \text{const.}$) the formal solution of (19.48) is given by

$$\mathbf{C} = \exp\left\{ \frac{t}{\mathrm{i}\hbar} H \right\} \mathbf{C}(0). \tag{19.49}$$

From the solution of the eigenvalue problem

$$H\mathbf{C}_\lambda = \lambda \mathbf{C}_\lambda \tag{19.50}$$

(eigenvalues λ and corresponding eigenvectors $\mathbf{C}_\lambda$) we build the linear combination

$$\mathbf{C} = \sum_\lambda a_\lambda \mathbf{C}_\lambda \mathrm{e}^{\frac{\lambda}{\mathrm{i}\hbar} t}. \tag{19.51}$$

The amplitudes a_λ can be calculated from the set of linear equations

$$\mathbf{C}(0) = \sum_\lambda a_\lambda \mathbf{C}_\lambda. \tag{19.52}$$

In the following we calculate the time evolution numerically using the fourth-order Runge–Kutta method. This allows also the treatment of a time-dependent Hamiltonian later on.

19.3.1 Simple Two-Level System

The two-level system is the simplest model of interacting states and is very often used in physics (Fig. 19.2).

The interaction matrix of a two-level system is

$$H = \begin{pmatrix} E_1 & V \\ V & E_2 \end{pmatrix} \tag{19.53}$$

and the equations of motion are

$$\begin{aligned} \mathrm{i}\hbar\dot{C}_1 &= E_1 C_1 + V C_2 \\ \mathrm{i}\hbar\dot{C}_2 &= E_2 C_2 + V C_1 \end{aligned}. \tag{19.54}$$

Equations (19.54) can be solved analytically but this involves some lengthy expressions. Let us therefore concentrate on two limiting cases:
(a) For $E_1 = E_2$ we have

$$\ddot{C}_1 = -\frac{V^2}{\hbar^2} C_1 \tag{19.55}$$

which is solved by an oscillating coefficient

$$C_1 = \cos\left(\frac{V}{\hbar} t\right) \tag{19.56}$$

with period

$$T = \frac{2\pi\hbar}{V}. \tag{19.57}$$

(b) For $V \ll |\Delta E| = |E_1 - E_2|$ perturbation theory for the small quantity $V/\Delta E$ gives the following approximations:

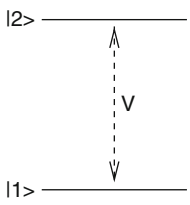


Fig. 19.2 Two level system model

$$\begin{aligned}\lambda_1 &\approx E_1 - \tfrac{V^2}{\Delta E} \\ \lambda_2 &\approx E_2 + \tfrac{V^2}{\Delta E}\end{aligned} \tag{19.58}$$

$$\begin{aligned}\mathbf{C}_1 &\approx \begin{pmatrix} 1 \\ \frac{V}{\Delta E} \end{pmatrix} \\ \mathbf{C}_2 &\approx \begin{pmatrix} \frac{-V}{\Delta E} \\ 1 \end{pmatrix}\end{aligned} . \tag{19.59}$$

For initial values $\mathbf{C}(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ the amplitudes $a_{1,2}$ are calculated from

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} a_1 - a_2 \frac{V}{\Delta E} \\ a_1 \frac{V}{\Delta E} + a_2 \end{pmatrix} \tag{19.60}$$

which gives in lowest order

$$\begin{aligned}a_1 &\approx 1 - \tfrac{V^2}{\Delta E^2} \\ a_2 &\approx \tfrac{V^2}{\Delta E^2}\end{aligned} . \tag{19.61}$$

The approximate solution is

$$\mathbf{C} = \begin{pmatrix} (1 - \frac{V^2}{\Delta E^2}) \mathrm{e}^{\frac{1}{i\hbar}(E_1 - \frac{V^2}{\Delta E^2})t} + \frac{V^2}{\Delta E^2} \mathrm{e}^{\frac{1}{i\hbar}(E_2 + \frac{V^2}{\Delta E^2})t} \\ \frac{V}{\Delta E} \mathrm{e}^{\frac{1}{i\hbar}(E_1 - \frac{V^2}{\Delta E^2})t} - \frac{V}{\Delta E} \mathrm{e}^{\frac{1}{i\hbar}(E_2 + \frac{V^2}{\Delta E^2})t} \end{pmatrix} \tag{19.62}$$

and the occupation probability of the initial state is (Fig. 19.3)

$$|C_1|^2 \approx 1 - 2\frac{V^2}{\Delta E^2} + 2\frac{V^2}{\Delta E^2} \cos\left(\left(\Delta E + 2\frac{V^2}{\Delta E}\right) t\right). \tag{19.63}$$

19.3.2 Three-State Model (Superexchange)

Consider two isoenergetic states i and f which do not interact directly but via coupling to an intermediate state v (Fig. 19.4).
The interaction matrix is

$$H = \begin{pmatrix} 0 & V_1 & 0 \\ V_1 & E_2 & V_2 \\ 0 & V_2 & 0 \end{pmatrix}. \tag{19.64}$$

For simplification we choose $V_1 = V_2$.

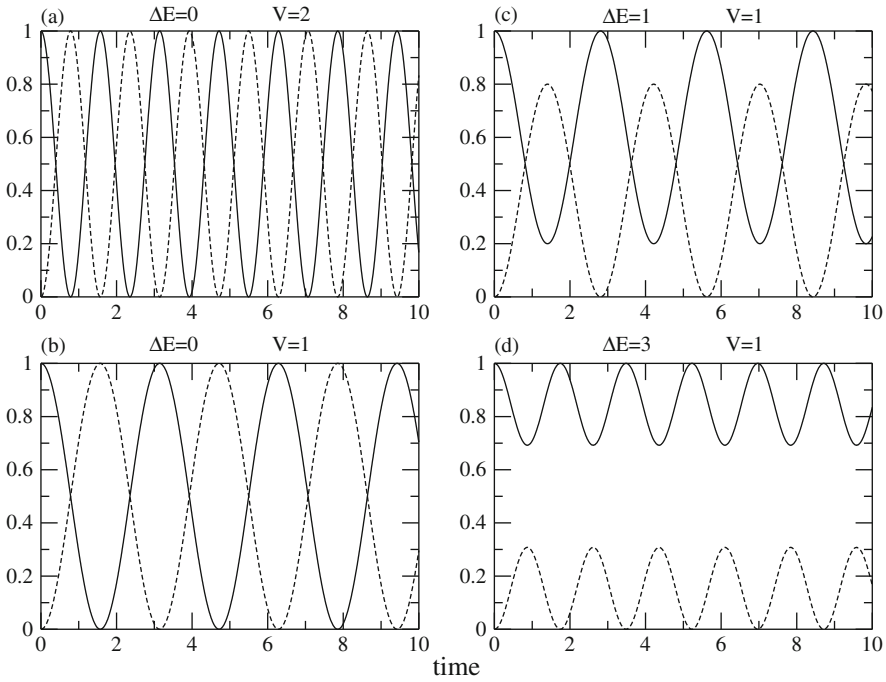


Fig. 19.3 Numerical simulation of a two-level system. The equations of motion of the two-level system (19.54) are integrated with the fourth-order Runge–Kutta method. For two resonant states the occupation probability of the initial state shows oscillations with the period (19.57) proportional to V^{-1}. With increasing energy gap $E_2 - E_1$ the amplitude of the oscillations decreases

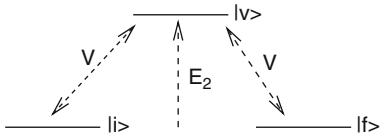


Fig. 19.4 Superexchange model

Let us first consider the special case of a resonant intermediate state $E_2 = 0$:

$$H = \begin{pmatrix} 0 & V & 0 \\ V & 0 & V \\ 0 & V & 0 \end{pmatrix}. \tag{19.65}$$

Obviously one eigenvalue is $\lambda_1 = 0$ and the corresponding eigenvector is

$$\mathbf{C}_1 = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}. \tag{19.66}$$

The two remaining eigenvalues are solutions of

$$0 = \det \begin{vmatrix} -\lambda & V & 0 \\ V & -\lambda & V \\ 0 & V & -\lambda \end{vmatrix} = \lambda(-\lambda^2 + 2V^2) \tag{19.67}$$

which gives

$$\lambda_{2,3} = \pm\sqrt{2}V. \tag{19.68}$$

The eigenvectors are

$$\mathbf{C}_{2,3} = \begin{pmatrix} 1 \\ \pm\sqrt{2} \\ 1 \end{pmatrix}. \tag{19.69}$$

From the initial values

$$\mathbf{C}(0) = \begin{pmatrix} a_1 + a_2 + a_3 \\ \sqrt{2}a_2 - \sqrt{2}a_3 \\ -a_1 + a_2 + a_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{19.70}$$

the amplitudes are calculated as

$$a_1 = \frac{1}{2}\, a_2 = a_3 = \frac{1}{4} \tag{19.71}$$

and finally the solution is

$$\begin{aligned} \mathbf{C} &= \frac{1}{2}\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} + \frac{1}{4}\begin{pmatrix} 1 \\ \sqrt{2} \\ 1 \end{pmatrix} \mathrm{e}^{\frac{1}{\mathrm{i}\hbar}\sqrt{2}Vt} + \frac{1}{4}\begin{pmatrix} 1 \\ -\sqrt{2} \\ 1 \end{pmatrix} \mathrm{e}^{-\frac{1}{\mathrm{i}\hbar}\sqrt{2}Vt} \\ &= \begin{pmatrix} \frac{1}{2} + \frac{1}{2}\cos\frac{\sqrt{2}V}{\hbar}t \\ \frac{\sqrt{2}}{2}\mathrm{i}\sin\frac{\sqrt{2}V}{\hbar}t \\ -\frac{1}{2} + \frac{1}{2}\cos\frac{\sqrt{2}V}{\hbar}t \end{pmatrix}. \end{aligned} \tag{19.72}$$

