Numerical Analysis and Simulation of Ordinary Differential Equations

Roland Pulch

Lecture in Winter Term 2011/12

University of Wuppertal
Applied Mathematics/Numerical Analysis

Contents:

Literature:

J. Stoer, R. Bulirsch: Introduction to Numerical Analysis. Springer, Berlin 2002. (Chapter 7)

J. Stoer, R. Bulirsch: Numerische Mathematik 2. Springer, Berlin 2005. (Kapitel 7)

A. Quarteroni, R. Sacco, F. Saleri: Numerical Mathematics. Springer, New York 2007. (Chapter 11)

Contents

Chapter 1

ODE Models in Science

This lecture deals with the numerical solution of systems of *ordinary differential equations* (ODEs), i.e.,

$$y'(x) = f(x, y(x)),$$

or written component-wise

$$\begin{aligned} y_1'(x) &= f_1(x, y_1(x), \ldots, y_n(x)) \\ y_2'(x) &= f_2(x, y_1(x), \ldots, y_n(x)) \\ &\vdots \\ y_n'(x) &= f_n(x, y_1(x), \ldots, y_n(x)). \end{aligned}$$

Additional conditions are required to achieve a unique solution. On the one hand, *initial value problems* (IVPs) demand

$$y(x_0) = y_0$$

at a specific initial point x_0 together with a predetermined value $y_0 \in \mathbb{R}^n$. Figure 1 outlines the task. On the other hand, *boundary value problems* (BVPs) impose a condition on an initial state as well as a final state, i.e.,

$$r(y(a), y(b)) = 0$$

with a given function $r : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ and an interval $[a, b]$. For example, periodic boundary conditions read $y(a) - y(b) = 0$.

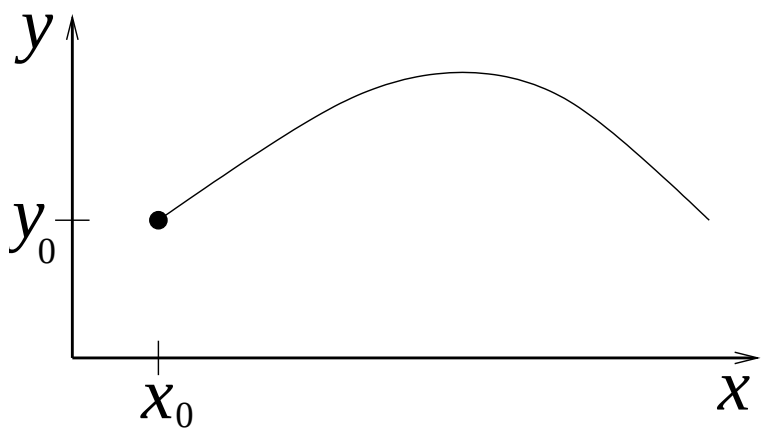

Figure 1: Initial value problem of an ODE.

An ODE of nth order reads

$$z^{(n)}(x) = g(x, z(x), z'(x), z''(x), \ldots, z^{(n-1)}(x)).$$

We obtain an equivalent system of first order by arranging

$$y_1 := z, \quad y_2 := z', \quad y_3 := z'', \quad \ldots \quad , \quad y_n := z^{(n-1)}.$$

It follows the system

$$y_1' = y_2, \quad y_2' = y_3, \quad \ldots \quad , \quad y_{n-1}' = y_n, \quad y_n' = g(x, y_1, \ldots, y_n).$$

Thus we consider without loss of generality systems of first order only in this lecture.

We start with some examples of ODE models resulting in various applications ranging from science (chemical reaction kinetics) to classical mechanics (N-body problem) and electrical engineering (electric circuits). In all cases, mathematical models are used to describe (approximatively) real processes. Due to simplifications and model assumptions, the exact solution of the ODE models represents an approximation of the real process. In most cases, the independent variable x represents the time.

1.1 Chemical reaction kinetics

The radioactive decay represents a process depending on time. For example, the decay of a radon isotope occurs via

$$\text{Rn-222} \quad \longrightarrow \quad \text{Po-218} \; + \; \text{He-4} \;\; (\alpha\text{-particle})$$

with the rate $T_{1/2} = 3.825$ days. Let n be the number of particles of the isotope. The corresponding ODE model reads

$$n'(t) = -kn(t), \quad n(0) = n_0,$$

where an initial value problem is formulated. The involved constant is $k = \ln 2/T_{1/2}$. In this simple example, the solution of the ODE can be determined analytically, i.e.,

$$n(t) = n_0 \mathrm{e}^{-kt}.$$

Although the number of particles is an integer in reality, it is reasonable to apply real numbers in the model. The radioactive decay can be seen as a unimolecular reaction.

Chemical processes typically include bimolecular reactions

$$\mathrm{A} \quad + \quad \mathrm{B} \quad \longrightarrow \quad \mathrm{C} \quad + \quad \mathrm{D}.$$

The special case B = C represents a catalysis. Let c_S be the concentration of the substance S. The corresponding system of ordinary differential equations reads

$$\begin{aligned} c'_\mathrm{A}(t) &= -k\, c_\mathrm{A}(t) c_\mathrm{B}(t) \\ c'_\mathrm{B}(t) &= -k\, c_\mathrm{A}(t) c_\mathrm{B}(t) \\ c'_\mathrm{C}(t) &= +k\, c_\mathrm{A}(t) c_\mathrm{B}(t) \\ c'_\mathrm{D}(t) &= +k\, c_\mathrm{A}(t) c_\mathrm{B}(t). \end{aligned} \tag{1.1}$$

The reaction rate coefficient $k > 0$ characterises the probability of the chemical reaction in case of a collision between the molecules A and B. The coefficient k can also be seen as velocity of the reaction. The physical unit of the parameter k is litre/(s mol). According initial conditions have to be specified for the system (1.1).

Now we consider a set of m general chemical reactions involving n different species $\mathrm{A}_1, \ldots, \mathrm{A}_n$ (molecules/atoms) in total

$$\alpha_{1j}\mathrm{A}_1 + \alpha_{2j}\mathrm{A}_2 + \cdots + \alpha_{nj}\mathrm{A}_n \quad \xrightarrow{k_j} \quad \gamma_{1j}\mathrm{A}_1 + \gamma_{2j}\mathrm{A}_2 + \cdots + \gamma_{nj}\mathrm{A}_n$$

for $j = 1, \ldots, m$ or, equivalently,

$$\sum_{i=1}^{n} \alpha_{ij}\mathrm{A}_i \quad \xrightarrow{k_j} \quad \sum_{i=1}^{n} \gamma_{ij}\mathrm{A}_i \qquad \text{for } \; j = 1, \ldots, m. \tag{1.2}$$

The parameters $\alpha_{ij}, \gamma_{ij} \in \mathbb{N}_0$ represent the stoichiometric constants. The jth reaction exhibits the rate coefficient $k_j \in \mathbb{R}^+$. Consequently, the resulting mathematical model reads

$$\frac{\mathrm{d}c_{\mathrm{A}_i}}{\mathrm{d}t} = \sum_{j=1}^{m} (\gamma_{ij} - \alpha_{ij}) k_j \prod_{l=1}^{n} {c_{\mathrm{A}_l}}^{\alpha_{lj}} \qquad \text{for } \; i = 1, \ldots, n \; ,$$

which represents a system of n ordinary differential equations for the unknown concentrations. The evaluation of the right-hand side can be done automatically, if the corresponding chemical reactions (1.2) are specified.

The hydrolysis of urea represents an example of a more complex chemical reaction. Thereby, urea is combining with water and results to ammonium carbonate. To achieve a sufficiently fast reaction, the help of the enzyme urease is necessary, since it decreases the energy of activation, i.e., the enzyme acts as a catalyser. The complete chemical reaction is given by the formula

$$(\mathrm{NH_2})_2\mathrm{CO} + 2\ \mathrm{H_2O} + \text{urease} \quad \longrightarrow \quad (\mathrm{NH_4})_2\mathrm{CO_3} + \text{urease}. \tag{1.3}$$

This relation represents a complex reaction, since it consists of three simpler reactions. In the following, we use the abbreviations: U: urea, E: urease (enzyme), UE: combination of urea and urease, A: ammonium carbonate. The reaction (1.3) includes the three parts

$$\begin{aligned} \mathrm{U} + \mathrm{E} \quad &\xrightarrow{k_1} \quad \mathrm{UE} \\ \mathrm{UE} \quad &\xrightarrow{k_2} \quad \mathrm{U} + \mathrm{E} \\ \mathrm{UE} + 2\ \mathrm{H_2O} \quad &\xrightarrow{k_3} \quad \mathrm{A} + \mathrm{E}. \end{aligned} \tag{1.4}$$

The parameters k_1, k_2, k_3 specify the velocities of the reactions. The three parts are complex reactions itself, i.e., they proceed as chains of simple reactions, which are not considered here.

We construct a mathematical model for this system of reactions. Let c_S be the concentration of the substance S with unit mol/litre (mol/l). The transient behaviour of the concentrations shall be determined. Since the reaction takes place in water and the concentrations of the other substances is relatively low, we assume the concentration of water to be constant in time (55.56 mol/l). The velocities of the reactions are

$$k_1 = 3.01 \, \frac{\text{l}}{\text{mol}\cdot\text{s}}, \quad k_2 = 0.02 \, \tfrac{1}{\text{s}}, \quad k_3 = 0.1 \, \tfrac{1}{\text{s}}. \tag{1.5}$$

Consequently, we obtain a system of four ODEs for the four unknown concentrations

$$\begin{array}{rcrcrcl} c'_{\text{U}} &=& -\, k_1 c_{\text{U}} c_{\text{E}} &+& k_2 c_{\text{UE}} & & \\ c'_{\text{E}} &=& -\, k_1 c_{\text{U}} c_{\text{E}} &+& k_2 c_{\text{UE}} &+& k_3 c_{\text{UE}} \\ c'_{\text{UE}} &=& k_1 c_{\text{U}} c_{\text{E}} &-& k_2 c_{\text{UE}} &-& k_3 c_{\text{UE}} \\ c'_{\text{A}} &=& & & & & k_3 c_{\text{UE}}. \end{array} \tag{1.6}$$

This system exhibits a unique solution for predetermined initial values. We apply the initial conditions

$$c_{\text{U}} = 0.1 \, \frac{\text{mol}}{\text{l}}, \quad c_{\text{E}} = 0.02 \, \frac{\text{mol}}{\text{l}}, \quad c_{\text{UE}} = c_{\text{A}} = 0. \tag{1.7}$$

Like in many other applications, an analytical solution of this system of ODEs is not feasible, i.e., we do not achieve an explicit formula for the unknown solution. Thus we employ a numerical simulation to determine a solution approximately. Figure 2 illustrates the results.

On the one hand, the concentration of urea decays to zero, since this substance is decomposed in the hydrolysis. On the other hand, the product ammonium carbonate is generated until no more urea is present. The concentration of the enzyme urease decreases at the beginning. According to an enzyme, the initial amount of urease is recovered at the end.

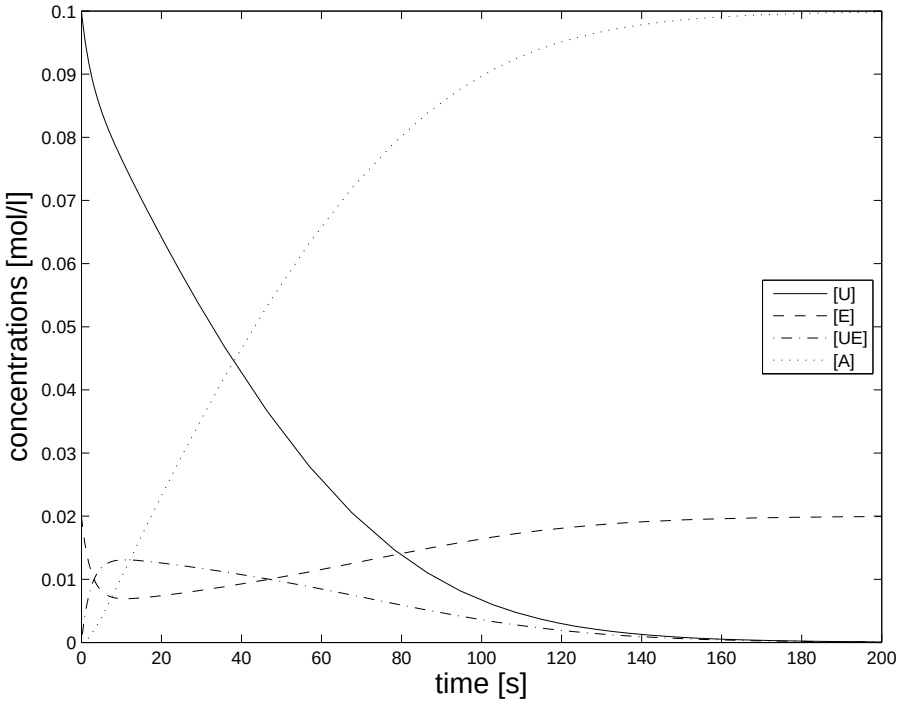

Figure 2: Simulation of the hydrolysis of urea.

1.2 Electric circuits

As a simple example of an electric circuit, we consider an electromagnetic oscillator, which consists of a capacitance C and inductance L and a resistance R, see Figure 3 (left). Kirchhoff's current law yiels the relation

$$I_C + I_L + I_R = 0.$$

Kirchhoff's voltage law implies $U := U_C = U_L = U_R$. Each basic element of the circuit exhibits a voltage-current relation

$$CU_C' = I_C, \quad LI_L' = U_L, \quad R = \frac{U_R}{I_R}.$$

It follows a linear system of two ODEs

$$\begin{array}{rcl} U' & = & -\frac{1}{C} I_L - \frac{1}{RC} U \\ I_L' & = & \frac{1}{L} U \end{array} \tag{1.8}$$

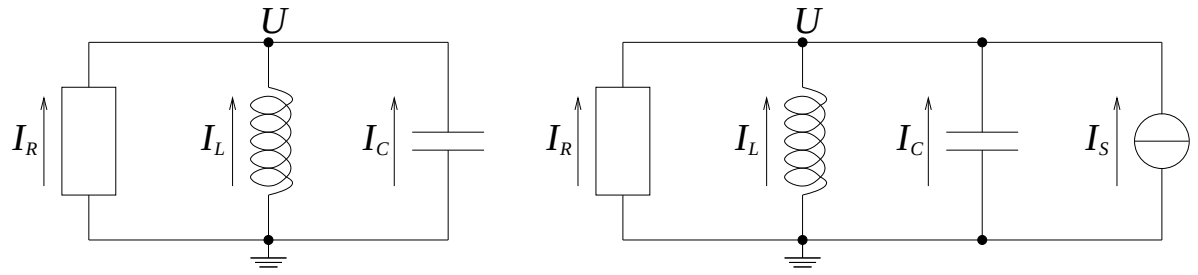

Figure 3: Electromagnetic oscillator with (right) and without (left) current source.

for the two unknown functions U and I_L. Further calculations yield an ODE of second order for the unknown voltage

$$U'' + \tfrac{1}{RC}U' + \tfrac{1}{LC}U = 0.$$

If the resistance is sufficiently large, the solution becomes a damped oscillation

$$U(t) = \mathrm{e}^{-\frac{1}{2RC}t}\left[A\sin\left(\tfrac{1}{\sqrt{LC}}t\right) + B\cos\left(\tfrac{1}{\sqrt{LC}}t\right)\right].$$

The constants A and B are determined by initial conditions.

The system (1.8) of ODEs is autonomous. We obtain a time-dependent system by adding an independent current source to the circuit, see Figure 3 (right). We apply the input

$$I_{\mathrm{in}}(t) = I_0 \sin(\omega_0 t).$$

The corresponding ODE model becomes

$$\begin{aligned} U' &= -\tfrac{1}{C}I_L - \tfrac{1}{RC}U - \tfrac{1}{C}I_{\mathrm{in}}(t) \\ I_L' &= \tfrac{1}{L}U. \end{aligned} \tag{1.9}$$

The system (1.9) exhibits periodic solutions with the rate $T = 2\pi/\omega_0$. Hence we can impose boundary conditions $U(0) = U(T)$ and $I_L(0) = I_L(T)$. Resonance occurs in the case $\omega_0 = 1/\sqrt{LC}$. Figure 4 shows the solutions of initial value problems corresponding to (1.8) and (1.9), respectively.

To demonstrate the model of a more complex electric circuit, we consider the Colpitts oscillator depicted in Figure 5. Mathematical modelling yields

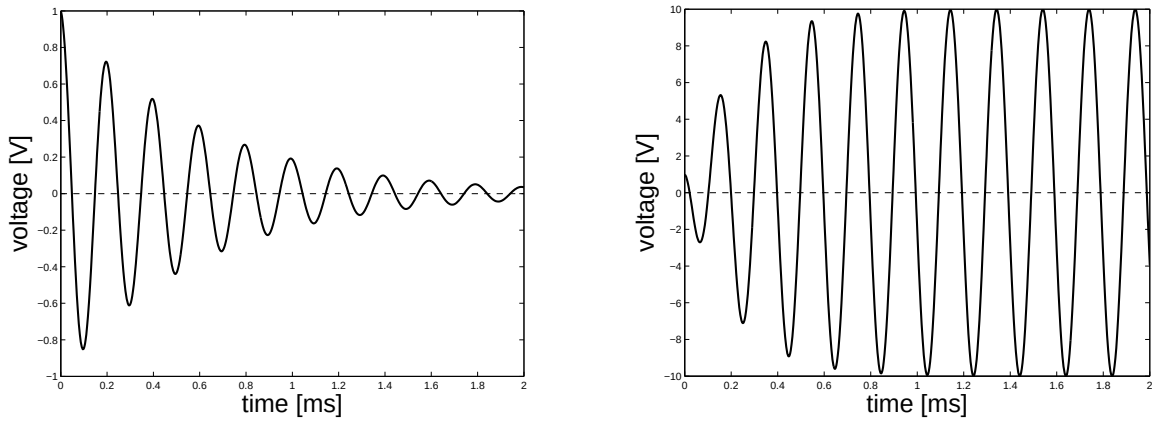

Figure 4: Solution U of ODE (1.8) (left) and ODE (1.9) (right).

an implicit system of four ODEs including four unknown node voltages:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & C_1+C_3 & -C_3 & -C_1 \\ 0 & -C_3 & C_2+C_3+C_4 & -C_2 \\ 0 & -C_1 & -C_2 & C_1+C_2 \end{pmatrix} \begin{pmatrix} U_1' \\ U_2' \\ U_3' \\ U_4' \end{pmatrix}$$
$$+ \begin{pmatrix} \frac{R_2}{L}(U_1-U_2) - R_2 U_{\mathrm{op}}' \\ \frac{1}{R_2}(U_1-U_{\mathrm{op}}) - \left(I_s + \frac{I_s}{b_c}\right) g(U_4-U_2) + I_s g(U_4-U_3) \\ \frac{1}{R_4}U_3 - \left(I_s + \frac{I_s}{b_e}\right) g(U_4-U_3) + I_s g(U_4-U_2) \\ \frac{1}{R_3}U_4 + \frac{1}{R_1}(U_4-U_{op}) + \frac{I_s}{b_e} g(U_4-U_3) + \frac{I_s}{b_c} g(U_4-U_2) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Several technical parameters appear in the system. The current-voltage relation corresponding to the bipolar transistor reads

$$g(U) := \exp\left(\frac{U}{U_{\mathrm{th}}}\right) - 1.$$

Thus the system is nonlinear.

