Contents

Part I
Ordinary Differential Equations

In this part, we discuss the standard numerical techniques used to integrate systems of ordinary differential equations (ODEs).

Chapter 1
Initial Value Problem

An initial value problem (IVP) is a system of differential equation

$$\frac{d\mathbf{x}}{dt} = f(t, \mathbf{x}(t)), \tag{1.1}$$

$\mathbf{x}(t) = (x_1(t), x_2(t), \ldots x_n(t))^T$, $f \in [a,b] \times \mathbb{R}^n \to \mathbb{R}^n$, together with specified value

$$\mathbf{x}(a) = \mathbf{x}_0, \tag{1.2}$$

called the initial condition. We are interested in a numerical approximation of the continuously differentiable solution $\mathbf{x}(t)$ of the IVP (1.1)–(1.2) over the time interval $t \in [a,b]$. To this aim we subdivide the interval $[a,b]$ into M equal subintervals and select *the mesh points* t_j [5, 2]

$$t_j = a + jh, \qquad j = 0, 1, \ldots, M, \quad h = \frac{b-a}{M}. \tag{1.3}$$

The value h is called *a step size*.

1.1 Explicit Euler's Method

The simplest method to approximate IVP (1.1)–(1.2) was devised by Leonhard Euler in 1768. The idea of the method is straightforward: From the initial condition (1.2) we know that at $t = t_0 = a$ the slope of the solution curve $d\mathbf{x}/dt$ is $f(t_0, \mathbf{x}_0)$. Therefore we can try to obtain the next approximation $\mathbf{x}_1 := \mathbf{x}(t_1)$ at a small time h later by adding $h\, f(t_0, \mathbf{x}_0)$ to $\mathbf{x}_0$, namely

$$\mathbf{x}_1 = \mathbf{x}_0 + h\, f(t_0, \mathbf{x}_0).$$

Now we can take another step forward in the same way, using the slope $f(t_1, \mathbf{x}_1)$, corresponding to the new time $t_1 = t_0 + h$, i.e.,

$$\mathbf{x}_2 = \mathbf{x}_1 + h\, f(t_1, \mathbf{x}_1)\,.$$

The process is repeated and generates a sequaence of points $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_M$ that approximates the solution $\mathbf{x}(t)$. The general step of the Euler's method is [4, 5]

$$\begin{aligned} \mathbf{x}_{j+1} &= \mathbf{x}_j + h\, f(t_j, \mathbf{x}_j)\,, \\ t_{j+1} &= t_j + h, \quad j = 0, 1, \ldots, M-1. \end{aligned} \tag{1.4}$$

Notice that the Euler method (1.4) is *an explicit method*, i.e., $\mathbf{x}_{j+1}$ is given *explicitly* in terms of known quantities such as $\mathbf{x}_j$ and $f(t_j, \mathbf{x}_j)$.

From geometrical point of view, one starts at the point $(t_0, \mathbf{x}_0)$ of the $(t, \mathbf{x})$-plane and is moving along the tangent line to the solution $\mathbf{x}(t)$ and will end up at the point $(t_1, \mathbf{x}_1)$. Now this point is used to compute the next slope $f(t_1, \mathbf{x}_1)$ and to locate the next approximation point $(t_2, \mathbf{x}_2)$ etc.

Example 1

Let us use Euler's method (1.4) to solve approximately a simple IVP [1]

$$\dot{x} = x, \quad \text{over} \quad t \in [0, 1], \quad x(0) = 1. \tag{1.5}$$

The exact solution is $x(t) = \exp(t)$, so we can calculate the correct value at the end of the time interval, i.e.,

$$x(1) = e = 2.71828...$$

Let us find the numerical approximation of (1.5) for different step sizes h=$\{0.1, 1e\text{-}2, 1e\text{-}3, 1e\text{-}4, 1e\text{-}5\}$ and calculate the difference between obtained numerical value at the end of the time interval x_{end} and the exact value e. The results are shown in Table (1.1) The presented results demonstrate that the error at the end of interval is

Table 1.1 Numerical results, obtained by Euler's methhod for Eq. (1.5)

h	x_{end}	$\lvert x_{end} - 1\rvert$
0.1	2.5937	0.12
1e-2	2.7048	1.35e-2
1e-3	2.7169	1.4e-3
1e-4	2.7181	1.35e-4
1e-5	2.7183	1.35e-5

propotional to the step size h.

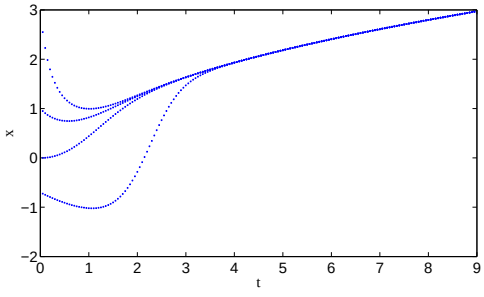

Fig. 1.1 Numerical solution of Eq. (1.6) over the interval $[0, 9]$ by the method (1.4) with the step size $h = 0.05$ for four different initial values $x_0 = \{-0.7, 0, 1, 3\}$.

Example 2

Now let us solve a nonlinaer IVP [1]

$$\dot{x} = t - x^2, \quad \text{over} \quad t \in [0, T], \quad x(0) = x_0 \tag{1.6}$$

for different values of x_0 and T using the method (1.4). Figure 1.1 shows numerical solutions of Eq. (1.6) over the time interval $[0, 9]$ for four different initial values $x_0 = \{-0.7, 0, 1, 3\}$. One can see that the solutions corresponding to different x_0 converge to the same curve. But if we compute the solution again for a longer time interval, say $t \in [0, 900]$ for, e.g., $x_0 = 0$, the numerical solution starts to oscillate from some time moment on (see Fig. 1.2 (a)) and the oscillations character becomes chaotic. This effect indicate the instability of the Euler's method at least at the choosen value of the time step. However, the effect disappears if we repeat the calculation with a smaller h (see Fig. 1.2 (b) for details).

The presented examples raises a number of questions. One of these is the question of *convergence*. That is, as the step size h tends to zero, do the values of the numerical solution approach the corresponding values of the actual solution? Assuming that the answer is affirmative, there remains the important practical question of how rapidly the numerical approximation converges to the solution. In other words, how small a step size is needed to guarantee a given level of accuracy? We discuss these questions below.

1.2 Numerical errors

Generally there are two major sources of error associated with a numerical integration scheme for ODE's, namely, *truncation error* and *rounding error*, which is due to finite precision of floating-point arithmetic [4, 2].

For the sake of simplicity, let us suppose that the IVP (1.1)–(1.2) is posed for the first order ODE, i.e., $\mathbf{x}(t)$ is a scalar value.The *local discretization (truncation) error* of IVP (1.1)–(1.2) is the error committed in the single step from t_j to t_{j+1} and is defined by [4]

$$\varepsilon_{j+1} = \frac{1}{h}\big(\mathbf{x}(t_{j+1}) - \mathbf{x}(t_j)\big) - f(t_j, \mathbf{x}_j)\,, \qquad j = 0, \ldots, M\,. \tag{1.7}$$

In addition, the integration scheme is *consistent* [4], if

$$\max_{t_j} \|\varepsilon_j\| \to 0\,, \quad \text{for} \quad h_{\max} \to 0\,, \tag{1.8}$$

where $h_{\max} = \max_j h_j$ and $h_j = t_{j+1} - t_j$.

The explicit Euler method (1.4) is based on a truncated Taylor series expansion, i.e., if we expand $\mathbf{x}(t)$ in the neighborhood of $t = t_j$, we get

$$\mathbf{x}(t_j + h) = \mathbf{x}(t_j) + h\mathbf{x}'(t_j) + \frac{h^2}{2!}\mathbf{x}''(t_j) + \ldots\,. \tag{1.9}$$

Thus, every time we take a step using Euler's method (1.4), we incur a truncation error of $\mathscr{O}(h^2)$, i.e., the local truncation error for the Euler method is proportional to the square of the step size h and the proportionality factor depends on the second derivative of the solution .

The local truncation error (1.7) is different from the *global discretization (truncation) error* e_j, which is defined as the difference between the true solution and the computed solution, i.e.,

$$e_j = \mathbf{x}(t_j) - \mathbf{x}_j\,, \qquad j = 0, \ldots, M, \tag{1.10}$$

where $\mathbf{x}(t_j)$ denotes the exact solution on the step j and $\mathbf{x}_j$ stands for its numerical approximation. The concept of the global discretization error is connected with the notion of *convergency* of the method, namely, the numerical scheme is convergent,

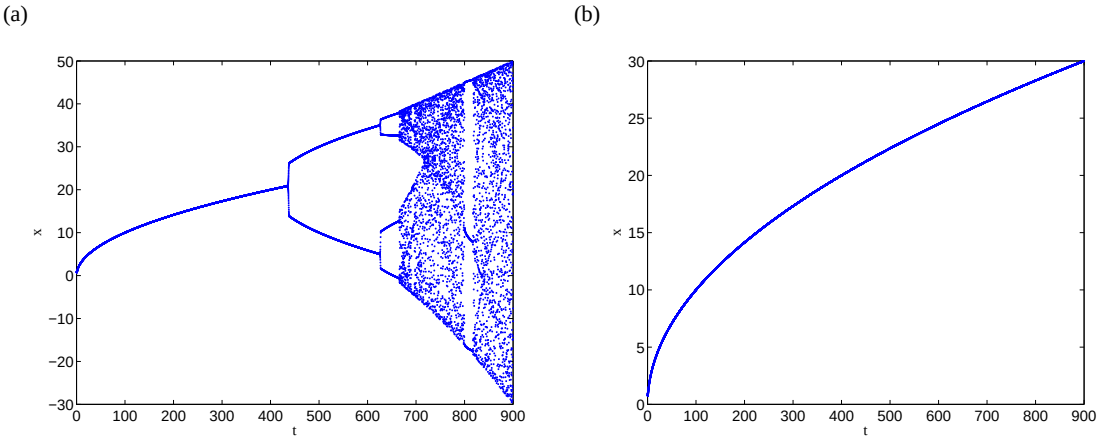

Fig. 1.2 Numerical implementation of the Euler's method for Eq. (1.6) over a long time interval, $T = 900$ and initial value $x_0 = 0$. (a) with $h = 0.05$, illustrating numerical instability of the scheme (1.4). (b) The instability desappeares for a smaller step size $h = 0.025$.

if

$$\max_{t_j} \|e_j\| \to 0\,, \quad \text{for} \quad h_{\max} \to 0\,. \tag{1.11}$$

Moreover, one can also say that the scheme possesses *the order of convergency p*, if

$$\max_{t_j} \|e_j\| \leq K h_{\max}^p\,, \qquad h_{\max} \to 0\,, \tag{1.12}$$

where K is some constant value.

If we are interested in study of the behavior of the error for various step sizes we can also consider the *final global error*, which is the global truncation error at the end of the integration intervall $[a, b]$ [2], i.e.,

$$E(\mathbf{x}(b), h) = \mathbf{x}(b) - \mathbf{x}_M. \tag{1.13}$$

In most cases, we do not know the exact solution and hence the final global error (1.13) is not possible to be evaluated. However, if we neglect round-off errors, it is reasonable to assume that the global error after M time step is M times the local truncation error (1.7), since M is proportional to $1/h$, $E(\mathbf{x}(b), h)$ should be proportional to ε_{j+1}/h. For example, for the Euler's method (1.4) the accumulated error would be

$$E(\mathbf{x}(b), h) = \sum_{j=1}^{M} \frac{h^2}{2}\mathbf{x}'' \approx M\mathbf{x}''\frac{h^2}{2} = \frac{hM}{2}\mathbf{x}''h = \frac{b-a}{2}\mathbf{x}''h = \mathcal{O}(h)$$

Thus, the explicit Euler method is of the *first order of convergency*.

Let us consider two approximations, made by the method (1.4), using the steps sizes h and $h/2$. Then we obtain

$$E(\mathbf{x}(b), h) \approx Kh, \qquad K = \text{const}\,,$$

and

$$E(\mathbf{x}(b), \frac{h}{2}) \approx K\frac{h}{2} = \frac{1}{2}Kh \approx \frac{1}{2}E(\mathbf{x}(b), h)\,.$$

Hence if the step size in Euler's method (1.4) is reduced by a factor of $1/2$ we can expect that the final global truncation error (1.13) will be reduced by the same factor (see also Example 2 of Section (1.1)).

