Chapter 1
Some Basic Remarks

1.1 Motivation

Computational Physics aims at solving physical problems by means of numerical methods developed in the field of numerical analysis [1, 2]. According to I. JACQUES and C. JUDD [3], it is defined as:

> Numerical analysis is concerned with the development and analysis of methods for the numerical solution of practical problems.

Although the term *practical problems* remained unspecified in this definition, it is certainly necessary to reflect on ways to find approximate solutions to complex problems which occur regularly in natural sciences. In fact, in most cases it is not possible to find analytic solutions and one must rely on good approximations. Let us give some examples.

Consider the definite integral

$$\int_a^b \mathrm{d}x \exp\left(-x^2\right) \ , \tag{1.1}$$

which, for instance, may occur when it is required to calculate the probability that an event following a normal distribution takes on a value within the interval $[a, b]$, where $a, b \in \mathbb{R}$. In contrast to the much simpler integral

$$\int_a^b \mathrm{d}x \exp(x) = \exp(b) - \exp(a) \ , \tag{1.2}$$

the integral (1.1) cannot be solved analytically because there is no elementary function which differentiates to $\exp\left(-x^2\right)$. Hence, we have to approximate this integral in such a way that the approximation is accurate enough for our purpose. This example illustrates that even mathematical expressions which appear quite

B.A. Stickler, E. Schachinger, *Basic Concepts in Computational Physics*,
DOI 10.1007/978-3-319-27265-8_1

simple at first glance may need a closer inspection when a numerical estimate for the expression is required. In fact, most numerical methods we will encounter within this book have been designed before the invention of modern computers or calculators. However, the applicability of these methods has increased and is still increasing drastically with the development of even more powerful machines. We give another example, namely the oscillation of a pendulum. We know from basic mechanics [4–8] that the time evolution of a frictionless pendulum of mass m and length ℓ in a gravitational field is modeled by the differential equation

$$\ddot{\theta} + \frac{g}{\ell} \sin(\theta) = 0 . \tag{1.3}$$

The solution of this equation describes the oscillatory motion of the pendulum around the origin O within a two-dimensional plane (Fig. 1.1). Here θ is the angular displacement and g is the acceleration due to gravity. Furthermore, a common situation is described by initial conditions of the form:

$$\begin{cases} \theta(0) = \theta_0 , \\ \dot{\theta}(0) = 0 . \end{cases} \tag{1.4}$$

For small initial angular displacements, $\theta_0 \ll 1$, we set in Eq. (1.3) $\sin(\theta) \approx \theta$ and obtain the differential equation of the harmonic oscillator:

$$\ddot{\theta} + \frac{g}{\ell}\theta = 0 . \tag{1.5}$$

Together with the initial conditions (1.4) we arrive at the solution

$$\theta(t) = \theta_0 \cos(\omega t) , \tag{1.6}$$

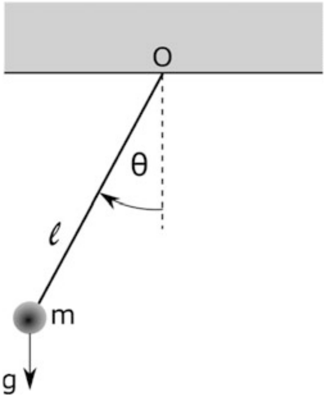


Fig. 1.1 Schematic illustration of the pendulum

with $\omega = \sqrt{g/\ell}$. The period τ of the pendulum follows immediately:

$$\tau = 2\pi\sqrt{\frac{\ell}{g}}\ . \tag{1.7}$$

However, if the approximation of a small angular displacement $\theta_0 \ll 1$ is not applicable, expressions (1.6) and (1.7) will not be valid. Thus, it is advisable to apply energy conservation in order to arrive at analytic results. The total energy of the pendulum is given by:

$$E = \frac{1}{2}mv^2 + mg\ell\left[1 - \cos\left(\theta\right)\right] = \frac{1}{2}mv_0^2 + mg\ell\left[1 - \cos\left(\theta_0\right)\right]\ . \tag{1.8}$$