1.3 Multibody problem

We consider the two-body problem for two particles with masses m_1, m_2. Let $\vec{X}_i = (x_i, y_i, z_i)$ be the location of the ith mass. The locations and the

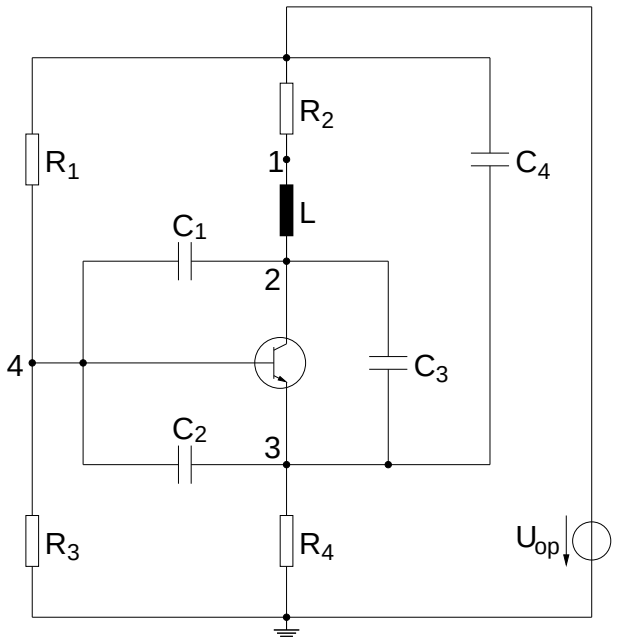

Figure 5: Electric circuit of the Colpitts oscillator.

velocities of the particles depend on time. The gravitation generates forces between the masses. Newton's laws of motion yield the ODEs of second order

$$\begin{aligned} m_1\vec{X}_1''(t) &= G\tfrac{m_1m_2}{|\vec{X}_1(t)-\vec{X}_2(t)|^3}(\vec{X}_2(t)-\vec{X}_1(t)) \\ m_2\vec{X}_2''(t) &= G\tfrac{m_1m_2}{|\vec{X}_1(t)-\vec{X}_2(t)|^3}(\vec{X}_1(t)-\vec{X}_2(t)) \end{aligned}$$

with the gravitational constant $G > 0$. Introducing the velocities $\vec{V}_i := \vec{X}_i'$ implies a system of first order

$$\begin{aligned} \vec{X}_1' &= \vec{V}_1 \\ \vec{V}_1' &= G\tfrac{m_2}{|\vec{X}_1-\vec{X}_2|^3}(\vec{X}_2-\vec{X}_1) \\ \vec{X}_2' &= \vec{V}_2 \\ \vec{V}_2' &= G\tfrac{m_1}{|\vec{X}_1-\vec{X}_2|^3}(\vec{X}_1-\vec{X}_2) \end{aligned}$$

including twelve ODEs. The system is autonomous. Initial conditions for $\vec{X}_i(0), \vec{V}_i(0)$ have to be specified. Figure 6 depicts the trajectories of a two-body problem with different masses $m_1 > m_2$. The movement typically proceeds approximatively along ellipses.

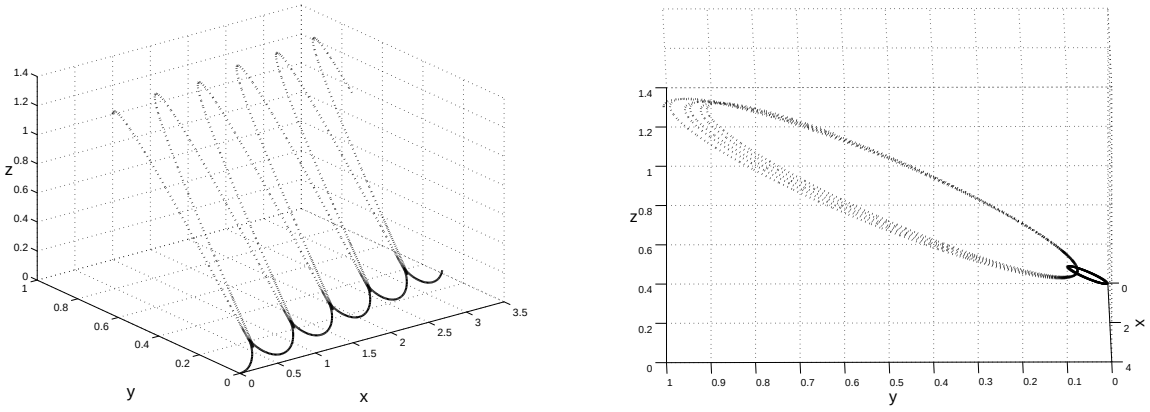

Figure 6: Trajectories (locations) of a two-body problem with masses $m_1 > m_2$ from two different viewpoints (solid line: first body, points: second body).

Moreover, the two-body problem can be solved analytically. In constrast, we arrange the N-body problem now, where N masses $m_1, \ldots, m_N$ are involved. Let $\vec{F}_{ij}$ be the gravitational force on the ith mass caused by the jth mass. Newton's laws of motion imply

$$m_i \vec{X}_i'' = \sum_{j=1, j\neq i}^{N} \vec{F}_{ij} = \sum_{j=1, j\neq i}^{N} G \frac{m_i m_j}{|\vec{X}_j - \vec{X}_i|^3} (\vec{X}_j - \vec{X}_i)$$

for $i = 1, \ldots, N$. It follows a system of $6N$ ODEs of first order

$$\begin{aligned} \vec{X}_i' &= \vec{V}_i \\ \vec{V}_i' &= G \sum_{j=1, j\neq i}^{N} \frac{m_j}{|\vec{X}_j - \vec{X}_i|^3} (\vec{X}_j - \vec{X}_i) \qquad \text{for } i = 1, \ldots, N. \end{aligned}$$

The N-body problem cannot be solved analytically. Thus we require numerical methods to solve the problem.

1.4 Further models

In the previous sections, we have considered problems in the fields of chemical reactions, electrical engineering and mechanics. Systems of ODEs also appear in the following applications:

- biology (predator-prey models, epidemic models, etc.),
- simulation of war battles (Lanchester's combat models),
- semi-discretisation of partial differential equations,
- and others.

In financial mathematics, for example, modelling yields stochastic (ordinary) differential equations. Numerical methods for the stochastic differential equations represent improvements of the techniques for ODEs. Hence knowledge on ODE methods is necessary to deal with stochastic systems.

Further reading on ODE models:

P. Deuflhard, F. Bornemann: Scientific Computing with Ordinary Differential Equations. Springer, New York 2002.

M. Braun: Differential Equations and Their Applications. (4th edition) Springer, Berlin 1993.

Chapter 2

Short Synopsis on Theory of ODEs

In this chapter, we review some basic results on existence and uniqueness corresponding to solutions of ODEs. Further interesting properties are also considered.

2.1 Linear differential equations

An initial value problem of a linear (inhomogeneous) ODE reads

$$y'(x) = a(x)y(x) + b(x), \qquad y(x_0) = y_0.$$

The corresponding solution exhibits the formula

$$y(x) = \exp\left(\int_{x_0}^{x} a(s)\,\mathrm{d}s\right) \cdot \left(y_0 + \int_{x_0}^{x} \exp\left(-\int_{x_0}^{s} a(t)\,\mathrm{d}t\right) b(s)\,\mathrm{d}s\right),$$

which can be verified straightforward. A more explicit formula of the solution is only obtained if the involved integrals can be solved analytically.

In case of linear (inhomogeneous) systems of ODEs, the initial value problem becomes

$$y'(x) = A(x)y(x) + b(x), \qquad y(x_0) = y_0$$

with predetermined functions $A : \mathbb{R} \to \mathbb{R}^{n\times n}$ and $b : \mathbb{R} \to \mathbb{R}^n$. Numerical methods are required to solve the system. We obtain a formula of the

solution in case of constant coefficients $A \in \mathbb{R}^{n\times n}$, i.e.,

$$y(x) = \exp(A(x - x_0)) \cdot \left(y_0 + \int_{x_0}^{x} \exp(-A(s - x_0))b(s)\ \mathrm{d}s \right).$$

The involved integral over a vector is evaluated component-wise. The matrix exponential is defined by

$$\exp(At) := \sum_{k=0}^{\infty} \frac{t^k}{k!} A^k,$$

where the sum converges for each $t \in \mathbb{R}$ with respect to an arbitrary matrix norm. In general, the matrix exponential cannot be evaluated analytically. Thus a numerical scheme is necessary. Further investigations show that numerical techniques avoiding the matrix exponential have to be preferred for solving the linear system of ODEs.

In conclusion, an analytical solution of linear ODEs is not always feasible. Hence numerical methods yield the corresponding solutions. Of course, this holds even more in case of nonlinear ODEs.

2.2 Existence and uniqueness

We consider initial value problems of systems of ODEs

$$y'(x) = f(x, y(x)), \qquad y(x_0) = y_0 \tag{2.1}$$

for functions $f : G \to \mathbb{R}^n$ with $G \subseteq \mathbb{R} \times \mathbb{R}^n$ and $(x_0, y_0) \in G$. A function y represents a solution of this problem if and only if

$$y(x) = y_0 + \int_{x_0}^{x} f(s, y(s))\ \mathrm{d}s \tag{2.2}$$

holds for all relevant x.

The theorem of Peano just requires a continuous right-hand side f. However, this theorem yields only the existence and not the uniqueness of a solution. To apply numerical methods, we need both properties.

In the following, we assume the Lipschitz-condition

$$\|f(x,y) - f(x,z)\| \leq L \cdot \|y - z\| \tag{2.3}$$

for all x, y, z located in G with a constant $L > 0$. The involved vector norm is arbitrary. Concerning the uniqueness of a solution to an initial value problem (2.1), it holds the following result.

Theorem 1 *Let $G \subseteq \mathbb{R} \times \mathbb{R}^n$ be an open set and let $f : G \to \mathbb{R}^n$ be a continuous function satisfying the Lipschitz-condition (2.3). Consider two solutions $\varphi, \psi : I \to \mathbb{R}^n$ of the ODE system $y' = f(x,y)$ on an interval $I \subseteq \mathbb{R}$. If $\varphi(x_0) = \psi(x_0)$ holds for some $x_0 \in I$, then it follows $\varphi(x) = \psi(x)$ for all $x \in I$.*

Outline of the proof:

Let $\varphi, \psi : I \to \mathbb{R}^n$ be two solutions of $y' = f(x,y)$. We show that the condition $\varphi(\hat{x}) = \psi(\hat{x})$ for an arbitrary $\hat{x} \in I$ implies $\varphi \equiv \psi$ in a neighbourhood of $\hat{x}$. It holds

$$\varphi(x) = \varphi(\hat{x}) + \int_{\hat{x}}^{x} f(s, \varphi(s))\, \mathrm{d}s, \qquad \psi(x) = \psi(\hat{x}) + \int_{\hat{x}}^{x} f(s, \psi(s))\, \mathrm{d}s.$$

The Lipschitz condition yields

$$\begin{aligned} \|\varphi(x) - \psi(x)\| &\leq \left| \int_{\hat{x}}^{x} \|f(s, \varphi(s)) - f(s, \psi(s))\|\, \mathrm{d}s \right| \\ &\leq L \left| \int_{\hat{x}}^{x} \|\varphi(s) - \psi(s)\|\, \mathrm{d}s \right|. \end{aligned}$$

We define

$$M(x) := \sup\{\|\varphi(s) - \psi(s)\| : |s - \hat{x}| \leq |x - \hat{x}|\}.$$

It follows

$$\|\varphi(t) - \psi(t)\| \leq L|t - \hat{x}|M(t) \leq L|x - \hat{x}|M(x)$$

for $|t - \hat{x}| \leq |x - \hat{x}|$ and thus

$$M(x) \leq L|x - \hat{x}|M(x).$$

For $|x - \hat{x}| < 1/(2L)$, we obtain $M(x) \leq \frac{1}{2}M(x)$ and thus $M(x) = 0$ for those x.

Now we consider the assumption $\varphi(x_0) = \psi(x_0)$. Let

$$x_1 := \sup\left\{s \in I : \varphi|_{[x_0,s]} = \psi|_{[x_0,s]}\right\}.$$

Since both functions are continuous, it follows $\varphi(x_1) = \psi(x_1)$. If x_1 is not equal to the right boundary of the interval, then a contradiction appears with respect to the previous result, which states that the functions are equal in a complete neighbourhood of x_1. The same argumentation can be applied to the left boundary $x \leq x_0$. □

The theorem of Picard-Lindelöf yields a result on the existence.

Theorem 2 (Picard-Lindelöf) *Let $G \subseteq \mathbb{R}\times\mathbb{R}^n$ be an open set and let $f : G \to \mathbb{R}^n$ be a continuous function satisfying the Lipschitz-condition (2.3). Then for each $(x_0, y_0) \in G$ it exists a real number $\varepsilon > 0$ and a solution $\varphi : [x_0 - \varepsilon, x_0 + \varepsilon] \to \mathbb{R}^n$ of the initial value problem (2.1).*

Outline of the proof:

It exists $r > 0$ such that the set

$$V := \{(x, y) \in \mathbb{R} \times \mathbb{R}^n : |x - x_0| \leq r, \ \ \|y - y_0\| \leq r\}$$

satisfies $V \subset G$. Since f is continuous and V is compact, it exists $M > 0$ such that

$$\|f(x, y)\| \leq M \qquad \text{for all } \ (x, y) \in V.$$

We define $\varepsilon := \min\{r, r/M\}$ and $I := [x_0 - \varepsilon, x_0 + \varepsilon]$.

A function φ satisfies the initial value problem if and only if

$$\varphi(x) = y_0 + \int_{x_0}^{x} f(s, \varphi(s)) \, \mathrm{d}s$$

holds for all $x \in I$. We define functions $\varphi_k : I \to \mathbb{R}^n$ via the iteration

$$\varphi_{k+1}(x) := y_0 + \int_{x_0}^{x} f(s, \varphi_k(s)) \, \mathrm{d}s$$

using the starting function $\varphi_0(x) \equiv y_0$. For $x \in I$, it follows

$$\|\varphi_{k+1}(x) - y_0\| \leq \left| \int_{x_0}^{x} \|f(s, \varphi_k(s))\| \, \mathrm{d}s \right| \leq M|x - x_0| \leq M\varepsilon \leq r$$

provided that φ_k lies in V. By induction, the functions φ_k are well-defined.

Furthermore, it follows by induction

$$\|\varphi_k(x) - \varphi_{k-1}(x)\| \leq ML^{k-1} \frac{|x - x_0|^k}{k!} \qquad \text{for each } \; x \in I.$$

Hence it holds

$$\|\varphi_k(x) - \varphi_{k-1}(x)\| \leq \frac{M}{L} \cdot \frac{(L\varepsilon)^k}{k!}$$

uniformly in I. The right-hand side exhibits terms of exponential series for $\mathrm{e}^{L\varepsilon}$. It follows that $(\varphi_k)_{k\in\mathbb{N}}$ is a Cauchy-sequence uniformly for $x \in I$. Consequently, the sequence $(\varphi_k)_{k\in\mathbb{N}}$ converges uniformly to a continuous function φ. Moreover, we obtain

$$\|f(x, \varphi(x)) - f(x, \varphi_k(x))\| \leq L \cdot \|\varphi(x) - \varphi_k(x)\|.$$

Thus the sequence $(f(x, \varphi_k(x))_{k\in\mathbb{N}}$ converges uniformly to $f(x, \varphi(x))$. It follows

$$\begin{aligned} \varphi(x) &= \lim_{k\to\infty} \varphi_k(x) = y_0 + \lim_{k\to\infty} \int_{x_0}^{x} f(s, \varphi_k(s)) \, \mathrm{d}s \\ &= y_0 + \int_{x_0}^{x} \lim_{k\to\infty} f(s, \varphi_k(s)) \, \mathrm{d}s = y_0 + \int_{x_0}^{x} f(s, \varphi(s)) \, \mathrm{d}s \end{aligned}$$

and the proof is completed. □

The theorem of Picard-Lindelöf also includes a method for the construction of the solution by the iteration. We analyse this iteration further. We define

$$F(\varphi) := y_0 + \int_{x_0}^{x} f(s, \varphi(s)) \, \mathrm{d}s.$$

The fixed point $\varphi = F(\varphi)$ represents a solution of the initial value problem (2.1). The corresponding iteration reads $\varphi_{k+1} = F(\varphi_k)$. We obtain for

$x_0 \le x \le x_1$:

$$\begin{aligned}
\|F(\varphi)(x) - F(\psi)(x)\| &\le \int_{x_0}^{x} \|f(s,\varphi(s)) - f(s,\psi(s))\| \,\mathrm{d}s \\
&\le L \int_{x_0}^{x} \|\varphi(s) - \psi(s)\| \,\mathrm{d}s \\
&\le L \int_{x_0}^{x_1} \|\varphi(s) - \psi(s)\| \,\mathrm{d}s \\
&\le L(x_1 - x_0) \max_{s\in[x_0,x_1]} \|\varphi(s) - \psi(s)\|.
\end{aligned}$$

It follows

$$\max_{s\in[x_0,x_1]} \|F(\varphi)(s) - F(\psi)(s)\| \le L(x_1 - x_0) \max_{s\in[x_0,x_1]} \|\varphi(s) - \psi(s)\|.$$

Hence the mapping F is contractive with respect to the maximum norm if $x_1 - x_0 < \frac{1}{L}$ holds. The theorem of Banach implies the convergence of the Picard-Lindelöf iteration and the existence of a unique fixed point.

However, the iteration requires a subsequent solution of integrals, which makes it disadvantageous in practice. Furthermore, we may be forced to use small subintervals.

Finally, we cite a theorem concerning the maximum interval of the existence of a solution to the initial value problem (2.1).

Theorem 3 *Let the assumptions of Theorem 2 be fulfilled. Then it exists a maximum interval (α, β) with $\alpha < x_0 < \beta$ such that a unique solution $\varphi : (\alpha, \beta) \to \mathbb{R}^n$ of the initial value problem (2.1) exists. It holds either $\beta = \infty$ or $\beta < \infty$ together with*

$$\overline{\{(x, \varphi(x)) : x \in [x_0, \beta)\}} \cap \{(x, y) \in G : x = \beta\} = \emptyset.$$

Analogue conditions follow for α.

If a function $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ is continuous everywhere, then $\beta < \infty$ implies that the solution of the initial value problem becomes unbounded near $x = \beta$.

2.3 Perturbation analysis

We analyse the condition of initial value problems of ODEs, i.e., the sensitivity of the solutions in dependence on the data. The data are the initial values y_0 and the right-hand side f. (Differences in the value x_0 can be described by different right-hand sides.)

We consider the solution $y(x)$ of the initial value problem (2.1) and the solution $z(x)$ of the perturbed initial value problem

$$z'(x) = f(x, z(x)) + \delta(x), \qquad z(x_0) = z_0.$$

Let the function $\delta(x)$ be continuous. We estimate the resulting difference $y(x) - z(x)$ between the unperturbed and the perturbed solution in terms of the perturbations

$$\rho := \|y_0 - z_0\|, \qquad \varepsilon := \max_{t \in [x_0, x_1]} \|\delta(t)\|$$

using some vector norm.

The estimate is based on the following version of Gronwall's lemma.

Lemma 1 *Assume that $m(x)$ is a non-negative, continuous function and that $\rho, \varepsilon \geq 0$, $L > 0$. Then the integral inequality*

$$m(x) \leq \rho + \varepsilon(x - x_0) + L \int_{x_0}^{x} m(s)\, \mathrm{d}s \tag{2.4}$$

implies the estimate

$$m(x) \leq \rho \mathrm{e}^{L(x-x_0)} + \frac{\varepsilon}{L} \left(\mathrm{e}^{L(x-x_0)} - 1 \right). \tag{2.5}$$

Proof:

At first we define the function

$$q(x) := \mathrm{e}^{-Lx} \int_{x_0}^{x} m(t)\, \mathrm{d}t,$$

which is differentiable as $m(x)$ is continuous and has the derivative

$$q'(x) = -L\mathrm{e}^{-Lx} \int_{x_0}^{x} m(t)\ \mathrm{d}t + \mathrm{e}^{-Lx} m(x).$$

Solving for $m(x)$, we get the relations

$$\begin{aligned} m(x) &= \mathrm{e}^{Lx} q'(x) + L \int_{x_0}^{x} m(t)\ \mathrm{d}t, && (2.6) \\ &= \mathrm{e}^{Lx} q'(x) + L\mathrm{e}^{Lx} q(x) = (\mathrm{e}^{Lx} q(x))'. && (2.7) \end{aligned}$$

We now insert (2.6) in (2.4) and obtain

$$\mathrm{e}^{Lx} q'(x) \leq \rho + \varepsilon(x - x_0) \tag{2.8}$$

and solving for $q'(x)$ we get

$$q'(x) \leq (\rho - \varepsilon x_0)\mathrm{e}^{-Lx} + \varepsilon x \mathrm{e}^{-Lx}.$$

Hence, performing integration, the inequality

$$q(x) \leq -\frac{\rho - \varepsilon x_0}{L} \left(\mathrm{e}^{-Lx} - \mathrm{e}^{-Lx_0}\right) + \varepsilon \int_{x_0}^{x} t\mathrm{e}^{-Lt}\ \mathrm{d}t \tag{2.9}$$

holds, where the integral can be calculated via integration by parts

$$\begin{aligned} \int_{x_0}^{x} t\mathrm{e}^{-Lt}\ \mathrm{d}t &= -\frac{1}{L} t\mathrm{e}^{-Lt}\Big|_{x_0}^{x} + \frac{1}{L} \int_{x_0}^{x} \mathrm{e}^{-Lt}\ \mathrm{d}t \\ &= -\frac{1}{L}\left(x\mathrm{e}^{-Lx} - x_0\mathrm{e}^{-Lx_0}\right) - \frac{1}{L^2}\left(\mathrm{e}^{-Lx} - \mathrm{e}^{-Lx_0}\right). \end{aligned}$$

Finally, inserting (2.8),(2.9) into (2.7) we end up with

$$\begin{aligned} m(x) &\leq -(\rho - \varepsilon x_0)\left(1 - \mathrm{e}^{L(x-x_0)}\right) - \varepsilon x + \varepsilon x_0 \mathrm{e}^{L(x-x_0)} \\ &\quad - \frac{\varepsilon}{L}\left(1 - \mathrm{e}^{L(x-x_0)}\right) + (\rho + \varepsilon(x - x_0)) \\ &= \rho\mathrm{e}^{L(x-x_0)} + \frac{\varepsilon}{L}\left(\mathrm{e}^{L(x-x_0)} - 1\right), \end{aligned}$$

which is the statement (2.5). □

Using $m(x) := \|y(x) - z(x)\|$, the assumptions of Gronwall's lemma are fulfilled, because it holds

$$y(x) - z(x) = y_0 - z_0 - \int_{x_0}^{x} \delta(s)\,\mathrm{d}s + \int_{x_0}^{x} f(s, y(s)) - f(s, z(s))\,\mathrm{d}s$$

and thus

$$\begin{aligned}
\|y(x) - z(x)\| &\leq \|y_0 - z_0\| + \int_{x_0}^{x} \|\delta(s)\|\,\mathrm{d}s \\
&\quad + \int_{x_0}^{x} \|f(s, y(s)) - f(s, z(s))\|\,\mathrm{d}s \\
&\leq \|y_0 - z_0\| + \left(\max_{t\in[x_0,x_1]} \|\delta(t)\|\right)(x - x_0) \\
&\quad + L\int_{x_0}^{x} \|y(s) - z(s)\|\,\mathrm{d}s
\end{aligned}$$

for $x_0 \leq x \leq x_1$. Thus Gronwall's lemma yields

$$\|y(x) - z(x)\| \leq \rho \mathrm{e}^{L(x-x_0)} + \frac{\varepsilon}{L}\left(\mathrm{e}^{L(x-x_0)} - 1\right)$$

for $x_0 \leq x \leq x_1$. We recognise that the problem is well-posed, since it depends continuously on the data. Nevertheless, the difference can increase exponentially for increasing x. This is not always the case but may happen.