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

$$\lambda_{2,3} = \pm\sqrt{\frac{E_2^2}{4} + V^2} + \frac{E_2}{2} \approx \frac{E_2}{2} \pm \frac{E_2}{2}\left(1 + \frac{4V^2}{E_2^2}\right) \tag{19.73}$$

$$\lambda_2 \approx E_2 + \frac{2V^2}{E_2} \quad \lambda_3 \approx -\frac{2V^2}{E_2} \tag{19.74}$$

and the eigenvectors by

$$\mathbf{C}_2 \approx \begin{pmatrix} 1 \\ \frac{E_2}{V} + \frac{2V}{E_2} \\ 1 \end{pmatrix} \qquad \mathbf{C}_3 \approx \begin{pmatrix} 1 \\ -\frac{2V}{E_2} \\ 1 \end{pmatrix}. \tag{19.75}$$

From the initial values

$$\mathbf{C}(0) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} a_1 + a_2 + a_3 \\ a_2\lambda_2 + a_3\lambda_3 \\ -a_1 + a_2 + a_3 \end{pmatrix} \tag{19.76}$$

we calculate the amplitudes

$$a_1 = \frac{1}{2} \qquad a_2 \approx \frac{V^2}{E_2^2} \qquad a_3 \approx \frac{1}{2}\left(1 - \frac{2V^2}{E_2^2}\right) \tag{19.77}$$

and finally the solution

$$\mathbf{C} \approx \begin{pmatrix} \frac{1}{2}(1 + \mathrm{e}^{-\frac{1}{\mathrm{i}\hbar}\frac{2V^2}{E_2}t}) \\ \frac{V}{E_2}\mathrm{e}^{\frac{1}{\mathrm{i}\hbar}E_2 t} - \frac{2V}{E_2}\mathrm{e}^{-\frac{1}{\mathrm{i}\hbar}\frac{2V^2}{E_2}t} \\ \frac{1}{2}(-1 + \mathrm{e}^{-\frac{1}{\mathrm{i}\hbar}\frac{2V^2}{E_2}t}) \end{pmatrix}. \tag{19.78}$$

The occupation probability of the initial state is

$$|C_1|^2 = \frac{1}{4}|1 + \mathrm{e}^{-\frac{1}{\mathrm{i}\hbar}\frac{2V^2}{E_2}t}|^2 = \cos^2\left(\frac{V^2}{\hbar E_2}t\right) \tag{19.79}$$

which shows that the system behaves like a two-state system with an effective interaction of (Fig. 19.5)

$$V_{\mathrm{eff}} = \frac{V^2}{E_2}. \tag{19.80}$$

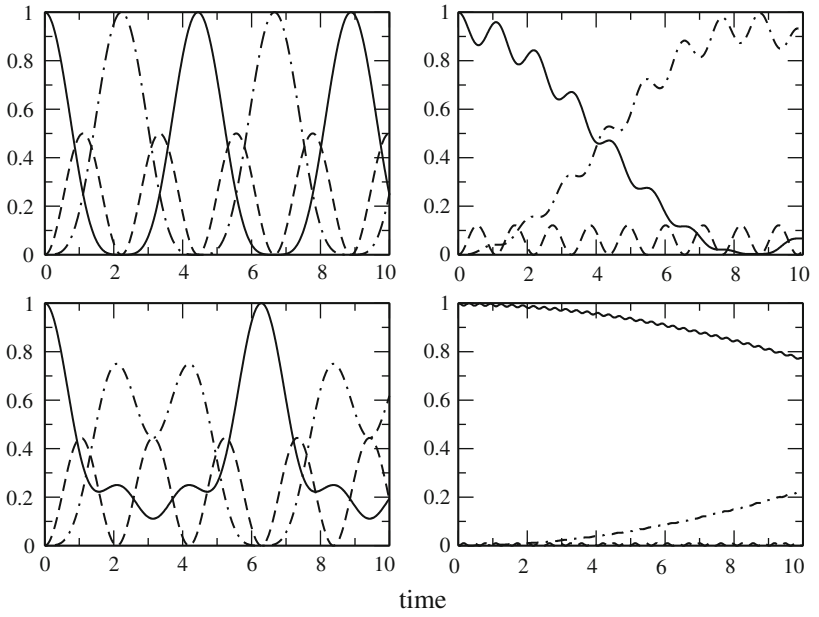


Fig. 19.5 Numerical simulation of the superexchange model. The equations of motion for the model equation (19.64) are solved numerically with the fourth-order Runge–Kutta method. The energy gap is varied to study the transition from the simple oscillation with $\omega = \sqrt{2}V/\hbar$ (19.72) to the effective two-level system with $\omega = V_{\text{eff}}/\hbar$ (19.79). 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

19.3.3 Ladder Model for Exponential Decay

We consider now a simple model for exponential decay [116, 117]. State 0 interacts with a manifold of states $(1 \cdots n)$, which do not interact with each other and are equally spaced (Fig. 19.6):

$$H = \begin{pmatrix} 0 & V & \cdots & V \\ V & E_1 & & \\ \vdots & & \ddots & \\ V & & & E_n \end{pmatrix} \qquad E_j = E_1 + (j-1)\Delta E \tag{19.81}$$

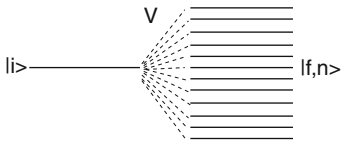


Fig. 19.6 Ladder model

The equations of motion are

$$\begin{aligned} \mathrm{i}\hbar\dot{C}_0 &= V\sum_{j=1}^{n} C_j \\ \mathrm{i}\hbar\dot{C}_j &= E_j C_j + V C_0. \end{aligned} \tag{19.82}$$

For the special case $\Delta E = 0$ we simply have

$$\ddot{C}_0 = -\frac{V^2}{\hbar^2} n C_0 \tag{19.83}$$

with an oscillating solution

$$C_0 \sim \cos\left(\frac{V\sqrt{n}}{\hbar}t\right). \tag{19.84}$$

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

$$C_j = u_j \mathrm{e}^{\frac{E_j}{\mathrm{i}\hbar}t} \tag{19.85}$$

and have

$$\mathrm{i}\hbar\dot{u}_j \mathrm{e}^{\frac{E_j}{\mathrm{i}\hbar}t} = V C_0. \tag{19.86}$$

Integration gives

$$u_j = \frac{V}{\mathrm{i}\hbar}\int_{t_0}^{t} \mathrm{e}^{-\frac{E_j}{\mathrm{i}\hbar}t'} C_0(t')\mathrm{d}t' \tag{19.87}$$

and therefore

$$C_j = \frac{V}{\mathrm{i}\hbar}\int_{t_0}^{t} \mathrm{e}^{\mathrm{i}\frac{E_j}{\hbar}(t'-t)} C_0(t')\mathrm{d}t'. \tag{19.88}$$

With the definition

$$E_j = j * \hbar\Delta\omega \tag{19.89}$$

we have

$$\dot{C}_0 = \frac{V}{\mathrm{i}\hbar}\sum_{j=1}^{n} C_j = -\frac{V^2}{\hbar^2}\sum\int_{t_0}^{t} \mathrm{e}^{\mathrm{i}j\Delta\omega(t'-t)} C_0(t')\mathrm{d}t'. \tag{19.90}$$

Replaced the sum by an integral

$$\omega = j\Delta\omega \tag{19.91}$$

and extend the integration range to $-\infty \cdots \infty$. Then the sum becomes approximately a delta function

$$\sum_{j=-\infty}^{\infty} e^{\mathrm{i}j\Delta\omega(t'-t)}\Delta j \to \int_{-\infty}^{\infty} e^{\mathrm{i}\omega(t'-t)}\frac{\mathrm{d}\omega}{\Delta\omega} = \frac{2\pi}{\Delta\omega}\delta(t'-t) \tag{19.92}$$

and hence the result is an exponential decay law (Fig. 19.7)

$$\dot{C}_0 = -\frac{2\pi V^2}{\Delta\omega}C_0 = -\frac{2\pi V^2}{\hbar}\rho(E)C_0 \tag{19.93}$$

with the density of final states

$$\rho(E) = \frac{1}{\hbar\Delta\omega} = \frac{1}{\Delta E}. \tag{19.94}$$

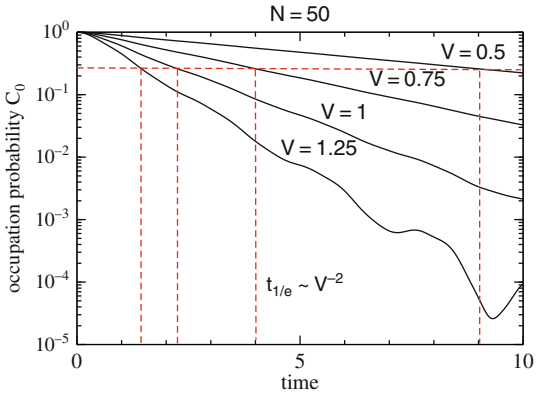


Fig. 19.7 Numerical solution of the ladder model. The time evolution of the ladder model equation (19.82) is calculated with the fourth-order Runge–Kutta method for $N = 50$ states and different values of the coupling V.

19.4 Time-Dependent Models

Now we study models with time-dependent Hamiltonian $H(t)$. Models of this type arise if nuclear motion or external fields are described as classical quantities.