1.3 Heun's Method

We have seen that the Euler method (1.4) is not sufficiently accurate to be an efficient problem-solving procedure, e.g., the rate of convergence scaling linearly with the step size h, so it is desirable to develop more accurate methods. To this end, let us consider IVP (1.1)–(1.2). Integrating both sides of Eq. (1.1) over one time step from t_j to t_{j+1} we obtain the *exact* relation

$$\mathbf{x}(t_{j+1}) - \mathbf{x}(t_j) = \int_{t_j}^{t_{j+1}} f(t, \mathbf{x}(t))\, dt\,. \tag{1.14}$$

Now a numerical integration method can be used to approximate the definite integral in Eq. (1.14). From the geometrical point of view, the right-hand side of (1.14) corresponds to the area S under the curve $f(t, \mathbf{x}(t))$, between t_j and t_{j+1}. For example, Euler's method (1.4) consists of approximation of right-hand side of (1.14) by the area of the rectangle S_r with the height $f(t_j, \mathbf{x}(t_j))$ and width h, i.e., one obtains Eq. (1.4), namely

$$\mathbf{x}_{j+1} = \mathbf{x}_j + S_r = \mathbf{x}_j + h\, f(t_j, \mathbf{x}(t_j))\,.$$

Clearly, a better approximation to the area S can be obtained if we use the trapezium with area

$$S_t = \frac{h}{2}\Big(f(t_j, \mathbf{x}(t_j)) + f(t_{j+1}, \mathbf{x}(t_{j+1}))\Big)\,,$$

yielding

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \frac{h}{2}\Big(f(t_j, \mathbf{x}(t_j)) + f(t_{j+1}, \mathbf{x}(t_{j+1}))\Big)\,. \tag{1.15}$$

Notice that the r.h.s. of Eq. (1.15) contains the yet unknown value $\mathbf{x}_{j+1}$. In order to overcome this difficulty we use the Euler's approximation (1.4) to replace $f(t_{j+1}, \mathbf{x}(t_{j+1}))$ with $f(t_{j+1}, \mathbf{x}_j + h\, f(t_j, \mathbf{x}(t_j)))$. After it is substituted into Eq. (1.15), the resulting expression is called *Heun's, trapezoid* or *improved Euler's method*:

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \frac{h}{2}\Big(f\big(t_j, \mathbf{x}(t_j)\big) + f\big(t_j + h, \mathbf{x}_j + h\, f(t_j, \mathbf{x}(t_j))\big)\Big)\,. \tag{1.16}$$

The improved Euler formula [2, 1] is an example of a two-stage method: First, Euler's method (1.4) is used as a *prediction*, and then the trapezoidal rule is used as a *correction*,i.e.,

$$\begin{aligned} \mathbf{y}_{j+1} &= \mathbf{x}_j + h\, f(t_j, \mathbf{x}_j)\,, \\ t_{j+1} &= t_j + h\,, \\ \mathbf{x}_{j+1} &= \mathbf{x}_j + h\, f(t_{j+1}, \mathbf{y}_{j+1})\,. \end{aligned} \tag{1.17}$$

The local truncation error ε_{j+1} for the trapezoidal formula (1.18) is $\mathscr{O}(h^3)$ as opposed to $\mathscr{O}(h^2)$ for the Euler's method (1.4) [5, 2]. It can also be shown that for a finite interval, the global truncation error for (1.18) is bounded by $\mathscr{O}(h^2)$, so this method is *a second order method*. Indeed, if we take into acount only the local error [3, 2]

$$\varepsilon_{j+1} = -\frac{h^3}{12}\mathbf{x}''(t_j)\,,$$

after M steps the accumulated error $E(\mathbf{x}(b), h)$ for the method (1.18) is

$$E(\mathbf{x}(b),h) = -\sum_{j=1}^{M} \frac{h^3}{12}\mathbf{x}'' \approx -\frac{b-a}{12}\mathbf{x}''h^2 = \mathscr{O}(h^2)\,.$$

Again, if we perform two computations using the step sizes h and $h/2$ we obtain

$$E(\mathbf{x}(b),h) = Kh^2\,, \qquad K = \text{const}\,.$$

and

$$E(\mathbf{x}(b),\frac{h}{2}) \approx K\frac{h^2}{4} = \frac{1}{4}E(\mathbf{x}(b),h)\,.$$

Thus, if the step size in Heun's method is reduced by a factor of $1/2$, we can expect that the final global truncation error would by reduced by a factor of $1/4$.

Example 1

Use Euler's method (1.4) and Heun's method (1.18) to solve the IVP for the ODE, describing the behavior of the simple harmonic oscillator

$$\ddot{x} + \omega^2 x = 0, \qquad x(0) = 0, \quad \dot{x}(0) = v_0\,, \tag{1.18}$$

over the time interval $t \in [0,T]$, and where the frequency ω and initial velocity v_0 are given constants. The exact solution

$$x(t) = \frac{v_0}{\omega}\sin(\omega t)\,. \tag{1.19}$$

represents simple harmonic motion: sinusoidal oscillations about the equilibrium point, with a constant amplitude and a constant frequency.
First of all we rewrite Eq. (1.18) as a system of first-order ODE's:

$$\begin{aligned} &\dot{x} = y\,, \\ &\dot{y} = -\omega^2 x\,, \qquad x(0) = 0\,, \quad y(0) = v_0\,. \end{aligned} \tag{1.20}$$

We begin with analysis of system (1.20) using the ideas of *phase space* [7]. If we multiply both sides of the first equation of (1.20) by $\omega^2 x$ and both sides of the second equation of the system by y and add the two together we get the following relation

$$y\dot{y} + \omega^2 x\dot{x} = 0\,.$$

Notice that the l.h.s. of the relation above is the time derivative, so one can rewrite the last relation as

$$\frac{d}{dt}\left(\frac{1}{2}y^2 + \frac{1}{2}\omega^2 x^2\right) = 0 \Leftrightarrow \frac{1}{2}y^2 + \frac{1}{2}\omega^2 x^2 := I_1\,, \tag{1.21}$$

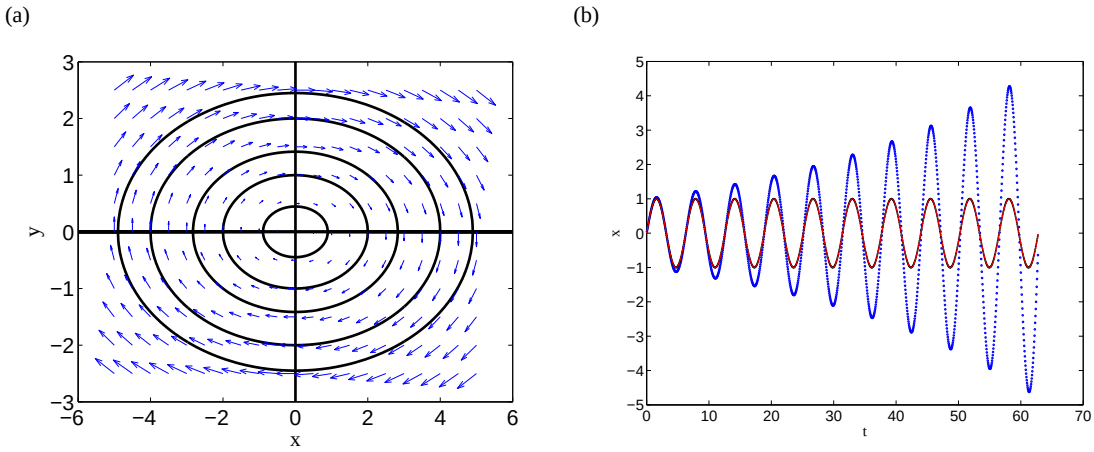

Fig. 1.3 (a) Phase diagramm for the linear oscillator (1.18), corresponding to the different values of the energy $I_1 = \{0.1, 0.5, 1, 3\}$ and $\omega = 0.5$. (b) Numerical solution of (1.18) over the interval $[0, 20\pi]$ by methods (1.4) (blue points) and (1.18) (red line) with the step size $h = 0.05$. The black curve corresponds to the exact solution of Eq. (1.18).

where $I_1 = \text{const}$ is usually called a *constant of motion* or a *first integral*, which can be interpreted as the *mechanical energy* of the system. From the geometrical point of view, one can speak about *phase curves* that form a set of ellipses in the *phase space* with coordinates (x, y). These ellipses cut the x-axis at $x = \pm\sqrt{2I_1}/\omega$ and the y-axis at $y = \pm\sqrt{2I_1}$. The origin of the phase plane corresponds to an equilibrium point of the motion (see Fig. 1.3 (a)).

Now we can solve system (1.20) numerically, using approximation methods (1.4) and (1.18) to the system (1.20). For the sake of simplisity we choose $v_0 = 1$, $\omega = 1$, so that the exact solution of (1.18) is just $x(t) = \sin(t)$, and integrate the system (1.20) over the time interval $t \in [0, 20\pi]$ with the time step $h = 0.05$. Figure 1.3 (b) shows a comparison between two methods, indicating the much better accuracy of the Heun's method (1.18).

1.4 Runge-Kutta Methods

The Runge-Kutta methods are an important family of iterative methods for the approximation of solutions of ODE's, that were develoved around 1900 by the german mathematicians C. Runge (1856–1927) and M.W. Kutta (1867–1944). We start with the considereation of the explicit methods.

Let us consider the IVP (1.1)–(1.2). The family of explicit Runge–Kutta (RK) methods of the m'th stage is given by [5, 3]

$$\mathbf{x}(t_{n+1}) := \mathbf{x}_{n+1} = \mathbf{x}_n + h\sum_{i=1}^{m} c_i k_i, \tag{1.22}$$

where

$$\begin{aligned}
k_1 &= f(t_n, \mathbf{x}_n),\\
k_2 &= f(t_n + \alpha_2 h, \mathbf{x}_n + h\beta_{21} k_1(t_n, \mathbf{x}_n)),\\
k_3 &= f(t_n + \alpha_3 h, \mathbf{x}_n + h(\beta_{31} k_1(t_n, \mathbf{x}_n) + \beta_{32} k_2(t_n, \mathbf{x}_n))),\\
&\vdots\\
k_m &= f(t_n + \alpha_m h, \mathbf{x}_n + h\sum_{j=1}^{m-1} \beta_{mj} k_j).
\end{aligned}$$

To specify a particular method, we need to provide the integer m (the number of stages), and the coefficients α_i (for $i = 2,3,...,m$), β_{ij} (for $1 \le j < i \le m$), and c_i (for $i = 1,2,...,m$). These data are usually arranged in a co-called *Butcher tableau* (after John C. Butcher) [5, 3]:

Table 1.2 The Butcher tableau.

0					
α_2	β_{21}				
α_3	β_{31}	β_{32}			
$\vdots$	$\vdots$	$\vdots$	$\ddots$		
$\vdots$	$\vdots$	$\vdots$			
α_m	β_{m1}	β_{m2}	.....	β_{mm-1}	
	c_1	c_2	.....	c_{m-1}	c_m

Examples

1. Let $m = 1$. Then

$$\begin{aligned}
k_1 &= f(t_n, \mathbf{x}_n),\\
\mathbf{x}_{n+1} &= \mathbf{x}_n + h c_1 f(t_n, \mathbf{x}_n).
\end{aligned}$$

   On the other hand, the Taylor expansion yields

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\dot{\mathbf{x}}\big|_{t_n} + \cdots = \mathbf{x}_n + h f(t_n, \mathbf{x}_n) + \mathscr{O}(h^2) \Rightarrow c_1 = 1.$$

   Thus, the first-stage RK-method is equivalent to the explicit Euler's method (1.4). Note that the method (1.4) is of the first order of accuracy. Thus we can speak about the RK method of the first order.

2. Now consider the case $m = 2$. In this case Eq. (1.22) is equivalent to the system

$$\begin{aligned} k_1 &= f(t_n, \mathbf{x}_n)\,, \\ k_2 &= f(t_n + \alpha_2 h, \mathbf{x}_n + h\beta_{21} k_1)\,, \\ \mathbf{x}_{n+1} &= \mathbf{x}_n + h(c_1 k_1 + c_2 k_2)\,. \end{aligned} \tag{1.23}$$

Now let us write down the Taylor series expansion of $\mathbf{x}$ in the neighborhood of t_n up to the h^2 term, i.e.,

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\left.\frac{d\mathbf{x}}{dt}\right|_{t_n} + \frac{h^2}{2}\left.\frac{d^2\mathbf{x}}{dt^2}\right|_{t_n} + \mathscr{O}(h^3)\,.$$

However, we know that $\dot{\mathbf{x}} = f(t, \mathbf{x})$, so that

$$\frac{d^2\mathbf{x}}{dt^2} := \frac{df(t,\mathbf{x})}{dt} = \frac{\partial f(t,\mathbf{x})}{\partial t} + f(t,\mathbf{x})\frac{\partial f(t,\mathbf{x})}{\partial \mathbf{x}}\,.$$

Hence the Taylor series expansion can be rewritten as

$$\mathbf{x}_{n+1} - \mathbf{x}_n = h\,f(t_n, \mathbf{x}_n) + \frac{h^2}{2}\left(\frac{\partial f}{\partial t} + f\frac{\partial f}{\partial \mathbf{x}}\right)\Bigg|_{(t_n,\mathbf{x}_n)} + \mathscr{O}(h^3)\,. \tag{1.24}$$

On the other hand, the term k_2 in the proposed RK method can also expanded to $\mathscr{O}(h^3)$ as

$$k_2 = f(t_n + \alpha_2 h, \mathbf{x}_n + h\beta_{21}k_1) = h\,f(t_n,\mathbf{x}_n) + h\,\alpha_2\left.\frac{\partial f}{\partial t}\right|_{(t_n,\mathbf{x}_n)} + h\beta_{21}\,f\left.\frac{\partial f}{\partial \mathbf{x}}\right|_{(t_n,\mathbf{x}_n)} + \mathscr{O}(h^3)\,.$$