If the perturbation appears only in the initial values ($\delta \equiv 0 \Rightarrow \varepsilon = 0$), then the corresponding estimate reads

$$\|y(x) - z(x)\| \leq \|y(x_0) - z(x_0)\| \cdot \mathrm{e}^{L(x-x_0)} \qquad \text{for each } \; x \geq x_0. \tag{2.10}$$

This estimate implies again that the solution $y(x)$ depends continuously on its initial value $y(x_0) = y_0$ for fixed x. Moreover, the dependence becomes smooth for a smooth right-hand side f. We denote the dependence of the solution on the inital values via $y(x; y_0)$.

Theorem 4 *Suppose that f is continuous with respect to x and that the partial derivatives of f with respect to y exist and are continuous. Then the solution $y(x; y_0)$ is smooth with respect to y_0. The derivatives*

$$\Psi(x) := \frac{\partial y}{\partial y_0}(x; y_0) \in \mathbb{R}^{n\times n}$$

are the solution of the initial value problem of the matrix differential equation

$$\Psi'(x) = \frac{\partial f}{\partial y}(x, y(x; y_0)) \cdot \Psi(x), \qquad \Psi(x_0) = I \tag{2.11}$$

with the identity $I \in \mathbb{R}^{n \times n}$.

The proof can be found in: Hairer, Nørsett, Wanner: Solving Ordinary Differential Equations. Volume 1. Springer.

We just show the second statement of the theorem. Differentiating the original system of ODEs

$$\frac{\partial}{\partial x} y(x; y_0) = f(x, y(x; y_0))$$

with respect to the initial values yields

$$\begin{aligned} \frac{\partial}{\partial y_0} \frac{\partial}{\partial x} y(x; y_0) &= \frac{\partial}{\partial y_0} f(x, y(x; y_0)) \\ \frac{\partial}{\partial x} \frac{\partial y}{\partial y_0}(x; y_0) &= \frac{\partial f}{\partial y}(x, y(x; y_0)) \cdot \frac{\partial y}{\partial y_0}(x; y_0) \\ \frac{\partial}{\partial x} \Psi(x) &= \frac{\partial f}{\partial y}(x, y(x; y_0)) \cdot \Psi(x). \end{aligned}$$

The initial value $y(x_0; y_0) = y_0$ implies the initial condition $\frac{\partial y}{\partial y_0}(x_0; y_0) = I$.

The matrix differential equation consists of n separate systems of ODEs (with dimension n each). Moreover, the matrix differential equation exhibits a linear structure. The matrix differential equation can be solved numerically in combination with the original system of ODEs (2.1). Alternatively, numerical differentiation is feasible.

Chapter 3

One-Step Methods

We consider numerical methods for the initial value problems introduced in the previous chapter. We start with one-step methods, whereas multi-step methods are discussed in a later chapter.

3.1 Preliminaries

We want to solve an initial value problem (2.1) of a system of ODEs numerically in some interval $x \in [x_0, x_{\text{end}}]$. All numerical methods for initial value problems, which we consider in this lecture, apply a finite set of grid points

$$x_0 < x_1 < x_2 < x_3 < \cdots < x_{N-1} < x_N = x_{\text{end}}.$$

A feasible choice are equidistant grid points

$$x_i := x_0 + ih \qquad \text{with } h := \frac{x_{\text{end}} - x_0}{N} \qquad \text{for } i = 0, 1, \ldots, N.$$

Numerical solutions $y_i \approx y(x_i)$ are computed successively. In a one-step method, the dependence of the values is just

$$y_0 \longrightarrow y_1 \longrightarrow y_2 \longrightarrow \cdots \longrightarrow y_{N-1} \longrightarrow y_N.$$

In contrast, a multi-step method with k steps exhibits the dependence

$$y_{i-k}, y_{i-k+1}, \ldots, y_{i-2}, y_{i-1} \quad \longrightarrow \quad y_i \qquad \text{for } i = k, k+1, \ldots, N.$$

Thereby, the first values $y_1, \ldots, y_{k-1}$ have to be provided by another scheme in case of $k > 1$. Remark that a one-step method represents a special case of a multi-step method with $k = 1$.

A general one-step method can be written in the form

$$y_{i+1} = y_i + h_i \Phi(x_i, y_i, h_i), \tag{3.1}$$

where the function Φ depends on the scheme as well as the right-hand side function f.

3.2 Elementary integration schemes

Most of the methods for the initial value problem (2.1) are based on an approximation of the corresponding integral equation (2.2). In the interval $[x_0, x_0 + h]$, we obtain

$$\begin{aligned} y(x_0 + h) &= y_0 + \int_{x_0}^{x_0+h} f(s, y(s)) \, \mathrm{d}s \\ &= y_0 + h \int_0^1 f(x_0 + sh, y(x_0 + sh)) \, \mathrm{d}s. \end{aligned} \tag{3.2}$$

Now the integral on the right-hand side is replaced by a quadrature rule. The problem is that the function y, which appears in the integrand, is unknown a priori.

Since h is small, we consider simple quadrature rules. We discuss the following four examples, see Figure 7:

(a) rectangle (left-hand):

The approximation becomes

$$y_1 = y_0 + h f(x_0, y_0).$$

This scheme is called the (explicit) Euler method. It is the most simple method, which is feasible. Given the initial value $y(x_0) = y_0$, the approximation y_1 is computed directly by just a function evaluation of f.

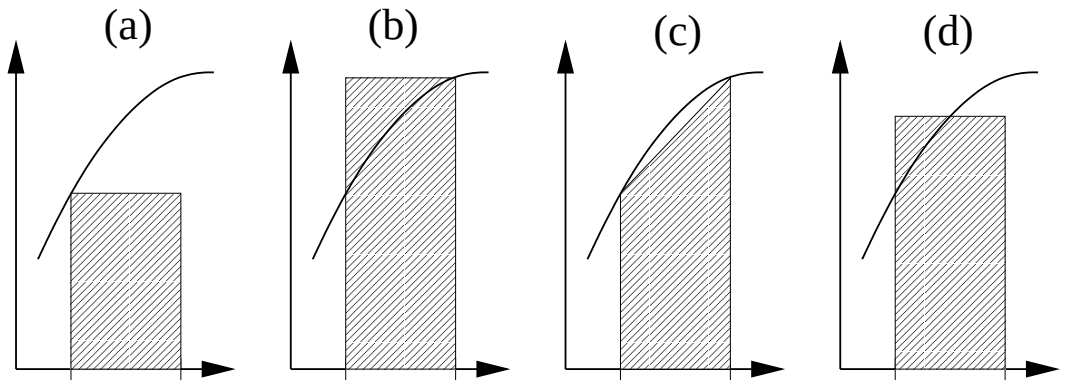

Figure 7: Elementary quadrature rules: (a) rectangle (left-hand), (b) rectangle (right-hand), (c) trapezoidal rule, (d) midpoint rule.

(b) rectangle (right-hand):

Now the scheme reads

$$y_1 = y_0 + hf(x_0 + h, y_1). \tag{3.3}$$

This technique is called the implicit Euler method. The unknown value y_1 appears on both sides of the relation. In general, we cannot achieve an explicit formula for y_1. The formula (3.3) represents a nonlinear system of algebraic equations for the unknown y_1, i.e., the value y_1 is defined implicitly. For example, a Newton iteration yields an approximative solution. Hence the computational effort of one integration step becomes much larger than in the explicit Euler method.

(c) trapezoidal rule:

If the integral is approximated by a trapezoid, the technique becomes

$$y_1 = y_0 + \frac{h}{2}\left(f(x_0, y_0) + f(x_0 + h, y_1)\right).$$

This approach results again in an implicit method. The computational effort of one integration step is nearly the same as in the implicit Euler method. However, the accuracy of the approximations is better in general, since trapezoids yield better approximations than rectangles in the quadrature.

(d) midpoint rule:

The midpoint rule applies a specific rectangle. It follows

$$y_1 = y_0 + hf(x_0 + \tfrac{1}{2}h, y(x_0 + \tfrac{1}{2}h)). \tag{3.4}$$

This scheme is not feasible yet, since both y_1 and $y(x_0 + \frac{1}{2}h)$ are unknown. We require an additional equation. For example, an approximation of the intermediate value $y(x_0+\frac{1}{2}h)$ can be computed by the explicit Euler method. The resulting technique reads

$$\begin{cases} y_{1/2} &= y_0 + \frac{h}{2}f(x_0, y_0) \\ y_1 &= y_0 + hf(x_0 + \frac{1}{2}h, y_{1/2}). \end{cases}$$

or, equivalently,

$$y_1 = y_0 + hf(x_0 + \tfrac{h}{2}, y_0 + \tfrac{h}{2}f(x_0, y_0)). \tag{3.5}$$

The method is explicit, since we can compute successively $y_{1/2}$ and y_1 without solving nonlinear systems. Just two function evaluations of f are required. This scheme is called the modified Euler method or the method of Collatz. Although the method applies the Euler method first, the final approximation y_1 is significantly more accurate than in the Euler method in general.

The accuracy of the above methods follows from latter discussions.

Explicit Euler method (revisited)

We consider the explicit Euler method more detailed. This scheme can be motivated by two other approaches. First, replacing the derivative in the ODE $y' = f(x, y)$ by the common difference quotient (of first order) yields

$$\frac{y(x_0 + h) - y(x_0)}{h} \doteq f(x_0, y(x_0)) \quad \Rightarrow \quad y_1 = y_0 + hf(x_0, y_0).$$

Second, we consider the tangent of $y(x)$ corresponding to the point (x_0, y_0) as approximation of the solution. The tangent is

$$t(x) = y(x_0) + (x - x_0)y'(x_0) = y(x_0) + (x - x_0)f(x_0, y(x_0)).$$

It follows

$$y_1 := t(x_0 + h) = y_0 + hf(x_0, y_0),$$

i.e., the explicit Euler method.

For example, we solve the initial value problem $y' = \frac{1}{2y}$, $y(\frac{1}{4}) = \frac{1}{2}$, $x \in [\frac{1}{4}, 2]$. The solution is just $y(x) = \sqrt{x}$. Figure 8 illustrates the numerical solutions following form the Euler method. We recognise that the accuracy becomes better the more steps N are applied.

3.3 Consistency and convergence

We consider a general explicit one-step method of the form (3.1) with the increment function Φ.

Different notations are used to analyse the accuracy of the approximations y_{i+1} in comparison to the exact solution $y(x_i)$. On a local scale, we arrange the following definition.

Definition 1 (local discretisation error) *Let $y(x)$ be the exact solution of the ODE-IVP $y' = f(x, y)$, $y(x_0) = y_0$ and $y_1 = y_0 + h\Phi(x_0, y_0, h)$ denote the numerical approximation of one step with $h > 0$. The local discretisation error is then defined as*

$$\tau(h) := \frac{y(x_0 + h) - y_1}{h}. \tag{3.6}$$

The definition (3.6) of the local error can be interpreted in three different ways:

- the difference between the exact solution and the numerical approximation (discretisation error after one step starting from the exact solution) scaled by the step size h.

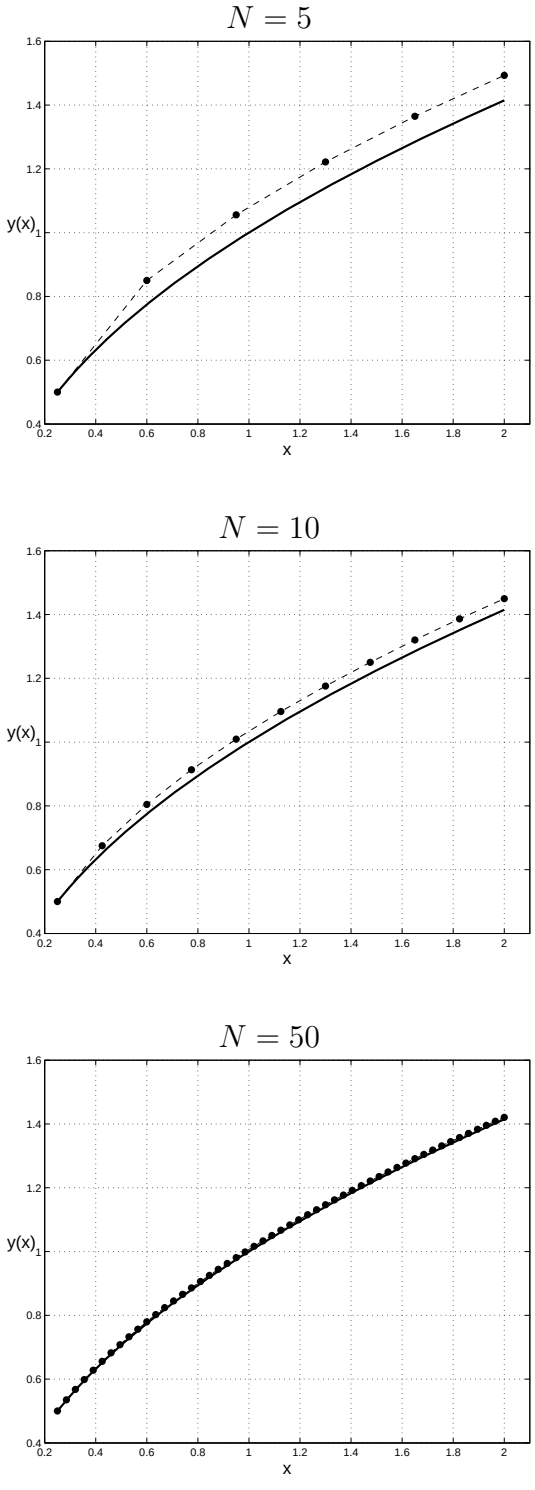

Figure 8: Solution of $y' = \frac{1}{2y}$, $y(\frac{1}{4}) = \frac{1}{2}$ (solid line) and numerical approximation (points) resulting from the explicit Euler method using N steps.

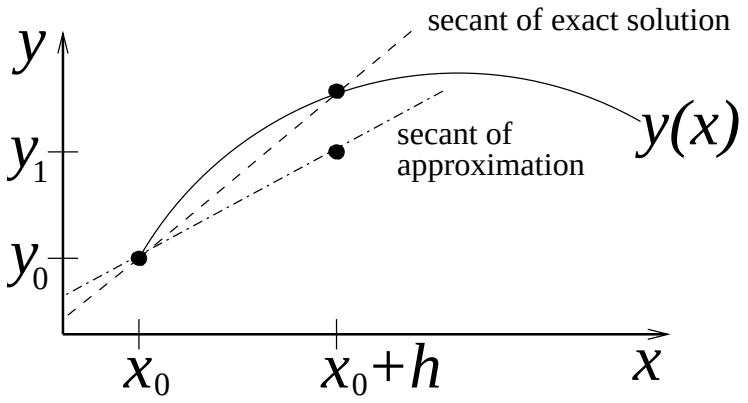

Figure 9: Secants of exact solution and numerical approximation.

- the difference in the gradients of the respective secants

$$\tau(h) = \underbrace{\frac{y(x_0+h)-y_0}{h}}_{\text{exact solution}} - \underbrace{\frac{y_1-y_0}{h}}_{\text{approximation}} .$$

  The secants are illustrated in Fig. 9. If $\tau(h) \to 0$ holds, then both secants become the tangent $t(x) = y(x_0) + (x - x_0)y'(x_0)$ in the limit.

- the defect

$$\tau(h) = \frac{y(x_0+h)-y_0}{h} - \Phi(x_0, y_0, h), \tag{3.7}$$

  which results from inserting the exact solution into the formula of the approximation.

Example 1: Local discretisation error of the explicit Euler method

Taylor expansion yields assuming $y \in C^2$

$$\begin{aligned} y(x_0+h) &= y(x_0) + hy'(x_0) + \tfrac{1}{2}h^2 y''(x_0 + \vartheta(h)h) \\ &= y_0 + hf(x_0, y_0) + \tfrac{1}{2}h^2 y''(x_0 + \vartheta(h)h) \end{aligned}$$

with $0 < \vartheta(h) < 1$.

The local discretisation error becomes

$$\begin{aligned}\tau(h) &= \tfrac{1}{h}(y(x_0+h)-y_1)\\ &= \tfrac{1}{h}(y(x_0+h)-y_0-hf(x_0,y_0))\\ &= \tfrac{1}{2}hy''(x_0+\vartheta(h)h).\end{aligned}$$

It follows $\tau(h)=\mathcal{O}(h)$.

Example 2: Local discretisation error of the implicit Euler method

For simplicity, we assume a bounded right-hand side, i.e., $\|f\| \leq M$. On the one hand, the implicit Euler method implies

$$y_1 = y_0 + hf(x_0+h,y_1) = y_0 + hf(x_0+h, y_0+hf(x_0+h,y_1)).$$

Multidimensional Taylor expansion yields

$$\begin{aligned}y_1 &= y_0 + h\left[f(x_0,y_0) + \tfrac{\partial f}{\partial x}(x_0,y_0)h + \tfrac{\partial f}{\partial y}(x_0,y_0)hf(x_0+h,y_1) + \mathcal{O}(h^2)\right]\\ &= y_0 + hf(x_0,y_0) + \mathcal{O}(h^2).\end{aligned}$$

On the other hand, the Taylor expansion of the exact solution from above can be used. It follows

$$\begin{aligned}\tau(h) &= \tfrac{1}{h}(y(x_0+h)-y_1)\\ &= \tfrac{1}{h}(y_0+hf(x_0,y_0)+\mathcal{O}(h^2)-(y_0+hf(x_0,y_0)+\mathcal{O}(h^2))) = \mathcal{O}(h).\end{aligned}$$

Again we obtain $\tau(h)=\mathcal{O}(h)$ like in the explicit Euler method.

Based on the property of the local discretisation error, we define the consistency.

Definition 2 (consistency) *A method (or its increment function Φ) is called consistent, if the local discretisation error tends to zero uniformly in x, y for $h \to 0$:*

$$\|\tau(h)\| \leq \sigma(h) \qquad \textit{with} \qquad \lim_{h\to 0}\sigma(h) = 0.$$

The method is consistent of (at least) order p, if

$$\|\tau(h)\| = \mathcal{O}(h^p).$$

Consistency of one-step methods can be easily characterised by the following property.

Lemma 2 *Let the right-hand side f of the ODEs $y' = f(x, y)$ be continuous in x and satisfy the Lipschitz-condition (2.3) with respect to y. Then it follows the equivalence*

$$\Phi \text{ is consistent} \quad \Leftrightarrow \quad \lim_{h \to 0} \Phi(x, y, h) = f(x, y).$$

Proof:

Let z be the solution of the ODE-IVP $z'(x) = f(x, z(x))$, $z(\xi) = \eta$. Due to the definition of τ and the mean value theorem of differentiation

$$\|\tau(\xi, \eta, h)\| = \|z'(\xi + \theta h) - \Phi(\xi, \eta, h)\|$$

for some $\theta \in (0, 1)$. Since f and z' are continuous in x, both functions are uniformly continuous in an interval $[a, b]$. It follows

$$\lim_{h \to 0} z'(\xi + \theta h) = z'(\xi) = f(\xi, \eta)$$

uniformly and thus

$$\lim_{h \to 0} \|\tau(\xi, \eta, h)\| = \|f(\xi, \eta) - \lim_{h \to 0} \Phi(\xi, \eta, h)\|.$$

This relation shows the statement. □

The order of consistency describes the quality of the numerical approximation after a single step. However, we are interested in the quality of the approximation after N steps, where the final point x_{end} is reached. This motivates the following definition.