19.4.1 Landau–Zener Model

This model describes crossing of two states, for instance, for colliding atoms or molecules [118, 119]. It is assumed that the interaction V is constant near the crossing point and that the nuclei move classically with constant velocity (Fig. 19.8)

$$H = \begin{pmatrix} 0 & V \\ V & \Delta E(t) \end{pmatrix} \qquad \Delta E(t) = \Delta E_0 + vt. \tag{19.95}$$

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

$$P = \frac{2\pi V^2}{\hbar \frac{\partial}{\partial t}\Delta E}. \tag{19.96}$$

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 (Fig. 19.9):

$$P_{\mathrm{LZ}} = 1 - \exp\left(-\frac{2\pi V^2}{\hbar \frac{\partial}{\partial t}\Delta E}\right). \tag{19.97}$$

In case of collisions multiple crossing of the interaction region has to be taken into account (Fig. 19.10)

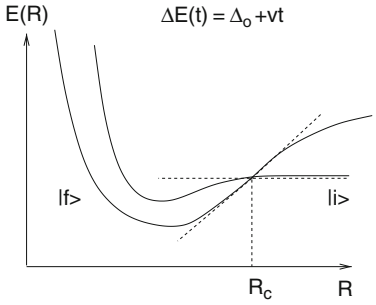


Fig. 19.8 Slow atomic collision

19.4.2 Two-State System with Time-Dependent Perturbation

Consider a two-state system with an oscillating perturbation (for instance, an atom or molecule in a laser field) (Fig. 19.11)

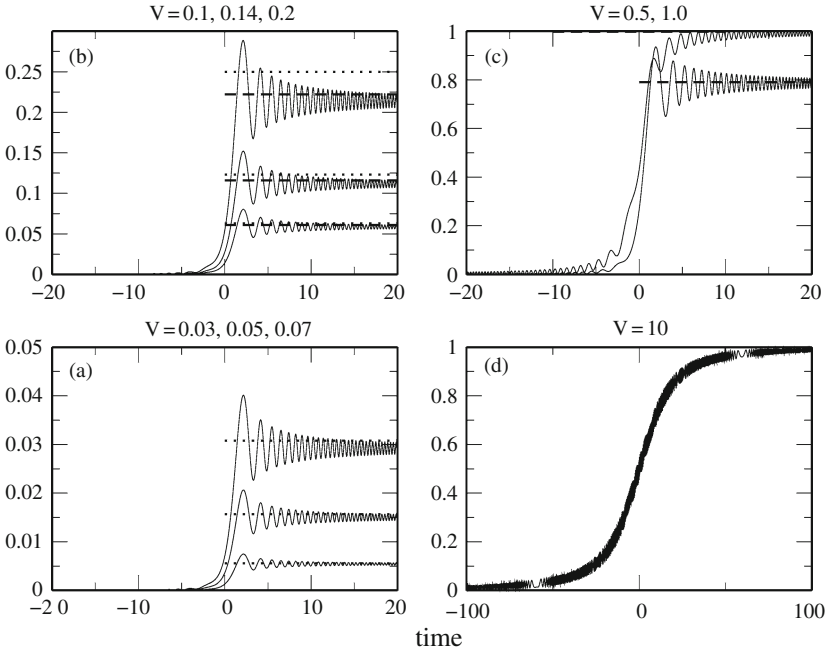


Fig. 19.9 Numerical solution of the Landau–Zener model. Numerical calculations (*solid curves*) are compared with the Landau–Zener probability ((19.97), *dashed lines*) and the approximation ((19.96), *dotted lines*) The velocity is $\mathrm{d}\Delta E/\mathrm{d}t = 1$

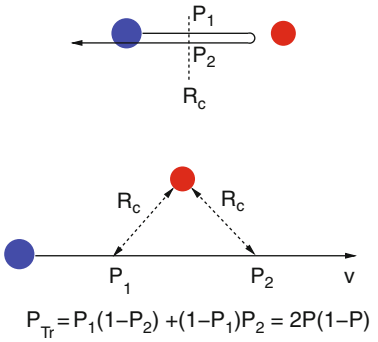


Fig. 19.10 Multiple passage of the interaction region

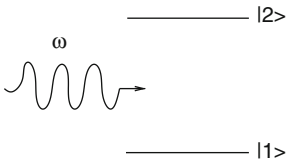


Fig. 19.11 Two-level system in an oscillating field

$$H = \begin{pmatrix} E_1 & V(t) \\ V(t) & E_2 \end{pmatrix} \qquad V(t) = V_0 \cos \omega t. \tag{19.98}$$

The equations of motion are

$$\begin{aligned} \mathrm{i}\hbar \dot{C}_1 &= E_1 C_1 + V(t) C_2 \\ \mathrm{i}\hbar \dot{C}_2 &= V(t) C_1 + E_2 C_2 \end{aligned}. \tag{19.99}$$

After the substitution

$$\begin{aligned} C_1 &= e^{\frac{E_1}{\mathrm{i}\hbar} t} u_1 \\ C_2 &= e^{\frac{E_2}{\mathrm{i}\hbar} t} u_2 \end{aligned} \tag{19.100}$$

they become

$$\begin{aligned} \mathrm{i}\hbar \dot{u}_1 &= V(t) \mathrm{e}^{\frac{E_2 - E_1}{\mathrm{i}\hbar} t} u_2 \\ \mathrm{i}\hbar \dot{u}_2 &= V(t) \mathrm{e}^{\frac{E_1 - E_2}{\mathrm{i}\hbar} t} u_1 \end{aligned}. \tag{19.101}$$

For small times we have approximately

$$u_1 \approx 1 \qquad u_2 \approx 0 \tag{19.102}$$

and with the definition

$$\omega_{21} = \frac{E_2 - E_1}{\hbar} \tag{19.103}$$

we find

$$\dot{u}_2 \approx \frac{V_0}{2\mathrm{i}\hbar} \left(e^{\mathrm{i}\omega t} + e^{-\mathrm{i}\omega t} \right) e^{\mathrm{i}\omega_{21} t}. \tag{19.104}$$

We neglect the fast oscillating term (this is the so-called rotating wave approximation)

$$u_2 \approx \frac{V_0}{2\mathrm{i}\hbar} \frac{e^{\mathrm{i}(\omega_{21} - \omega)t} - 1}{\omega_{21} - \omega} \tag{19.105}$$

and the transition probability

$$|u_2|^2 \approx \frac{V_0^2}{4\hbar^2} \frac{\sin^2 \left(\frac{\omega_{21} - \omega}{2} t \right)}{(\omega_{21} - \omega)^2} \tag{19.106}$$

shows resonance behavior at $\omega = \omega_{21}$. The transition probability per time is approximately given by the Golden rule expression

$$\frac{|u_2(t)|^2}{t} \approx \frac{2\pi}{\hbar}\left(\frac{V_0}{2}\right)^2 \delta(\hbar\omega - \hbar\omega_{21}). \tag{19.107}$$

At larger times the system oscillates between the two states.[7] Applying the random-phase approximation we neglect the perturbation component with positive frequency

$$\mathrm{i}\hbar\dot{u}_1 = V_0 \mathrm{e}^{\mathrm{i}(\omega_{21}-\omega)t} u_2 \tag{19.108}$$

$$\mathrm{i}\hbar\dot{u}_2 = V_0 \mathrm{e}^{-\mathrm{i}(\omega_{21}-\omega)t} u_1 \tag{19.109}$$

and substitute

$$u_1 = a_1 \mathrm{e}^{\mathrm{i}(\omega_{21}-\omega)t} \tag{19.110}$$

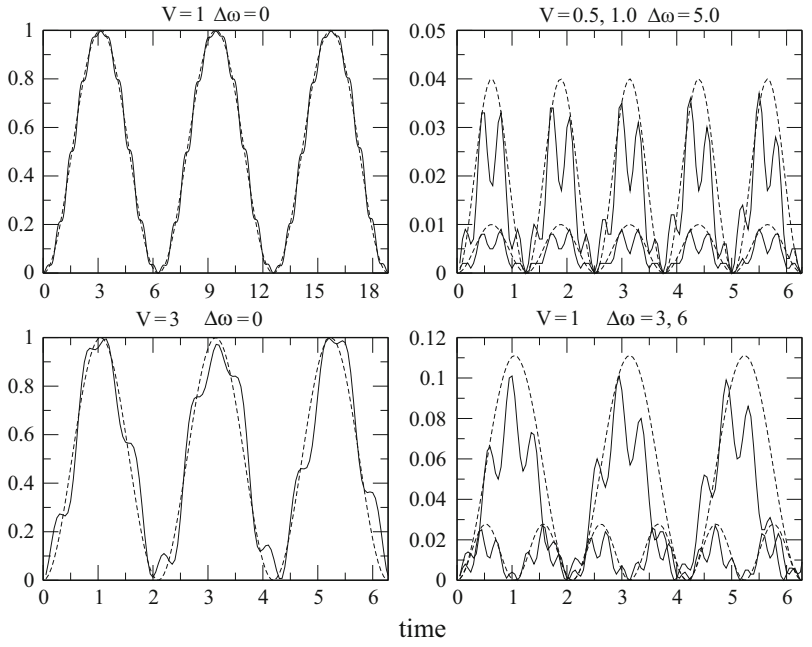


Fig. 19.12 Simulation of a two-state system in an oscillating field. The equations of motion (19.99) are integrated with the fourth-order Runge-Kutta method. At resonance the system oscillates between the two states with the frequency $V/\hbar$. The *dashed curves* show the corresponding solution of a two-level system with constant coupling (Sect. 19.4.2).

[7] These are so-called Rabi oscillations.

to have

$$\mathrm{i}\hbar(\dot{a_1} + a_1\mathrm{i}(\omega_{21} - \omega))\mathrm{e}^{\mathrm{i}(\omega_{21}-\omega)t} = V_0\mathrm{e}^{\mathrm{i}(\omega_{21}-\omega)t}u_2 \tag{19.111}$$

$$\mathrm{i}\hbar\dot{u}_2 = V_0\mathrm{e}^{-\mathrm{i}(\omega_{21}-\omega)t}\mathrm{e}^{\mathrm{i}(\omega_{21}-\omega)t}a_1 \tag{19.112}$$

or

$$\mathrm{i}\hbar\dot{a}_1 = \hbar(\omega_{21} - \omega)a_1 + V_0u_2 \tag{19.113}$$

$$\mathrm{i}\hbar\dot{u}_2 = V_0a_1 \tag{19.114}$$

which shows that the system behaves approximately like a two-level system with a constant interaction V_0 and an energy gap $\hbar(\omega_{12} - \omega) = E_2 - E_1 - \hbar\omega$ (Fig. 19.12).