Here v is the velocity of the point mass m and v_0 and θ_0 are defined by the initial conditions (1.4). Since $\dot{\theta}(0) = 0$ we have

$$\begin{aligned} E &= mg\ell\left[1 - \cos\left(\theta_0\right)\right] \\ &= 2mg\ell\sin^2\left(\frac{\theta_0}{2}\right)\ , \end{aligned} \tag{1.9}$$

where we made use of the relation: $1 - \cos(x) = 2\sin^2(x/2)$. We use this result in Eq. (1.8) and arrive at:

$$\frac{1}{2}v^2 = 2g\ell\left[\sin^2\left(\frac{\theta_0}{2}\right) - \sin^2\left(\frac{\theta}{2}\right)\right]\ . \tag{1.10}$$

Since $v^2 = \ell^2\dot{\theta}^2$ we have

$$\dot{\theta} = 2\sqrt{\frac{g}{\ell}}\sqrt{\sin^2\left(\frac{\theta_0}{2}\right) - \sin^2\left(\frac{\theta}{2}\right)}\ . \tag{1.11}$$

Separation of variables yields

$$\sqrt{\frac{g}{\ell}}t = \frac{1}{2k}\int_0^\theta \frac{\mathrm{d}\varphi}{\sqrt{1 - \frac{1}{k^2}\sin^2\left(\frac{\varphi}{2}\right)}}\ , \tag{1.12}$$

with $k = \sin\left(\theta_0/2\right)$. For $t = \tau$ we have $\theta = \theta_0$ and we obtain for the period

$$\tau = \frac{2}{k}\sqrt{\frac{\ell}{g}}\int_0^{\theta_0} \frac{\mathrm{d}\varphi}{\sqrt{1 - \frac{1}{k^2}\sin^2\left(\frac{\varphi}{2}\right)}}\ . \tag{1.13}$$

Let us transform the above integral into a more convenient form with help of the substitution $k\sin(\alpha) = \sin(\varphi/2)$. Thus, $\alpha \in [0, \pi/2]$ and a straightforward calculation yields:

$$\begin{aligned}\tau &= 4\sqrt{\frac{\ell}{g}}\int_0^{\frac{\pi}{2}} \frac{\mathrm{d}\alpha}{\sqrt{1-k^2\sin^2(\alpha)}}\\ &= 4\sqrt{\frac{\ell}{g}}K_1(k)\,. \end{aligned} \tag{1.14}$$

The function $K_1(k)$ introduced in (1.14) for $k \in \mathbb{R}$ is referred to as the complete elliptic integral of the first kind [9–12]. All these manipulations did not really result in a simplification of the problem at hand because we are still confronted with the integral in Eq. (1.14) which cannot be evaluated without the use of additional approximations which will, in the end, result in a numerical solution of the problem. A natural way to proceed would be to expand the complete elliptic integral in a power series up to order N, where N is chosen in such a way that the *truncation error* $R_N(k)$ becomes negligible. We can find the desired expression in any text on special functions [9, 11, 12]. It reads

$$\begin{aligned}K_1(k) &= \frac{\pi}{2}\sum_{n=0}^{\infty}\left[\frac{(2n)!}{2^{2n}(n!)^2}\right]^2 k^{2n}\\ &= \frac{\pi}{2}\sum_{n=0}^{N}\left[\frac{(2n)!}{2^{2n}(n!)^2}\right]^2 k^{2n} + R_N(k)\,. \end{aligned} \tag{1.15}$$

Imagine now the inverse problem: the period τ is given and the initial angle θ_0 is unknown. Again, we could expand the integrand in a power series and solve the corresponding polynomial for θ_0. However, such an approach would be very inefficient due to two reasons: first of all, we are confronted with the impossibility of finding analytically the roots of a polynomial of order $N > 4$[1] and, secondly, at which value of N should we truncate the power series if θ_0 is unknown? A glance in a book on special functions might give us a better, i.e. more convenient, alternative. Indeed, the inverse function of the elliptic integral $K_1(k)$ with respect to k can be given explicitly in terms of JACOBI elliptic functions [9–12]. Series expansions of these functions have been developed such that we can approximate θ_0 by truncating the respective series.