Now, substituting this relation for k_2 into the last equation of (1.23), we achieve the following expression:

$$\mathbf{x}_{n+1} - \mathbf{x}_n = h\,(c_1 + c_2)\,f(t_n,\mathbf{x}_n) + h^2\,c_2\,\alpha_2\left.\frac{\partial f}{\partial t}\right|_{(t_n,\mathbf{x}_n)} + h^2\,c_2\beta_{21}\,f\left.\frac{\partial f}{\partial \mathbf{x}}\right|_{(t_n,\mathbf{x}_n)} + \mathscr{O}(h^3)\,.$$

Making comparision the last equation and Eq. (1.24) we can write down the system of algebraic equations for unknown coefficients

$$\begin{aligned} c_1 + c_2 &= 1\,, \\ c_2\,\alpha_2 &= \frac{1}{2}\,, \\ c_2\,\beta_{21} &= \frac{1}{2}\,. \end{aligned}$$

The system involves four unknowns in three equations. That is, one additional condition must be supplied to solve the system. We discuss two useful choices, namely

a) Let $\alpha_2 = 1$. Then $c_2 = 1/2$, $c_1 = 1/2$, $\beta_{21} = 1$. The corresponding Butcher tableau reads:

$$\begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & 1/2 & 1/2 \end{array}$$

Thus, in this case the two-stages RK method takes the form

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}\Big(f(t_n, \mathbf{x}_n) + f(t_n + h, \mathbf{x}_n + h f(t_n, \mathbf{x}_n)) \Big),$$

and is equivalent to the Heun's method (1.18), so we refer the last method to as RK-method of the second order.

b) Now let $\alpha_2 = 1/2$. In this case $c_2 = 1$, $c_1 = 0$, $\beta_{21} = 1/2$. The corresponding Butcher tableau reads:

$$\begin{array}{c|cc} 0 & & \\ 1/2 & 1/2 & \\ \hline & 0 & 1 \end{array}$$

In this case the second-order RK method (1.22) can be written as

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\, f\big(t_n + \frac{h}{2}, \mathbf{x}_n + \frac{h}{2} f(t_n, \mathbf{x}_n)\big)$$

and is called the *RK2 method*.

RK4 Methods

One member of the family of Runge–Kutta methods (1.22) is often referred to as *RK4 method* or *classical RK method* and represents one of the solutions corresponding to the case $m = 4$. In this case, by matching coefficients with those of the Taylor series one obtains the following system of equations [2]

$$
\begin{aligned}
c_1 + c_2 + c_3 + c_4 &= 1\,, \\
\beta_{21} &= \alpha_2\,, \\
\beta_{31} + \beta_{32} &= \alpha_3\,, \\
c_2\alpha_2 + c_3\alpha_3 + c_4\alpha_4 &= \frac{1}{2}\,, \\
c_2\alpha_2^2 + c_3\alpha_3^2 + c_4\alpha_4^2 &= \frac{1}{3}\,, \\
c_2\alpha_2^3 + c_3\alpha_3^3 + c_4\alpha_4^3 &= \frac{1}{4}\,, \\
c_3\alpha_2\beta_{32} + c_4(\alpha_2\beta_{42} + \alpha_3\beta_{43}) &= \frac{1}{6}\,, \\
c_3\alpha_2\alpha_3\beta_{32} + c_4\alpha_4(\alpha_2\beta_{42} + \alpha_3\beta_{43}) &= \frac{1}{8}\,, \\
c_3\alpha_2^2\beta_{32} + c_4(\alpha_2^2\beta_{42} + \alpha_3^2\beta_{43}) &= \frac{1}{12}\,, \\
c_4\alpha_2\beta_{32}\beta_{43} &= \frac{1}{24}\,.
\end{aligned}
$$

The system involves thirteen unknowns in eleven equations. That is, two additional condition must be supplied to solve the system. The most useful choices is [3]

$$
\alpha_2 = \frac{1}{2}\,, \qquad \beta_{31} = 0\,.
$$

The corresponding Butcher tableau is presented in Table 1.3. The tableau 1.3 yields

Table 1.3 The Butcher tableau corresponding to the RK4 method.

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	1/3	1/3	1/6

the equivalent corresponding equations defining the classical RK4 method:

$$
\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{6}\big(k_1 + 2k_2 + 2k_3 + k_4\big), \tag{1.25}
$$

where

$$
\begin{aligned}
k_1 &= f(t_n, \mathbf{x}_n),\\
k_2 &= f(t_n + \frac{h}{2}, \mathbf{x}_n + \frac{h}{2}k_1),\\
k_3 &= f(t_n + \frac{h}{2}, \mathbf{x}_n + \frac{h}{2}k_2),\\
k_4 &= f(t_n + h, \mathbf{x}_n + hk_3).
\end{aligned}
$$

This method is reasonably simple and robust and is a good general candidate for numerical solution of ODE's when combined with an intelligent adaptive step-size routine or an embedded methods (,e.g., so-called Runge-Kutta-Fehlberg methods (RKF45)).

Remark:

Notice that except for the classical method (1.25), one can also construct other RK4 methods. We mention only so-called $3/8$-*Runge-Kutta method.* The Brutcher tableau, corresponding to this method is presented in Table 1.4.

Table 1.4 The Butcher tableau corresponding to the $3/8$- Runge-Kutta method.

0				
1/3	1/3			
2/3	-1/3	1		
1	1	-1	1	
	1/8	3/8	3/8	1/8

Geometrical interpretation of the RK4 method

Let us consider a curve $\mathbf{x}(t)$, obtained by (1.25) over a single time step from t_n to t_{n+1}. The next value of approximation $\mathbf{x}_{n+1}$ is obtained ty integrating the slope function, i.e.,

$$
\mathbf{x}_{n+1} - \mathbf{x}_n = \int_{t_n}^{t_{n+1}} f(t, \mathbf{x})\, dt\,. \tag{1.26}
$$

Now, if the Simpson's rule is applied, the approximation to the integral of the last equation reads [4]

$$
\int_{t_n}^{t_{n+1}} f(t, \mathbf{x})\, dt \approx \frac{h}{6}\left(f(t_n, \mathbf{x}(t_n)) + 4\, f(t_n + \frac{h}{2}, \mathbf{x}(t_n + \frac{h}{2})) + f(t_{n+1}, \mathbf{x}(t_{n+1}))\right). \tag{1.27}
$$

On the other hand, the values k_1, k_2, k_3 and k_4 are approximations for slopes of the curve $\mathbf{x}$, i.e., k_1 is the slope of the left end of the interval, k_2 and k_3 describe two estimations of the slope in the middle of the time interval, whereas k_4 corresponds to the slope at the right. Hence, we can choose $f(t_n, \mathbf{x}(t_n)) = k_1$ and $f(t_{n+1}, \mathbf{x}(t_{n+1})) = k_4$, whereas for the value in the middle we choose the average of k_2 and k_3, i.e.,

$$f(t_n + \frac{h}{2}, \mathbf{x}(t_n + \frac{h}{2})) = \frac{k_2 + k_3}{2}.$$

Then Eq. (1.26) becomes

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{6}\left(k_1 + \frac{4(k_2 + k_3)}{2} + k_4\right),$$

which is equivalent to the RK4 schema (1.25).

Stage versus Order

The local truncation error (1.7) for the method (1.25) can be estimated from the error term for the Simpson's rule (1.27) and equals [4, 2]

$$\varepsilon_{n+1} = -h^5 \frac{\mathbf{x}^{(4)}}{2880}.$$

Now we can estimate the final global error (1.13), if we suppose that only the error above is presented. After M steps the accumulated error for the RK4 method reads

$$E(\mathbf{x}(b), h) = -\sum_{k=1}^{M} h^5 \frac{\mathbf{x}^{(4)}}{2880} \approx \frac{b-a}{2880}\mathbf{x}^{(4)} h = \mathcal{O}(h^4).$$

That is, the RK4 method (1.25) is of the fourth order. Now, let us compare two appximations, obtained using the time steps h and $h/2$. For the step size h we have

$$E(\mathbf{x}(b), h) \approx K h^4,$$

with $K = \text{const}$. Hence, for the step $h/2$ we get

$$E(\mathbf{x}(b), \frac{h}{2}) = K \frac{h^4}{16} \approx \frac{1}{16} E(\mathbf{x}(b), h).$$

That is, if the step size in (1.25) is reduced by the factor of two, the global error of the method will be reduced by the factor of $1/16$.

Remark:

In general there are two ways to improve the accuracy:

1. One can reduce the time step h, i.e., the amount of steps increases;
2. The method of the higher convergency order can be used.

However, increasing of the convergency order p is reasonable only up to some limit, given by so-called *Butcher barrier* [5], which says, that the amount of stages m grows faster, as the order p. In other words, *for $m \geq 5$ there are no explicit RK methods with the convergency order $p = m$ (the corresponding system is unsolvable)*. Hence, in order to reach convergency order five one needs six stages. Notice that further increasing of the stage $m = 7$ leads to the convergency order $p = 5$ as well.

1.4.1 Adaptive stepsize control and embedded methods

As mentioned above, one way to guarantee accuracy in the solution of (1.1)–(1.1) is to solve the problem twice using step sizes h and $h/2$. To illustrate this approach, let us consider the RK method of the order p and denote an exact solution at the point $t_{n+1} = t_n + h$ by $\widetilde{\mathbf{x}}_{n+1}$, whereas $\mathbf{x}_1$ and $\mathbf{x}_2$ represent the approximate solutions, corresponding to the step sizes h and $h/2$. Now let us perform one step with the step size h and after that two steps each of size $h/2$. In this case the true solution and two numerical approximations are related by

$$\begin{aligned}
\widetilde{\mathbf{x}}_{n+1} &= \mathbf{x}_1 + C h^{p+1} + \mathscr{O}(h^{p+2}), \\
\widetilde{\mathbf{x}}_{n+1} &= \mathbf{x}_2 + 2C \left(\frac{h}{2}\right)^{p+1} + \mathscr{O}(h^{p+2}).
\end{aligned}$$

That is,

$$|\mathbf{x}_1 - \mathbf{x}_2| = C h^{p+1} \left(1 - \frac{1}{2^p}\right) \Leftrightarrow C = \frac{|\mathbf{x}_1 - \mathbf{x}_2|}{(1 - 2^{-p}) h^{p+1}}.$$

Substituing the relation for C in the second estimate for the true solution we get

$$\widetilde{\mathbf{x}}_{n+1} = \mathbf{x}_2 + \varepsilon + \mathscr{O}(h^{p+2}),$$

where

$$\varepsilon = \frac{|\mathbf{x}_1 - \mathbf{x}_2|}{2^p - 1}$$

can be considered as a convenient *indicator* of the truncation error. That is, we have improved our estimate to the order $p + 1$. For example, for $p = 4$ we get

$$\widetilde{\mathbf{x}}_{n+1} = \mathbf{x}_2 + \frac{|\mathbf{x}_1 - \mathbf{x}_2|}{15} + \mathscr{O}(h^6).$$

This estimate is accurate to fifth order, one order higter than with the original step h. However, this method is not efficient. First of all, it requires a significant amount

of computation (we should solve the equation three times at each time step). The second point is, that we have no possibility to control the truncation error of the method (higher order means not always higher accuracy).
However we can use an estimate ε for the *step size control*, namely we can compare ε with some *desired accuracy* ε_0 (see Fig 1.4).

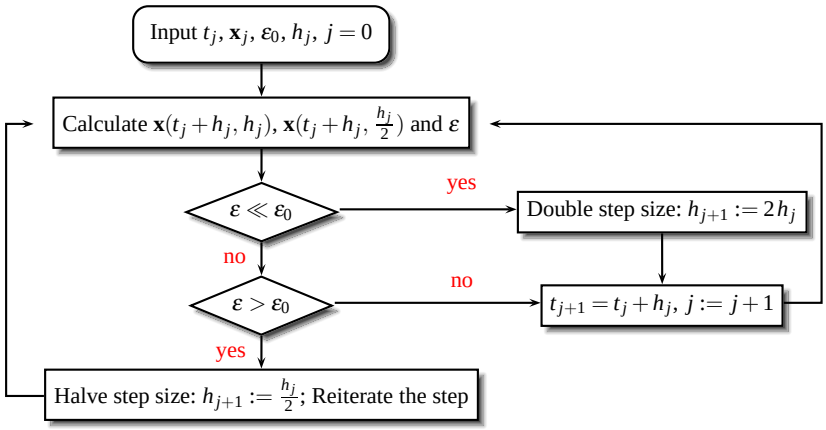

Fig. 1.4 Flow diagramm of the step size control by use of the step doubling method.

Alternatively, using the estimate ε, we can try to formulate the following problem of the *adaptive step size control*, namely: Using the given values $\mathbf{x}_j$ and t_j, find the largest possible step size h_{new}, so that the truncation error after the step with this step size remains below some given desired accuracy ε_0, i.e,

$$Ch_{new}^{p+1} \le \varepsilon_0 \Leftrightarrow \left(\frac{h_{new}}{h}\right)^{p+1} \frac{|\mathbf{x}_1 - \mathbf{x}_2|}{1-2^{-p}} \le \varepsilon_0 .$$

That is,

$$h_{new} = h\left(\frac{\varepsilon_0}{\varepsilon}\right)^{1/p+1} .$$

Then if the two answers are in close agreement, the approximation is accepted. If $\varepsilon > \varepsilon_0$ the step size has to be decreased, whereas the relation $\varepsilon < \varepsilon_0$ means, that the step size has to be increased in the next step.