Definition 3 (global discretisation error and convergence)
The global discretisation error of a method using a grid $x_0 < x_1 < \cdots < x_N$ is defined by the difference

$$e_N = y(x_N) - y_N. \tag{3.8}$$

For $N \to \infty$, we assume $|h| \to 0$ with $|h| := \max_{i=0,\ldots,N-1} |x_{i+1} - x_i|$. The method is called convergent, if for fixed $x = x_N$ it holds

$$\lim_{N\to\infty} e_N = 0.$$

The method is convergent of (at least) order p, if it holds

$$e_N = \mathcal{O}(|h|^p).$$

Concerning consistency and convergence, we prove the following theorem.

Theorem 5 (convergence of one-step methods) *Let f be continuous and satisfy the Lipschitz-condition (2.3). Consider a one-step scheme with increment function Φ, which is consistent of order p, i.e.,*

$$\|\tau(h)\| = \mathcal{O}(h^p).$$

Then the global discretisation error is bounded by

$$\|e_N\| \leq c \cdot |h|^p \cdot \frac{\exp(L|x_N - x_0|) - 1}{L}$$

with a constant $c > 0$ and $|h| = \max\{h_0, h_1, \ldots, h_{N-1}\}$ for $h_i := x_{i+1} - x_i$.

Proof:

The one-step scheme generates the sequence $y_1, \ldots, y_N$. We define auxilary ODE-IVPs by

$$u_i'(x) = f(x, u_i(x)), \quad u(x_i) = y_i \qquad \text{for } i = 0, 1, \ldots, N-1.$$

The auxilary solutions are sketched in Figure 10. The global error can be written as

$$\begin{aligned} e_N &:= y(x_N) - y_N \quad = \quad u_0(x_N) - y_N \\ &= u_{N-1}(x_N) - y_N + \sum_{i=0}^{N-2} u_i(x_N) - u_{i+1}(x_N). \end{aligned}$$

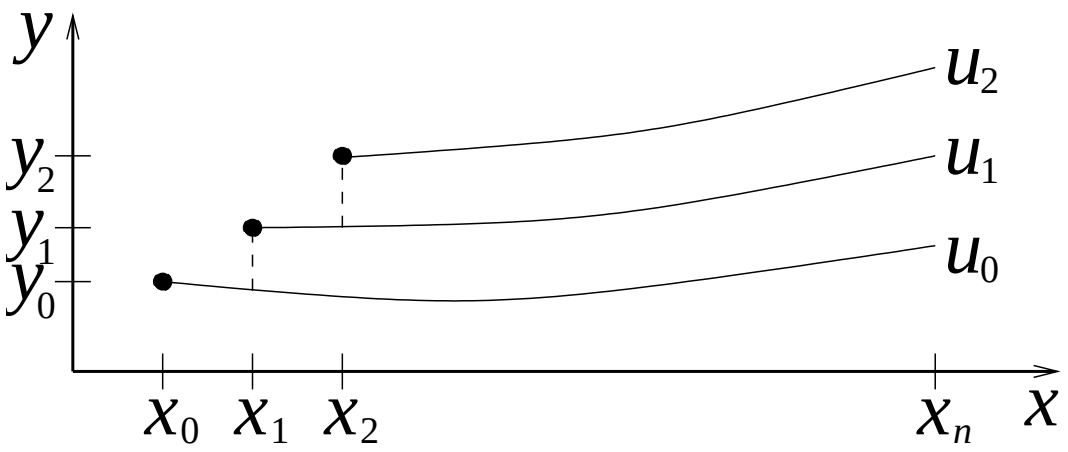

Figure 10: Lady Windermere's Fan.

We obtain the estimate

$$\|e_N\| \le \|u_{N-1}(x_N) - y_N\| + \sum_{i=0}^{N-2} \|u_i(x_N) - u_{i+1}(x_N)\|.$$

Since the solutions u_i satisfy the same system of ODEs for different initial values, we can apply the relation (2.10). It follows

$$\begin{aligned} \|e_N\| &\le \|u_{N-1}(x_N) - y_N\| + \sum_{i=0}^{N-2} \|u_i(x_{i+1}) - u_{i+1}(x_{i+1})\| \mathrm{e}^{L|x_N - x_{i+1}|} \\ &= \sum_{i=0}^{N-1} \|u_i(x_{i+1}) - y_{i+1}\| \mathrm{e}^{L|x_N - x_{i+1}|}. \end{aligned}$$

The norms on the right-hand side correspond to the local errors after one step. Since we assume a consistent method, it holds

$$\|u_i(x_{i+1}) - y_{i+1}\| \le c \cdot h_i^{p+1} \le c \cdot |h|^p \cdot h_i$$

uniformly with a constant $c > 0$ and $h_i := x_{i+1} - x_i$. Thus we obtain

$$\begin{aligned} \|e_N\| &\le c \cdot |h|^p \sum_{i=0}^{N-1} h_i \mathrm{e}^{L|x_N - x_{i+1}|} \le c \cdot |h|^p \int_{x_0}^{x_N} \mathrm{e}^{L|x_N - t|} \, \mathrm{d}t \\ &= c \cdot |h|^p \cdot \frac{\mathrm{e}^{L|x_N - x_0|} - 1}{L}, \end{aligned}$$

which completes the proof. □

This theorem demonstrates that consistency is sufficient for convergence. Moreover, the order of consistency coincides with the order of convergence. The consistency can be determined by analysing the increment function Φ of the one-step method. Vice versa, convergent methods exist, which are not consistent. Hence consistency is not necessary for convergence. However, inconsistent methods are not used in practice.

Comparison to numerical quadrature:

Assume that we want to compute the integral

$$I(g) := \int_a^b g(x)\,\mathrm{d}x$$

of a function $g \in C^2[a,b]$ approximately by trapezoidal rule. Let

$$M := \max_{a\le x\le b} |g''(x)|.$$

We apply a grid $x_i = a + ih$ with step size $h = \frac{b-a}{N}$. Let $T_{x_i}^{x_{i+1}}(g)$ be the area of one trapezoid constructed in the interval $[x_i, x_{i+1}]$. It holds

$$\left| \int_{x_i}^{x_i+h} g(x)\,\mathrm{d}x - T_{x_i}^{x_{i+1}}(g) \right| = \tfrac{1}{12}h^3|g''(\xi)| \le \tfrac{1}{12}h^3 M =: R(h).$$

The value $R(h) = \mathcal{O}(h^3)$ or $R(h)/h = \mathcal{O}(h^2)$ can be seen as a local error of the trapezoidal rule. For the global error E_N, we obtain

$$\begin{aligned} E_N &:= \left| \int_a^b g(x)\,\mathrm{d}x - \sum_{i=1}^N T_{x_{i-1}}^{x_i}(g) \right| = \left| \sum_{i=1}^N \int_{x_{i-1}}^{x_i} g(x)\,\mathrm{d}x - T_{x_{i-1}}^{x_i}(g) \right| \\ &\le \sum_{i=1}^N \left| \int_{x_{i-1}}^{x_i} g(x)\,\mathrm{d}x - T_{x_{i-1}}^{x_i}(g) \right| \le \sum_{i=1}^N \frac{1}{12}h^3 M = N\frac{1}{12}h^3 M = \frac{b-a}{12}h^2 M. \end{aligned}$$

Thus it holds $E_N = \mathcal{O}(h^2)$. Remark that $N = \frac{b-a}{h}$, i.e., E_N can be written in dependence on the step size h. We recognise that the order of the global error $E_N = \mathcal{O}(h^2)$ coincides with the order of the local error $R(h)/h = \mathcal{O}(h^2)$ (provided that we define the local error as $R(h)/h$ — and not as $R(h)$).

3.4 Taylor methods for ODEs

The analysis of the order of consistency indicates an approach for numerical techniques based on Taylor expansions. For simplicity, we consider an autonomous system of ODEs, i.e.,

$$y'(x) = f(y(x)), \quad y(x_0) = y_0.$$

An arbitrary intial value problem of ODEs $y' = f(x, y)$ can be transformed to an equivalent autonomous system via

$$Y'(t) = F(Y(t)) \qquad \text{with} \quad Y(t) := \begin{pmatrix} y(t) \\ x(t) \end{pmatrix}, \quad F(Y) := \begin{pmatrix} f(x, y) \\ 1 \end{pmatrix}$$

with initial conditions $x(t_0) = x_0$, $y(t_0) = y_0$.

Moreover, we discuss a scalar autonomous ODE $y(x) = f(y(x))$ now. Given a solution $y \in C^{p+1}$, Taylor expansion yields

$$\begin{aligned} y(x_0 + h) &= y(x_0) + hy'(x_0) + \tfrac{h^2}{2!}y''(x_0) + \cdots + \tfrac{h^p}{p!}y^{(p)}(x_0) \\ &\quad + \tfrac{h^{p+1}}{(p+1)!}y^{(p+1)}(x_0 + \vartheta(h)h) \end{aligned} \tag{3.9}$$

with $0 < \vartheta(h) < 1$. For a sufficiently smooth right-hand side f, we can replace the derivatives of the unknown solution. It holds

$$\begin{aligned} y' &= f(y) \\ y'' &= f'(y)y' = f'(y)f(y) \\ y''' &= (f''(y)y')f(y) + f'(y)(f'(y)y') = f''(y)f(y)^2 + f'(y)^2 f(y) \\ &\vdots \end{aligned}$$

and thus

$$\begin{aligned} y'(x_0) &= f(y_0) \\ y''(x_0) &= f'(y_0)f(y_0) \\ y'''(x_0) &= f''(y_0)f(y_0)^2 + f'(y_0)^2 f(y_0) \\ &\vdots \quad . \end{aligned}$$

Since the initial value $y(x_0) = y_0$ is given, we define one-step methods $y_1 = y_0 + h\Phi(y_0, h)$ via

$$\begin{aligned}
\Phi_1(y,h) &= f(y) \\
\Phi_2(y,h) &= f(y) + \tfrac{h}{2} f'(y) f(y) \\
\Phi_3(y,h) &= f(y) + \tfrac{h}{2} f'(y) f(y) + \tfrac{h^2}{6}\left[f''(y) f(y)^2 + f'(y)^2 f(y)\right] \\
&\vdots
\end{aligned}$$

based on the Taylor expansion (3.9). The method specified by Φ_1 is just the explicit Euler method. Due to this construction, the pth method exhibits the local discretisation error

$$\tau_p(h) = \frac{y(x_0+h) - y(x_0)}{h} - \Phi_p(y_0, h) = \frac{h^p}{(p+1)!} y^{(p+1)}(x_0 + \vartheta(h)h)$$

i.e., $\tau_p(h) = \mathcal{O}(h^p)$. It follows that the method is consistent of order p.

However, the number of required derivatives increases rapidly in case of systems of ODEs:

$$\begin{aligned}
f &: \ n \text{ components} \\
\tfrac{\partial f}{\partial y} &: \ n^2 \text{ components} \\
\tfrac{\partial^2 f}{\partial y^2} &: \ n^3 \text{ components} \\
&\vdots \\
\tfrac{\partial^k f}{\partial y^k} &: \ n^{k+1} \text{ components.}
\end{aligned}$$

Hence the computational effort becomes large for higher orders. Moreover, the computation of derivatives of higher order via numerical differentiation becomes more and more affected by roundoff errors.

In conclusion, Taylor methods of order $p > 1$ are not used in practice.

3.5 Runge-Kutta methods

The most important class of one-step schemes are Runge-Kutta methods. The idea is to replace the integral in (2.2) by a quadrature rule with nodes $c_1, \dots, c_s \in [0,1]$ and (outer) weights $b_1, \dots, b_s \in \mathbb{R}$. Without loss of generality, we assume $c_1 \leq c_2 \leq \cdots \leq c_s$. It follows a finite sum

$$y_1 = y_0 + h \sum_{i=1}^{s} b_i f(x_0 + c_i h, y(x_0 + c_i h)).$$

The problem is that the intermediate values $y(x_0 + c_i h)$ are unknown a priori. We achieve according approximations by an integral relation again

$$y(x_0 + c_i h) = y_0 + h \int_0^{c_i} f(x_0 + sh, y(x_0 + sh)) \, \mathrm{d}s.$$

The involved integrals are substituted by quadrature formulas. To avoid the generation of new unknowns, the same nodes $c_1, \dots, c_s$ as before are used. Just new (inner) weights a_{ij} are introduced. We obtain the approximations

$$z_i = y_0 + h \sum_{j=1}^{s} a_{ij} f(x_0 + c_j h, z_j) \tag{3.10}$$

for $i = 1, \dots, s$. The resulting final approximation becomes

$$y_1 = y_0 + h \sum_{i=1}^{s} b_i f(x_0 + c_i h, z_i).$$

The general relations (3.10) represent a nonlinear system for the unknowns $z_1, \dots, z_s$. If these intermediate values have been determined, then we can compute y_1 directly via s evaluations of the function f.

Considering (3.10), a natural requirement is that a constant function $f \equiv 1$ ($y(x_0 + c_i h) = y_0 + c_i h$) is resolved exactly. We obtain the conditions

$$c_i = \sum_{j=1}^{s} a_{ij} \qquad \text{for each } \; i = 1, \dots, s. \tag{3.11}$$

This condition means that the sum of the weights is equal to the (relative) length of the corresponding subinterval.

A Runge-Kutta scheme is uniquely determined by its coefficients. The coefficients can be written in a so-called Butcher-tableau:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array} \qquad \text{resp.} \qquad \begin{array}{c|c} c & A \\ \hline & b^\top \end{array}$$

with $c \in \mathbb{R}^s$, $b \in \mathbb{R}^s$, $A \in \mathbb{R}^{s \times s}$.

Examples: Schemes from Sect. 3.2

(a): expl. Euler method, (b): impl. Euler method, (c): trapezoidal rule, (d): method of Collatz (midpoint rule):

(a) $\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$ (b) $\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$ (c) $\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$ (d) $\begin{array}{c|cc} 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & 0 & 1 \end{array}$

Example: Gauss-Runge-Kutta methods

For the nodes c_i and the weights b_i, we apply a Gaussian quadrature. The Gaussian quadrature exhibits the order $2s$, i.e., it holds

$$\sum_{i=1}^{s} b_i p(c_i) = \int_0^1 p(x)\,\mathrm{d}x \qquad \text{for all } \; p \in \mathbb{P}_{2s-1}$$

($\mathbb{P}_m$: polynomials up to degree m). The weights a_{ij} are determined such that for each $i = 1, \ldots, s$ it holds

$$\sum_{j=1}^{s} a_{ij} p(c_j) = \int_0^{c_i} p(x)\,\mathrm{d}x \qquad \text{for all } \; p \in \mathbb{P}_{s-1}.$$

In the simple case $s = 1$, it follows directly $c_1 = \frac{1}{2}$, $b_1 = 1$ and $a_{11} = \frac{1}{2}$. The resulting Runge-Kutta method is

$$\begin{aligned} z_1 &= y_0 + \tfrac{h}{2} f(x_0 + \tfrac{h}{2}, z_1), \\ y_1 &= y_0 + h f(x_0 + \tfrac{h}{2}, z_1). \end{aligned} \tag{3.12}$$

This approach corresponds to the midpoint rule (3.4), where the approximation $z_1 \doteq y(x_0 + \frac{1}{2}h)$ is determined by the implicit Euler method.

The Butcher tableau of the case $s = 2$ reads:

$$\begin{array}{c|cc} \frac{3-\sqrt{3}}{6} & \frac{1}{4} & \frac{3-2\sqrt{3}}{12} \\ \frac{3+\sqrt{3}}{6} & \frac{3+2\sqrt{3}}{12} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

If the matrix $A = (a_{ij})$ is full, then the Runge-Kutta method is implicit. A nonlinear system (3.10) of $s \cdot n$ algebraic equations has to be solved. In contrast, we want to achieve an explicit method now. The corresponding condition reads $a_{ij} = 0$ for $i \leq j$. Thus A becomes a strictly lower triangular matrix. The Butcher-tableau exhibits the form:

$$\begin{array}{c|ccccc} 0 & 0 & 0 & \cdots & \cdots & 0 \\ c_2 & a_{21} & 0 & \ddots & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & & \ddots & 0 & 0 \\ c_s & a_{s1} & \cdots & \cdots & a_{s,s-1} & 0 \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array}$$

In particular, it follows $c_1 = 0$ due to (3.11) and thus $z_1 = y_0$. Now the computation of the intermediate values reads

$$z_i = y_0 + h \sum_{j=1}^{i-1} a_{ij} f(x_0 + c_j h, z_j).$$

The computational effort for an explicit Runge-Kutta method just consists in the s evaluations of the right-hand side f. Explicit methods correspond

to successive extrapolations using the intermediate values. Implicit methods can be seen as an interpolation based on the intermediate values.

Examples: Some well-known explicit Runge-Kutta methods

Method of Heun (left), Kutta-Simpson rule (middle) and classical Runge-Kutta method (right):

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{3} & \frac{1}{3} & & \\ \frac{2}{3} & 0 & \frac{2}{3} & \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array} \qquad \begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ 1 & -1 & 2 & \\ \hline & \frac{1}{6} & \frac{4}{6} & \frac{1}{6} \end{array} \qquad \begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{array}$$

An equivalent notation of Runge-Kutta schemes results from the definition of the increments k_i via

$$k_i = f(x_0 + c_i h, z_i) = f\left(x_0 + c_i h, y_0 + h \sum_{j=1}^{s} a_{ij} k_j\right) \tag{3.13}$$

for $i = 1, \ldots, s$. Now the Runge-Kutta method reads

$$\begin{aligned} k_i &= f\left(x_0 + c_i h, y_0 + h \sum_{j=1}^{s} a_{ij} k_j\right), \quad i = 1, \ldots, s\,, \\ y_1 &= y_0 + h \sum_{i=1}^{s} b_i k_i. \end{aligned} \tag{3.14}$$

Thereby, the increments k_i are unknown a priori.

Order conditions

A Runge-Kutta method is determined by its coefficients c_i, b_i, a_{ij}. We derive conditions on these coefficients such that the one-step method becomes consistent of some order p. We consider autonomous scalar ODEs $y' = f(y)$. It follows

$$\begin{aligned} y'' &= f'y' = f'f, \\ y''' &= f''y'f + f'f'y' = f''f^2 + (f')^2 f. \end{aligned}$$

Taylor expansion of the exact solution yields

$$\begin{aligned} y(x_0+h) &= y(x_0) + hy'(x_0) + \tfrac{h^2}{2}y''(x_0) + \tfrac{h^3}{6}y'''(x_0) + \mathcal{O}(h^4) \\ &= y_0 + hf(y_0) + \tfrac{h^2}{2}f'(y_0)f(y_0) \\ &\quad + \tfrac{h^3}{6}\left[f''(y_0)f(y_0)^2 + f'(y_0)^2 f(y_0)\right] + \mathcal{O}(h^4). \end{aligned}$$

In the following, we use the abbreviations $f = f(y_0)$, $f' = f'(y_0)$, etc. We assume that the Runge-Kutta method fulfills the fundamental condition (3.11). A Taylor expansion of the function f in the increments (3.13) implies for $i = 1, \ldots, s$

$$\begin{aligned} k_i &= f + f'h\left(\sum_{j=1}^{s} a_{ij}k_j\right) + \tfrac{1}{2}f''h^2\left(\sum_{j=1}^{s} a_{ij}k_j\right)^2 + \mathcal{O}(h^3) \\ &= f + f'h\left(\sum_{j=1}^{s} a_{ij}\left(f + f'h\left(\sum_{l=1}^{s} a_{jl}k_l\right) + \mathcal{O}(h^2)\right)\right) \\ &\quad + \tfrac{1}{2}f''h^2\left(\sum_{j=1}^{s} a_{ij}\left(f + \mathcal{O}(h)\right)\right)^2 + \mathcal{O}(h^3) \\ &= f + f'h\left(\sum_{j=1}^{s} a_{ij}\left(f + f'h\left(\sum_{l=1}^{s} a_{jl}\left(f + \mathcal{O}(h)\right)\right) + \mathcal{O}(h^2)\right)\right) \\ &\quad + \tfrac{1}{2}f''h^2\left(fc_i + \mathcal{O}(h)\right)^2 + \mathcal{O}(h^3) \\ &= f + f'h\left(\sum_{j=1}^{s} a_{ij}\left(f + f'fhc_j + \mathcal{O}(h^2)\right)\right) + \tfrac{1}{2}f''f^2h^2c_i^2 + \mathcal{O}(h^3) \\ &= f + hf'fc_i + h^2(f')^2 f\left(\sum_{j=1}^{s} a_{ij}c_j\right) + \tfrac{1}{2}h^2f''f^2c_i^2 + \mathcal{O}(h^3). \end{aligned}$$

The approximation obtained by the Runge-Kutta method becomes

$$\begin{aligned} y_1 &= y_0 + h\sum_{i=1}^{s} b_i k_i \\ &= y_0 + hf\left(\sum_{i=1}^{s} b_i\right) + h^2 f' f\left(\sum_{i=1}^{s} b_i c_i\right) + h^3 (f')^2 f\left(\sum_{i,j=1}^{s} b_i a_{ij} c_j\right) \\ &\quad + \tfrac{1}{2} h^3 f'' f^2 \left(\sum_{i=1}^{s} b_i c_i^2\right) + \mathcal{O}(h^4). \end{aligned}$$

A comparison to the Taylor expansion of the exact solution shows the conditions for consistency up to order $p = 3$. We also cite the conditions for order $p = 4$:

$p = 1:$	$\sum_{i=1}^{s} b_i = 1$
$p = 2:$	$\sum_{i=1}^{s} b_i c_i = \frac{1}{2}$
$p = 3:$	$\sum_{i=1}^{s} b_i c_i^2 = \frac{1}{3}$
	$\sum_{i,j=1}^{s} b_i a_{ij} c_j = \frac{1}{6}$

$p = 4:$	$\sum_{i=1}^{s} b_i c_i^3 = \frac{1}{4}$
	$\sum_{i,j=1}^{s} b_i a_{ij} c_i c_j = \frac{1}{8}$
	$\sum_{i,j=1}^{s} b_i a_{ij} c_j^2 = \frac{1}{12}$
	$\sum_{i,j,l=1}^{s} b_i a_{ij} a_{jl} c_l = \frac{1}{24}$

The conditions for consistency can be derived up to an arbitrary order p. In case of explicit Runge-Kutta methods, the sums just involve the non-zero coefficients. For a desired order p of consistency, we like to apply a Runge-Kutta method with relatively low number of stages s. In case of implicit schemes, a method with s stages exhibits the maximum order $p = 2s$ in case of Gauss-Runge-Kutta methods. In case of explicit schemes, Table 1 gives corresponding informations.

stage number s	1	2	3	4	5	6	7	8	9	10	11	$\cdots$	17
maximum order p	1	2	3	4	4	5	6	6	7	7	8	$\cdots$	10

order p	1	2	3	4	5	6	7	8
minimum stage number s	1	2	3	4	6	7	9	11
number of order conditions	1	2	4	8	17	37	85	200

Table 1: Order and number of stages in explicit Runge-Kutta methods.