This example helped to illustrate that we depend on numerical approximations of definite expressions in a multitude of cases. Even if an numerically approximate solution has been found for a particular problem it will be adamant to check quite

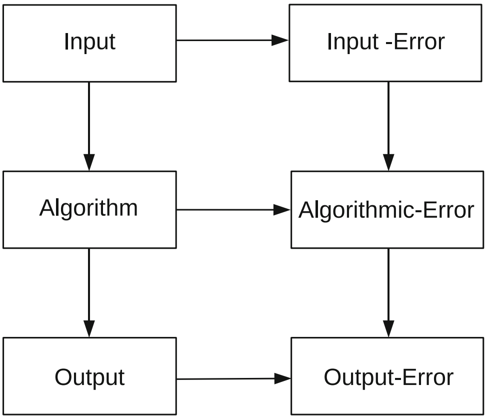
[1] The roots of a real valued polynomial of order $N = 3$ or 4 are referred to as CARDANO's or FERRARI's solutions [13], respectively.

carefully if the approach was (i) justified within the required accuracy, and (ii) if it allowed to improve the induced error of the result. The second point is known as the *stability* of a routine. We will discuss this topic in more detail in Sect. 1.4.

Throughout this book we will be confronted with numerous methods which will allow approximate solutions of problems similar to the two examples illustrated above. First of all, we would like to specify the properties we expect these methods to have. Primarily, the method is to be formulated as an unambiguous mathematical recipe which can be applied to the set of problems it was designed for. Its applicability should be well defined and it should allow to determine an estimate for the error. Moreover, infinite repetition of the procedure should approximate the exact result to arbitrary accuracy. In other words, we want the method to be well defined in *algorithmic* form. Consequently, let us define an algorithm as *a sequence of logical and arithmetic operations (addition, subtraction, multiplication or division) which allows to approximate the solution of the problem under consideration within any accuracy desired.* This implies, of course, that numerical errors will be unavoidable.

Let us classify the occurring errors based on the structure every numerical routine follows: We have input-errors, algorithmic-errors, and output-errors as indicated schematically in Fig. 1.2. This structural classification can be refined: input-errors are divided into roundoff errors and measurement errors contained in the input data; algorithmic-errors consist of roundoff errors during evaluation and of methodological errors due to mathematical approximations; finally, output errors are, in fact, roundoff errors. In Sects. 1.2 and 1.3 we will concentrate on roundoff errors and methodological errors. Since in most cases measurement errors cannot be influenced by the theoretical physicist concerned with numerical modeling, this particular part will not be discussed in this book. However, we will discuss the stability of numerical routines, i.e. the influence of slight modifications of the input parameters on the outcome of a particular algorithm in Sect. 1.4.

Fig. 1.2 Schematic classification of the errors occurring within a numerical procedure



1.2 Roundoff Errors

In fact, since every number is stored in a computer using a finite number of digits, we have to truncate every non-terminating number at some point. For instance, consider $\frac{2}{3} = 0.666666666666\ldots$ which will be stored as 0.6666666667 if the machine allows only ten digits. Actually, computers use binary arithmetic (for which even $0.1_{10} = 0.000110011001100\ldots_2$ is problematic[2]) but for the moment we shall ignore this fact since the above example suffices to illustrate the crucial point. Let $\mathrm{Fl}(x)$ denote the floating-point form of a number x within the numerical range of the machine. For the above example, i.e. a ten digit storage, we have

$$\mathrm{Fl}\left(\frac{2}{3}\right) = 0.6666666667\,. \tag{1.16}$$

This has the consequence that, for instance, $\mathrm{Fl}(\sqrt{3}) \cdot \mathrm{Fl}(\sqrt{3}) \neq \mathrm{Fl}(\sqrt{3} \cdot \sqrt{3})=3$. However, $\mathrm{Fl}(\sqrt{3}) \cdot \mathrm{Fl}(\sqrt{3}) \approx 3$ within the defined range. Before we continue our discussion on roundoff errors we have to introduce the concepts of the absolute and the relative error. We denote the true value of a quantity by y and its approximate value by $\bar{y}$. Then the absolute error ϵ_a is defined as

$$\epsilon_a = |y - \bar{y}|\,, \tag{1.17}$$

while the relative error ϵ_r is given by

$$\epsilon_r = \left|\frac{y - \bar{y}}{y}\right| = \frac{\epsilon_a}{|y|}\,, \tag{1.18}$$

provided that $y \neq 0$. In most applications, the relative error is more significant. This is illustrated in Table 1.1, where it is intuitively obvious that in the second case the approximate value is much better although the absolute error is the same for both examples.