Notice that because our estimate of error is not exact, we should put some "safety" factor $\beta \simeq 1$ [5, 3]. Usually, $\beta = 0.8, 0.9$. The flow diagramm, corresponding to the the adaptive step size control is shown on Fig. 1.5

Notice one additional technical point. The choise of the desired error ε_0 depends on the IVP we are interested in. In some applications it is convinient to set ε_0 propotional to h [3]. In this case the exponent $1/p+1$ in the estimate of the new time step is no longer correct (if h is reduced from a too-large value, the new predicted value h_{new} will fail to meet the desired accuracy, so instead of $1/p+1$ we should scale with $1/p$ (see [3] for details)). That is, the optimal new step size can be written as

$$h_{new} = \begin{cases} \beta h\left(\frac{\varepsilon_0}{\varepsilon}\right)^{1/p+1}, & \varepsilon \ge \varepsilon_0, \\ \beta h\left(\frac{\varepsilon_0}{\varepsilon}\right)^{1/p}, & \varepsilon < \varepsilon_0, \end{cases} \tag{1.28}$$

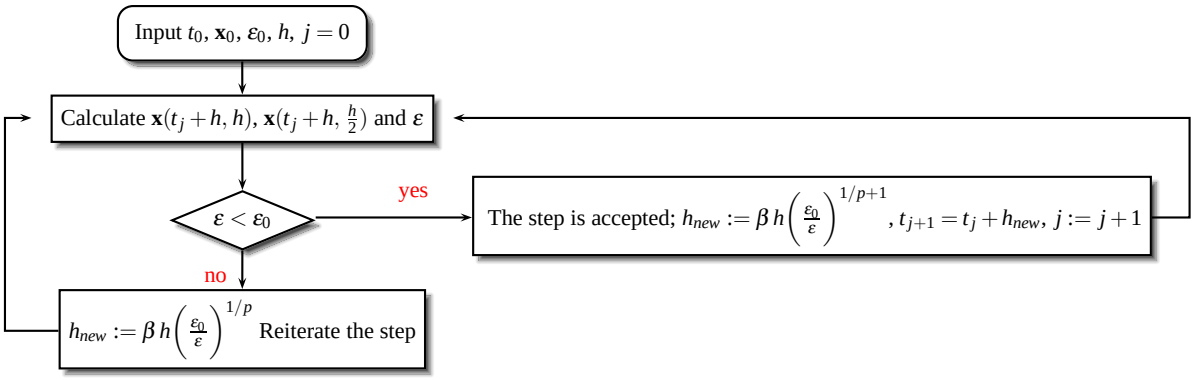

Fig. 1.5 Flow diagramm of the adaptive step size control by use of the step doubling method.

where β is a "safety" factor.

Runge-Kutta-Fehlberg method

The alternative stepsize adjustment algorithm is based on the *embedded Runge-Kutta formulas*, originally invented by Fehlberg and is called *the Runge-Kutta-Fehlberg methods (RKF45)* [5, 4]. At each step, two different approximations for the solution are made and compared. Usually an fourth-order method with five stages is used together with an fifth-order method with six stages, that uses all of the points of the first one. The general form of a fifth-order Runge-Kutta with six stages is

$$\begin{aligned} k_1 &= f(t,\mathbf{x}),\\ k_2 &= f(t+\alpha_2 h, \mathbf{x}+h\beta_{21}k_1),\\ &\vdots\\ k_6 &= f(t+\alpha_6 h, \mathbf{x}+h\sum_{j=1}^{5}\beta_{6j}k_j)\,. \end{aligned}$$

The embedded fourth-order formula is

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\sum_{i=1}^{6} c_i k_i + \mathscr{O}(h^5)\,.$$

And a better value for the solution is determined using a Runge-Kutta method of fifth-order:

$$\mathbf{x}^*_{n+1} = \mathbf{x}_n + h\sum_{i=1}^{6} c^*_i k_i + \mathscr{O}(h^6)$$

The two particlular choises of unknown parametrs of the method are given in Tables 1.5–1.6.

The error estimate is

$$\varepsilon = |\mathbf{x}_{n+1} - \mathbf{x}^*_{n+1}| = \sum_{i=1}^{6}(c_i - c^*_i)k_i\,.$$

Table 1.5 Fehlberg parameters of the Runge-Kutta-Fehlberg 4(5) method.

1/4	1/4					
3/8	3/32	9/32				
12/13	1932/2197	-7200/2197	7296/2197			
1	439/216	-8	3680/513	-845/4104		
1/2	-8/27	2	-3544/2565	1859/4104	-11/40	
	25/216	0	1408/2565	2197/4104	-1/5	
	16/135	0	6656/12825	28561/56430	-9/50	2/55

Table 1.6 Cash-Karp parameters of the Runge-Kutta-Fehlberg 4(5) method.

1/5	1/5					
3/10	3/40	9/40				
3/5	3/10	-9/10	6/5			
1	-11/54	5/2	-70/27	35/27		
7/8	1631/55296	175/512	575/13828	44275/110592	253/4096	
	37/378	0	250/621	125/594	512/1771	
	2825/27648	0	18575/48384	13525/55296	277/14336	1/4

As was mentioned above, if we take the current step h and produce an error ε, the corresponding "optimal" step h_{opt} is estimated as

$$h_{opt} = \beta\, h\left(\frac{\varepsilon_{tol}}{\varepsilon}\right)^{0.2},$$

where ε_{tol} is a desired accuracy and β is a "safety" factor, $\beta \simeq 1$. Then if the two answers are in close agreement, the approximation is accepted. If $\varepsilon > \varepsilon_{tol}$ the step size has to be decreased, whereas the relation $\varepsilon < \varepsilon_{tol}$ means, that the step size are to be increased in the next step.
Using Eq. (1.28), the optimal step can be often written as

$$h_{opt} = \begin{cases} \beta\, h\left(\frac{\varepsilon_{tol}}{\varepsilon}\right)^{0.2}, & \varepsilon \geq \varepsilon_{tol}, \\ \beta\, h\left(\frac{\varepsilon_{tol}}{\varepsilon}\right)^{0.25}, & \varepsilon < \varepsilon_{tol}, \end{cases}$$

1.4.2 Examples

1.4.2.1 Lotka-Volterra competition model

The Lotka–Volterra competition equations are a simple model of the population dynamics of species competing for some common resource. For given two populations with sizes x and y the model equations are [4]

$$\begin{aligned} \dot{x} &= ax(b - x - cy), \\ \dot{y} &= dy(e - y - fx), \end{aligned} \tag{1.29}$$

Here, positive constant c represents the effect species two has on the population of species one and positive constant f describes the effect species one has on the population of species two. Let us analyse the system (1.29) using parameters

$$a = 0.004,\, b = 50,\, c = 0.75,\, d = 0.001,\, e = 100,\; f = 3\,.$$

Fixed points

Equations (1.29) have four fixed points (x^*, y^*):

$$(0,0)\,, \quad (0,e) = (0,100)\,, \quad (b,0) = (50,0)\,, \quad \left(\frac{b-ce}{1-fc}, \frac{bf-1}{cf-1}\right) = (20,40).$$

Linear stability

In order to analyse the linear stability of (1.29) one derives the corresponding Jacobian

$$J = \begin{pmatrix} ab - 2ax^* - acy^* & -acx^* \\ -dfy^* & de - 2y^* - dfx^* \end{pmatrix}$$

Now one can calculate J for all fixed point values (x^*, y^*) and derive the eigenvalues (λ_1, λ_2) (see Table 1.7).

Table 1.7 Eigenvalues and linear stability analysis for four fixed points of the system (1.29)

(x^*, y^*)	(λ_1, λ_2)	stability
$(0, 0)$	$(0.2, 0.1)$	-
$(0, 100)$	$(-0.1, -0.1)$	+
$(50, 0)$	$(-0.2, -0.05)$	+
$(20, 40)$	$(0.027, -0.14)$	-

Numerical results

Table (1.29) indicates that the trivial fixed point, coresponding to the case, that both populations die out, is unstable. Furthermore, the fixed point $(20, 40)$, corresponding to the case, that both populations survive, is unstable too. That is, both populations will neither die out or survive. Which population will survive (or die out) depends on initial conditions (see Fig. 1.6).

1.4.2.2 Predator-Prey Model

Now let us consider another model of Lotka-Volterra type, which describes prey's (x) and predator's (y) population dynamics in the presence of one another. Equations are:

$$\dot{x} = ax - bxy, \qquad (1.30)$$
$$\dot{y} = cxy - dy.$$

Here $a > 0$ and $c > 0$ are prey's and predator's growth rates whereas $d > 0$ and $b > 0$ describe prey's and predator's death rates respectively.
A typical numerical coefficients are $a = 2$, $b = 0.02$, $c = 0.0002$, $d = 0.8$.
The model (1.30) predicts a cyclical relationship between predator and prey numbers. To see this effect, first we find two fixed points (x^*, y^*) of the system. The fixed points are

$$(0,0)\,, \qquad \left(\frac{d}{c}, \frac{a}{b}\right) = (4 \cdot 10^3, 10^2)\,.$$

The Jacobian of (1.30) is

$$J = \begin{pmatrix} a - by^* & -bx^* \\ cy^* & cx^* - d \end{pmatrix}$$

Now one can calculate eigenvalues for both fixed points (see Table 1.8). Furthermore, one can also

Table 1.8 Eigenvalues and linear stability analysis for four fixed points of the system (1.29)

(x^*, y^*)	(λ_1, λ_2)	stability
$(0,0)$	$(a, -d)$	no, saddle point
$\left(\frac{d}{c}, \frac{a}{b}\right)$	$(i\sqrt{ad}, -i\sqrt{ad})$	neutral stable

calculate the first integral V of the system (1.30):

$$c\dot{x} + b\dot{y} - \frac{d\dot{x}}{x} - \frac{a\dot{y}}{y} = 0 \Rightarrow V := cx + by - d\ln(x) - a\ln(y) = \text{const}\,.$$

The total derivative of V with respect to time reads

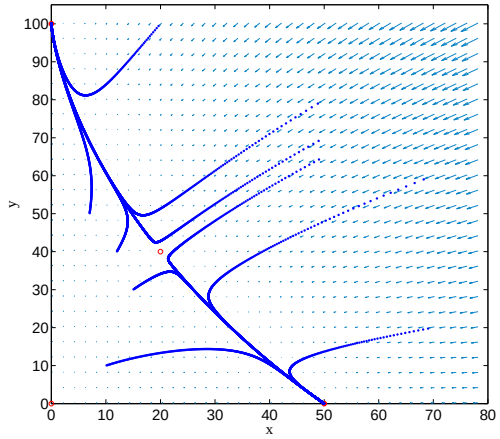

Fig. 1.6 Numerical solution of (1.29) over the time interval $t \in [0, 150]$ on the phase plane (x, y) by the classical RK4 method with the step size $h = 0.025$ for different initial values. Open red circles denote unstable fixed points, whereas filled red circles represent stable fixed points.

(a) (b)

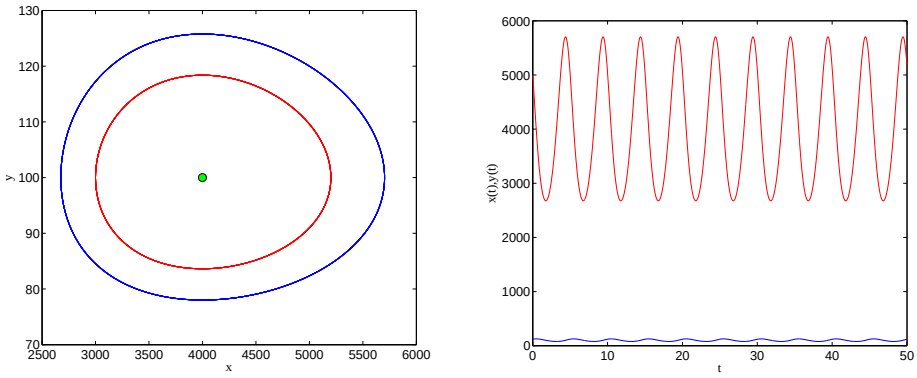

Fig. 1.7 Numerical solution of the system (1.30), calculated with RK4 method. (a) Solutions on the phase plane, corresponding to two different initial conditions: $(5\cdot 10^3, 120)$ and $(3\cdot 10^3, 10^2)$. (b) A cyclical relationship between predator and prey numbers, calculated for the initial condition $(5\cdot 10^3, 120)$.

$$\frac{dV}{dt} = \left(c - \frac{d}{x}\right) x \left(a - by\right) + \left(b - \frac{a}{y}\right) y \left(cx - d\right) = 0$$

That is, solutions of (1.30) can not leave levels of V. This is illustrated on Fig. 1.7 (a), where two numerical solutions, corresponding to two different initial conditions $(5\cdot 10^3, 120)$ and $(3\cdot 10^3, 10^2)$ are presented. The green point denotes the neutral stable fixed point. Oscillations of both populations, corresponding to the initial value $(5\cdot 10^3, 120)$ is presented on Fig. 1.7 (b).
Now suppose that prey's growth rate is periodic in time, e.g,

$$a := a\left(1 + \varepsilon \sin(\omega t)\right),$$

where $\varepsilon \in [0, 1)$ and let be $\omega = \pi$. In this case, depending on control parameter ε, quasiperiodic or even chaotic behaviour can be expected. Figure 1.8 illustates an example of quasiperiodic behaviour.