19.5 Description of a Two-State System with the Density Matrix Formalism

We consider now a two-state system which is coupled to a thermal bath. This model is relevant not only for coherent optical excitation but also for NMR phenomena [120].

19.5.1 Density Matrix Formalism

The density matrix formalism is very suitable for the description of an ensemble of quantum systems or the average evolution of a quantum system in contact with a thermal bath [113].

19.5.1.1 Density Matrix for an Ensemble of Systems

Consider a thermal ensemble of systems. Their wave functions are expanded with respect to basis functions $|\psi_s\rangle$ as

$$|\psi\rangle = \sum C_s|\psi_s\rangle. \tag{19.115}$$

The ensemble average of an operator A is given by

$$\overline{\langle A\rangle} = \overline{\langle\psi A\psi\rangle} = \overline{\langle\sum C_s^*\psi_s A C_{s'}\psi_{s'}\rangle} \tag{19.116}$$

$$= \sum \overline{C_s^*C_{s'}}A_{ss'} = \mathrm{tr}(\rho A) \tag{19.117}$$

with the statistical operator

$$\rho_{s's} = \sum \overline{C_s^*C_{s'}}. \tag{19.118}$$

19.5.1.2 Characterization of the Elements of the Density Matrix

The wave function of a N-state system is a linear combination

$$|\psi\rangle = C_1|\psi_1\rangle + C_2|\psi_2\rangle + \cdots C_N|\psi_N\rangle. \tag{19.119}$$

The diagonal elements of the density matrix are the occupation probabilities

$$\rho_{11} = \overline{|C_1|^2} \qquad \rho_{22} = \overline{|C_2|^2} \cdots \qquad \rho_{NN} = \overline{|C_N|^2}. \tag{19.120}$$

The non-diagonal elements measure the correlation of two states[8]

$$\rho_{12} = \rho_{21}^* = \overline{C_2^* C_1}, \cdots \tag{19.121}$$

19.5.1.3 Equations of Motion for the Density Matrix

The expansion coefficients of

$$|\psi\rangle = \sum C_s|\psi_s\rangle \tag{19.122}$$

can be obtained from the scalar product

$$C_s = \langle\psi_s|\psi\rangle. \tag{19.123}$$

Hence we have

$$C_s^* C_{s'} = \langle\psi|\psi_s\rangle\langle\psi_{s'}|\psi\rangle = \langle\psi_{s'}|\psi\rangle\langle\psi|\psi_s\rangle \tag{19.124}$$

which can be considered to be the s', s matrix element of the operator $|\psi\rangle\langle\psi|$

$$C_s^* C_{s'} = (|\psi\rangle\langle\psi|)_{s's}\,. \tag{19.125}$$

The thermal average is the statistical operator

$$\rho_{s's} = \overline{C_s^* C_{s'}} = \overline{|\psi\rangle\langle\psi|}_{s's} \to \rho = \overline{|\psi\rangle\langle\psi|}. \tag{19.126}$$

From the Schrödinger equation

$$\mathrm{i}\hbar|\dot{\psi}\rangle = H|\psi\rangle \tag{19.127}$$

we find

$$-\mathrm{i}\hbar\langle\dot{\psi}| = \langle H\psi| = \langle\psi|H \tag{19.128}$$

[8] They are often called the "coherence" of the two states.

and hence

$$\mathrm{i}\hbar\dot{\rho} = \mathrm{i}\hbar\left(\overline{|\dot{\psi}\rangle\langle\psi|} + \overline{|\psi\rangle\langle\dot{\psi}|}\right) = \overline{|H\psi\rangle\langle\psi|} - \overline{|\psi\rangle\langle H\psi|}. \tag{19.129}$$

Since the Hamiltonian H is identical for all members of the ensemble we end up with the Liouville–von Neumann equation:

$$\mathrm{i}\hbar\dot{\rho} = H\rho - \rho H = [H, \rho]. \tag{19.130}$$

With respect to a finite basis this becomes explicitly:

$$\mathrm{i}\hbar\dot{\rho}_{ii} = \sum_j H_{ij}\rho_{ji} - \rho_{ij}H_{ji} = \sum_{j\neq i} H_{ij}\rho_{ji} - \rho_{ij}H_{ji} \tag{19.131}$$

$$\begin{aligned}\mathrm{i}\hbar\dot{\rho}_{ik} &= \sum_j H_{ij}\rho_{jk} - \rho_{ij}H_{jk} \\ &= (H_{ii} - H_{kk})\rho_{ik} + H_{ik}(\rho_{kk} - \rho_{ii}) + \sum_{j\neq i,k}(H_{ij}\rho_{jk} - \rho_{ij}H_{jk}).\end{aligned} \tag{19.132}$$

19.5.1.4 Two-State System

Consider a two-state system in a pulsed laser field with Gaussian envelope:

$$H_{12} = \mu E_0 e^{-t^2/t_\mathrm{p}^2}\cos(\omega_L t) \tag{19.133}$$

The equations of motion for the two-state system are

$$\begin{aligned}\mathrm{i}\hbar\dot{\rho}_{11} &= H_{12}\rho_{21} - \rho_{12}H_{21} \\ \mathrm{i}\hbar\dot{\rho}_{22} &= H_{21}\rho_{12} - \rho_{21}H_{12} \\ \mathrm{i}\hbar\dot{\rho}_{12} &= (H_{11} - H_{22})\rho_{12} + H_{12}(\rho_{22} - \rho_{11}) \\ -\mathrm{i}\hbar\dot{\rho}_{21} &= (H_{11} - H_{22})\rho_{21} + H_{21}(\rho_{22} - \rho_{11}).\end{aligned} \tag{19.134}$$

Obviously we have

$$\rho_{11} + \rho_{22} = \text{const} \tag{19.135}$$

and

$$\mathrm{i}\hbar\frac{\partial}{\partial t}(\rho_{11} - \rho_{22}) = 2H_{12}\rho_{21} - 2H_{21}\rho_{12} \tag{19.136}$$

$$\mathrm{i}\hbar\dot{\rho}_{12} = (H_{11} - H_{22})\rho_{12} + H_{12}(\rho_{22} - \rho_{11}) \tag{19.137}$$

$$-\mathrm{i}\hbar\dot{\rho}_{21} = (H_{11} - H_{22})\rho_{21} + H_{21}(\rho_{22} - \rho_{11}). \tag{19.138}$$

The equations of motion can be written as a system of linear equations

$$\mathrm{i}\hbar \begin{pmatrix} \dot{\rho}_{11} \\ \dot{\rho}_{22} \\ \dot{\rho}_{12} \\ \dot{\rho}_{21} \end{pmatrix} = \begin{pmatrix} 0 & 0 & -H_{21} & H_{12} \\ 0 & 0 & H_{21} & -H_{12} \\ -H_{12} & H_{12} & H_{11} - H_{22} & 0 \\ H_{21} & -H_{21} & 0 & H_{22} - H_{11} \end{pmatrix} \begin{pmatrix} \rho_{11} \\ \rho_{22} \\ \rho_{12} \\ \rho_{21} \end{pmatrix} \tag{19.139}$$

or in symbolic form

$$\mathrm{i}\hbar\dot{\rho} = \widehat{L}\rho. \tag{19.140}$$

Hence the same numerical treatment as for the Schrödinger equation can be used (but with larger dimension).

The radiation which is emitted by the two-state system depends on the expectation value of the dipole moment μ and is given by

$$\begin{aligned} \mathrm{Tr}(\rho\mu) &= \mathrm{Tr}\left(\begin{pmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{pmatrix} \begin{pmatrix} 0 & \mu \\ \mu & 0 \end{pmatrix} \right) \\ &= \mathrm{Tr} \begin{pmatrix} \mu\rho_{12} & \mu\rho_{11} \\ \mu\rho_{22} & \mu\rho_{21} \end{pmatrix} = \mu(\rho_{12} + \rho_{21}) = \mu x. \end{aligned} \tag{19.141}$$

19.5.2 Analogy to Nuclear MagneticResonance

The time evolution of the two-state system can be alternatively described with three real variables

$$\begin{aligned} x &= 2\Re(\rho_{12}) = \rho_{12} + \rho_{12}^* \\ y &= -2\Im(\rho_{12}) = \tfrac{1}{\mathrm{i}}(\rho_{12}^* - \rho_{12}) \\ z &= \rho_{11} - \rho_{22} \end{aligned} \tag{19.142}$$

which parametrize the density matrix according to[9]

$$\begin{aligned} \begin{pmatrix} \rho_{11} & \rho_{12} \\ \rho_{21} & \rho_{22} \end{pmatrix} &= \begin{pmatrix} \frac{1+z}{2} & \frac{x-iy}{2} \\ \frac{x+iy}{2} & \frac{1-z}{2} \end{pmatrix} = \frac{1 + x\sigma_x + y\sigma_y + z\sigma_z}{2} \\ &= \frac{1 + \mathbf{x}\boldsymbol{\sigma}}{2}. \end{aligned} \tag{19.143}$$