Let us have a look at the relative error of an arbitrary number stored to the k-th digit: We can write an arbitrary number y in the form $y = 0.d_1 d_2 d_3 \ldots d_k d_{k+1} \ldots 10^n$ with $d_1 \neq 0$ and $n \in \mathbb{Z}$. Accordingly, we write its approximate value as $\bar{y} = 0.d_1 d_2 d_3 \ldots d_k 10^n$, where k is the maximum number of digits stored by the

Table 1.1 Illustration of the significance of the relative error

	y	$\bar{y}$	ϵ_a	ϵ_r
(1)	0.1	0.09	0.01	0.1
(2)	1000.0	999.99	0.01	0.00001

[2] A disastrous effect of this binary approximation of 0.1 was discussed by T. Chartier [14].

machine. Hence we obtain for the relative error

$$\begin{aligned}\epsilon_r &= \left|\frac{0.d_1d_2d_3\dots d_kd_{k+1}\dots 10^n - 0.d_1d_2d_3\dots d_k10^n}{0.d_1d_2d_3\dots d_kd_{k+1}\dots 10^n}\right| \\ &= \left|\frac{0.d_{k+1}d_{k+2}\dots 10^{n-k}}{0.d_1d_2d_3\dots 10^n}\right| \\ &= \left|\frac{0.d_{k+1}d_{k+2}\dots}{0.d_1d_2d_3\dots}\right| 10^{-k} \\ &\le \frac{1}{0.1}10^{-k} \\ &= 10^{-k+1}\,. \end{aligned} \tag{1.19}$$

In the last steps we employed that, since $d_1 \neq 0$, we have $0.d_1d_2d_3\dots \geq 0.1$ and accordingly $0.d_{k+1}d_{k+2}\dots < 1$. If the last digit would have been rounded (for $d_{k+1} \geq 5$ we set $d_k = d_k + 1$ otherwise d_k remains unchanged) instead of a simple truncation, the relative error of a variable y would be $\epsilon_r = 0.5 \cdot 10^{-k+1}$.

Whenever an arithmetic operation is performed, the errors of the variables involved is transferred to the result [15]. This can occur in an advantageous or disadvantageous way, where we understand disadvantageous as an increase in the relative error. Particular care is required when two nearly identical numbers are subtracted (*subtractive cancellation*) or when a large number is divided by a, in comparison, small number. In such cases the roundoff error will increase dramatically. We note that it might be necessary to avoid such operations in our aim to design an algorithm which is required to produce reasonable results. An illustrative example and its remedy will be discussed in Sect. 1.3. However, before proceeding to the next section we introduce a lower bound to the accuracy which is achievable with a non-ideal computer, the *machine-number*. The machine-number is smallest positive number η which can be added to another number, such that a change in the result is observed. In particular,

$$\eta = \min_{\delta}\left\{\delta > 0 \middle| 1 + \delta > 1\right\}\,. \tag{1.20}$$

For a (nonexistent) super-computer, which is capable of saving as much digits as desired, η would be arbitrarily small. A typical value for double-precision in FORTRAN or C is $\eta \approx 10^{-16}$.

1.3 Methodological Errors

A methodological error is introduced into the routine whenever a complex mathematical expression is replaced by an approximate, simpler one. We already came across an example when we regarded the series representation of the elliptic integral (1.12) in Sect. 1.1. Although we could evaluate the series up to an arbitrary order N, we are definitely not able to sum up the coefficients to infinite order. Hence, it is not possible to get rid of methodological errors whenever we have to deal with expressions we cannot evaluate analytically. Another intriguing example is the numerical differentiation of a given function. The standard approximation of a derivative reads

$$f'(x_0) = \left.\frac{\mathrm{d}}{\mathrm{d}x} f(x)\right|_{x=x_0} \approx \frac{f(x_0+h)-f(x_0)}{h} \,. \tag{1.21}$$

This approximation is referred to as *finite difference* and will be discussed in more detail in Chap. 2. One would, in a first guess, expect that the obtained value gets closer to the true value of the derivative $f'(x_0)$ with decreasing values of h. From a calculus point of view, this is correct since by definition

$$\left.\frac{\mathrm{d}}{\mathrm{d}x} f(x)\right|_{x=x_0} = \lim_{h\to 0} \frac{f(x_0+h)-f(x_0)}{h} \,. \tag{1.22}$$

However, this is not the case numerically. In particular, one can find a value $\hat{h}$ for which the relative error is minimal, while for values $h < \hat{h}$ and $h > \hat{h}$ the approximation obtained is worse in comparison. The reason is that for small values of h the roundoff errors dominate the result since $f(x_0 + h)$ and $f(x_0)$ almost cancel while $1/h$ is very small. For $h > \hat{h}$, the methodological error, i.e. the replacement of a derivative by a finite difference, controls the result.