3.6 Dense output

The numerical method yields a sequence of approximations $y_0, y_1, \ldots, y_N$ corresponding to a grid $x_0 < x_1 < \cdots < x_N$. The number N of the steps should be relatively small, since the computational effort is proportional to N. The appropriate choice of the step sizes $h_i = x_{i+1} - x_i$ will be discussed in the next subsection.

Let the approximations $y_0, y_1, \ldots, y_N$ be already determined. Often we require more solution values on a finer grid (for example to visualise/plot the solution). We want to calculate these values with a low additional effort. An appropriate approach is the dense output, i.e., a continuous approximation $\tilde{y}(x)$ is contructed using the (maybe coarse) data (x_i, y_i) from the numerical integration method. Without loss of generality, we consider a scalar ODE in the following.

First idea: Interpolation

We can arrange a cubic spline interpolant $\tilde{y} \in C^2$ based on the data (x_i, y_i). However, the complete data has to be computed first in this case. Alternatively, we perform a cubic Hermite interpolation to obtain $\tilde{y} \in C^1$. For $x \in [x_i, x_{i+1}]$, the interpolant corresponding to the exact solution is

$$u(x_i + \theta h_i) = y(x_i)p_1(\theta) + y(x_{i+1})p_2(\theta) + h_i y'(x_i)p_3(\theta) + h_i y'(x_{i+1})p_4(\theta).$$

The interpolant corresponding to the available data is

$$\tilde{y}(x_i + \theta h_i) = y_i p_1(\theta) + y_{i+1}p_2(\theta) + h_i f(x_i, y_i)p_3(\theta) + h_i f(x_{i+1}, y_{i+1})p_4(\theta),$$

where the ODE $y' = f(x, y)$ has been applied. The basis polynomials read

$$\begin{aligned} p_1(\theta) &= 1 - 3\theta^2 + 2\theta^3, & p_2(\theta) &= 3\theta^2 - 2\theta^3, \\ p_3(\theta) &= \theta - 2\theta^2 + \theta^3, & p_4(\theta) &= -\theta^2 + \theta^3, \end{aligned} \qquad \text{for } 0 \le \theta \le 1.$$

The evaluation of the Hermite interpolant $\tilde{y}$ can be done online during the integration. Moreover, the function evaluations $f(x_i, y_i), f(x_{i+1}, y_{i+1})$ are available from the (explicit) Runge-Kutta method.

We determine the accuracy of the approximation. On the one hand, it holds

$$|y(x) - u(x)| \le \frac{1}{384} \left(\max_{s \in [x_i, x_{i+1}]} |y^{(4)}(s)| \right) h_i^4 = \mathcal{O}(h_i^4)$$

for $x \in [x_i, x_{i+1}]$. On the other hand, we obtain using the Lipschitz-condition $|f(x_i, y_i) - f(x_i, y(x_i))| \le L \cdot |y_i - y(x_i)|$

$$\begin{aligned} |u(x_i + \theta h_i) - \tilde{y}(x_i + \theta h_i)| \;\le\; & |y(x_i) - y_i| \cdot |p_1(\theta)| \\ & + |y(x_{i+1}) - y_{i+1}| \cdot |p_2(\theta)| \\ & + h_i \cdot L \cdot |y(x_i) - y_i| \cdot |p_3(\theta)| \\ & + h_i \cdot L \cdot |y(x_{i+1}) - y_{i+1}| \cdot |p_4(\theta)|. \end{aligned}$$

The definition

$$\hat{p}_l := \max_{\theta \in [0,1]} |p_l(\theta)| \qquad \text{for } l = 1, 2, 3, 4$$

yields

$$|u(x) - \tilde{y}(x)| \le |y(x_i) - y_i| \cdot (\hat{p}_1 + h_i L \hat{p}_3) + |y(x_{i+1}) - y_{i+1}| \cdot (\hat{p}_2 + h_i L \hat{p}_4).$$

for all $x \in [x_i, x_{i+1}]$. In case of a Runge-Kutta method with consistency order p, the convergence of the scheme implies $|y(x_i) - y_i| = \mathcal{O}(|h|^p)$ and thus $|u(x) - \tilde{y}(x)| = \mathcal{O}(|h|^p)$. It follows

$$|y(x) - \tilde{y}(x)| = \mathcal{O}(|h|^4) + \mathcal{O}(|h|^p).$$

Hence an order p of consistency implies a dense output, which approximates the exact solution with order $q = \min\{4, p\}$.

Second idea: Continuous Runge-Kutta method

The second strategy of dense output is to use a Runge-Kutta scheme as a basis for a continuous extension. Thereby, the constant weights b_i are replaced by polynomials $b_i(\theta)$ in $\theta \in (0,1)$. Let the scheme be defined by

$$\begin{aligned} \tilde{y}(x_0+\theta h) &= y_0 + h\sum_{i=1}^{s} b_i(\theta) k_i, \qquad 0<\theta<1, \\ k_i &= f\left(x_0 + c_i h, y_0 + h\sum_{j=1}^{s} a_{ij} k_j\right), \qquad i=1,\dots,s. \end{aligned} \tag{3.15}$$

Furthermore, the Runge-Kutta scheme shall satisfy the node relation (3.11). We determine the order conditions of the continuous extension for the orders $p = 1, 2, 3$. (The approximation for $y(x_0 + \theta h)$ has to be consistent of order p for all $\theta \in (0,1)$.) We rewrite the dense output scheme as

$$\begin{aligned} \tilde{y}(x_0+\theta h) &= y_0 + \theta h\sum_{i=1}^{s} \frac{b_i(\theta)}{\theta} k_i, \qquad 0<\theta<1, \\ k_i &= f\left(x_0 + \frac{c_i}{\theta}\theta h, y_0 + \theta h\sum_{j=1}^{s} \frac{a_{ij}}{\theta} k_j\right), \qquad i=1,\dots,s. \end{aligned}$$

These formulas represent an ordinary Runge-Kutta method with step size θh and new coefficients

$$\tilde{b}_i(\theta) := \frac{b_i(\theta)}{\theta}, \qquad \tilde{c}_i(\theta) := \frac{c_i}{\theta}, \qquad \tilde{a}_{ij}(\theta) := \frac{a_{ij}}{\theta}.$$

For each θ, the new coefficients have to satisfy the usual order conditions. Thus we obtain:

$$p=1: \quad \sum_{i=1}^{s} \tilde{b}_i(\theta) = 1 \quad \Rightarrow \quad \sum_{i=1}^{s} \frac{b_i(\theta)}{\theta} = 1 \quad \Rightarrow \quad \sum_{i=1}^{s} b_i(\theta) = \theta$$

$$p=2: \quad \sum_{i=1}^{s} \tilde{b}_i(\theta)\tilde{c}_i(\theta) = \frac{1}{2} \quad \Rightarrow \quad \sum_{i=1}^{s} \frac{b_i(\theta)}{\theta}\cdot\frac{c_i}{\theta} = \frac{1}{2} \quad \Rightarrow \quad \sum_{i=1}^{s} b_i(\theta) c_i = \frac{\theta^2}{2}$$

$$p=3: \quad \sum_{i=1}^{s} \tilde{b}_i(\theta)\tilde{c}_i(\theta)^2 = \frac{1}{3} \quad \Rightarrow \quad \sum_{i=1}^{s} \frac{b_i(\theta)}{\theta}\cdot\frac{c_i^2}{\theta^2} = \frac{1}{3} \quad \Rightarrow \quad \sum_{i=1}^{s} b_i(\theta) c_i^2 = \frac{\theta^3}{3}$$

$$\sum_{i,j=1}^{s} \tilde{b}_i(\theta)\tilde{a}_{ij}(\theta)\tilde{c}_j(\theta) = \frac{1}{6} \quad \Rightarrow \quad \sum_{i,j=1}^{s} \frac{b_i(\theta)}{\theta} \cdot \frac{a_{ij}}{\theta} \cdot \frac{c_j}{\theta} = \frac{1}{6} \quad \Rightarrow \quad \sum_{i,j=1}^{s} b_i(\theta)a_{ij}c_j = \frac{\theta^3}{6}$$

The generalisation to higher orders is obvious. In general, the maximum order for the dense output scheme will be lower than the maximum order for the pointwise (original) method.

Example:

We use the classical Runge-Kutta method with four stages. The scheme is consistent of order 4. We determine the polynomials $b_i(\theta)$ such that the dense output scheme features the order 3.

The order conditions imply the equations:

$$\begin{aligned} b_1(\theta) + b_2(\theta) + b_3(\theta) + b_4(\theta) &= \theta \\ \tfrac{1}{2}b_2(\theta) + \tfrac{1}{2}b_3(\theta) + b_4(\theta) &= \tfrac{\theta^2}{2} \\ \tfrac{1}{4}b_2(\theta) + \tfrac{1}{4}b_3(\theta) + b_4(\theta) &= \tfrac{\theta^3}{3} \\ \tfrac{1}{4}b_3(\theta) + \tfrac{1}{2}b_4(\theta) &= \tfrac{\theta^3}{6} \end{aligned}$$

Solving this linear system yields

$$b_1(\theta) = \theta - \frac{3\theta^2}{2} + \frac{2\theta^3}{3}, \quad b_2(\theta) = b_3(\theta) = \theta^2 - \frac{2\theta^3}{3}, \quad b_4(\theta) = -\frac{\theta^2}{2} + \frac{2\theta^3}{3}.$$

Now the dense output scheme (3.15) can be applied, since all involved coefficients are determined. It holds $b_i(0) = 0$ and $b_i(1) = b_i$ for each i. Hence the approximating function $\tilde{y}$ is globally continuous and just piecewise smooth.

3.7 Step-Size Control

In a numerical integration, the approximations $y_k \doteq y(x_k)$ are computed successively by some numerical method. We would like to obtain an automatic selection of the step sizes $h_k := x_{k+1} - x_k$ such that the corresponding error in the approximations remains sufficiently small.

Let $y = (y_1, \ldots, y_n)^\top$ be the components of the solution. We assume that a given numerical scheme exhibits a consistency order of p, i.e., the corresponding approximation $y^h \doteq y(x_0 + h)$ satisfies

$$y_i^h - y_i(x_0 + h) = \mathcal{O}(h^{p+1}) = C_i h^{p+1} + \mathcal{O}(h^{p+2}) \tag{3.16}$$

with constants $C_i \neq 0$ for each component. A similar numerical technique is used to compute an approximation $\hat{y}^h$ of a higher order

$$\hat{y}_i^h - y_i(x_0 + h) = \mathcal{O}(h^{p+2}). \tag{3.17}$$

In Runge-Kutta methods, embedded schemes are usually employed. Several possibilities exist in case of multi-step methods. Richardson extrapolation can be applied with respect to both one-step and multi-step methods. Thereby, an approximation is computed using step size h as usual and another approximation is calculated with two steps of size $\frac{h}{2}$, for example. Both values yield an approximation of a higher order in a corresponding extrapolation.

We want to estimate the error $y^h - y(x_0 + h)$ of the lower order method. Combining (3.16) and (3.17) yields

$$y_i^h - y_i(x_0 + h) = y_i^h - \hat{y}_i^h - (y_i(x_0 + h) - \hat{y}_i^h) = y_i^h - \hat{y}_i^h + \mathcal{O}(h^{p+2}). \tag{3.18}$$

Thus $\hat{y}^h - y^h$ represents an estimator for the local error of order $p + 1$. Applying (3.16) and (3.18), it follows

$$y_i^h - \hat{y}_i^h = C_i h^{p+1} + \mathcal{O}(h^{p+2}). \tag{3.19}$$

We assume that we have already performed an integration step of the size h_{used}. Now we want to estimate an appropriate step size h_{opt} to re-

peat the integration. The properties (3.16) and (3.19) imply approximately

$$\begin{aligned} y_i^{h_{\text{used}}} - \hat{y}_i^{h_{\text{used}}} &\doteq C_i h_{\text{used}}^{p+1}, \\ y_i^{h_{\text{opt}}} - y_i(x_0 + h_{\text{opt}}) &\doteq C_i h_{\text{opt}}^{p+1}. \end{aligned}$$

Eliminating the constant C_i yields

$$\frac{|y_i^{h_{\text{opt}}} - y_i(x_0 + h_{\text{opt}})|}{|y_i^{h_{\text{used}}} - \hat{y}_i^{h_{\text{used}}}|} = \left(\frac{h_{\text{opt}}}{h_{\text{used}}}\right)^{p+1}. \tag{3.20}$$

The error estimate of the done step is given by

$$\eta_i := |y_i^{h_{\text{used}}} - \hat{y}_i^{h_{\text{used}}}| \tag{3.21}$$

for $i = 1, \ldots, n$. The error corresponding to the new step size shall satisfy

$$|y_i^{h_{\text{opt}}} - y_i(x_0 + h_{\text{opt}})| = \text{TOL} \tag{3.22}$$

for some given absolute tolerance $\text{TOL} > 0$ in all components. We do not want that the error is smaller than TOL, since a smaller error implies smaller step sizes and thus more computational effort due to more steps. Inserting the last two relations in equation (3.20) implies

$$h_{\text{opt},i} = h_{\text{used}} \cdot \sqrt[p+1]{\frac{\text{TOL}}{\eta_i}},$$

where each component exhibits a different step size. The size for the new step is chosen as

$$h_{\text{new}} = \delta \cdot \min_{i=1,\ldots,n} h_{\text{opt},i}$$

including some safety factor $\delta = 0.9$, for example. To avoid oscillating step sizes, the restriction

$$\sigma \; h_{\text{used}} \leq h_{\text{new}} \leq \theta \; h_{\text{used}}$$

is imposed with $0 < \sigma < 1 < \theta$ (e.g. $\sigma = \frac{1}{5}, \theta = 5$).

If $h_{\text{new}} < h_{\text{used}}$ holds, then we have not been sufficiently accurate in the done integration step with respect to our demand (3.22). Consequently, the step is repeated using h_{new} instead of h_{used}. If $h_{\text{new}} \geq h_{\text{used}}$ is satisfied,

then the integration step is accepted and the next step is done using h_{new} as suggested step size.

Often the tolerance is defined relatively with respect to the magnitude of the solution. Given some relative tolerance RTOL > 0 and absolute tolerance ATOL > 0, we arrange

$$\text{TOL} = \text{ATOL} + \text{RTOL} \cdot |y_i^{h_{\text{used}}}|.$$

The absolute part ATOL is required in case of $|y_i^{h_{\text{used}}}| \approx 0$. Typical values are RTOL $= 10^{-3}$ and ATOL $= 10^{-6}$, for example.

Using the modulus $|\cdot|$ like above, i.e., some kind of maximum norm, exhibits a lack of smoothness, which sometimes causes problems in the simulations. In practice, the scaled norm

$$\text{ERR} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{\hat{y}_i^{h_{\text{used}}} - y_i^{h_{\text{used}}}}{\text{ATOL} + \text{RTOL} \cdot |y_i^{h_{\text{used}}}|}\right)^2} \tag{3.23}$$

ia applied, which represents a kind of weighted Euclidean norm. Note that denominators are always positive in (3.23). The condition (3.22) corresponds to ERR $= 1$ now. The new step size becomes

$$h_{\text{new}} = \delta \cdot h_{\text{used}} \cdot \frac{1}{\sqrt[p+1]{\text{ERR}}}$$

using some safety factor δ again.

The estimation is done for the error in the method of order p, whereas the result of the method of order $p + 1$ is only used in the error estimation. However, the approximation of order $p + 1$ is often applied as the output of the algorithm after each integration step. This is reasonable, since the method of order $p + 1$ is usally more accurate.

The above approach controls the local error in each integration step. However, we like to select the step sizes such that the global error (3.8) satisfies a predetermined accuracy. Yet there are no satisfactory strategies to control the global error. Hence numerical integrators of common software packages (e.g. MATLAB) just perform a step size selection based on the local error.

Embedded techniques

It remains to choose the two numerical methods in the estimation of the local error. In case of Runge-Kutta methods, embedded schemes are applied, since the additional computational work for the second approximation is relatively low.

The Butcher tableau of an embedded scheme reads

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \\ \hline & \hat{b}_1 & \hat{b}_2 & \cdots & \hat{b}_s \end{array}$$

with two sets of weights b_i and $\hat{b}_i$, respectively. The corresponding approximations are

$$\begin{aligned} y^h &= y_0 + h(b_1 k_1 + \cdots + b_s k_s), \\ \hat{y}^h &= y_0 + h(\hat{b}_1 k_1 + \cdots + \hat{b}_s k_s). \end{aligned}$$

If the data $k_1, \ldots, k_s$ for computing the approximation y^h is available, then the second approximation $\hat{y}^h$ can be calculated with nearly no additional effort.

In case of explicit Runge-Kutta methods, the class of the Runge-Kutta-Fehlberg methods represents embedded schemes.

Example: Runge-Kutta-Fehlberg 2(3)

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{4} & \frac{1}{4} & & & \\ \frac{27}{40} & -\frac{189}{800} & \frac{729}{800} & & \\ 1 & \frac{214}{891} & \frac{1}{33} & \frac{650}{891} & \\ \hline & \frac{214}{891} & \frac{1}{33} & \frac{650}{891} & 0 \\ \hline & \frac{533}{2106} & 0 & \frac{800}{1053} & -\frac{1}{78} \end{array}$$

Chapter 4

Multistep Methods

In this chapter, we investigate multistep methods, i.e., several old approximations are used to construct a new approximation. In contrast to one-step techniques, consistency alone is not sufficient for the convergence of these methods.

4.1 Techniques based on numerical quadrature

We introduce an important class of multistep schemes now. The strategy is based on the integral equation (2.2). A polynomial interpolation is arranged and the exact integral of the polynomial yields an approximation.