The equations of motion for x, y, z are

$$\begin{aligned} \mathrm{i}\hbar\dot{z} &= 2(H_{12}\rho_{21} - H_{21}\rho_{12}) \\ \mathrm{i}\hbar\dot{x} &= (H_{11} - H_{22})(\rho_{12} - \rho_{21}) + (H_{12} - H_{21})(\rho_{22} - \rho_{11}) \\ \mathrm{i}\hbar\dot{y} &= \mathrm{i}(H_{11} - H_{22})(\rho_{12} + \rho_{21}) + \mathrm{i}(H_{12} + H_{21})(\rho_{22} - \rho_{11}) \end{aligned} \tag{19.144}$$

[9] The Pauli matrices $\sigma_{x,y,z}$ are explained in (12.111).

and with the definitions

$$V' = \Re(H_{12}) = \frac{H_{12} + H_{12}^*}{2} \quad V'' = \Im(H_{12}) = \frac{H_{12} - H_{12}^*}{2\mathrm{i}}$$
$$\Delta = H_{11} - H_{22} \tag{19.145}$$

we have finally

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} 0 & -\frac{\Delta}{\hbar} & -2\frac{V''}{\hbar} \\ \frac{\Delta}{\hbar} & 0 & -2\frac{V'}{\hbar} \\ 2\frac{V''}{\hbar} & 2\frac{V'}{\hbar} & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{19.146}$$

which can be written as a vector product

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} \frac{2V'}{\hbar} \\ -\frac{2V''}{\hbar} \\ \frac{\Delta}{\hbar} \end{pmatrix} \times \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \boldsymbol{\omega} \times \begin{pmatrix} x \\ y \\ z \end{pmatrix}. \tag{19.147}$$

For a spin-$\frac{1}{2}$ system we can interpret this equation in the following way: The expectation value of the spin vector is

$$\begin{aligned} \frac{\hbar}{2}\langle\psi|\boldsymbol{\sigma}|\psi\rangle &= \frac{\hbar}{2}\begin{pmatrix} C_1^* & C_2^* \end{pmatrix}\begin{pmatrix} \sigma_x \\ \sigma_y \\ \sigma_z \end{pmatrix}\begin{pmatrix} C_1 \\ C_2 \end{pmatrix} \\ &= \hbar \begin{pmatrix} \frac{C_1^* C_2 + C_2^* C_1}{2} \\ \frac{C_1^* C_2 - C_2^* C_1}{2i} \\ \frac{|C_1|^2 - |C_2|^2}{2} \end{pmatrix} \end{aligned} \tag{19.148}$$

and for an ensemble of spin-$\frac{1}{2}$ particles the ensemble average is

$$\frac{\hbar}{2}\langle\boldsymbol{\sigma}\rangle = \hbar \begin{pmatrix} \Re(\rho_{12}) \\ -\Im(\rho_{12}) \\ \frac{1}{2}(\rho_{11} - \rho_{22}) \end{pmatrix} = \frac{\hbar}{2}\begin{pmatrix} x \\ y \\ z \end{pmatrix}. \tag{19.149}$$

Thus $\mathbf{m} = \gamma\frac{\hbar}{2}\begin{pmatrix} x \\ y \\ z \end{pmatrix}$ is the average magnetization vector. The Hamiltonian of a spin-$\frac{1}{2}$ particle in a magnetic field is

$$H = -\gamma\frac{\hbar}{2}\boldsymbol{\sigma}\mathbf{B}. \tag{19.150}$$

Assume a typical NMR experiment with a constant field along the z-axis and a rotating field in the xy-plane

$$\mathbf{B} = \begin{pmatrix} B_1 \cos(\omega_\mathrm{f} t) \\ B_1 \sin(\omega_\mathrm{f} t) \\ B_0 \end{pmatrix}. \tag{19.151}$$

Here the Hamiltonian becomes

$$H = -\gamma \frac{\hbar}{2} \begin{pmatrix} B_0 & B_1 \mathrm{e}^{-\mathrm{i}\omega_\mathrm{f} t} \\ B_1 \mathrm{e}^{\mathrm{i}\omega_\mathrm{f} t} & -B_0 \end{pmatrix} \tag{19.152}$$

and from comparison we find

$$\omega_z = \frac{\Delta}{\hbar} = -\gamma B_0 = -\Omega_0 \tag{19.153}$$

$$H_{12} = V' + \mathrm{i}V'' = -\gamma \frac{\hbar}{2} B_1 \mathrm{e}^{-\mathrm{i}\omega_\mathrm{f} t}. \tag{19.154}$$

The equation of motion for the magnetization is

$$\dot{\mathbf{m}} = \gamma \frac{\hbar}{2} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} -\gamma B_1 \cos(\omega_\mathrm{f} t) \\ -\gamma B_1 \sin(\omega_\mathrm{f} t) \\ -\gamma B_0 \end{pmatrix} \times \gamma \frac{\hbar}{2} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{19.155}$$

or in the more conventional form

$$\frac{d\mathbf{m}}{dt} = \gamma \mathbf{m} \times \mathbf{B}. \tag{19.156}$$

19.5.3 Relaxation Processes—Bloch Equations

19.5.3.1 Phenomenological Description

Interaction with the environment will be described with phenomenological relaxation terms. Two different contributions have to be considered:

- Dephasing (loss of coherence) $\propto \mathrm{e}^{-t/T_2}$ with a time constant of T_2 (for NMR this is the spin–spin relaxation time)

$$\frac{\mathrm{d}}{\mathrm{d}t}_{|\mathrm{Rel}} \rho_{12} = -\frac{1}{T_2} \rho_{12} \tag{19.157}$$

- Thermalization $\rho_{22} - \rho_{11} \to \rho_{22}^{\mathrm{eq}} - \rho_{11}^{\mathrm{eq}}$ with time constant T_1 (for NMR this is the spin–lattice relaxation time)

$$\frac{\mathrm{d}}{\mathrm{d}t}_{|\mathrm{Rel}}(\rho_{22}-\rho_{11}) = -\frac{1}{T_1}((\rho_{22}-\rho_{11}) - (\rho_{22}^{\mathrm{eq}}-\rho_{11}^{\mathrm{eq}})). \tag{19.158}$$

Within the vector model this gives the Bloch equations [121] which are used to describe NMR phenomena

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \boldsymbol{\omega} \times \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \begin{pmatrix} \frac{x}{T_2} \\ \frac{y}{T_2} \\ \frac{z-z^{\mathrm{eq}}}{T_1} \end{pmatrix} \tag{19.159}$$

or

$$\frac{\mathrm{d}\mathbf{m}}{\mathrm{d}t} = \gamma \mathbf{m} \times \mathbf{B} - \widehat{R}(\mathbf{m}-\mathbf{m}_{\mathrm{eq}}) \qquad \widehat{R} = \begin{pmatrix} \frac{1}{T_2} & 0 & 0 \\ 0 & \frac{1}{T_2} & 0 \\ 0 & 0 & \frac{1}{T_1} \end{pmatrix}. \tag{19.160}$$

More general relaxation processes for systems with many states can be described with a more general relaxation operator

$$\mathrm{i}\hbar\dot{\rho} = [H,\rho] - \mathrm{i}\hbar\widehat{\Gamma}(\rho-\rho_{\mathrm{eq}}). \tag{19.161}$$

19.5.3.2 Free Precession

Consider the special case $B_z = \text{const}\ B_x = B_y = 0$. With $m_\pm = m_x \pm i m_y$ and the Larmor frequency $\Omega_0 = \gamma B_0$ the equations of motion are

$$\begin{aligned} \dot{m}_+ &= -i\Omega_0 m_+ - \tfrac{m_+}{T_2} \\ \dot{m}_z &= -\tfrac{m_z - m_0}{T_1} \end{aligned} \tag{19.162}$$

with the solution

$$\begin{aligned} m_+ &= m_+(0)\mathrm{e}^{-i\Omega_0 t - t/T_2} \\ m_z &= m_0 + (m_z(0) - m_0)\mathrm{e}^{-t/T_1}\,. \end{aligned} \tag{19.163}$$

The corresponding density matrix is diagonal

$$H = \begin{pmatrix} H_{11} & 0 \\ 0 & H_{22} \end{pmatrix} \tag{19.164}$$

and the equations of motion are

$$\begin{aligned} \mathrm{i}\hbar\tfrac{\partial}{\partial t}(\rho_{11}-\rho_{22}) &= -\tfrac{(\rho_{11}-\rho_{22})-(\rho_{11}^{eq}-\rho_{22}^{eq})}{T_1} \\ \mathrm{i}\hbar\tfrac{\partial}{\partial t}\rho_{12} &= \Delta\rho_{12} - \mathrm{i}\hbar\tfrac{1}{T_2}\rho_{12} \end{aligned} \tag{19.165}$$

with the solution

$$\begin{array}{c}(\rho_{11}-\rho_{22}) = (\rho_{11}^{\mathrm{eq}}-\rho_{22}^{\mathrm{eq}}) + [(\rho_{11}(0)-\rho_{22}(0)) - (\rho_{11}^{\mathrm{eq}}-\rho_{22}^{\mathrm{eq}})]\mathrm{e}^{-t/T_1} \\ \rho_{12} = \rho_{12}(0)\mathrm{e}^{-\mathrm{i}\frac{\Delta}{\hbar}t - t/T_2}\end{array} . \quad (19.166)$$

19.5.3.3 Stationary Solution with Monochromatic Excitation

For the monochromatic rotating field

$$\mathbf{B} = \begin{pmatrix} B_1\cos(\omega_\mathrm{f} t) \\ B_1\sin(\omega_\mathrm{f} t) \\ B_0 \end{pmatrix} \quad H_{12} = V_0\mathrm{e}^{-\mathrm{i}\omega_\mathrm{f} t} \quad (19.167)$$

the solution of the Bloch equations

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \boldsymbol{\omega} \times \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \begin{pmatrix} \frac{x}{T_2} \\ \frac{y}{T_2} \\ \frac{z-z^{\mathrm{eq}}}{T_1} \end{pmatrix} \quad (19.168)$$

can be found explicitly. After transforming to a coordinate system which rotates along the z-axis with frequency ω_0