We give one further example [16] in order to illustrate the interplay between methodological errors and roundoff errors. We regard the, apparently nonhazardous, numerical solution of a quadratic equation

$$ax^2 + bx + c = 0 \,, \tag{1.23}$$

where $a, b, c \in \mathbb{R}$, $a \neq 0$. The well known solutions read

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \,. \tag{1.24}$$

Cautious because of the above examples, we immediately diagnose the danger of a subtractive cancellation in the expression of x_1 for $b > 0$ or in x_2 for $b < 0$, and rewrite the above expression for x_1:

$$x_1 = \frac{(-b + \sqrt{b^2 - 4ac})}{2a} \frac{(-b - \sqrt{b^2 - 4ac})}{(-b - \sqrt{b^2 - 4ac})} = \frac{2c}{-b - \sqrt{b^2 - 4ac}} \,. \tag{1.25}$$

For x_2 we obtain

$$x_2 = \frac{2c}{-b + \sqrt{b^2 - 4ac}} \ . \tag{1.26}$$

Consequently, if $b > 0$ x_1 should be calculated using Eq. (1.25) and if $b < 0$ Eq. (1.26) should be used to calculate x_2. Moreover, the above expressions can be cast into one expression by setting

$$x_1 = \frac{q}{a} \qquad \text{and} \qquad x_2 = \frac{c}{q} \ , \tag{1.27}$$

with

$$q = -\frac{1}{2}\left[b + \operatorname{sgn}(b)\sqrt{b^2 - 4ac}\right] \ . \tag{1.28}$$

Thus, Eqs. (1.27) and (1.28) can be used to calculate x_1 and x_2 for any sign of b.

1.4 Stability

When a new numerical method is designed *stability* is the third crucial point after roundoff errors and methodological errors [17]. We give an introductory definition:

> An algorithm, equation or, even more general, a problem is referred to as unstable or ill-conditioned if small changes in the input cause a large change in the output.

It will be followed by a couple of elucidating examples [3].[3] To be more specific, let us now, for instance, consider the following system of equations

$$\begin{aligned} x + y &= 2.0, \\ x + 1.01y &= 2.01 \ . \end{aligned} \tag{1.29}$$

These equations are easily solved and give $x = 1.0$ and $y = 1.0$. To make our point we consider now the case in which the right hand side of the second equation of (1.29) is subjected to a small perturbation, i.e. we consider in particular the following system of equations

$$\begin{aligned} x + y &= 2.0, \\ x + 1.01y &= 2.02 \ . \end{aligned} \tag{1.30}$$

[3] Although unstable behavior is not desirable in the first place the discovery of unstable systems was the birth of a specific branch in physics called *Chaos Theory*. We briefly comment on this point at the end of this section.

The corresponding solution is $x = 0.0$ and $y = 2.0$. We observe that a relative change of 0.05 % on the right hand side of the second equation in (1.29) resulted in a 100 % relative change of the solution. Moreover, if the coefficient of y in the second equation of (1.29) were 1.0 instead of 1.01, which corresponds to a relative change of 1 %, the equations would be unsolvable. This is a behavior typical for ill-conditioned problems which, for obvious reasons, should be avoided whenever possible.

We give a second example: We consider the following initial value problem

$$\begin{cases} \ddot{y} - 10\dot{y} - 11y = 0 \ , \\ y(0) = 1, \qquad \dot{y}(0) = -1 \ . \end{cases} \tag{1.31}$$

The general solution is readily obtained to be of the form

$$y = A\exp(-x) + B\exp(11x) \ , \tag{1.32}$$

with numerical constants A and B. The initial conditions yield the unique solution

$$y = \exp(-x) \ . \tag{1.33}$$

The initial conditions are now changed by two small parameters $\delta, \epsilon > 0$ to give:

$$y(0) = 1 + \delta \qquad \text{and} \qquad \dot{y}(0) = -1 + \epsilon \ . \tag{1.34}$$