We consider the initial value problem $y' = f(x, y),\ y(x_0) = y_0$, see (2.1). For the following discussions, we assume a scalar ODE, since the strategy can be applied in each component of a system separately. Let the approximations

$$(x_{i-k+1}, y_{i-k+1}), (x_{i-k+2}, y_{i-k+2}), \ldots, (x_{i-1}, y_{i-1}), (x_i, y_i) \tag{4.1}$$

be given for some integer $k \geq 1$. We want to construct a new approximation (x_{i+1}, y_{i+1}). Choosing some integer $l \geq 1$, the exact solution satisfies the

integral equation

$$\begin{aligned} y(x_{i+1}) &= y(x_{i-l+1}) + \int_{x_{i-l+1}}^{x_{i+1}} y'(s)\,\mathrm{d}s \\ &= y(x_{i-l+1}) + \int_{x_{i-l+1}}^{x_{i+1}} f(s, y(s))\,\mathrm{d}s. \end{aligned} \tag{4.2}$$

Now we approximate the integrand $f(x, y(x))$. We arrange the polynomial $p_{k,i} \in \mathbb{P}_{k-1}$, which interpolates the data

$$(x_j, f(x_j, y_j)) \qquad \text{for } \; j = i-k+1, i-k+2, \ldots, i-1, i.$$

Consequently, it holds

$$p_{k,i}(x_j) = f(x_j, y_j) \qquad \text{for } \; j = i-k+1, i-k+2, \ldots, i-1, i.$$

The interpolating polynomial is unique. Using a Lagrange basis

$$L_{i,j}(x) = \prod_{\nu=1, \nu\neq j}^{k} \frac{x - x_{i-\nu+1}}{x_{i-j+1} - x_{i-\nu+1}} \qquad \text{for } \; j = 1, \ldots, k\ ,$$

the polynomial becomes with $f_i := f(x_i, y_i)$

$$p_{k,i}(x) = \sum_{j=1}^{k} f_{i-j+1} L_{i,j}(x).$$

Due to the assumption $p_{k,i}(x) \approx f(x, y(x))$ in the considered domain, the new approximation becomes due to (4.2)

$$y_{i+1} = y_{i-l+1} + \sum_{j=1}^{k} f_{i-j+1} \int_{x_{i-l+1}}^{x_{i+1}} L_{i,j}(s)\,\mathrm{d}s.$$

Since the Lagrange polynomials are known, the integral can be evaluated exactly.

In most cases, it holds $l \leq k$, i.e., the interval of the interpolation contains the interval of the integration (to the left-hand side). Fig. 11 illustrates this strategy. We have achieved an explicit k-step method.

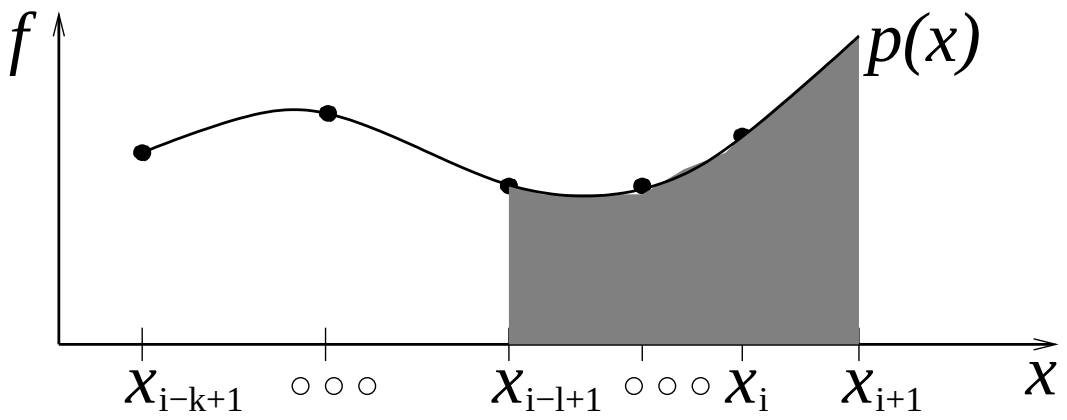

Figure 11: Construction of multistep method by quadrature.

In case of an equidistant grid $x_i = x_0 + ih$, the integrals of the Lagrange polynomials are independent of the index i

$$\begin{aligned}\int_{x_{i-l+1}}^{x_{i+1}} L_{i,j}(s)\,\mathrm{d}s &= \int_{x_{i-l+1}}^{x_{i+1}} \prod_{\nu\neq j} \frac{s - x_{i-\nu+1}}{x_{i-j+1} - x_{i-\nu+1}}\,\mathrm{d}s \\ &= h\int_{1-l}^{1} \prod_{\nu\neq j} \frac{x_0 + (i+u)h - (x_0 + (i-\nu+1)h)}{x_0 + (i-j+1)h - (x_0 + (i-\nu+1)h)}\,\mathrm{d}u \\ &= h\int_{1-l}^{1} \prod_{\nu\neq j} \frac{u+\nu-1}{\nu-j}\,\mathrm{d}u.\end{aligned}$$

It follows a method

$$y_{i+1} = y_{i-l+1} + h\sum_{j=1}^{k} \beta_j f(x_{i-j+1}, y_{i-j+1})$$

with the constant coefficents

$$\beta_j := \int_{1-l}^{1} \prod_{\nu=1,\nu\neq j}^{k} \frac{u+\nu-1}{\nu-j}\,\mathrm{d}u \qquad \text{for } j = 1,\ldots,k.$$

An implicit multistep method results, if we include the unknown new approximation (x_{i+1}, y_{i+1}) in the interpolation. Let $q_{k,i} \in \mathbb{P}_k$ be the interpolating polynomial of the data

$$(x_j, f(x_j, y_j)) \qquad \text{for } j = i-k+1, i-k+2, \ldots, i-1, i, i+1.$$

It follows

$$q_{k,i}(x_j) = f(x_j, y_j) \qquad \text{for } \; j = i-k+1, i-k+2, \ldots, i-1, i, i+1.$$

The corresponding Lagrange polynomials become

$$L^*_{i,j}(x) = \prod_{\nu=0, \nu \neq j}^{k} \frac{x - x_{i-\nu+1}}{x_{i-j+1} - x_{i-\nu+1}} \qquad \text{for } \; j = 0, 1, \ldots, k$$

and thus

$$q_{k,i}(x) = \sum_{j=0}^{k} f_{i-j+1} L^*_{i,j}(x).$$

We write $q_{k,i}(x; y_{i+1})$ to emphasize that the polynomial depends on the new approximation, which is unknown a priori. We obtain

$$y_{i+1} = y_{i-l+1} + \int_{x_{i-l+1}}^{x_{i+1}} q_{k,i}(s; y_{i+1}) \, \mathrm{d}s.$$

This relation represents a nonlinear equation for the unknown y_{i+1}. Hence this approach yields an implicit method with k steps.

In case of equidistant step sizes, the method reads

$$y_{i+1} = y_{i-l+1} + h \sum_{j=0}^{k} \beta^*_j f(x_{i-j+1}, y_{i-j+1})$$

with corresponding coefficients

$$\beta^*_j := \int_{1-l}^{1} \prod_{\nu=0, \nu \neq j}^{k} \frac{u + \nu - 1}{\nu - j} \, \mathrm{d}u \qquad \text{for } \; j = 0, 1, \ldots, k.$$

Equivalently, we can write

$$y_{i+1} - h\beta^*_0 f(x_{i+1}, y_{i+1}) = y_{i-l+1} + h \sum_{j=1}^{k} \beta^*_j f(x_{i-j+1}, y_{i-j+1}),$$

where the right-hand side involves known data and the left-hand side contains the unknown new approximation.

	β_1	β_2	β_3	β_4
$k=1$	1			
$k=2$	$\frac{3}{2}$	$-\frac{1}{2}$		
$k=3$	$\frac{23}{12}$	$-\frac{16}{12}$	$\frac{5}{12}$	
$k=4$	$\frac{55}{24}$	$-\frac{59}{24}$	$\frac{37}{24}$	$-\frac{9}{24}$

	β_0^*	β_1^*	β_2^*	β_3^*	β_4^*
$k=1$	$\frac{1}{2}$	$\frac{1}{2}$			
$k=2$	$\frac{5}{12}$	$\frac{8}{12}$	$-\frac{1}{12}$		
$k=3$	$\frac{9}{24}$	$\frac{19}{24}$	$-\frac{5}{24}$	$\frac{1}{24}$	
$k=4$	$\frac{251}{720}$	$\frac{646}{720}$	$-\frac{264}{720}$	$\frac{106}{720}$	$-\frac{19}{720}$

Table 2: Coefficients in Adams-Bashforth (left) and Adams-Moulton (right).

Adams methods

A popular family of multistep techniques are the Adams methods, which result from the choice $l = 1$ in (4.2). Hence the integration is just done in the subinterval $[x_i, x_{i+1}]$.

The explicit schemes are the Adams-Bashforth methods. These k-step methods read

$$y_{i+1} = y_i + h \sum_{j=1}^{k} \beta_j f(x_{i-j+1}, y_{i-j+1}) \tag{4.3}$$

in case of equidistant step sizes. The implicit schemes are the Adams-Moulton methods. The k-step scheme exhibits the formula

$$y_{i+1} = y_i + h \sum_{j=0}^{k} \beta_j^* f(x_{i-j+1}, y_{i-j+1}). \tag{4.4}$$

Table 2 shows the coefficients of these methods in the cases $k = 1, 2, 3, 4$. The one-step Adams-Bashforth method coincides with the explicit Euler scheme, whereas the one-step Adams-Moulton method yields the trapezoidal rule.

Nyström methods and Milne methods

We obtain further important multistep schemes by the choice $l = 2$ in (4.2). The corresponding explicit techniques are called Nyström methods. For example, the selection $k = 1$ (now $k < l$) yields the explicit midpoint rule

$$y_{i+1} = y_{i-1} + 2hf(x_i, y_i), \tag{4.5}$$

which is a two-step method. Alternatively, the implicit techniques are called Milne methods. For equidistant step sizes, the case $k = 1$ results in the explicit midpoint rule again, i.e., the term f_{i+1} cancels out. The choice $k = 2$ yields the Milne-Simpson rule

$$y_{i+1} = y_{i-1} + h\tfrac{1}{3}\left(f(x_{i-1}, y_{i-1}) + 4f(x_i, y_i) + f(x_{i+1}, y_{i+1})\right),$$

which represents an implicit scheme. This method agrees to the Simpson rule applied in numerical quadrature.

Remark that choices $l \geq 3$ in (4.2) are not important in practice. Moreover, the number of steps ($\max\{k, l\}$) is often not larger than 5 in corresponding software packages.

4.2 Linear difference schemes

We consider a scalar ODE and equidistant step sizes for simplicity. The multistep methods from the previous section represent specific cases of linear multistep schemes

$$\sum_{l=0}^{k} \alpha_l y_{i+l} = h \sum_{l=0}^{k} \beta_l f(x_{i+l}, y_{i+l}). \tag{4.6}$$

Remark that the ordering of the coefficients is opposite to Sect. 4.1, since y_{i+k} represents the new approximation now. It holds $\alpha_k \neq 0$, whereas $\alpha_0 = 0$ is feasible, see the Adams methods with $k > 1$, for example. A general (nonlinear) multistep scheme reads

$$\sum_{l=0}^{n} a_l \; y_{i+l} = hF(x_i, y_{i-m}, \ldots, y_{i+n}) \tag{4.7}$$

with a function F depending also on the right-hand side f of the system of ODEs (2.1). We assume $a_0, a_n \neq 0$ in (4.7). The integers n, m are determined by the method (n is not the dimension of the ODE system here). To analyse the stability of a multistep method, it is sufficient to investigate the linear difference scheme in the left-hand side of (4.7).

We apply complex numbers in the following. A linear difference equation of order n reads

$$L(u_j) :\equiv \sum_{s=0}^{n} a_s \, u_{j+s} = c_{j+n} \qquad \text{for } \; j = 0, 1, 2, \dots \; , \tag{4.8}$$

where the coefficients $a_0, \dots, a_n \in \mathbb{C}$ and $c_i \in \mathbb{C}$ for $i > n-1$ are arbitrary except for the assumption $a_0, a_n \neq 0$. The mapping L is called a difference operator. We want to determine sequences $(u_i)_{i \in \mathbb{N}_0} \subset \mathbb{C}$, which satisfy the difference equation (4.8). An initial condition

$$u_i = v_i \qquad \text{for } \; i = 0, \dots, n-1 \tag{4.9}$$

with predetermined values $v_0, v_1, \dots, v_{n-1} \in \mathbb{C}$ is required. The solution of the initial value problem (4.8),(4.9) results to

$$u_{j+n} = \frac{1}{a_n} \left(- \sum_{s=0}^{n-1} a_s \, u_{j+s} + c_{j+n} \right) \qquad \text{for } \; j = 0, 1, 2, \dots \tag{4.10}$$

and can be computed successively.

The homogeneous difference equation corresponding to (4.8) is

$$L(u_j) = 0 \qquad \text{for } \; j = 0, 1, 2, \dots \; . \tag{4.11}$$

The solution of an initial value problem is given by (4.10) with $c_i = 0$ for all i. Since the operator L is linear: $L(\alpha u_j^{(1)} + \beta u_j^{(2)}) = \alpha L(u_j^{(1)}) + \beta L(u_j^{(2)})$, the solutions form a linear space.

Definition 4 *The sequences $(u_i^{(\nu)})_{i \in \mathbb{N}_0} \subset \mathbb{C}$ for $\nu = 1, \dots, r$ are called linear independent, if the relation*

$$\sum_{\nu=1}^{r} \alpha_\nu \, u_i^{(\nu)} = 0 \qquad \textit{for all } \; i \in \mathbb{N}_0$$

implies $\alpha_\nu = 0$ for all $\nu = 1, \dots, r$. A set of n linear independent solutions of the homogeneous difference equation (4.11) is a fundamental system.

Theorem 6 *Let $(u_i^{(\nu)})_{i\in\mathbb{N}_0}$ for $\nu = 1,\dots,n$ be a fundamental system of (4.11). Then each solution $(v_i)_{i\in\mathbb{N}_0}$ of (4.11) exhibits a unique representation*

$$v_i = \sum_{\nu=1}^{n} \alpha_\nu u_i^{(\nu)}$$

with coefficients $\alpha_1,\dots,\alpha_n \in \mathbb{C}$.

Proof:

Since the elements of the fundamental system are linearly independent, it follows that the n vectors $(u_0^{(\nu)},\dots,u_{n-1}^{(\nu)})^\top \in \mathbb{C}^n$ for $\nu = 1,\dots,n$ are linearly independent. Thus the matrix

$$A := \begin{pmatrix} u_0^{(1)} & \cdots & u_0^{(n)} \\ \vdots & & \vdots \\ u_{n-1}^{(1)} & \cdots & u_{n-1}^{(n)} \end{pmatrix} \in \mathbb{C}^{n\times n} \tag{4.12}$$

is regular. Let $v = (v_0,\dots,v_{n-1})^\top \in \mathbb{C}^n$. The linear system $Ax = v$ with $x = (\alpha_1,\dots,\alpha_n)^\top$ exhibits a unique solution, which represents the desired coefficients. Remark that initial value problems of (4.11) exhibit unique solutions. □

Now we show the existence of a fundamental system for the homogeneous difference equation (4.11) by construction.

Definition 5 *The polynomial*

$$p_n(x) := \sum_{s=0}^{n} a_s x^s = a_0 + a_1 x + \cdots + a_{n-1}x^{n-1} + a_n x^n \tag{4.13}$$

is called the characteristic polynomial of the difference operator L in (4.8).

Let $x_1,\dots,x_m \in \mathbb{C}$ be the pairwise different roots (zeros) of the characteristic polynomial with the multiplicities $r_1,\dots,r_m$:

$$p_n(x) = a_n(x-x_1)^{r_1}(x-x_2)^{r_2}\cdots(x-x_m)^{r_m}.$$

Thus it holds $r_1 + \cdots + r_m = n$. In particular, it follows

$$\frac{\mathrm{d}^k}{\mathrm{d}x^k}(p_n(x))\bigg|_{x=x_\mu} = 0 \qquad \text{for } k = 0, \ldots, r_\mu - 1 \tag{4.14}$$

and each $\mu = 1, \ldots, m$. The assumption $a_0 \neq 0$ implies $x_\mu \neq 0$ for each root.

Theorem 7 *A fundamental system of (4.11) is given by*

$$\begin{aligned}
(u_i^{(1,\mu)}) &:= (x_\mu{}^i) \\
(u_i^{(2,\mu)}) &:= (i x_\mu{}^i) \\
(u_i^{(3,\mu)}) &:= (i(i-1) x_\mu{}^i) \\
&\vdots \\
(u_i^{(r_\mu,\mu)}) &:= (i(i-1)\cdots(i-r_\mu+2) x_\mu{}^i)
\end{aligned}$$

for $\mu = 1, \ldots, m$.

Proof:

We insert a sequence of the set into the difference operator L

$$\begin{aligned}
L(u_i^{(k+1,\mu)}) &= \sum_{s=0}^{n} a_s u_{i+s}^{(k+1,\mu)} \\
&= \sum_{s=0}^{n} a_s (i+s)(i+s-1)\cdots(i+s-k+1) x_\mu^{i+s} \\
&= x_\mu^k \cdot \sum_{s=0}^{n} a_s (i+s)(i+s-1)\cdots(i+s-k+1) x_\mu^{i+s-k} \\
&= x_\mu^k \cdot \tfrac{\mathrm{d}^k}{\mathrm{d}x^k}(x^i p_n(x))\Big|_{x=x_\mu}
\end{aligned}$$

for $k = 0, 1, \ldots, r_\mu - 1$. The latter equality follows from

$$\begin{aligned}
\frac{\mathrm{d}^k}{\mathrm{d}x^k}\left(x^i \cdot \sum_{s=0}^{n} a_s x^s\right) &= \frac{\mathrm{d}^k}{\mathrm{d}x^k}\left(\sum_{s=0}^{n} a_s x^{i+s}\right) = \sum_{s=0}^{n} a_s \tfrac{\mathrm{d}^k}{\mathrm{d}x^k} x^{i+s} \\
&= \sum_{s=0}^{n} a_s (s+i)(s+i-1)\cdots(s+i-k+1) x^{i+s-k}.
\end{aligned}$$

The Leipniz rule yields

$$\frac{\mathrm{d}^k}{\mathrm{d}x^k}\left(x^i p_n(x)\right) = \sum_{l=0}^{k} \binom{k}{l} \cdot \left(\frac{\mathrm{d}^{k-l}}{\mathrm{d}x^{k-l}} x^i\right) \cdot \left(\frac{\mathrm{d}^l}{\mathrm{d}x^l} p_n(x)\right).$$

The property (4.14) yields $L(u_i^{(k+1,\mu)}) = 0$ for all i.

It remains to show that the n sequences are linearly independent. We observe the square matrix formed by the values $u_i^{(k+1,\mu)}$ for $i = 0, 1, \ldots, n-1$. The structure agrees to a Van-der-Monde matrix (cf. polynomial interpolation) and thus the matrix is regular. It follows that the system of sequences is linearly independent. □

We have shown the existence of a fundamental system $(u_i^{(\nu)})$ for $\nu = 1, \ldots, n$. We also achieve a standardised fundamental system $(w_i^{(\nu)})$ for $\nu = 1, \ldots, n$ characterised by the initial conditions

$$w_{i-1}^{(\nu)} = \begin{cases} 1 & \text{if } i = \nu, \\ 0 & \text{if } i \neq \nu, \end{cases} \qquad \text{for } i, \nu = 1, \ldots, n.$$

We obtain the standardised system via

$$w_i^{(\nu)} = \sum_{j=1}^{n} \alpha_j^{(\nu)} u_i^{(j)},$$

where the coefficients $x^{(\nu)} = (\alpha_1^{(\nu)}, \ldots, \alpha_n^{(\nu)})^\top$ follow from the linear system $Ax^{(\nu)} = e_\nu$ with the transformation matrix (4.12) and the νth unit vector $e_\nu = (0, \ldots, 0, 1, 0, \ldots, 0)^\top$.

Lemma 3 *Let $(u_i^{(\nu)})$ for $\nu = 0, 1, \ldots, n-1$ be the standardised fundamental system of the homogeneous difference equation (4.11). Then the solution (u_i) of the initial value problem (4.8), (4.9) of the inhomogeneous difference equation is given by*

$$u_i = \sum_{\nu=0}^{n-1} v_\nu u_i^{(\nu)} + \frac{1}{a_n} \sum_{k=0}^{i-n} c_{k+n} u_{i-k-1}^{(n-1)} \qquad \text{for } i = 0, 1, 2, \ldots \tag{4.15}$$

with the definitions $u_j^{(n-1)} = 0$ for $j < 0$ and $c_j = 0$ for $j < n$.