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \cos(\omega_\mathrm{f} t) & \sin(\omega_\mathrm{f} t) & 0 \\ -\sin(\omega_\mathrm{f} t) & \cos(\omega_\mathrm{f} t) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad (19.169)$$

the equation of motion simplifies to

$$\begin{pmatrix} \dot{x}' \\ \dot{y}' \\ \dot{z}' \end{pmatrix} = \begin{pmatrix} -\frac{1}{T_2} & \Omega_0-\omega_\mathrm{f} & 0 \\ -\Omega_0+\omega_\mathrm{f} & -\frac{1}{T_2} & -\frac{2V_0}{\hbar} \\ 0 & \frac{2V_0}{\hbar} & -\frac{1}{T_1} \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{z^{\mathrm{eq}}}{T_1} \end{pmatrix} \quad (19.170)$$

with the stationary solution (Fig. 19.13)

$$\frac{z^{\mathrm{eq}}}{1+4\frac{V_0^2}{\hbar^2}T_1T_2 + T_2^2(\omega_\mathrm{f}-\Omega_0)^2} \begin{pmatrix} 2T_2^2\frac{V_0}{\hbar}(\omega_\mathrm{f}-\Omega_0) \\ -2T_2\frac{V_0}{\hbar} \\ 1+T_2^2(\omega_\mathrm{f}-\Omega_0)^2 \end{pmatrix} . \quad (19.171)$$

Saturation appears for

$$4\frac{V_0^2}{\hbar^2}T_1T_2 \gg 1 + (\omega_\mathrm{f}-\Omega_0)^2T_2^2 . \quad (19.172)$$

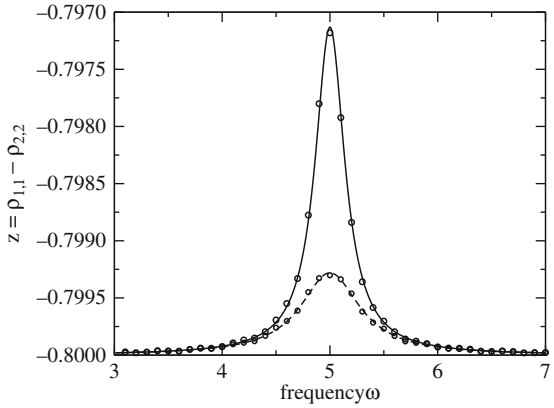


Fig. 19.13 Resonance line. The equations of motion of the two-level system including relaxation terms are integrated with the fourth-order Runge–Kutta until a steady state is reached. Parameters are $\omega_0 = 5, z_{\mathrm{eq}} = -0.8, V = 0.01$, and $T_1 = T_2 = 3.0, 6.9$. The change of the occupation difference is shown as a function of frequency (*circles*) and compared with the steady-state solution (19.171)

The width of the Lorenz line depends on the intensity (saturation broadening) (Fig. 19.14)

$$\Delta\omega = \frac{1}{T_2} \to \frac{1}{T_2}\sqrt{1 + 4\frac{V_0^2}{\hbar^2}T_1T_2}. \tag{19.173}$$

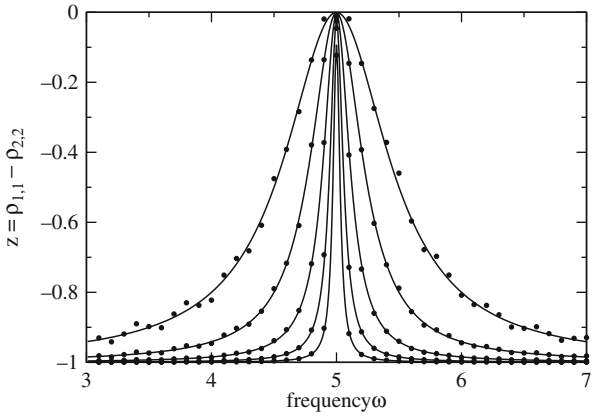


Fig. 19.14 Power saturation and broadening. The resonance line is investigated as a function of the coupling strength V and compared with the stationary solution (19.171) to observe the broadening of the line width (19.173). Parameters are $\omega_0 = 5, z_{\mathrm{eq}} = -1.0, T_1 = T_2 = 100$, and $V = 0.5, 0.25, 0.125, 0.0625, 0.03125$

19.5.3.4 Excitation by a Resonant Pulse

For a resonant pulse $\omega_f = \Omega_0$ with envelope $V_0(t)$ the equation of motion in the rotating system is

$$\begin{pmatrix} \dot{x}' \\ \dot{y}' \\ \dot{z}' \end{pmatrix} = \begin{pmatrix} -\frac{1}{T_2} & 0 & 0 \\ 0 & -\frac{1}{T_2} & -\frac{2V_0(t)}{\hbar} \\ 0 & \frac{2V_0(t)}{\hbar} & -\frac{1}{T_1} \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{z^{eq}}{T_1} \end{pmatrix}. \tag{19.174}$$

If the relaxation times are large compared to the pulse duration we have the approximate solution

$$x' = x'_0 \tag{19.175}$$

$$y' = \frac{y'_0 + iz'_0}{2} e^{i\Phi} + \frac{y'_0 - iz'_0}{2} e^{-i\Phi} \tag{19.176}$$

$$z' = \frac{z'_0 - ix'_0}{2} e^{i\Phi} + \frac{z'_0 + ix'_0}{2} e^{-i\Phi} \tag{19.177}$$

with the phase angle

$$\Phi = \int_{-\infty}^{t} \frac{2V_0(t')}{\hbar} dt'. \tag{19.178}$$

For a total phase of $\Phi(\infty) = \pi$ (π-pulse) the y- and z-component change their sign. The transition between $z = 1$ and $z = -1$ corresponds to a spin flip. On the other

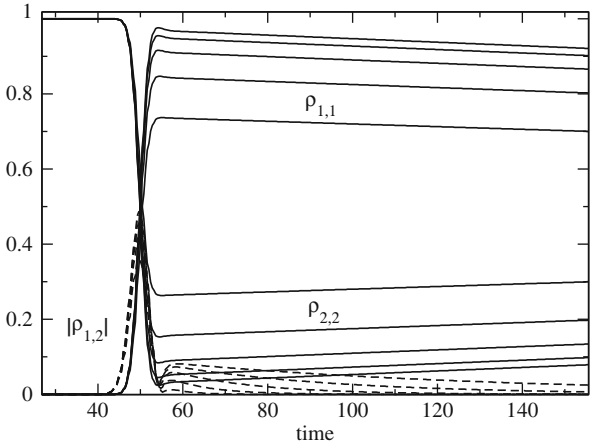


Fig. 19.15 Spin flip by a π-pulse. The equation of motion of the Bloch vector (19.174) is solved with the fourth-order Runge–Kutta for an interaction pulse with a Gaussian shape. The pulse is adjusted to obtain a spin flip, which is a simple model for the invert operation on a Qubit. The influence of dephasing processes is studied. Parameters are $T_1 = 2000, t_p = 3.75, V_0 = 0.5$, and $T_2 = 5, 10, 20, 40, 80$. The occupation probabilities of the two states (*solid curves*) and the coherence (*broken curves*) are shown for several values of the dephasing time

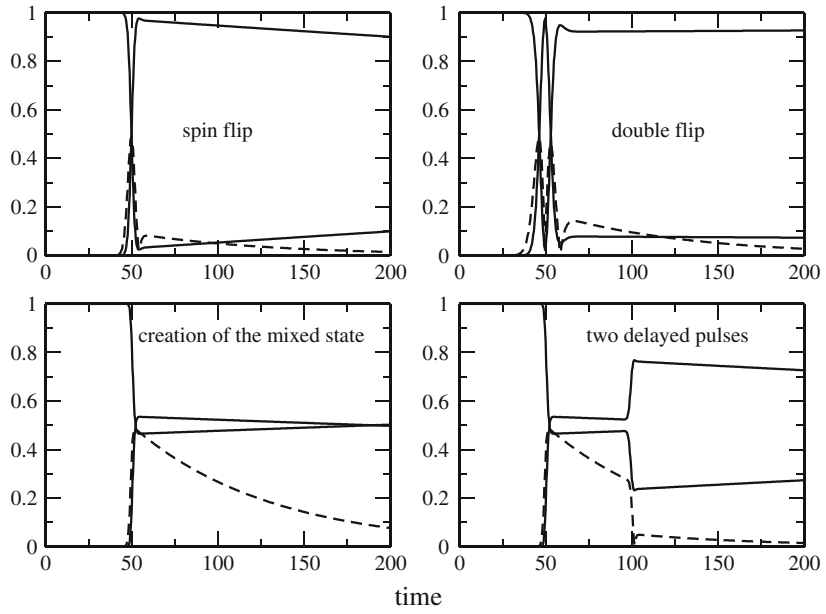


Fig. 19.16 Simulation of $\frac{N}{2}\pi$ pulses. The pulse duration is varied to obtain multiple spin flips or to create the coherently mixed state. Finally a simple measurement of the coherence decay with two delayed π-pulses is simulated, where the first pulse creates the coherently mixed state and the second pulse measures the remaining coherence after a variable delay time

hand a $\frac{\pi}{2}$-pulse converts a pure state into a completely mixed state and vice versa (Figs. 19.15, 19.16).

Problems

Problem 19.1 Schrödinger Equation

In this computer experiment we solve the Schrödinger 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 either a harmonic parabola or a fourth-order double well. The initial width and position of the wave packet can be varied under the constraint $V(x_0) = 0$.

Try to generate the time-independent 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 19.2 Two-Level System

In this computer experiment a two-level system is simulated. Amplitude and frequency of an external field can be varied as well as the energy gap between the two levels (see Fig. 19.3).
Compare the time evolution at resonance and away from it.