The unique solution which satisfies these initial conditions is:

$$\bar{y} = \left(1 + \frac{11\delta}{12} - \frac{\epsilon}{12}\right)\exp(-x) + \left(\frac{\delta}{12} + \frac{\epsilon}{12}\right)\exp(11x) \ . \tag{1.35}$$

We calculate the relative error

$$\begin{aligned} \epsilon_r &= \left|\frac{y - \bar{y}}{y}\right| \\ &= \left(\frac{11\delta}{12} - \frac{\epsilon}{12}\right) + \left(\frac{\delta}{12} + \frac{\epsilon}{12}\right)\exp(12x) \ , \end{aligned} \tag{1.36}$$

which indicates that the problem is ill-conditioned since for large values of x the second term definitely overrules the first one.

Another, but not less serious kind of problem is *induced instability*:

> A method is referred to as induced unstable if a small error at one point of the calculation induces a large error at some subsequent point.

Induced instability is particularly dangerous since small roundoff errors are unavoidable in most calculations. Hence, if some part of the whole algorithm is ill-conditioned, the final output will be dominated by the error induced in such a way. Again, an example will help to illustrate such behavior. The definite integral

$$I_n = \int_0^1 \mathrm{d}x\, x^n \exp(x-1) \ , \tag{1.37}$$

is considered. Integration by parts yields

$$I_n = 1 - nI_{n-1} \ . \tag{1.38}$$

This expression can be used to recursively calculate I_n from I_0, where

$$I_0 = 1 - \exp(-1) \ . \tag{1.39}$$

Although the recursion formula (1.38) is exact we will run into massive problems using it. The reason is easily illustrated:

$$\begin{aligned} I_n &= 1 - nI_{n-1} \\ &= 1 - n + n(n-1)I_{n-2} \\ &= 1 - n + n(n-1) - n(n-1)(n-2)I_{n-3} \\ &\ \ \vdots \\ &= 1 + \sum_{k=1}^{n-1}(-1)^k \frac{n!}{(n-k)!} + (-1)^{n-1} n! I_0 \ . \end{aligned} \tag{1.40}$$

Thus, the initial roundoff error included in the numerical value of I_0 is multiplied with $n!$. Note that for large n we have according to STIRLING's approximation

$$n! \approx \sqrt{2\pi} n^{n+\frac{1}{2}} \exp(-n) \ , \tag{1.41}$$

i.e. an initial error increases almost as n^n.

However, Eq. (1.38) can be reformulated to give

$$I_n = \frac{1}{n+1}(1 - I_{n+1}) \ , \tag{1.42}$$

and this opens an alternative method for a recursive calculation of I_n. We can start with some value $N \gg n$ and simply set $I_N = 0$. The error introduced in such a way may in the end not be acceptable, nevertheless, it decreases with every iteration step due to the division by n in Eq. (1.42).

Having discussed some basic features of stability in numerical algorithms we would like to add a few remarks on *Chaos Theory*. Chaos theory investigates dynamical processes which are very sensitive to initial conditions. One of the best known examples for such a behavior is the weather prediction. Although, POINCARÉ already observed chaotic behavior while working on the three body problem, one of the pioneers of chaos theory was E.N. LORENZ [18] (not to be confused with H. LORENTZ, who introduced the LORENTZ transformation). In 1961 he ran weather simulations on a computer of restricted capacity. However, when he tried to reproduce one particular result by restarting the calculation with new parameters calculated the days before, he observed that the outcome was completely different [19]. The reason was that the equations he dealt with were ill-conditioned, and the roundoff error he introduced by simply typing in the numbers of the graphical output, increased drastically, and, hence, produced a completely different result. Nowadays, various physical systems are known which indeed behave in such a way. Further examples are turbulences in fluids, oscillations in electrical circuits, oscillating chemical reactions, population growth in ecology, the time evolution of the magnetic field of celestial bodies, ...

It is important to note, that chaotic behavior induced in such systems is *deterministic*, yet *unpredictable*. This is due to the impossibility of an exact knowledge of the initial conditions required to predict, for instance, the weather over a reasonably long period. A feature which is referred to as the *butterfly effect*: a hurricane can form because a butterfly flapped its wings several weeks before. However, these effects have nothing to do with intrinsically probabilistic properties which are solely a feature of quantum mechanics. In contrast to this, in chaos theory, the future is uniquely determined by initial conditions, however, still unpredictable. This is often referred to as *deterministic chaos*.