1.4.2.3 Forced oscillations: Pohl's pendulum

The Pohl's wheel is a rotating pendulum with electromagnetic brake, spiral spring and variable stimulation, which can demonstrate harmonic oscillations as well as chaotic motion. The equation of motion of the wheel reads

$$J\ddot{\varphi} + K\dot{\varphi} + D\varphi - N\sin(\varphi) = \hat{F}\sin(\omega t + \Omega). \tag{1.31}$$

Here φ denotes the rotation angle, J is a inertia moment of the pendulum about the axis of rotation, K is a damping constant, D stays for the torque per unit angel, and, finally, $N = mgr\sin(\varphi)$ is a projection of the variable stimulation's moment (m is a external mass, r is a radius of the wheel). In addition, $\hat{F}\sin(\omega t + \Omega)$ is an external forsing of the amplitude $\hat{F}$, frequency ω and the free phase Ω.

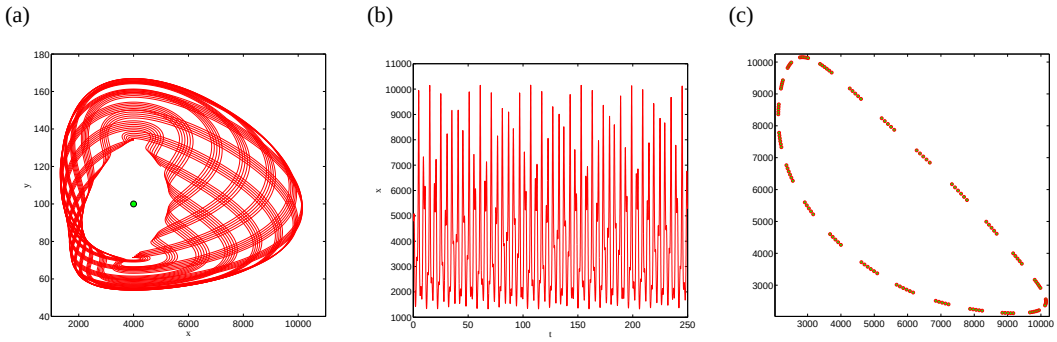

Fig. 1.8 Numerical solution for (1.30) calculated with RK4 method for the case $\varepsilon = 0.4$. Initial condition is $(5 \cdot 10^3, 120)$. (a) Solutions on phase plane; (b) Quasiperiodic oscillations of the preys population; (c) The first return map.

In order to solve Eq. (1.31) numerically we rewrite it to a system of first order ODE's. Substitution $x := \varphi$, $y := \dot{\varphi}$, $z := t$ leads to the system

$$\begin{aligned} \dot{x} &= y\,, \\ \dot{y} &= -ay - bx + c\sin(x) + d\sin(\omega z)\,, \\ \dot{z} &= 1\,, \end{aligned} \tag{1.32}$$

with

$$a := \frac{K}{J}\,, \quad b := \frac{D}{J}\,, \quad c := \frac{N}{J}\,, \quad d := \frac{\hat{F}}{J}\,.$$

We solve the system (1.32) with the classical RK4 method (1.25) with the time step $h = 0.025$ over the time interval $t \in [0, 150]$. Other parameters are

$$a = 0.799\,, \quad b = 9.44\,, \quad c = 14.68, \quad d = 2.1\,.$$

Furthermore, we use the frequency of the external forcing ω as a control parameter. We start at $\omega = 2.5$. The result is shown on Fig. 1.9 (a)-(c). Figure 1.9 (a) shows the solution of (1.32) on the phase space (φ, φ'). One can see, that the solution corresponds to forced oscillations with the period one (see Fig. 1.9 (b) as well). Period one oscillations can also be recognised from the first return map (Fig. 1.9 (c)). Now we increase the control parameter ω to $\omega = 2.32$. The results can be seen on Fig. 1.10 (a)–(c). In this case the system oscillates between two values, so one can speak about period two oscillations (or about periode-doubling bifurcation). Further increasing of ω leads to second periode-doubling bifurcation and period four oscillation sets in (see Fig. 1.11 (a)–(c)). Finally, we increase ω further and chaotic oscillations can be observed (see Fig. 1.12 (a)–(c)). The first return map, shown on Fig. (1.12) (c) indicates the structure of the chaotic motion: the n'th maximal value of φ is predicted by the $n-1$'th one.

1.4.2.4 Lorenz system

Let us consider the so-called *Lorenz equations*

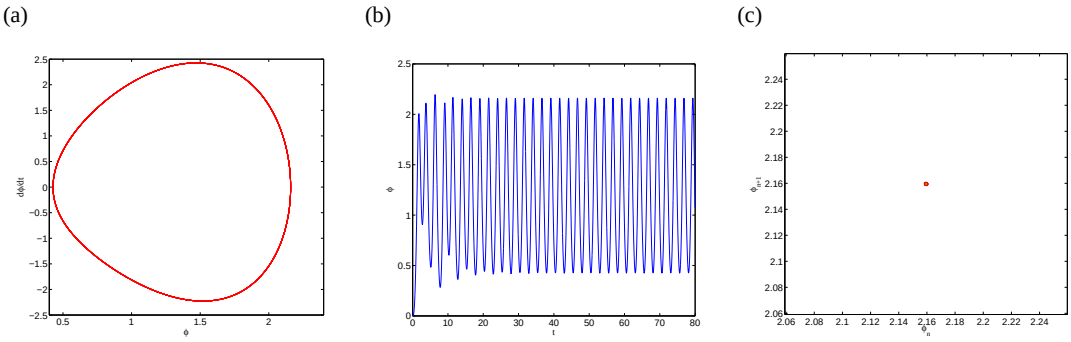

Fig. 1.9 Solution of Eq. (1.31) corresponding to $\omega = 2.5$. (a) Solution on the phase plane. (b) One period oscillations on the φ, t plot. (c) The first return map.

$$\begin{aligned} \dot{x} &= \sigma(y-x), \\ \dot{y} &= rx - x - xz, \\ \dot{z} &= xy - bz. \end{aligned} \tag{1.33}$$

Here $\sigma > 0$ is Prandtl number, $r > 0$ stays for normalized Rayleigh number, whereas $b > 0$ is a geometric factor. The function $x(t)$ is propotional to the intensity of convection motion, $y(t)$ is propotional to the temperature difference between ascending and descending currents and $z(t)$ is propotional to the distortion of vertical tmperature profile from the linear one.
This system was investigated by Ed Lorenz in 1963. Its purpose was to provide a simplifed model of atmospheric convection [6, 7].

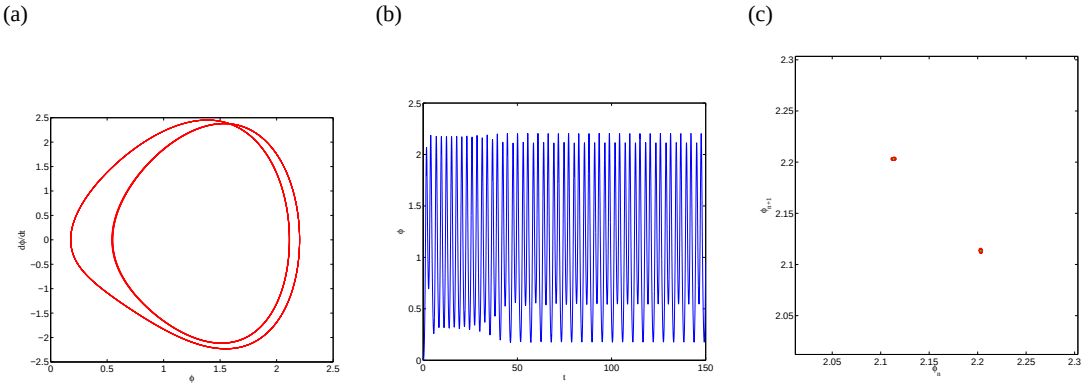

Fig. 1.10 Solution of Eq. (1.31), corresponding to the periode-doubling bifurcation for $\omega = 2.32$. (a) Solution on the phase plane. (b) Two period oscillations on the φ, t plane . (c) The first return map.

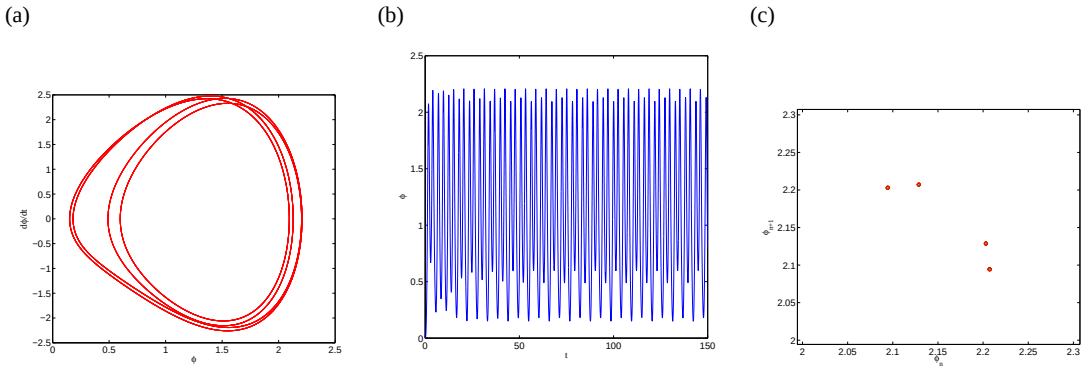

Fig. 1.11 Solution of Eq. (1.31), corresponding to the second periode-doubling bifurcation for $\omega = 2.3$. (a) Solution on the phase plane. (b) Period four oscillations on the φ, t plane. (c) The first return map.

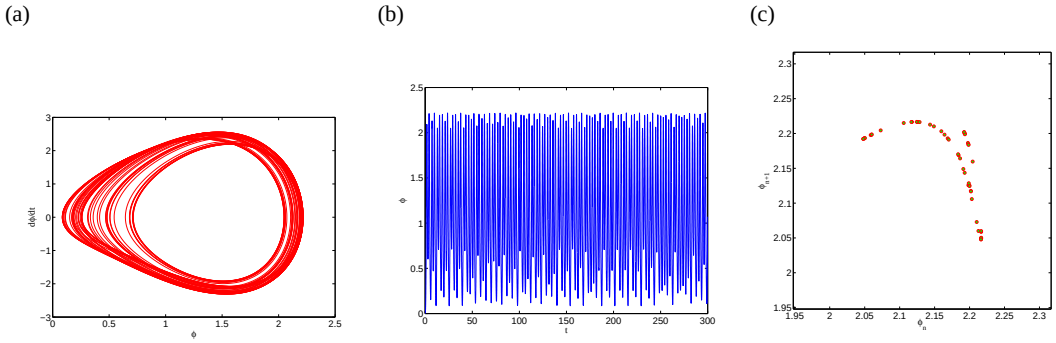

Fig. 1.12 Solution of Eq. (1.31), corresponding to the chaotic oscillation for $\omega = 2.25$.(a) Solution on the phase plane. (b) Chaotic oscillations on the φ, t plane. (c) The first return map, indicationg the chaotic regime.

Symmetry

The system (1.33) admits a symmetry

$$(x, y, z) \rightarrow (-x, -y, z)\,.$$

Fixed Points

The fixed points (x^*, y^*, z^*) are

(a) $x^* = y^* = z^* = 0$ – corresponds to the state of no convection;
(b) $C^+ = (\sqrt{b(r-1)}, \sqrt{b(r-1)}, r-1)$ and $C^- = (-\sqrt{b(r-1)}, -\sqrt{b(r-1)}, r-1)$ – correspond to the state of steady convection. Note that both solutions exist only for $r > 1$.

Linear Stability

The Jacobian of the system (1.33) reads

$$J(x^*, y^*, z^*) = \begin{pmatrix} -\sigma & \sigma & 0 \\ \sigma - z^* & -1 & -x^* \\ y^* & x^* & -b \end{pmatrix}$$

(a) The trivial solution $(x^*, y^*, z^*) = (0, 0, 0)$: In this case the matrix J can be written as 2×2 matrix,

$$J_0 = \begin{pmatrix} -\sigma & \sigma \\ r & -1 \end{pmatrix}$$

as an linearized equation for $z(t)$ is

$$\dot{z} = -bz$$

decoupled. The stability of (1.33) can be determined using the trace and the deteminant of J_0:

$$\mathrm{Sp}(J_0) = -\sigma - 1 < 0\,, \qquad \det(J_0) = \sigma(1-r) > 0 \Rightarrow r < 1\,.$$

That is, the trivial solution is stable if $r < 1$.