Problem 19.3 Three-Level System

In this computer experiment a three-level 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. 19.5).

Problem 19.4 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. 19.7).

Problem 19.5 Landau–Zener Model

This computer experiment simulates the Landau–Zener model. The coupling strength and the nuclear velocity can be varied (see Fig. 19.9).
Try to find parameters for an efficient crossing of the states.

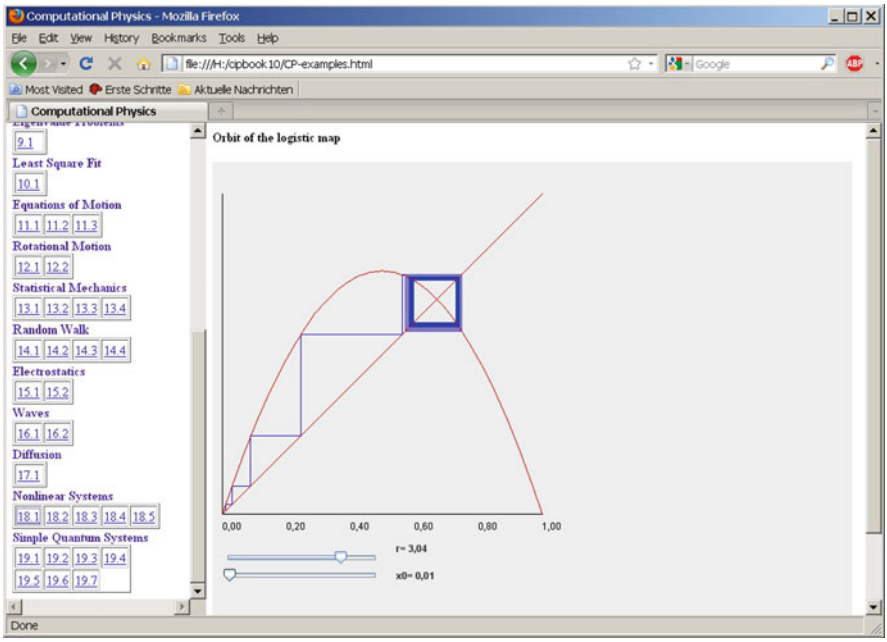
Problem 19.6 Resonance Line

In this computer experiment a two-level system with damping is simulated. The resonance curve is calculated from the steady-state occupation probabilities (see Figs. 19.13 and 19.14).
Study the dependence of the line width on the intensity (power broadening).

Problem 19.7 Spin Flip

The damped two-level system is now subject to an external pulsed field (see Figs. 19.15 and 19.16).
Try to produce a coherent superposition state ($\pi/2$ pulse) or a spin flip (π pulse).
Investigate the influence of decoherence.

Appendix: Performing the Computer Experiments



The computer experiments are realized as Java-applets which can be run in any browser that has the Java plugin installed without installing anything else. 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 an open source and allows quick generation of graphical user interfaces.

After downloading and unzipping the zipped file from extras.springer.com you have two options:

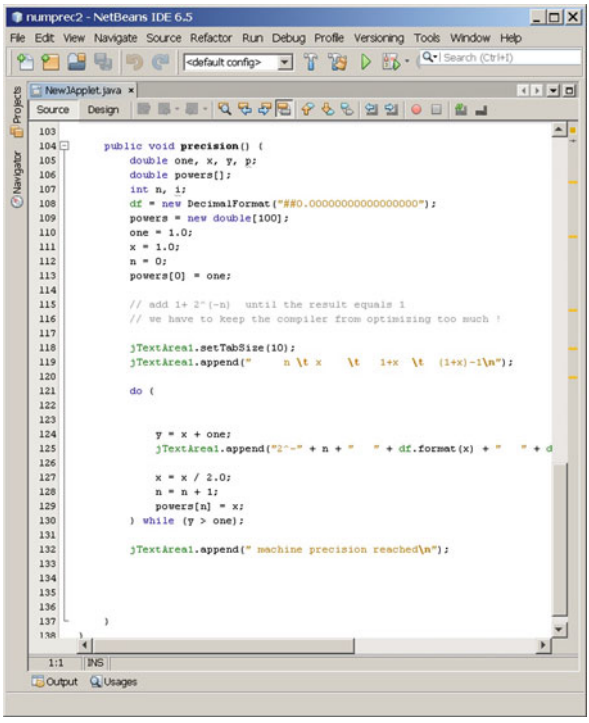
P.O.J. Scherer, *Computational Physics*, DOI 10.1007/978-3-642-13990-1,

Run a Program in your Browser

Open the file CP-examples.html in your browser. If the Java-plugin is installed properly you can start any one of the programs by simply clicking its number in the left-hand frame.

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 HTML/code/. You may have a look at the source code and compile and run it.



References

1. Institute for Electrical and Electronics Engineers, *IEEE Standard for Binary Floating-Point Arithmetic*. (ANSI/IEEE Std 754–1985)
2. J. Stoer, R. Bulirsch, *Introduction to Numerical Analysis*, 3rd revised edn. (Springer, New York, 2010). ISBN 978-1441930064
3. H. Jeffreys, B.S. Jeffreys, *Lagrange's Interpolation Formula, §9.011* in *Methods of Mathematical Physics*, 3rd edn. (Cambridge University Press, Cambridge, 1988), p. 260
4. H. Jeffreys, B.S. Jeffreys, *Divided Differences §9.012 in Methods of Mathematical Physics*, 3rd edn. (Cambridge University Press, Cambridge, England, 1988), pp. 260–264
5. E.H. Neville, Indian Math. Soc. **20**, 87 (1933)
6. I.J. Schoenberg, Quart. Appl. Math. 4, 45–99, 112–141 (1946)
7. G. Nürnberger, *Approximation by Spline Functions* (Springer, Berlin, 1989) ISBN 3-540-51618-2
8. L.F. Richardson, Phil. Trans. R. Soc. Lond. A **210**, 307–357 (1911)
9. 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
10. 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
11. G.H. Golub, C.F. Van Loan, *Matrix Computations*, 3rd edn. (Johns Hopkins, Baltimore, MD, 1976) ISBN 978-0-8018-5414-9
12. 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
13. J. Sherman, Winifred J. Morrison, Ann. Math. Stat. **20**, 621 (1949).
14. I.N. Bronshtein, K.A. Semendyayev, *Handbook of Mathematics*, 3rd edn. (Springer, New York, NY 1997), p. 892
15. H. Jeffreys, B.S. Jeffreys, *Methods of Mathematical Physics*, 3rd edn. (Cambridge University Press, Cambridge, 1988) pp. 305–306
16. W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Successive Overrelaxation (SOR)* in: *Numerical Recipes, The Art of Scientific Computing*, 3rd edn. (Cambridge University Press, Cambridge, 2007) pp. 1061–1065
17. M.R. Hestenes, E. Stiefel, J. Res. Natl. Bur. Stand. **49** ,435 (1952)
18. R. Fletcher, C. Reeves, Comput. J. **7**, 149 (1964)
19. C.G. Broyden, J. Inst. Math. Appl. **6**, 76 (1970)
20. R. Fletcher, Comput. J. **13**, 317 (1970)
21. D. Goldfarb, Math. Comput. **24**, 23 (1970)
22. D.F. Shanno, Math. Comput. **24**, 647 (1970)
23. Fredric j. Harris, Proc. IEEE **66** , 51 (1978)

24. G. Goertzel, Am. Math. Mon. **65**, 34 (1958)
25. E. I. Jury, *Theory and Application of the Z-Transform Method* (Krieger, 1973) ISBN 0-88275-122-0.
26. P. Duhamel, M. Vetterli, Signal Process **19**, 259 (1990)
27. H.J. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms* (Springer, Berlin, 1990).
28. J.W. Cooley, J.W. Tukey, Math. Comput. **19**, 297 (1965)
29. N. Metropolis, S. Ulam, J. Am. Stat. Assoc. **44**, 335 (1949)
30. G.S. Fishman, *Monte Carlo: Concepts, Algorithms, and Applications.* (Springer, New York, 1996) ISBN 038794527X
31. C.P. Robert, G. Casella, *Monte Carlo Statistical Methods*, 2nd edn. (Springer, New York, 2004) ISBN 0387212396
32. R.E. Caflisch, *Monte Carlo and Quasi-Monte Carlo Methods*, Acta Numerica, vol. 7 (Cambridge University Press, Cambridge, 1998) pp. 1–49
33. Richtmyer, *Principles of Modern Mathematical Physics* I (Springer, Berlin Heidelberg, New York, 1978)
34. J. Rice, *Mathematical Statistics and Data Analysis*, 2nd edn. (Duxbury Press, Belmont, 1995) ISBN 0-534-20934-3
35. G. Marsaglia, A. Zaman, Ann. Appl. Probab. **1**, 462 (1991)
36. G.E.P. Box, M.E. Muller, Ann. Math. Stat. **29**, 610 (1958)
37. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, J. Chem. Phys. **21**, 1087 (1953)
38. C. Lanczos, J. Res. Natl. Bureau Stand. **45**, 255 (1951)
39. J.A. Richards, *Remote Sensing Digital Image Analysis* (Springer, Berlin Heidelberg, 1993)
40. A.E. Garcia, Phys. Rev. Lett. **86**, 2696 (1992)
41. T.D. Romo et al., Proteins **22**, 311 (1995)
42. D.P. Derrarr et al., in *A Practical Approach to Microarray Data Analysis*, (Kluwer, 2003) pp. 91
43. J.C. Butcher, *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods* (Wiley, Chichester and New York, 1987)
44. W.B. Gragg, SIAM J. Num. Anal. **2**, 384 (1965)
45. L.F. Shampine, IMA J. Num. Anal. **3**, 383 (1983)
46. L.F. Shampine, L.S. Baca, Numer. Math. **41**, 165 (1983)
47. I.P. Omelyan, I.M. Mryglod, R. Folk, Comput. Phys. Comm. **151**, 272 (2003)
48. Shan-Ho Tsai et al., Braz. J. Phys. **34**, 384 (2004)
49. M. Tuckerman, B.J. Berne, J. Chem. Phys. **97**, 1990 (1992)
50. M.P. Allen, D.J. Tildesley, *Computer Simulation of Liquids* (Oxford University Press, 1989) ISBN 0-19-855645-4
51. R. Sonnenschein, A. Laaksonen, E. Clementi, J. Comput. Chem. **7**, 645 (1986)
52. I.P. Omelyan, Phys. Rev. **58**, 1169 (1998)
53. I.P. Omelyan, Comput. Phys. Comm. **109**, 171 (1998)
54. H. Goldstein, *Klassische Mechanik* (Akademische Verlagsgesellschaft, Frankfurt a.Main, 1974)
55. I.P. Omelyan, Comput. Phys. **12**, 97 (1998)
56. D.C. Rapaport, *The Art of Molecular Dynamics Simulation*. (Cambridge University Press, Cambridge, 2004) ISBN 0-521-44561-2.
57. D. Frenkel, B. Smit, *Understanding Molecular Simulation: from algorithms to applications* (Academic, San Diego, CA, 2002), ISBN 0-12-267351-4
58. J.M. Haile, *Molecular Dynamics Simulation: Elementary Methods*, (John Wiley & sons, New York, 2001) ISBN 0-471-18439-X
59. A. Leach, *Molecular Modelling: Principles and Applications*, 2nd edn. (Harlow: Prentice Hall, 2001) ISBN 978-0582382107