It has to be emphasized that chaos in physical systems is a consequence of the equations describing the processes and not a consequence of the numerical method used for modeling. Therefore, it is important to distinguish between the stability of a numerical method and the stability of a physical system in general.

We will come across chaotic behavior again in Sect. 6.3 where we discuss chaotic behavior in the dynamics of the double pendulum [4–8].

1.5 Concluding Remarks

In this chapter we dealt with the basic features of numerical errors one is always confronted with when developing an algorithm. One point we neglected in our discussion is the *computational cost*, i.e. the time a program needs to be executed. Although this is a very important point, it is beyond the scope of this book. However, one has to find a balance between the need of achieving the most accurate result and the computing time required to achieve it. The most accurate result is useless if the programmer does not get the result within his lifetime. D. ADAMS [20] put in a nutshell: the super-computer *Deep Thought* was asked to compute the answer to

"The Ultimate Question of Life, the Universe and Everything", quote:

> "How long?" he said.
> "Seven and a half million years."

Another quite crucial point, which we neglected so far, is the error analysis of a computational method which is based on random numbers (in fact it is pseudo-random numbers and this point will be discussed in the second part of this book). In this case, the situation changes completely, because, similar to experimental results, the observed values are distributed around a mean with a certain variance. Such results have to be interpreted within a statistical context. However, it turns out that for many problems the computational efficiency can be significantly increased using such methods. Typical applications are estimates of integrals or solutions to optimization problems. Such topics will be treated in the second part of this book.

References

1. Süli, E., Mayers, D.: An Introduction to Numerical Analysis. Cambridge University Press, Cambridge (2003)
2. Gautschi, W.: Numerical Analysis. Springer, Berlin/Heidelberg (2012)
3. Jacques, I., Judd, C.: Numerical Analysis. Chapman and Hall, London (1987)
4. Arnol'd, V.I.: Mathematical Methods of Classical Mechanics, 2nd edn. Graduate Texts in Mathematics, vol. 60. Springer, Berlin/Heidelberg (1989)
5. Fetter, A.L., Walecka, J.D.: Theoretical Mechanics of Particles and Continua. Dover, New York (2004)
6. Scheck, F.: Mechanics, 5th edn. Springer, Berlin/Heidelberg (2010)
7. Goldstein, H., Poole, C., Safko, J.: Classical Mechanics, 3rd edn. Addison-Wesley, Menlo Park (2013)
8. Fließbach, T.: Mechanik, 7th edn. Lehrbuch zur Theoretischen Physik I. Springer, Berlin/Heidelberg (2015)
9. Abramovitz, M., Stegun, I.A. (eds.): Handbook of Mathemathical Functions. Dover, New York (1965)
10. Olver, F.W.J., Lozier, D.W., Boisvert, R.F., Clark, C.W.: NIST Handbook of Mathematical Functions. Cambridge University Press, Cambridge (2010)
11. Mathai, A.M., Haubold, H.J.: Special Functions for Applied Scientists. Springer, Berlin/Heidelberg (2008)
12. Beals, R., Wong, R.: Special Functions. Cambridge Studies in Advanced Mathematics. Cambridge University Press, Cambridge (2010)
13. Clark, A.: Elements of Abstract Algebra. Dover, New York (1971)
14. Chartier, T.: Devastating roundoff error. Math. Horiz. **13**, 11 (2006). http://www.jstor.org/stable/25678616
15. Ueberhuber, C.W.: Numerical Computation 1: Methods, Software and Analysis. Springer, Berlin/Heidelberg (1997)
16. Burden, R.L., Faires, J.D.: Numerical Analysis. PWS-Kent Publishing Comp., Boston (1993)
17. Higham, N.J.: Accuracy and Stability of Numerical Algorithms, 2nd edn. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2002)
18. Lorenz, E.N.: Deterministic nonperiodic flow. J. Atmos. Sci. **20**, 130–141 (1963)
19. Roulstone, I., Norbury, J.: Invisible in the Storm: The Role of Mathematics in Understanding Weather. Prinecton University Press, Princeton (2013)
20. Adams, D.: The Hitchhiker's Guide to the Galaxy. Pan Books, London (1979)