(b) Stability of C^+ and C^-: Consider the case $r > 1$, so both nontrivial solutions exist. The characteristic polynomial reads

$$\lambda^3 + (\sigma + b + 1)\lambda^2 + (r + \sigma) b \lambda + 2b\sigma(r-1) = 0\,.$$

The eigenvalues consist of one real negative root and a pair of complex conjugate roots [6]. The complex roots can be found using the ansatz $\lambda = i\omega$. Substitution into characteristic polynomial leads to the expression for the crtical Rayleigh number r_H

$$r_H = \frac{\sigma + b + 3}{\sigma - b - 1}\,, \qquad \sigma > b + 1.$$

The third eigenvalue λ_3 can be found as

$$\lambda_3 = -(\sigma + b + 1) < 0$$

That is, the nontrivial solutions C^+ and C^- are stable for

$$1 < r < r_H\,, \qquad \sigma > b + 1.$$

The nontrivial solutions loose stability at r_H via the Hopf bifurcation. One can show that this bifurcation is subcritical [6]. That is, the limit circles are unstable and exist only for $r < r_H$.

Time behaviour for different r's

In his study, Lorenz chose the parameter values

$$b = \frac{8}{3}\,, \qquad \sigma = 10\,.$$

With this choise the steady state becomes unstable at

$$r = r_H = \frac{470}{19} \approx 27.74\ldots.$$

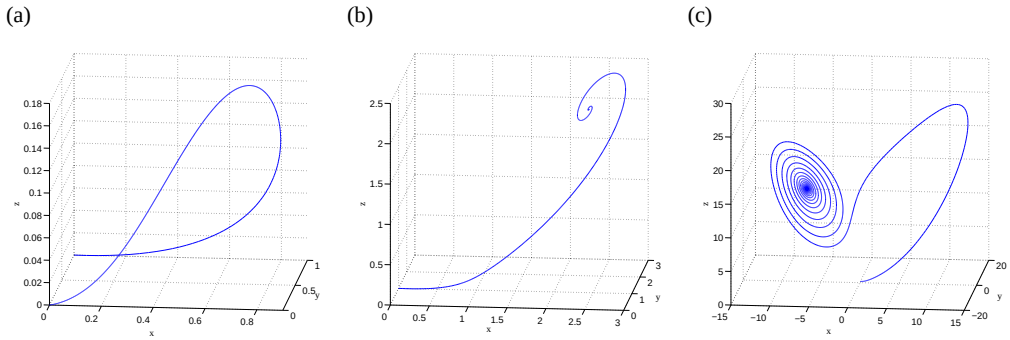

Fig. 1.13 Solutions of the Lorenz equations (1.33), corresponding to different values of r. (a) $r = 0.5$ - the origin is stable; (b) $r = 3$ - the origin is unstable. All trajectories converge to one of stable nontrivial fixed points C^+ or C^-; (c) $r = 16$ -the basin of attraction around C^+ and C^- are no longer distinct.

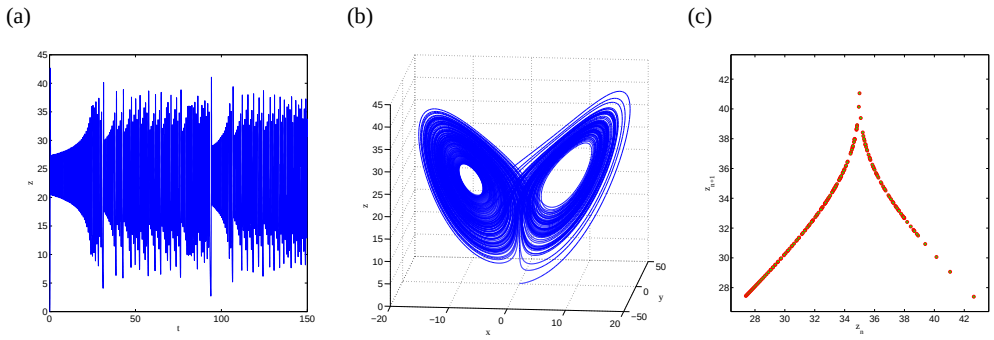

Fig. 1.14 (a) Solution of the Lorenz equations (1.33) on (t, z) plane, computed at $r = 26$. (b) Solution of (1.33) at $r = 26$ on the tree-dimensional phase space. (c) The Lorenz map.

The initial value was $(0, 1, 0)$.
Now let us summarize what happens to the solutions of (1.33) as r is increased [6, 7]:

- $0 < r < 1$: The origin is stable node (see Fig. 1.13 (a)).
- $1 < r < 24.74$: The origin becomes unstable and bifurcates into a pair of solutions C^+ and C^-. All trajectories converge to either one or another point (see Fig. 1.13 (b)). At $r \approx 13.926$ the origin becomes a homoclinic point, i.e., beyond this point trajectories can cross forward and backward between C^+ and C^- befor settle down to them (see Fig. 1.13 (c)).
- $r = 24.74$: Both C^+ and C^- become unstable via subcritical Hopf bifurcation.
- $r > 24.74$: After initial transient the solution settes into irregular oscillation and is aperiodic (see Fig. 1.14 (a)). On the phase space, the time spent wandering near sets around C^+ and C^- becomes infinite and the set becomes *a strange attractor* (see Fig. 1.14 (b)).

Lorenz map

Lorenz found a way to analyse the dynamics on the strange attractor. He has considered a projection of the three-dimensional phase space on the (t, z) plane. The idea was that if we consider the n'th local maximum of the function $z(t)$, z_n, it should predict z_{n+1}. To check this, one can estimate the local maxima of the function $z(t)$ and plot z_{n+1} versus z_n. The resulting function, presented on Fig. 1.14 (c) is now called *the Lorenz map*.

1.5 Predictor-Corrector Methods

The Runge-Kutta methods, introduced in Section 1.4 are referred to as *single-step methods*, because they use only the information from one previous point to evolve from x_n to x_{n+1}. In addition to single-step methods, there is also a broad class of so-called *multi-step* integration methods, which use information at more than one previous point to estimate solution at next point.

The main advantages of Runge-Kutta methods (1.22) are that they are easy to implement, rather stable, and "self-starting", (i.e., we do not have to treat the first few steps taken by a single-step integration method as special cases). On the other hand, the primary disadvantage of Runge-Kutta methods (1.22) compared to multi-step methods is that they require significantly more computer time than multi-step methods of comparable accuracy. In addition, the local truncation error of a milti-step method can be determinated and a correction term can be included, which improves the accuracy of the numarical approximation *at each step* [2]. One of the example of multi-step methods are the various *predictor-corrector methods*, which proceed by extrapolating a polynomial fit to the derivative from the previous points to the new point (the predictor step), then using this to interpolate the derivative (the corrector step) [3].

1.5.1 The Adams-Bashforth-Moulton method

Again, let us consider IVP (1.1)–(1.2). Integrating both sides of Eq. (1.1) over one time step from t_n to t_{n+1} we obtain the *exact* relation (1.14):

$$\mathbf{x}(t_{n+1}) - \mathbf{x}(t_n) = \int_{t_n}^{t_{n+1}} f(t, \mathbf{x}(t))\, dt\,.$$

Now a numerical integration method can be used to approximate the definite integral in the last equation. The Adams-Bashforth-Moulton method is a multi-step method that proceeds in two steps [2, 3]. The first step is called *the Adams-Bashforth predictor*. The predictor uses the Lagrange polynomial approximation for the function $f(t, \mathbf{x}(t))$ based on the nodes $(t_{n-3},\, f_{n-3})$, $(t_{n-2},\, f_{n-2})$, $(t_{n-1},\, f_{n-1})$ and $(t_n,\, f_n)$. After integrating over the interval $[t_n, t_{n+1}]$ the predictor reads

$$p_{n+1} = \mathbf{x}_n + \frac{h}{24}\Big(-9\, f_{n-3} + 37\, f_{n-2} - 59\, f_{n-1} + 55\, f_n\Big)\,. \tag{1.34}$$

The second step is *the Adams-Moulton corrector* and is developed similary. A seecond Lagrange polynomial for the function $f(t, \mathbf{x}(t))$ is constracted. In this case, it based on the points $(t_{n-2},\, f_{n-2})$, $(t_{n-1},\, f_{n-1})$, $(t_n,\, f_n)$ and the new point $(t_{n+1},\, f_{n+1}) = (t_{n+1},\, f(t_{n+1},\, p_{n+1}))$. After integrating over the interval $[t_n, t_{n+1}]$ the following relation for the corrector is obtained

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{24}\left(f_{n-2} - 5\, f_{n-1} + 19\, f_n + 9\, f_{n+1} \right). \tag{1.35}$$

Notice that the method (1.34)–(1.35) is not "self-starting", i.e., four initial points $(t_n, \mathbf{x}_n)$, $n = 0, 1, 2, , 3$ must be given in order to estimate the points $(t_n, \mathbf{x}_n)$ for $n \geq 4$.

Error Estimation

The local truncation error for both predictor (1.34) and corrector (1.35) terms are of the order $\mathcal{O}(h^5)$, namely [2]

$$\begin{aligned} \mathbf{x}(t_{n+1}) - p_{n+1} &= \frac{251}{720}\mathbf{x}^{(5)} h^5, \\ \mathbf{x}(t_{n+1}) - \mathbf{x}_{n+1} &= -\frac{19}{720}\mathbf{x}^{(5)} h^5. \end{aligned}$$

That is, for small values of h one can eliminate terms with fifth derivative and the error estimate reads

$$\mathbf{x}(t_{n+1}) - \mathbf{x}_{n+1} \approx \frac{-19}{270}\left(\mathbf{x}_{n+1} - p_{n+1} \right). \tag{1.36}$$

Equation (1.36) gives an estimation of the local truncation error based on the *two computed values* p_{n+1} and $\mathbf{x}_{n+1}$, but $\mathbf{x}^{(5)}$.

Example 1

Solve an IVP

$$\dot{x} = t^2 - x, \qquad x_0 := x(0) = 1 \tag{1.37}$$

over the time interval $t \in [0, 5]$ with the Adams-Bashforth-Moulton method (1.34)–(1.35) using the time step $h = 0.05$. The three starting x_1, x_2 and x_3 values can be calculated via the classical RK4 method. The exact solution of the problem is [2]

$$x(t) = t^2 - 2t + 2 - e^{-t}.$$

The result of the calculation is presented on Fig. 1.15.

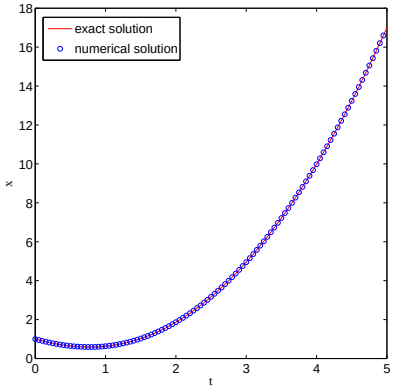

Fig. 1.15 Numerical solution of Eq. (1.37) over the interval $[0, 5]$ by the Adams-Bashforth-Moulton method (1.34)–(1.35) with the step size $h = 0.05$ (blue open circles). The exact solution of the problem is depicted by the red line.

Chapter 2
Boundary Value Problem

A boundary value problem (BVP) is a problem, typically an ODE or a PDE, which has values assigned on the physical boundary of the domain in which the problem is specified. Let us consider a genearal ODE of the form

$$\mathbf{x}^{(n)} = f(t, \mathbf{x}, \mathbf{x}', \mathbf{x}'', \cdots, \mathbf{x}^{(n-1)}), \qquad t \in [a, b] \tag{2.1}$$

At $t = a$ and $t = b$ the solution is supposed to satisfy

$$\begin{aligned} r_1(\mathbf{x}(a)\mathbf{x}'(a), \cdots, \mathbf{x}^{(n-1)}(a), \mathbf{x}(b)\mathbf{x}'(b), \cdots, \mathbf{x}^{(n-1)}(b)) = 0\,, \\ \vdots \\ r_n(\mathbf{x}(a)\mathbf{x}'(a), \cdots, \mathbf{x}^{(n-1)}(a), \mathbf{x}(b)\mathbf{x}'(b), \cdots, \mathbf{x}^{(n-1)}(b)) = 0\,. \end{aligned} \tag{2.2}$$

The resulting problem (2.1)–(2.2) is called a *two point boundary value problem* [4].
In order to be useful in applications, a BVP (2.1)–(2.2) should be *well posed*. This means that given the input to the problem there exists a unique solution, which depends continuously on the input. However, questions of existence and uniqueness for BVPs are much more difficult than for IVPs and there is no general theory.