60. T. Schlick, *Molecular Modeling and Simulation* (Springer, New York, NY, 2002) ISBN 0-387-95404-X
61. F. Schwabl, *Statistical Mech*anics (Springer, Berlin, 2003)
62. H. Risken, *The Fokker-Planck Equation* (Springer, Berlin Heidelberg, 1989)
63. E. Ising, Beitrag zur Theorie des Ferromagnetismus, Z. Phys. **31**, 253–258 (1925). doi:10.1007/BF02980577
64. K. Binder, in *"Ising model" Encyclopedia of Mathematics*, Suppl. vol. 2, ed. by R. Hoksbergen (Kluwer, Dordrecht, 2000), pp. 279–281
65. L. Onsager, Phys. Rev. **65**, 117 (1944)
66. B.M. McCoy, T.T. Wu, *The Two-Dimensional Ising Model* (Harvard University Press, Cambridge, MA, 1973) ISBN 0674914406
67. K. Pearson, The problem of the Random Walk. Nature **72**, 294 (1905)
68. A.A. Markov, in *Theory of Algorithms* [Translated by Jacques J. Schorr-Kon and PST staff] Imprint (Academy of Sciences of the USSR, Moscow, 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, vol. 42. Original title: Teoriya algorifmov. [QA248.M2943 Dartmouth College library. U.S. Dept. of Commerce, Office of Technical Services, number OTS 60–51085.]
69. A.A. Markov, in *Extension of the limit theorems of probability theory to a sum of variables connected in a chain* reprinted in Appendix B ed. by R. Howard. Dynamic Probabilistic Systems, vol. 1 (Wiley, Markov Chains, 1971)
70. P. Fluekiger, H.P. Luethi, S. Portmann, J. Weber, MOLEKEL 4.0 (Swiss National Supercomputing Centre CSCS, Manno, Switzerland, 2000)
71. W.L. Mattice, U.W. Suter, *Conformational Theory of Large Molecules* (Wiley Interscience, New York, NY, 1994). ISBN 0-471-84338-5
72. R. Brown, Phil. Mag. **4**, 161 (1828)
73. A. Einstein, Ann. Phys. **17**, 549 (1905)
74. A. Einstein, *Investigations on the Theory of Brownian Movement* (Dover, New York, NY, 1956)
75. C. Yu Zhu, Andreas Cangellaris, *Multigrid Finite Element Methods for Electromagnetic Field Modeling* (John Wiley & sons, New York, 2006), p. 132 ff. ISBN 0471741108
76. M.T. Heath, § 11.5.7 *Multigrid Methods. Scientific Computing: An Introductory Survey* (McGraw-Hill Higher Education, New York 2002) p. 478 ff. ISBN 007112229X
77. R.E. Bruccoleri, J. Novotny, M.E. Davis, K.A. Sharp, J. Comp. Chem. **18**, 268 (1997)
78. F. Fogolari, A. Brigo, H. Molinari, J. Mol. Recognit **15**, 377 (2002)
79. G.L. Gouy, J. Phys. **9**, 457 (1910)
80. D.L. Chapman, Philos. Mag. **25**, 475 (1913)
81. A. Nicholls, B. Honig, J. Comp. Chem. **12**, 435 (1990)
82. G. Wunsch, *Feldtheorie*, (VEB Technik, Berlin, 1973)
83. A.H. Boschitsch, M.O. Fenley, H.X. Zhou, J. Phys. Chem. B **106**, 2741 (2002)
84. A.H. Juffer et al., J. Phys. Chem. B **101**, 7664 (1997)
85. J.S. Bader et al., J. Chem. Phys.**106**, 2372 (1997)
86. T. Simonson, Rep. Prog. Phys. **66**, 737 (2003)
87. J.G. Kirkwood, J. Chem. Phys. **2**, 351 (1934)
88. *Solid-State Physics: An Introduction to Principles of Materials Science* (Advanced Texts in Physics) Harald Ibach, Hans Lüth (Springer, Berlin, 2003)
89. R. Courant, K. Friedrichs, H. Lewy, Math. Annalen **100**, 32 (1928)
90. Y.A Cengel, *Heat transfer-A Practical Approach*, 2nd edn. (McGraw Hill Professional, 2003) p. 26 ISBN 0072458933, 9780072458930, New York
91. Fourier, Joseph. (1878). *The Analytical Theory of Heat*. (Cambridge University Press, Cambridge, reissued by Cambridge University Press, 2009) ISBN 978-1-108-00178-6)
92. A. Fick, Phil. Mag. **10**, 30 (1855)

93. J. Crank, P. Nicolson, Proc. Camb. Phil. Soc. **43**, 50 (1947)
94. J.W. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods, Texts in Applied Mathematics*, vol. 22 (Springer, Berlin, 1995)
95. Diederich Hinrichsen, Anthony J. Pritchard, *Mathematical Systems Theory I—Modelling, State Space Analysis, Stability and Robustness* (Springer, Berlin, 2005) ISBN 0-978-3-540-441250
96. Khalil, H. K, *Nonlinear Systems* (Prentice Hall, Englewood Cliffs, NJ, 2001) ISBN 0-13-067389-7
97. Vasile I. Istratescu, in *Fixed Point Theory, An Introduction*, D. Reidel Publ. Comp. Dordrecht, Boston, London, 1981
98. S.H. Strogatz, *Nonlinear dynamics and Chaos: Applications to Physics, Biology, Chemistry, and Engineering* (Perseus, New York, NY, 2001) ISBN 0-7382-0453-6
99. J.D. Murray, *Mathematical Biology: I. An Introduction*, 3rd edn. 2 vols (Springer, Berlin, 2002) ISBN 0-387-95223-3
100. E. Renshaw, *Modelling Biological Populations in Space and Time* (C.U.P., Utah, 1991) ISBN 0-521-44855-7
101. P. Grindrod, Patterns and Waves, *The Theory and Applications of Reaction-Diffusion Equations* (Clarendon Press, Oxford, 1991)
102. P.C. Fife, in *Mathematical Aspects of Reacting and Diffusing Systems* (Springer, Berlin, 1979)
103. A.M. Lyapunov, *Stability of Motion* (Academic, New-York, London, 1966)
104. P.F. Verhulst, *Mémoires de l'Académie Royale des Sciences et Belles Lettres de Bruxelles* vol. 18 (Bruxelles, 1845) p. 1–42
105. A.J. Lotka, in *Elements of Physical Biology* (Williams and Wilkins, Baltimore, 1925)
106. V. Volterra, Mem. R. Accad. Naz. dei Lincei **2**, 31 (1926)
107. C.S. Holling, Canad. Entomol. **91**, 293 (1959)
108. C.S. Holling, Canad. Entomol. **91**, 385 (1959)
109. J.T. Tanner, Ecology **56**, 855 (1975)
110. A.M. Turing, Phil. Trans. R. Soc. B **237**, 37 (1952)
111. E. Schrödinger, Phys. Rev. **28**, 1049 (1926)
112. D. Hilbert, L. Nordheim, J. von Neumann, John Mathe. Annalen **98**, 1 (1927)
113. F. Schwabl, *Quantum Mechanics*, 4th edn. (Springer, Berlin, 2007)
114. L. Diosi, *A Short Course in Quantum Information Theory* (Springer, Berlin, 2007)
115. S.E. Koonin, C.M. Dawn, *Computational Physics* (Perseus Books, 1990) ISBN: 978-0201127799
116. M. Bixon, J. Jortner, J. Chem. Phys. **48**, 715 (1986)
117. B.I. Stepanov, V.P. Gribkovskii, in *Theory of Luminescence* [by] B.I. Stepanov, V.P. Gribkovskii, [Translated [from the Russian] by Scripta Technica Ltd., ed. by S. Chomet (lliffe, London, 1968] (Butterworth, London)
118. L. Landau, *Zur Theorie der Energieübertragung bei Stössen II*, Phys. Z. Sowjetunion 2, 46–51 (1932)
119. C. Zener, Proc. Royal Soc. Lond. A **137**(6), 696–702 (1932)
120. A. Yariv, *Quantum Electronics* (Wiley, New York, NY, 1975)
121. F. Bloch, Nuclear Induction, Phys. Rev. **70**, 460 (1946)

Index