Proof:

The first sum in (4.15) satisfies the homogeneous difference equation (4.8) as well as the initial conditions (4.9). We have to show that the second sum fulfills the inhomogeneous difference equation (4.8) with initial values identical to zero. Let

$$w_i := \frac{1}{a_n} \sum_{k=0}^{i-n} c_{k+n} u_{i-k-1}^{(n-1)}.$$

Due to the definition $u_j^{(n-1)} = 0$ for $j < 0$ and $u_0^{(n-1)} = \cdots = u_{n-2}^{(n-1)} = 0$ as well as $c_j = 0$ for $j < n$, the initial values are clearly zero. Moreover, we can write

$$w_i = \frac{1}{a_n} \sum_{k=-\infty}^{+\infty} c_{k+n} u_{i-k-1}^{(n-1)} \qquad \text{for } \; i = 0, 1, 2, \dots \; ,$$

since all new terms are equal to zero. It follows

$$\begin{aligned} L(w_i) &= \sum_{s=0}^{n} a_s w_{i+s} = \frac{1}{a_n} \sum_{s=0}^{n} a_s \sum_{k=-\infty}^{+\infty} c_{k+n} u_{i-k-1+s}^{(n-1)} \\ &= \frac{1}{a_n} \sum_{k=-\infty}^{+\infty} c_{k+n} \sum_{s=0}^{n} a_s u_{i-k-1+s}^{(n-1)} = \frac{1}{a_n} \sum_{k=0}^{i} c_{k+n} \sum_{s=0}^{n} a_s u_{i-k-1+s}^{(n-1)} \\ &= \frac{1}{a_n} \sum_{k=0}^{i} c_{k+n} L(u_{i-k-1}^{(n-1)}). \end{aligned}$$

Due to $L(u_j^{(n-1)}) = 0$ for all $j \geq 0$ and $u_l^{(n-1)} = \delta_{l,n-1}$ for all $l \leq n-1$, we obtain

$$L(u_{i-k-1}^{(n-1)}) = a_n \delta_{ik} \qquad \text{for } \; k = 0, 1, \dots, i \; \text{ and } \; i = 0, 1, 2, \dots \; .$$

Inserting this relation in the equation above yields $L(w_i) = c_{i+n}$. □

Definition 6 *The linear difference scheme (4.8) is stable, if and only if the corresponding characteristic polynomial (4.13) satisfies the root condition:*

$$\begin{aligned} &(i) \quad |x_\mu| \leq 1 \quad \textit{for all simple roots } \; (r_\mu = 1), \\ &(ii) \quad |x_\mu| < 1 \quad \textit{for all multiple roots } \; (r_\mu > 1). \end{aligned}$$

A fundamental system $(u_i^{(\nu)})$ for $\nu = 1, \ldots, n$ is bounded, if it holds

$$|u_i^{(\nu)}| \leq C \qquad \text{for all } \; i \in \mathbb{N}_0 \; \text{ and all } \; \nu = 1, \ldots, n$$

with some constant $C > 0$. For the specific fundamental system from Theorem 7, it follows $C \geq 1$.

Lemma 4 *A fundamental system of the linear difference scheme (4.8) is bounded if and only if all fundamental systems are bounded.*

Proof:

Let $(u_i^{(\nu)})$ for $\nu = 1, \ldots, n$ be a bounded fundamental system. Given an arbitrary fundamental system $(v_i^{(\nu)})$ for $\nu = 1, \ldots, n$, it holds

$$v_i^{(j)} = \sum_{\nu=1}^{n} \alpha_{\nu,j} u_i^{(\nu)} \qquad \text{for } \; j = 1, \ldots, n$$

with unique coefficients $\alpha_{\nu,j} \in \mathbb{C}$ due to Theorem 6. It follows

$$|v_i^{(j)}| \leq \sum_{\nu=1}^{n} |\alpha_{\nu,j}| \cdot |u_i^{(\nu)}| \leq n \left(\max_{\nu,j} |\alpha_{\nu,j}| \right) \max_{\nu=1,\ldots,n} |u_i^{(\nu)}| \leq n \left(\max_{\nu,j} |\alpha_{\nu,j}| \right) C.$$

Thus the arbitrary fundamental system is bounded. □

We have introduced the root condition, because we need the following property.

Theorem 8 *A fundamental system of the linear difference scheme (4.8) is bounded if and only if the corresponding characteristic polynomial (4.13) fulfills the root condition.*

Proof:

Due to Lemma 4, it is sufficient to investigate the fundamental system $(u_i^{(\nu)})$ for $\nu = 1, \ldots, n$ from Theorem 7.

Let the root condition be satisfied. For simple roots with $|x_\mu| \leq 1$, it follows

$$\left|u_i^{(1,\mu)}\right| = |{x_\mu}^i| = |x_\mu|^i \leq 1$$

for each i. For multiple roots $|x_\mu| < 1$, we obtain the stronger relation

$$\lim_{i\to\infty} u_i^{(k+1,\mu)} = 0 \qquad \text{for } \; k = 0, 1, \ldots, r_\mu - 1 \; ,$$

since the terms exhibit the form $u_i = q(i){x_\mu}^i$ with polynomials q. In particular, the sequences are bounded.

Vice versa, assume that the root condition is violated. If a root exhibits $|x_\mu| > 1$, then it follows

$$\left|u_i^{(1,\mu)}\right| = |{x_\mu}^i| = |x_\mu|^i \to \infty.$$

In case of a multiple root ($r_\mu \geq 2$) with $|x_\mu| = 1$, we obtain

$$\left|u_i^{(2,\mu)}\right| = |i{x_\mu}^i| = i \cdot |x_\mu|^i = i \to \infty.$$

In both cases, the fundamental system becomes unbounded. □

Stability often means the Lipschitz-continuous dependence on perturbations in the initial data. In case of a homogeneous linear difference equation (4.11) and initial values (4.9) zero, the solution becomes identical to zero. Initial values not equal to zero can be seen as a perturbation of this solution. Let $(u_i^{(\nu)})$ for $\nu = 0, 1, \ldots, n-1$ be the standardised fundamental system. If and only if the root condition is satisfied, then this system is bounded, i.e., $|u_i^{(\nu)}| \leq C$ with a constant $C > 0$. For initial values $v_0, v_1, \ldots, v_{n-1} \in \mathbb{C}$, the corresponding solution becomes

$$v_i = \sum_{\nu=0}^{n-1} v_\nu u_i^{(\nu)}.$$

It follows

$$|v_i| \leq \sum_{\nu=0}^{n-1} |v_\nu| \cdot |u_i^{(\nu)}| \leq C \sum_{\nu=0}^{n-1} |v_\nu|.$$

Thus the solution (v_i) depends Lipschitz-continuously on the perturbations $v_0, v_1, \ldots, v_{n-1}$.

Now consider two solutions (v_i) and (w_i) of the inhomogeneous linear difference equation (4.8). It holds

$$L(v_i - w_i) = L(v_i) - L(w_i) = c_{i+n} - c_{i+n} = 0,$$

i.e., the difference solves the homogeneous equation (4.11). Thus we can represent the difference by the standardised fundamental system

$$v_i - w_i = \sum_{\nu=0}^{n-1} (v_\nu - w_\nu) u_i^{(\nu)}.$$

It follows

$$|v_i - w_i| \leq C \sum_{\nu=0}^{n-1} |v_\nu - w_\nu|$$

for each i. We recognise the Lipschitz-continuous dependence on the initial data again.

4.3 Consistency, stability and convergence

We consider an initial value problem (2.1) of a scalar ODE. We apply an equidistant grid

$$x_i = x_0 + ih \quad \text{for } i = 0, 1, \ldots, N \quad \text{with } h := \frac{x_{\text{end}} - x_0}{N}.$$

Let $y_i := y(x_i)$ be the values of the exact solution, whereas u_i denotes the numerical approximations. Now we define a local discretisation error of a multistep method. The scheme (4.7) can be written in the form

$$\frac{1}{h} \sum_{s=0}^{n} a_s u_{i+s} - F(x_i, u_{i-m}, \ldots, u_{i+n}) = 0.$$

Inserting the exact solution $y(x)$ in this formula yields a defect, which is the local discretisation error.

Definition 7 (local discretisation error of multistep methods)
Let $y(x)$ be the exact solution of the ODE-IVP $y' = f(x,y),\ y(x_0) = y_0$. The local discretisation error of the multistep method (4.7) is defined as the defect

$$\tau(h) := \frac{1}{h}\sum_{s=0}^{n} a_s y(x_{i+s}) - F(x_i, y(x_{i-m}), \ldots, y(x_{i+n})). \tag{4.16}$$

This definition agrees to the local error of one-step methods, cf. (3.7).

For example, we consider an explicit linear multistep method (4.6). The approximation becomes ($\beta_k = 0$)

$$\alpha_k u_{i+k} + \sum_{l=0}^{k-1} \alpha_l u_{i+l} = h \sum_{l=0}^{k-1} \beta_l f(x_{i+l}, u_{i+l}).$$

If the involved initial values are exact ($u_{i+l} = y(x_{i+l})$ for $l = 0, \ldots, k-1$), then it holds

$$\alpha_k u_{i+k} + \sum_{l=0}^{k-1} \alpha_l y(x_{i+l}) = h \sum_{l=0}^{k-1} \beta_l f(x_{i+l}, y(x_{i+l})).$$

The exact solution satisfies

$$\alpha_k y(x_{i+k}) + \sum_{l=0}^{k-1} \alpha_l y(x_{i+l}) = h \sum_{l=0}^{k-1} \beta_l f(x_{i+l}, y(x_{i+l})) + h \cdot \tau(h).$$

It follows

$$\tau(h) = \frac{\alpha_k}{h}(y(x_{i+k}) - u_{i+k}).$$

The linear multistep method (4.6) can be normalized by setting $\alpha_k := 1$.

Again we define a consistency according to Def. 2.

Definition 8 (consistency of a multistep method)
The multistep method (4.7) is consistent if the local discretisation error from (4.16) satisfies

$$\lim_{h \to 0} \tau(h) = 0$$

uniformly in x, y. The method is consistent of (at least) order p, if it holds $\tau(h) = \mathcal{O}(h^p)$.

In our discussion, we include errors in the initial values as well as roundoff errors in each step of the method now. We always have these errors in practice. Thus we ask if the final approximation is still convergent in the presence of the errors. We consider an interval $[x_0, x_{\text{end}}]$ and equidistant step sizes $h = \frac{x_{\text{end}} - x_0}{N}$. The multistep method (4.7) becomes

$$\begin{aligned} u_i &= y_i + \rho_i \qquad \text{for } i = 0, 1, \ldots, m+n-1 \,, \\ \sum_{s=0}^{n} a_s u_{i+s} &= hF(x_i, u_{i-m}, \ldots, u_{i+n}) + h\rho_{i+n} \\ &\qquad \text{for } i = m, m+1, \ldots, N-n \end{aligned} \tag{4.17}$$

with the errors $\rho_0, \ldots, \rho_N$ and the exact solution $y_i = y(x_i)$.

According to (4.16), the local discretisation error exhibits the form

$$\tau_{i+n} = \frac{1}{h} \sum_{s=0}^{n} a_s y_{i+s} - F(x_i, y_{i-m}, \ldots, y_{i+n}) \qquad \text{for } i = m, \ldots, N-n.$$

We make the following assumptions:

(i) It exists a function $\rho(h) \geq 0$ for $h \geq 0$ such that

$$|\rho_i| \leq \rho(h) \qquad \text{for all } i = 0, \ldots, N. \tag{4.18}$$

(ii) If the right-hand side of the ODE becomes $f(x, y) \equiv 0$, then it follows

$$F(x_i, u_{i-m}, \ldots, u_{i+n}) \equiv 0 \qquad \text{for all } i = m, \ldots, N-n$$

and all $h \geq 0$.

(iii) The function F is Lipschitz-continuous: For all $u, v \in \mathbb{R}^{m+n+1}$, it holds

$$|F(x_i, v_{i-m}, \ldots, v_{i+n}) - F(x_i, u_{i-m}, \ldots, u_{i+n})| \leq K \sum_{\nu=-m}^{n} |v_{i+\nu} - u_{i+\nu}| \tag{4.19}$$

for each $i = m, \ldots, N-n$, where the constant $K \geq 0$ depends just on the right-hand side f (and its derivatives).

(iv) It exists a function $\tau(h) \geq 0$ for $h \geq 0$ with

$$|\tau_{i+n}| \leq \tau(h) \qquad \text{for all } \; i = m, \ldots, N-n. \tag{4.20}$$

In case of a linear multistep method (4.6), the assumption (iii) is satisfied if the right-hand side f exhibits a Lipschitz-condition (2.3):

$$\begin{aligned}
|F(x_i, v) - F(x_i, u)| &= \left| \sum_{\nu=-m}^{n} \beta_\nu f(x_{i+\nu}, v_{i+\nu}) - \sum_{\nu=-m}^{n} \beta_\nu f(x_{i+\nu}, u_{i+\nu}) \right| \\
&\leq \sum_{\nu=-m}^{n} |\beta_\nu| \cdot |f(x_{i+\nu}, v_{i+\nu}) - f(x_{i+\nu}, u_{i+\nu})| \\
&\leq \sum_{\nu=-m}^{n} |\beta_\nu| \cdot L \cdot |v_{i+\nu} - u_{i+\nu}| \\
&\leq L \left(\max_{j=-m,\ldots,n} |\beta_j| \right) \sum_{\nu=-m}^{n} |v_{i+\nu} - u_{i+\nu}| \\
&= L \left(\max_{j=-m,\ldots,n} |\beta_j| \right) \|v - u\|_1.
\end{aligned}$$

According to Def. 8, the consistency of a multistep scheme implies the existence of a function $\tau(h)$ from assumption (iv) with

$$\lim_{h \to 0} \tau(h) = 0.$$

The convergence of the method is defined as follows. (The same definition can be done for one-step methods, if the influence of errors $\rho_0, \ldots, \rho_N$ is considered.)

Definition 9 (convergence of multistep method)
Assume that the function $\rho(h)$ from (4.18) satisfies

$$\lim_{h \to 0} \rho(h) = 0.$$

The multistep method (4.17) is convergent, if it holds ($h = \frac{x_{\text{end}} - x_0}{N}$)

$$\lim_{h \to 0} \left(\max_{i=0,\ldots,N} |u_i - y(x_i)| \right) = 0.$$

The method is convergent of (at least) order p if

$$\max_{i=0,\ldots,N} |u_i - y(x_i)| = \mathcal{O}(h^p)$$

holds provided that $\rho(h) = \mathcal{O}(h^p)$ is satisfied.

The following theorem (Dahlquist 1956) connects consistency and convergence. However, the stability of the involved difference scheme is required.

Theorem 9 (convergence of multistep methods)
Let the multistep method (4.7) be consistent with respect to the ODE-IVP $y' = f(x, y)$, $y(x_0) = y_0$. The method (4.7) is convergent if and only if the corresponding linear difference scheme is stable.

Proof:

1.) We assume that the root condition from Def. 6 is violated. We construct an example, which does not converge. Consider $f(x, y) \equiv 0$, which implies the solution $y(x) \equiv 0$ for the IVP $y(x_0) = 0$. It follows $F \equiv 0$ due to our assumption (iii).

Let $\xi \in \mathbb{C}$ be a simple root of $p_n(x)$ with $|\xi| > 1$ or a multiple root with $|\xi| \geq 1$. We define the perturbations in (4.17) via

$$\rho_i := \begin{cases} h(\xi^i + \overline{\xi}^i) & \text{if } |\xi| > 1 \\ hi(\xi^i + \overline{\xi}^i) & \text{if } |\xi| = 1 \end{cases} \qquad \text{for } i = 0, \ldots, n-1$$

and $\rho_i := 0$ for $i = n, \ldots, N$. Since $n > 0$ is a fixed integer, it holds $\rho(h) = \mathcal{O}(h)$. The multistep method (4.17) becomes

$$u_i = \rho_i \quad \text{for } i = 0, \ldots, n-1, \qquad \sum_{s=0}^{n} a_s u_{i+s} = 0 \quad \text{for } i = n, \ldots, N-n.$$

Due to our construction, the solution of this difference scheme is just

$$u_i = \begin{cases} h(\xi^i + \overline{\xi}^i) & \text{if } |\xi| > 1 \\ hi(\xi^i + \overline{\xi}^i) & \text{if } |\xi| = 1 \end{cases} \qquad \text{for } i = 0, \ldots, N.$$

Remark that $u_i \in \mathbb{R}$ for all i. It follows

$$u_N = (x_{\text{end}} - x_0) \cdot \begin{cases} \frac{1}{N}(\xi^N + \overline{\xi}^N) & \text{if } |\xi| > 1, \\ (\xi^N + \overline{\xi}^N) & \text{if } |\xi| = 1. \end{cases}$$

Due to $\xi = |\xi|\mathrm{e}^{\mathrm{i}\varphi}$, $\xi^j = |\xi|^j\mathrm{e}^{\mathrm{i}j\varphi}$, $\xi^j + \overline{\xi}^j = 2|\xi|^j\cos(j\varphi)$, it follows for the final approximation $\lim_{h\to 0} u_N \neq 0$. Hence the convergence is violated.

2.) Vice versa, we assume that the root condition from Def. 6 is satisfied now. The global errors are $e_i := u_i - y_i$. We define

$$c_{i+n} := h(F(x_i, u_{i-m}, \ldots, u_{i+n}) - F(x_i, y_{i-m}, \ldots, y_{i+n})) + h\rho_{i+n} - h\tau_{i+n}.$$

Subtraction of (4.7) and the relation of the exact solution (see (4.16)) yields

$$\begin{aligned} e_{m+k} &= \rho_{m+k} \qquad \text{for } k = 0, 1, \ldots, n-1\ , \\ \sum_{s=0}^{n} a_s e_{i+s} &= c_{i+n} \qquad \text{for } i = m, m+1, \ldots, N-n\ . \end{aligned} \tag{4.21}$$

According to Lemma 3, the solution of this difference scheme exhibits the form

$$e_{i+m} = \sum_{k=0}^{n-1} e_{m+k} u_i^{(k)} + \frac{1}{a_n} \sum_{k=0}^{i-n} c_{k+m+n} u_{i-k-1}^{(n-1)} \tag{4.22}$$

for $i = 0, \ldots, N-m$, where $(u_i^{(k)})$ for $k = 0, 1, \ldots, n-1$ is the standardised fundamental system corresponding to $L(u_j) = 0$. Since the root condition is assumed, the fundamental system is bounded:

$$|u_i^{(k)}| \leq Q \qquad \text{for } k = 0, 1, \ldots, n-1 \quad \text{and all } i \in \mathbb{N}_0$$

with a constant $Q \geq 1$. Due to the assumptions (4.18),(4.19) and (4.20), it follows

$$|c_{k+m+n}| \leq h\left(K \sum_{\nu=0}^{m+n} |e_{k+\nu}| + \rho(h) + \tau(h)\right).$$

Now we estimate (4.22)

$$\begin{aligned} |e_{i+m}| \;\leq\; & Qn \max_{k=0,\ldots,n-1} |e_{m+k}| \\ & + \frac{Q}{|a_n|}(i-n+1)h\left((m+n+1)K \max_{r=0,\ldots,i+m} |e_r| + \rho(h) + \tau(h)\right) \end{aligned}$$

for $i = n, \ldots, N-m$. The definition $w_i := \max\{|e_0|, |e_1|, \ldots, |e_i|\}$ implies

$$|e_{i+m}| \leq Qnw_{m+n-1} + \frac{Q}{|a_n|}(i-n+1)h\left((m+n+1)Kw_{i+m} + \rho(h) + \tau(h)\right).$$

Since $(w_i)_{i\in\mathbb{N}_0}$ is a monotone increasing sequence and $n \geq 1$, $Q \geq 1$, it follows

$$w_{i+m} \leq Qnw_{m+n-1} + \frac{Q}{|a_n|} ih \left((m+n+1)Kw_{i+m} + \rho(h) + \tau(h) \right)$$

for $i = 0, 1, \ldots, N-m$. For $i = 0, 1, \ldots, n-1$, this estimate is clearly valid, since the first term on the right-hand side is already an upper bound. We define the constant $\gamma := \frac{Q}{|a_n|} K(m+n+1)$. The previous result yields

$$(1 - \gamma ih) w_{i+m} \leq Qnw_{m+n-1} + \tfrac{Q}{|a_n|} ih(\rho(h) + \tau(h))$$

for $i = 0, \ldots, N-m$. The condition $\gamma ih \leq \frac{1}{2}$ implies $1 - \gamma ih \geq \frac{1}{2}$. Since $|e_{i+m}| \leq w_{i+m}$ and $w_{m+n-1} \leq \rho(h)$ holds, it follows

$$|e_{i+m}| \leq 2Q \left(n\rho(h) + \frac{\rho(h) + \tau(h)}{2\gamma |a_n|} \right) \qquad \text{for } \ 0 \leq i \leq \frac{1}{2\gamma h}. \tag{4.23}$$

The restriction on i is equivalent to $x_0 \leq x_i \leq x_0 + \frac{1}{2\gamma}$. Hence the convergence is given in the interval $[x_0, x_0 + \frac{1}{2\gamma}]$.