2.1 Single shooting methods

2.1.1 Linear shooting method

Consider a linear two-point second-order BVP of the form

$$x''(t) = p(t)x'(t) + q(t)x(t) + r(t)\,, \qquad t \in [a, b] \tag{2.3}$$

with

$$x(a) = \alpha\,, \qquad x(b) = \beta\,.$$

The main idea of the method is to reduce the solution of the BVP (2.3) to the solution of an initial value problem [5, 2]. Namely, let us consider two special IVPs for two functions $u(t)$ and $v(t)$. Suppose that $u(t)$ is a solution of the IVP

$$u''(t) = p(t)\,u'(t) + q(t)\,u(t) + r(t)\,, \qquad u(a) = \alpha\,, \qquad u'(a) = 0$$

and $v(t)$ is the unique solution to the IVP

$$v''(t) = p(t)\,v'(t) + q(t)\,v(t)\,, \qquad v(a) = 0\,, \qquad v'(a) = 1\,.$$

Then the linear combination

$$x(t) = u(t) + c\,v(t)\,, \qquad c = \text{const}\,. \tag{2.4}$$

is a solution to BVP (2.3). The unknown constant c can be found from the boundary condition on the right end of the time interval, i.e.,

$$x(b) = u(b) + c\,v(b) = \beta \;\Rightarrow c = \frac{\beta - u(b)}{v(b)}\,.$$

That is, if $v(b) \neq 0$ the unique solution of (2.3) reads

$$x(t) = u(t) + \frac{\beta - u(b)}{v(b)}\,v(t)\,.$$

Example 1

Let us solve a BVP [2]

$$\begin{aligned} x''(t) &= \frac{2t}{1+t^2}\,x'(t) - \frac{2}{1+t^2}\,x(t) + 1\,, \\ x(0) &= 1.25, \qquad x(1) = -0.95\,. \end{aligned} \tag{2.5}$$

over the time interval $t \in [0, 4]$ using the linear shooting method (2.4).
According to Eq. (2.4) the solution of this equation has the form

$$x(t) = u(t) - \frac{0.95 + u(4)}{v(4)}\,v(t)\,,$$

where $u(t)$ and $v(t)$ are solutions of two IVPs

$$u''(t) = \frac{2t}{1+t^2}\,u'(t) + \frac{2}{1+t^2}\,u(t) + 1\,, \qquad u(0) = 1.25\,, \qquad u'(0) = 0$$

and

$$v''(t) = \frac{2t}{1+t^2}\,v'(t) + \frac{2}{1+t^2}\,v(t)\,, \qquad v(0) = 0\,, \qquad v'(0) = 1\,.$$

Numerical solution of the problem 2.5 as well as both dunctions $u(t)$ and $v(t)$ are presented on Fig. 2.1

2.1.2 Single shooting for general BVP

For a general BVP for a second-order ODE, the simple shooting method is stated as follows: Let

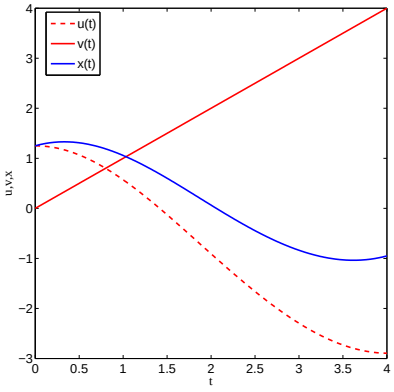

Fig. 2.1 Numerical solution of Eq. (2.5) over the interval $[0,4]$ by the linear shooting method (2.4).

$$\begin{aligned} x''(t) &= f(t,x(t),x'(t)), \qquad t \in [a,b] \\ x(a) &= \alpha, \qquad x(b) = \beta\,. \end{aligned} \tag{2.6}$$

be the BVP in question and let $x(t,s)$ denote the solution of the IVP

$$\begin{aligned} x''(t) &= f(t,x(t),x'(t)), \qquad t \in [a,b] \\ x(a) &= \alpha, \qquad x'(a) = s, \end{aligned} \tag{2.7}$$

where s is a parameter that can be varied. The IVP (2.7) is solved with different values of s with, e.g., RK4 method till the boundary condition on the right side $x(b) = \beta$ becomes fulfilled. As mentioned above, the solution $x(t,s)$ of (2.7) depends on the parameter s. Let us define a function

$$F(s) := x(b,s) - \beta\,.$$

If the BVP (2.6) has a solution, then the function $F(s)$ has a root, which is just the value of the slope $x'(a)$ giving the solution $x(t)$ of the BVP in question. The zeros of $F(s)$ can be found with, e.g., *Newton's method* [3].
The Newton's method is probably the best known method for finding numerical approximations to the zeroes of a real-valued function. The idea of the method is to use the first few terms of the Taylor series of a function $F(s)$ in the vicinity of a suspected root, i.e.,

$$F(s_n + h) = F(s_n) + F'(s_n)\,h + \mathcal{O}(h^2)\,.$$

where s_n is a n'th approximation of the root. Now if one inserts $h = s - s_n$, one obtains

$$F(s) = F(s_n) + F'(s_n)\,(s - s_n)\,.$$

As the next approximation s_{n+1} to the root we choose the zero of this function, i.e.,

$$F(s_{n+1}) = F(s_n) + F'(s_n)\,(s_{n+1} - s_n) = 0 \Rightarrow s_{n+1} = s_n - \frac{F(s_n)}{F'(s_n)}\,. \tag{2.8}$$

The derivative $F'(s_n)$ can be calculated using the forward difference formula

$$F'(s_n) = \frac{F(s_n + \delta s) - F(s_n)}{\delta s}$$

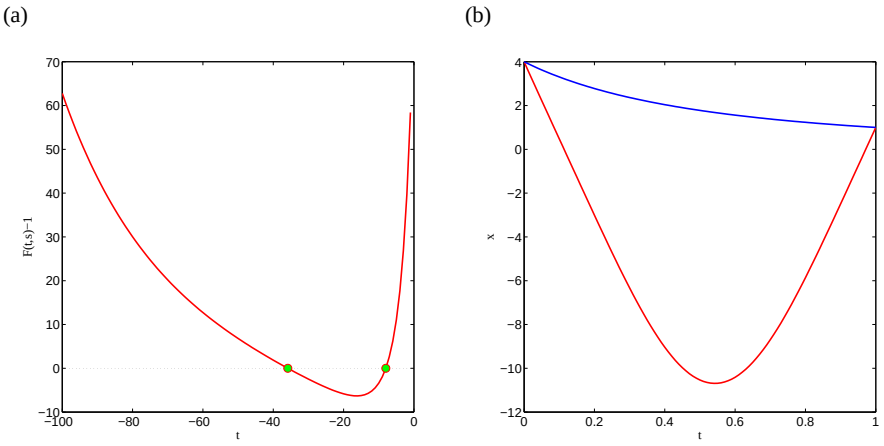

Fig. 2.2 Numerical solution of BVP (2.9) with single shooting method. (a) The Function $F(s) = x(t,s) - 1$ is presented. Green points depict two zeros of this function, which can be found with Newton's method. (b) Two solutions of (2.9) corresponding to two different values of parameter s (the red line corresponds to $s = -35.8$, whereas the blue one – to $s = -8.0$).

where δs is small. Notice that this procedure can be unstable near a horizontal asymptote or a local extremum.

Example 1

Consider a simple nonlinear BVP [5]

$$\begin{aligned} x''(t) &= \frac{3}{2}x(t)^2, \\ x(0) &= 4, \qquad x(1) = 1 \end{aligned} \tag{2.9}$$

over the interval $t \in [0, 1]$ and let us solve it numerically with the single shooting method discussed above. First of all we define a corresponding IVP

$$x''(t) = \frac{3}{2}x(t)^2 \qquad x(0) = 4\,, \quad x'(0) = s$$

over $t \in [0, 1]$ and solve it for different values of s, e.g., $s \in [-100, 0]$ with the classical RK4 method. The result of calculation is presented on Fig. 2.2 (a). One can see, that the function $F(s) = x(t,s) - 1$ admits two zeros, depicted on Fig. 2.2 (a) as green points. In order to find them we use the Newton's method, discussed above. The method gives an approximation to both zeros of the function $F(s)$: $s = \{-35.8, -8.0\}$, which give the right slope $x'(0)$. Both solutions, corresponding to two different values of s are presented on Fig. 2.2 (b).

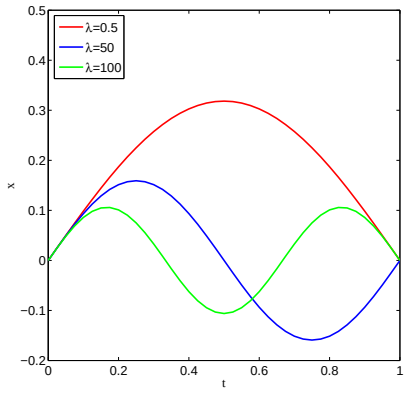

Fig. 2.3 Numerical solutions of Eq. (2.10) over the interval $[0,1]$ by single shooting method. First three eigenfunctions, corresponding to eigenvalues $\lambda = \{\pi^2, (2\pi)^2, (3\pi)^2\}$ are presented.

Example 2

Let us consider a linear eigenvalue problem of the form

$$x'' + \lambda x = 0, \qquad x(0) = x(1) = 0, \quad x'(0) = 1 \tag{2.10}$$

over $t \in [0,1]$ with the simple shooting method. The exact solution is

$$\lambda = n^2\pi^2, \quad n \in \mathbb{N}.$$

In order to apply the simple shooting method we consider a corresponding IVP of the first order with additional equation for the unknown function $\lambda(t)$:

$$x' = y, \qquad y' = -\lambda x, \quad \lambda' = 0$$

with

$$x(0) = 0, \quad x'(0) = 1, \quad \lambda(0) = s.$$

where s is a free shooting parameter. Here we choose $s = \{0.5, 50, 100\}$. Results of the shooting with these initial parameters are shown on Fig. 2.3. One can see, that numerical solutions correspond to first three eigenvalues $\lambda = \{\pi^2, (2\pi)^2, (3\pi)^2\}$.

Example 3

Consider a nonlinear BVP of the fourth order [4]

$$x^{(4)}(t) - (1+t^2)x''(t)^2 + 5x(t)^2 = 0, \qquad t \in [0,1] \tag{2.11}$$

with

$$x(0) = 1, \quad x'(0) = 0, \quad x''(1) = -2, \quad x'''(1) = -3.$$

Our goal is to solve this equation with the simple shooting method. To this end, first we rewrite the equation as a system of four ODE's of the first order:

$$
\begin{aligned}
x_1' &= x_2\,, \\
x_2' &= x_3\,, \qquad & x_1(0) = 1\,, \qquad & x_3(1) = -2\,, \\
x_3' &= x_4\,, \qquad & x_2(0) = 0\,, \qquad & x_4(1) = -3\,, \\
x_4' &= (1+t^2)x_3^2 - 5x_1^2\,.
\end{aligned}
$$

As the second step we consider correspondig IVP

$$
\begin{aligned}
x_1' &= x_2\,, \\
x_2' &= x_3\,, \qquad & x_1(0) = 1\,, \qquad & x_3(1) = p\,, \\
x_3' &= x_4\,, \qquad & x_2(0) = 0\,, \qquad & x_4(1) = q\,, \\
x_4' &= (1+t^2)x_3^2 - 5x_1^2
\end{aligned}
$$

with two free shooting parameters p and q. The solution of this IVP fulfilles following two requirements:

$$
\begin{aligned}
F_1(p,q) &:= x_3(1,p,q) + 2 = 0\,, \\
F_2(p,q) &:= x_4(1,p,q) + 3 = 0\,.
\end{aligned}
$$

That is, a system of nonlinear algebraic equations should be solved to find (p,q). The zeros of the system can be found with the Newton's method (2.8). In this case the iteration step reads

$$
s_{i+1} = s_i - \frac{F(s_i)}{DF(s_i)}
$$

where $s = (p,q)^T$, $F = (F_1, F_2)^T$ and

$$
DF(s_i) = \begin{pmatrix} \frac{\partial F_1}{\partial p} & \frac{\partial F_1}{\partial q} \\ \frac{\partial F_2}{\partial p} & \frac{\partial F_2}{\partial q} \end{pmatrix}
$$

is a Jacobian of the system and

$$
\begin{aligned}
\frac{\partial F_i}{\partial p} &= \frac{F_i(p+\Delta p, q) - F_i(p,q)}{\Delta p}\,, \\
\frac{\partial F_i}{\partial q} &= \frac{F_i(p, q+\Delta q) - F_i(p,q)}{\Delta q}\,,
\end{aligned}
$$

where $i = 1, 2$ and Δp, Δq are given values. Numerical solution of the problem in question is presented on Fig. 2.4.

2.2 Finite difference Method

One way to solve a given BVP over the time interval $t \in [a,b]$ numerically is to approximate the problem in question by *finite differences* [4, 5, 2]. We form a partition of the domain $[a,b]$ using *mesh points* $a = t_0, t_1, \ldots, t_N = b$, where

$$
t_i = a + i\,h\,, \qquad h = \frac{b-a}{N}\,, \qquad i = 0, 1, \ldots N\,.
$$

Difference quotient approximations for derivatives can be used to solve BVP in question [5, 2]. In particular, using a Taylor expansion in the vicinity of the point t_j, for the first derivative one obtains

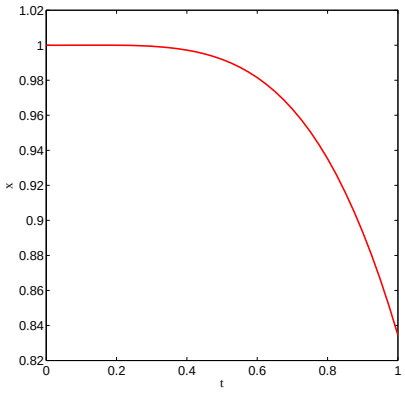

Fig. 2.4 Numerical solutions of (2.11) over the interval $[0,1]$ by single shooting method. Parameters are: $\Delta p = \Delta q = 0.05$, the time step $h = 0.025$, initial shooting parameters $(p_0, q_0) = (0,0)$.

a forward difference

$$x'(t_i) = \frac{x(t_{i+1}) - x(t_i)}{h} + \mathscr{O}(h)\,. \tag{2.12}$$

In a similar way one gets *a backward difference*

$$x'(t_i) = \frac{x(t_i) - x(t_{i-1})}{h} + \mathscr{O}(h)\,. \tag{2.13}$$

We can combine these two approaches and derive *a central difference*, which yields a more accurate approximation:

$$x'(t_i) = \frac{x(t_{i+1}) - x(t_{i-1})}{2h} + \mathscr{O}(h^2)\,. \tag{2.14}$$

The second derivative $x''(t_i)$ can be found in the same way using the linear combination of different Taylor expansions. For example, a central difference reads

$$x''(t_i) = \frac{x(t_{i+1}) - 2x(t_i) + x(t_{i-1})}{h^2} + \mathscr{O}(h^2)\,. \tag{2.15}$$

2.2.1 Finite Difference for linear BVP

Let us consider a linear BVP of the second order (2.3)

$$x'' = p(t)x'(t) + q(t)x(t) + r(t)\,, \quad t \in [a,b]\,, \quad x(a) = \alpha\,, \quad x(b) = \beta\,.$$

and introduce the notation $x(t_i) = x_i$, $p(t_i) = p_i$, $q(t_i) = q_i$ and $r(t_i) = r_i$. Then, using Eq. (2.14) and Eq. (2.15) one can rewrite Eq. (2.3) as *a difference equation*

$$\begin{aligned} x_0 &= \alpha\,, \\ \frac{x_{i+1} - 2x_i + x_{i-1}}{h^2} &= p_i \frac{x_{i+1} - x_{i-1}}{2h} + q_i x_i + r_i\,, \qquad i = 1, \ldots, N-1\,, \\ x_N &= \beta\,. \end{aligned}$$

Now we can multiply both sides of the second equation with h^2 and collect terms, involving x_{i-1}, x_i and x_{i+1}. As result we get a system of linear equations

$$\left(1+\frac{h}{2}p_i\right)x_{i-1}-(2+h^2q_i)x_i+\left(1-\frac{h}{2}p_i\right)x_{i+1}=h^2r_i\,,\qquad i=1,2,\dots N-1\,.$$

or, in matrix notation

$$A\mathbf{x}=b\,, \tag{2.16}$$

or, more precisely

$$\begin{pmatrix}
-(2+h^2q_1) & 1-\frac{h}{2}p_1 & 0 & \dots & \dots & 0\\
1+\frac{h}{2}p_2 & -(2+h^2q_2) & 1-\frac{h}{2}p_2 & 0 & \dots & 0\\
0 & 1+\frac{h}{2}p_3 & -(2+h^2q_3) & 1-\frac{h}{2}p_3 & \dots & 0\\
\dots & \dots & \dots & \dots & \dots & \dots\\
\dots & \dots & \dots & \dots & \dots & 1-\frac{h}{2}p_{N-2}\\
0 & \dots & \dots & 0 & 1+\frac{h}{2}p_{N-1} & -(2+h^2q_{N-1})
\end{pmatrix}
\begin{pmatrix} x_1\\ x_2\\ x_3\\ \cdot\\ \cdot\\ x_{N-1}\end{pmatrix}=
\begin{pmatrix} h^2r_1-\gamma_1\\ h^2r_2\\ h^2r_3\\ \cdot\\ \cdot\\ h^2r_{N-1}-\gamma_N\end{pmatrix},$$

where

$$\gamma_1=\alpha\left(\frac{h}{2}p_1+1\right),\qquad \gamma_N=\beta\left(1-\frac{h}{2}p_{N-1}\right).$$

Our goal is to find unknown vector **x**. To this end we should invert the matrix A. This matrix has a band structure and is *tridiagonal*. For matrices of this kind a tridiagonal matrix algorithm (TDMA), also known als *Thomas algorithm* can be used (see Appendix A for details).

Example

Solve a linear BVP [4]

$$\begin{aligned} -x''(t)-(1+t^2)x(t)&=1\,,\\ x(-1)=x(1)&=0\end{aligned} \tag{2.17}$$

over $t\in[-1,1]$ with finite difference method. First we introduce discrete set of nodes $t_i=-1+ih$ with given time step h. According to notations used in previous section, $p(t)=0$, $q(t)=-(1+t^2)$, $r(t)=-1$, $\alpha=\beta=0$. Hence, the linear system (2.16) we are interested in reads

$$\begin{pmatrix}
-(2+h^2q_1) & 1 & 0 & \dots & \dots & 0\\
1 & -(2+h^2q_2) & 1 & 0 & \dots & 0\\
0 & 1 & -(2+h^2q_3) & 1 & \dots & 0\\
\dots & \dots & \dots & \dots & \dots & \dots\\
\dots & \dots & \dots & \dots & \dots & 1\\
0 & \dots & \dots & 0 & 1 & -(2+h^2q_{N-1})
\end{pmatrix}
\begin{pmatrix} x_1\\ x_2\\ x_3\\ \cdot\\ \cdot\\ x_{N-1}\end{pmatrix}=-
\begin{pmatrix} h^2\\ h^2\\ h^2\\ \cdot\\ \cdot\\ h^2\end{pmatrix}.$$

The numerical solution of the problem in question is presented on Fig. 2.5.

2.2.2 Finite difference for linear eigenvalue problems

Consider a Sturm-Liouville problem of the form

$$-x''(t)+q(t)x(t)=\lambda\,v(t)x(t)\,, \tag{2.18}$$

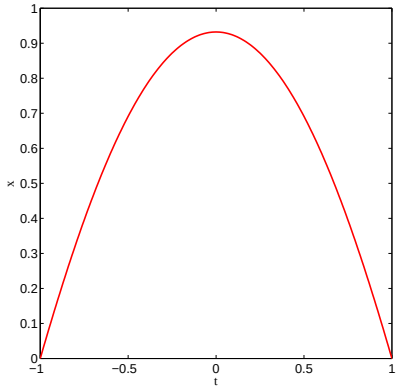

Fig. 2.5 Numerical solutions of (2.17) over the interval $[-1, 1]$ by finite difference method.

over $t \in [a, b]$ with

$$x(a) = 0\,, \qquad x(b) = 0\,.$$

Introducing notation $x_i := x(t_i)$, $q_i := q(t_i)$, $v_i := v(t_i)$, we can write down a difference equation for Eq. (2.18)

$$\begin{aligned} x_0 &= 0 \\ -\frac{x_{i+1} - 2x_i + x_{i-1}}{h^2} + q_i x_i - \lambda\, v_i x_i &= 0\,, \qquad i = 1, \ldots N-1\,, \\ x_N &= 0\,. \end{aligned}$$

If $v_i \neq 0$ for all i we can rewrite the difference equation above as an eigenvalue problem

$$(A - \lambda\, I)\, x = 0 \tag{2.19}$$

for a tridiagonal matrix A

$$A = \begin{pmatrix} \frac{2}{h^2 v_1} + \frac{q_1}{v_1} & \frac{-1}{h^2 v_1} & 0 & \ldots & \ldots & 0 \\ \frac{-1}{h^2 v_2} & \frac{2}{h^2 v_2} + \frac{q_2}{v_2} & \frac{-1}{h^2 v_2} & \ldots & \ldots & 0 \\ 0 & \frac{-1}{h^2 v_3} & \frac{2}{h^2 v_3} + \frac{q_3}{v_3} & \frac{-1}{h^2 v_3} & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \frac{-1}{h^2 v_{N-2}} \\ 0 & \ldots & \ldots & 0 & \frac{-1}{h^2 v_{N-1}} & \frac{2}{h^2 v_{N-1}} + \frac{q_3}{v_{N-1}} \end{pmatrix}$$

and a vector $x = (x_1, x_2, \ldots, x_{N-1})^T$.

Appendix A
Tridiagonal matrix algorithm (TDMA)

The tridiagonal matrix algorithm (TDMA), also known als *Thomas algorithm*, is a simplified form of Gaussian elimination that can be used to solve tridiagonal system of equations

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = y_i, \qquad i = 1, \ldots n, \tag{A.1}$$

or, in matrix form ($a_1 = 0$, $c_n = 0$)

$$\begin{pmatrix} b_1 & c_1 & 0 & \ldots & \ldots & 0 \\ a_2 & b_2 & c_2 & \ldots & \ldots & 0 \\ 0 & a_3 & b_3 & c_3 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & c_{n-1} \\ 0 & \ldots & \ldots & 0 & a_n & b_n \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_n \end{pmatrix}$$

The TDMA is based on the Gaussian elimination procedure and consist of two parts: a forward elimination phase and a backward substitution phase [3]. Let us consider the system (A.1) for $i = 1 \ldots n$ and consider following modification of first two equations:

$$\mathrm{Eq}_{i=2} \cdot b_1 - \mathrm{Eq}_{i=1} \cdot a_2$$

which relults in

$$(b_1 b_2 - c_1 a_2) x_2 + c_2 b_1 x_3 = b_1 y_2 - a_2 y_1 .$$

The effect is that x_1 has been eliminated from the second equation. In the same manner one can eliminate x_2, using the modified second equation and the third one (for $i = 3$):

$$(b_1 b_2 - c_1 a_2)\mathrm{Eq}_{i=3} - a_3(\text{mod. } \mathrm{Eq}_{i=2}),$$

which would give

$$(b_3(b_1 b_2 - c_1 a_2) - c_2 b_1 a_3) x_3 + c_3(b_1 b_2 - c_1 a_2) x_4 = y_3(b_1 b_2 - c_1 a_2) - (y_2 b_1 - y_1 a_2) a_3$$

If the procedure is repeated until the n'th equation, the last equation will involve the unknown function x_n only. This function can be then used to solve the modified equation for $i = n - 1$ and so on, until all unknown x_i are found (backward substitution phase). That is, we are looking for a backward ansatz of the form:

$$x_{i-1} = \gamma_i x_i + \beta_i . \tag{A.2}$$

If we put the last ansatz in Eq. (A.1) and solve the resulting equation with respect to x_i, the following relation can be obtained:

$$x_i = \frac{-c_i}{a_i\gamma_i + b_i} x_{i+1} + \frac{y_i - a_i\beta_i}{a_i\gamma_i + b_i} \tag{A.3}$$

This relation possesses the same form as Eq. (A.2) if we identify

$$\boxed{\gamma_{i+1} = \frac{-c_i}{a_i\gamma_i + b_i}, \qquad \beta_{i+1} = \frac{y_i - a_i\beta_i}{a_i\gamma_i + b_i}}. \tag{A.4}$$

Equation (A.4) involves the recursion formula for the coefficients γ_i and β_i for $i = 2, \ldots, n-1$. The missing values γ_1 and β_1 can be derived from the first ($i = 1$) equation (A.1):

$$x_1 = \frac{y_1}{b_1} - \frac{c_1}{b_1} x_2 \Rightarrow \gamma_2 = -\frac{c_1}{b_1}, \beta_2 = \frac{1}{b_1} \Rightarrow \boxed{\gamma_1 = \beta_1 = 0.}$$

The last what we need is the value of the function x_n for the first backward substitution. We can obtain if we put the ansatz

$$x_{n-1} = \gamma x_n + \beta_n$$

into the last ($i = n$) equation (A.1):

$$a_n(\gamma x_n + \beta_n) + b_n x_n = y_n,$$

yielding

$$x_n = \frac{y_n - a_n\beta_n}{a_n\gamma_n + b_n}.$$

One can get this value directly from Eq. (A.2), if one formal puts

$$x_{n+1} = 0.$$

Altogether, the TDMA can be written as:

1. `Set` $\gamma_1 = \beta_1 = 0$;
2. `Evaluate for` $i = 1, \ldots, n-1$

$$\gamma_{i+1} = \frac{-c_i}{a_i\gamma_i + b_i}, \qquad \beta_{i+1} = \frac{y_i - a_i\beta_i}{a_i\gamma_i + b_i};$$

3. `Set` $x_{n+1} = 0$;
4. `Find for` $i = n+1, \ldots, 2$

$$x_{i-1} = \gamma_i x_i + \beta_i.$$

The algorithm admits $\mathcal{O}(n)$ operations instead of $\mathcal{O}(n^3)$ required by Gaussian elimination.

Limitation

The TDMA is only applicable to matrices that are diagonally dominant, i.e.,

$$|b_i| > |a_i| + |c_i|, \qquad i = 1, \ldots, n.$$

References

1. David Acheson. *From Calculus to Chaos*. Oxford University Press, New York, 1997.
2. John H. Mathews and Kurtis D. Fink. *Numerical Methods Using Matlab*. Prentice Hall, New York, 1999.
3. William H. Press, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, www.nr.com, 1993.
4. Hans Rudolf Schwarz and Norbert Koeckler. *Numerische Mathematik*. Teubner, Wiesbaden, 2006.
5. Josef Stoer and Roland Bulirsch. *Numerische Mathematik 2*. Springer, Berlin, 2000.
6. Steven H. Strogaz. *Nonlinear Dynamics and Chaos*. Perseus Books Publishing, NewYork, 1994.
7. Michael Tabor. *Chaos and Integrability in Nonlinear Dynamics: An Introduction*. A Wiley-Interscience Publication, 1989.