The same estimate holds in an arbitrary interval $[\hat{x}, \hat{x} + \frac{1}{2\gamma}] \subset [x_0, x_{\text{end}}]$ given corresponding initial errors. In particular, the estimate (4.23) is valid in the smaller interval $[x_0, x_0 + \frac{1}{4\gamma}]$. The final values of this interval can be seen as initial values for the next interval $[x_0 + \frac{1}{4\gamma}, x_0 + \frac{2}{4\gamma}]$. The new initial errors are bounded and converge to zero for $h \to 0$ due to (4.23). Successively, the intervals

$$[x_0, x_0 + \tfrac{1}{4\gamma}], [x_0 + \tfrac{1}{4\gamma}, x_0 + \tfrac{2}{4\gamma}], [x_0 + \tfrac{2}{4\gamma}, x_0 + \tfrac{3}{4\gamma}], \ldots, [x_0 + \tfrac{R}{4\gamma}, x_{\text{end}}]$$

are considered, where R is the largest integer below the value $4\gamma(x_{\text{end}} - x_0)$. The inequality (4.23) shows $|e_{i+m}| \leq C_1\rho(h) + C_2\tau(h)$ with constants C_1, C_2. The constants are independent of the position within the global interval $[x_0, x_{\text{end}}]$. W.l.o.g. let $C_1 > 1$, i.e., $C_1 \neq 1$. By induction, it follows

$$|e_j| \leq C_1^{R+1}\rho(h) + C_2\tau(h) \sum_{l=0}^{R} C_1^l = C_1^{R+1}\rho(h) + C_2 \frac{1 - C_1^{R+1}}{1 - C_1} \tau(h) \tag{4.24}$$

for $j = 0, 1, \ldots, N$. Hence the convergence is fulfilled in the global interval $[x_0, x_{\text{end}}]$, which concludes the second part of the proof. $\square$

Remarks:

- The first part of the proof does not apply the consistency of the method. Indeed, the consistency is not necessary for the convergence, whereas the stability is necessary for the convergence. There exist methods, which are convergent but not consistent. However, a convergent linear multistep method (4.6) can be shown to be consistent.
- If $\tau(h) = \mathcal{O}(h^p)$ and $\rho(h) = \mathcal{O}(h^p)$ holds, then the estimate (4.24) yields $|e_i| = \mathcal{O}(h^p)$ for $i = 0, \ldots, N$. Thus consistency of (at least) order p implies the convergence of (at least) order p.
- Theorem 9 also holds in case of systems of ODEs $y' = f(x, y)$ with $y : [x_0, x_{\text{end}}] \to \mathbb{R}^n$. The required modifications in the above proof are straightforward. A corresponding proof can be found in the book of Stoer/Bulirsch.
- The statement of Theorem 9 cannot be generalised directly to the case of non-constant step sizes $h_i := x_{i+1} - x_i$. The step size selection has to fulfill certain properties, which still guarantee a kind of stability. Thus local error control, see Sect. 3.7, is more critical in case of multistep methods.

4.4 Analysis of multistep methods

Given some linear multistep method (4.6), we like to know if the method is convergent and (if yes) the corresponding order. Of course this property depends on the choice of the coefficients, which are the degree of freedom here.

In the previous section, we recognised that a linear multistep method is convergent (of order p), if and only if the method is stable and consistent (of order p). Thus to verify the convergence of a linear multistep method, two properties have to be checked:

- *Stability of the method*: Does the corresponding difference scheme satisfy the root condition (see Def. 6)? This can be verified straightforward by determining the roots of the characteristic polynomial.
- *Consistency of the method*: We require conditions to confirm the consistency and to detect the corresponding order. Corresponding formulas can be obtained similar to the procedure used in Sect. 3.5 for one-step methods, which yields order conditions for the coefficients.

One-step methods

A general (explicit) one-step method can be written in the form

$$y_1 = y_0 + h\Phi(x_0, y_0, h),$$

where the function Φ depends on the right-hand side f. The corresponding homogeneous linear difference scheme reads $y_1 - y_0 = 0$. The characteristic polynomial becomes $p(\lambda) = \lambda - 1$. Just the simple root $\lambda_1 = 1$ appears. Hence a one-step method of this form always satisfies the root condition, i.e., it is stable.

Stability of methods based on quadrature

As an example, we verify the stability of the linear k-step methods introduced in Sect. 4.1. The schemes exhibit the form (4.6) with

$$y_{i+k} - y_{i+r} = h\left[\beta_0 f_i + \beta_1 f_{i+1} + \cdots + \beta_{k-1} f_{i+k-1} + \beta_k f_{i+k}\right],$$

where the quadrature is done in the interval $[x_{i+r}, x_{i+k}]$ $(r < k)$ and the interpolation in the interval $[x_i, x_{i+k}]$. Let $n := k - r$. The characteristic polynomial (see Def. 5) becomes

$$p_n(\lambda) = \lambda^n - 1.$$

For the Adams methods, it holds $r = k - 1$ and $n = 1$. The characteristic polynomial just exhibits the simple root $\lambda = 1$. Thus the root condition is satisfied. For $n > 1$, the roots of the characteristic polynomial read

$$\lambda_j = \mathrm{e}^{\mathrm{i}2\pi\frac{j-1}{n}} \qquad \text{for } \; j = 1, 2, \ldots, n.$$

We obtain n simple roots satisfying $|\lambda_j| = 1$. Hence the root condition is fulfilled again.

Order conditions

Now we derive conditions for the consistency of a linear k-step method up to an arbitrary order p. The local discretisation error (4.16) can be written as

$$\tau(h) = \frac{1}{h}\left(\sum_{l=0}^{k} \alpha_l y(x+lh) - h\sum_{l=0}^{k} \beta_l y'(x+lh)\right). \tag{4.25}$$

Taylor expansion yields

$$\begin{aligned}
y(x+lh) &= \sum_{q=0}^{p} y^{(q)}(x)\cdot\frac{(lh)^q}{q!} + \mathcal{O}(h^{p+1}) \\
&= y(x) + \sum_{q=1}^{p} y^{(q)}(x)\cdot\frac{(lh)^q}{q!} + \mathcal{O}(h^{p+1}), \\
y'(x+lh) &= \sum_{q=0}^{p-1} y^{(q+1)}(x)\cdot\frac{(lh)^q}{q!} + \mathcal{O}(h^{p}) \\
&= \sum_{q=1}^{p} y^{(q)}(x)\cdot\frac{(lh)^{q-1}}{(q-1)!} + \mathcal{O}(h^{p}).
\end{aligned}$$

Inserting these expansions in the local error (4.25) results in

$$\begin{aligned}
\tau(h) &= \frac{1}{h}\left(\sum_{l=0}^{k}\alpha_l\left[y(x) + \sum_{q=1}^{p} y^{(q)}(x)\frac{(lh)^q}{q!} + \mathcal{O}(h^{p+1})\right]\right. \\
&\quad \left. + h\sum_{l=0}^{k}\beta_l\left[\sum_{q=1}^{p} y^{(q)}(x)\frac{(lh)^{q-1}}{(q-1)!} + \mathcal{O}(h^{p})\right]\right) \\
&= \frac{y(x)}{h}\sum_{l=0}^{k}\alpha_l + \frac{1}{h}\sum_{l=0}^{k}\left[\sum_{q=1}^{p}\frac{y^{(q)}(x)}{q!}(\alpha_l l^q h^q + q\beta_l l^{q-1}h^q)\right] + \mathcal{O}(h^p) \\
&= \frac{y(x)}{h}\sum_{l=0}^{k}\alpha_l + \sum_{q=1}^{p}\frac{y^{(q)}(x)}{q!}\left[\sum_{l=0}^{k}(\alpha_l l^q + q\beta_l l^{q-1})h^{q-1}\right] + \mathcal{O}(h^p).
\end{aligned}$$

We can read the order conditions from the above formula. For consistency of order $p = 1$, we need $\tau(h) = \mathcal{O}(h)$. It follows the conditions of order 1

$$\sum_{l=0}^{k} \alpha_l = 0 \qquad \text{and} \qquad \sum_{l=0}^{k} (\alpha_l l - \beta_l) = 0. \tag{4.26}$$

The additional conditions for order $p > 1$ become

$$\sum_{l=1}^{k} \alpha_l l^q = q \sum_{l=1}^{k} \beta_l l^{q-1} \qquad \text{for } \; q = 2, \ldots, p.$$

Remark that the first condition in (4.26) for consistency of order $p = 1$ is equivalent to $p_n(1) = 0$. Thus a consistent linear multistep method always implies the root $x = 1$ of the characteristic polynomial.

If a method is consistent of exactly order p (i.e. it holds $\tau(h) = \mathcal{O}(h^p)$, $\tau(h) \neq \mathcal{O}(h^{p+1})$), then the local error exhibits the form

$$\tau(h) = h^p y^{(p+1)}(x) \frac{1}{(p+1)!} \left[\sum_{l=1}^{k} (\alpha_l l^{p+1} - (p+1)\beta_l l^p) \right] + \mathcal{O}(h^{p+1}).$$

Thus this error depends on the magnitude of a higher-order derivative of the solution.

Example: Adams-Moulton methods

We determine the order of consistency for the first two Adams-Moulton methods. The coefficients can be found in Tab. 2.

The first method is the trapezoidal rule

$$-y_i + y_{i+1} = h \left[\tfrac{1}{2} f_i + \tfrac{1}{2} f_{i+1} \right].$$

The involved coefficients are $\alpha_0 = -1$, $\alpha_1 = 1$, $\beta_0 = \beta_1 = \frac{1}{2}$. It follows

$$\sum_{l=0}^{1} \alpha_l = -1 + 1 = 0$$

and

$$\sum_{l=0}^{1} (\alpha_l l - \beta_l) = (-1) \cdot 0 - \tfrac{1}{2} + 1 \cdot 1 - \tfrac{1}{2} = 0.$$

Thus the order of the method satisfies $p \geq 1$. The condition for $p = 2$ becomes

$$\sum_{l=1}^{1}(\alpha_l l^2 - 2\beta_l l^1) = 1 \cdot 1^2 - 2 \cdot \tfrac{1}{2} \cdot 1 = 0.$$

It follows $p \geq 2$. The condition for $p = 3$ is violated due to

$$\sum_{l=1}^{1}(\alpha_l l^3 - 3\beta_l l^2) = 1 \cdot 1^3 - 3 \cdot \tfrac{1}{2} \cdot 1^2 = -\tfrac{1}{2} \neq 0.$$

The trapezoidal rule is consistent of the exact order $p = 2$.

The second method reads

$$-y_{i+1} + y_{i+2} = h\left[-\tfrac{1}{12}f_i + \tfrac{8}{12}f_{i+1} + \tfrac{5}{12}f_{i+2}\right].$$

The coefficients are $\alpha_0 = 0$, $\alpha_1 = -1$, $\alpha_2 = 1$, $\beta_0 = -\frac{1}{12}$, $\beta_1 = \frac{8}{12}$, $\beta_2 = \frac{5}{12}$. The conditions of order $p = 1$

$$\sum_{l=0}^{2}\alpha_l = 0 + (-1) + 1 = 0$$

and

$$\sum_{l=0}^{2}(\alpha_l l - \beta_l) = 0 \cdot 0 - (-\tfrac{1}{12}) + (-1) \cdot 1 - \tfrac{8}{12} + 1 \cdot 2 - \tfrac{5}{12} = 0.$$

It follows the order $p \geq 1$. The condition of order $p = 2$ is verified via

$$\sum_{l=1}^{2}(\alpha_l l^2 - 2\beta_l l^1) = (-1) \cdot 1^2 - 2 \cdot \tfrac{8}{12} \cdot 1^1 + 1 \cdot 2^2 - 2\tfrac{5}{12} \cdot 2^1 = 0.$$

It follows the order $p \geq 2$. The condition of order $p = 3$ becomes

$$\sum_{l=1}^{2}(\alpha_l l^3 - 3\beta_l l^2) = (-1) \cdot 1^3 - 3 \cdot \tfrac{8}{12} \cdot 1^2 + 1 \cdot 2^3 - 3 \cdot \tfrac{5}{12} \cdot 2^2 = 0.$$

It follows $p \geq 3$. The condition or order $p = 4$ is violated:

$$\sum_{l=1}^{2}(\alpha_l l^4 - 4\beta_l l^3) = (-1) \cdot 1^4 - 4 \cdot \tfrac{8}{12} \cdot 1^3 + 1 \cdot 2^4 - 4 \cdot \tfrac{5}{12} \cdot 2^3 = -1 \neq 0.$$

Hence the method is exactly of the order $p = 3$. It can be shown that the k-step Adams-Moulton method is consistent of order $p = k + 1$ exactly.

It is natural to ask for the optimal order of convergence of a linear k-step scheme (4.6) for fixed k. Without loss of generality, we assume $\alpha_k = 1$. Thus we obtain $2k+1$ degrees of freedom by the coefficients $\alpha_0, \ldots, \alpha_{k-1}$ and $\beta_0, \ldots, \beta_k$. We can construct a method, which is consistent of order $p = 2k$, since $p + 1$ conditions have to be satisfied. However, our difference scheme has to be stable to achieve a convergent method. The root condition implies k contraints. A consistent scheme exhibits the root $\lambda = 1$, which already satisfies the root condition. Hence $k - 1$ contraints remain. We expect that the optimal order becomes $p \approx 2k - (k-1) = k+1$. The following theorem of Dahlquist (1956/59) presents the exact result.

Theorem 10 (first Dahlquist barrier) *A linear k-step method, which fulfills the stability condition, exhibits the maximum order*

$$\begin{array}{ll} k+2 & \textit{if } k \textit{ is even,} \\ k+1 & \textit{if } k \textit{ is odd,} \\ k & \textit{if } \beta_k/\alpha_k \leq 0 \textit{ (especially for explicit schemes).} \end{array}$$

For comparison, an implicit Runge-Kutta method with s stages exhibits $s^2 + s$ coefficients. (The nodes follow from the inner weights due to (3.11).) An explicit Runge-Kutta scheme has about $\frac{s^2}{2} + s$ degrees of freedom. No additional stability criterion has to be satisfied. The optimal order of convergence in case of fixed s becomes $p = 2s$ for implicit methods (Gauss-Runge-Kutta) and $p \leq s$ for explicit methods. Remark that the maximal order increases linearly with the stage number, whereas the number of coefficients grows quadratically.

Finally, we show a result mentioned in the previous subsection.

Theorem 11 *A convergent linear multistep method (4.6) is consistent.*

Proof:

We have to show the two conditions (4.26).

We consider the ODE-IVP $y' = 0$, $y(0) = 1$ with the exact solution $y(x) \equiv 1$. The linear multistep method becomes

$$\alpha_k u_{i+k} + \alpha_{k-1} u_{i+k-1} + \cdots + \alpha_1 u_{i+1} + \alpha_0 u_i = 0. \tag{4.27}$$

For $x = 1$ and $h_N := \frac{1}{N}$, the approximation at the point $x = 1$ is just u_N. The convergence of the method implies

$$\lim_{N \to \infty} u_N = y(1) = 1.$$

Thus the limit $i \to \infty$ in (4.27) yields (because k is constant)

$$\sum_{l=0}^{k} \alpha_l = \alpha_k + \alpha_{k-1} + \cdots + \alpha_1 + \alpha_0 = 0,$$

which is the first condition from (4.26).

Now we consider the ODE-IVP $y' = 1$, $y(0) = 0$, where the solution becomes $y(x) \equiv x$. Let $x = 1$ and $h_N := \frac{1}{N}$ We try the ansatz $u_i = i h_N K$ with some constant $K \in \mathbb{R}$. Inserting the ansatz in the linear difference scheme yields

$$\sum_{l=0}^{k} \alpha_l (i + l) h_N K = h_N \sum_{l=0}^{k} \beta_l.$$

The previous result implies

$$K \sum_{l=0}^{k} \alpha_l l = \sum_{l=0}^{k} \beta_l \qquad \Rightarrow \qquad K = \left(\sum_{l=0}^{k} \beta_l \right) / \left(\sum_{l=0}^{k} \alpha_l l \right). \tag{4.28}$$

Thus we have found a solution of the difference equation. Setting $i = N$ yields $u_N = K$. The convergence of the method ensures

$$K = \lim_{N\to\infty} u_N = y(1) = 1.$$

It remains to verify that the constant K exists. The characteristic polynomial of the method is

$$p(\lambda) = \sum_{l=0}^{k} \alpha_l \lambda^l, \qquad p'(\lambda) = \sum_{l=1}^{k} \alpha_l l \lambda^{l-1}.$$

The condition from above yields $p(1) = 0$. Since a convergent method is stable due to Theorem 9, the root $\lambda = 1$ is simple. It follows $p'(1) \neq 0$ and the denominator in (4.28) is not equal to zero. Hence the second condition from (4.26) is shown. □

4.5 Techniques based on numerical differentiation

We introduce another class of implicit multistep methods by using numerical differentiation.

BDF methods

Given the ODE $y' = f(x, y)$, we can replace the derivative on the left-hand side directly by a difference formula, which corresponds to a numerical differentiation. Using the difference quotient yields

$$y'(x_0 + h) = \tfrac{1}{h}\left[y(x_0 + h) - y(x_0)\right] + \mathcal{O}(h).$$

Together with $y'(x_0 + h) = f(x_0 + h, y(x_0 + h))$, we obtain the numerical method

$$y_1 = y_0 + hf(x_0 + h, y_1),$$

which is just the implicit Euler scheme.

This approach can be generalised to a k-step method as follows: Given the old data (x_{i-k+l}, y_{i-k+l}) for $l = 1, \ldots, k$, we arrange the interpolating

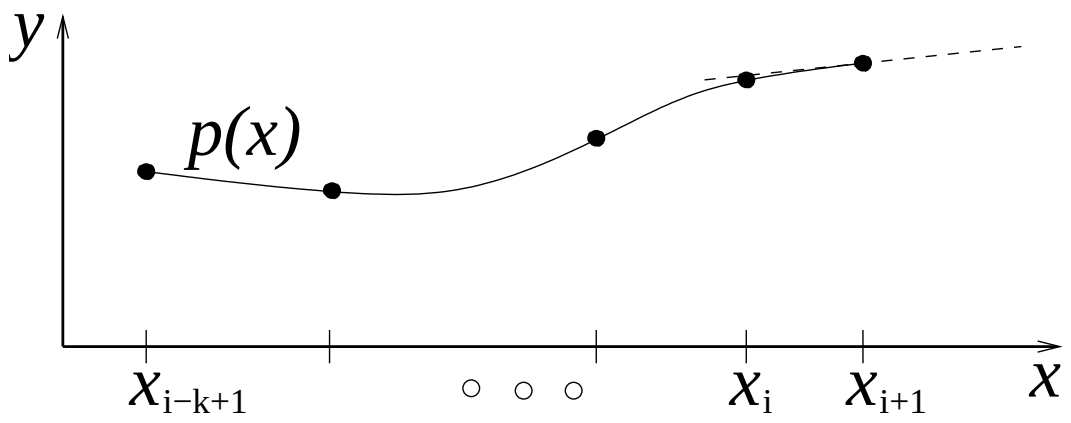

Figure 12: Construction of multistep method by numerical differentiation.

polynomial $p \in \mathbb{P}_k$ with

$$p(x_{i-k+l}) = y_{i-k+l} \qquad \text{for } \; l = 1, \ldots, k, k+1.$$

Thereby, the unknown value y_{i+1} is included in the interpolation, which makes the method implicit. The strategy is outlined in Fig. 12. The unknown value is determined by the demand

$$p'(x_{i+1}) = f(x_{i+1}, y_{i+1}),$$

which corresponds to a collocation method. The resulting techniques are called *backward differentiation formulas* (*BDF*).

The interpolating polynomial exhibits the form

$$p(x) = \sum_{j=0}^{k} y_{i+1-j} L_j(x)$$

with the Lagrange polynomials

$$L_j(x) = \prod_{\nu=0, \nu \neq j}^{k} \frac{x - x_{i+1-\nu}}{x_{i+1-j} - x_{i+1-\nu}}.$$

We obtain

$$p'(x_{i+1}) = \sum_{j=0}^{k} y_{i+1-j} L_j'(x_{i+1}) = f(x_{i+1}, y_{i+1}).$$

	α_0	α_1	α_2	α_3	α_4
$k=1$	1	-1			
$k=2$	$\frac{3}{2}$	-2	$\frac{1}{2}$		
$k=3$	$\frac{11}{6}$	-3	$\frac{3}{2}$	$-\frac{1}{3}$	
$k=4$	$\frac{25}{12}$	-4	3	$-\frac{4}{3}$	$\frac{1}{4}$

Table 3: Coefficients in BDF method.

In case of equidistant step sizes ($x_l = x_0 + lh$), the Lagrange polynomials can be transformed to

$$\tilde{L}_j(u) = \prod_{\nu=0,\nu\neq j}^{k} \frac{u+\nu-1}{\nu-j} \qquad \text{with } x = x_i + uh.$$

The new polynomials are independent of the index i. The resulting k-step method exhibits the form

$$\alpha_0 y_{i+1} + \alpha_1 y_i + \cdots + \alpha_{k-1} y_{i-k+2} + \alpha_k y_{i-k+1} = h f(x_{i+1}, y_{i+1}) \tag{4.29}$$

with constant coefficients

$$\alpha_j = \tilde{L}_j'(1) \qquad \text{for } j = 0, \ldots, k.$$

(Recall that it holds $\mathrm{d}x = h\mathrm{d}u$.) Table 3 illustrates the coefficients of the first four BDF methods.

Remark that all coefficients are determined by the approach based on the polynomial interpolation and the collocation technique. We do not have further degrees of freedom to satisfy the stability, i.e., the root condition. Fortunately, it turns out that the BDF methods are stable up to $k \leq 6$ (unstable for all $k \geq 7$).

Concerning consistency, it holds the following theorem.

Theorem 12 *The k-step BDF method is consistent of order k.*

Outline of the proof:

The polynomial p interpolates data corresponding to the solution $y(x)$. Let the data be exactly the values of the solution $y(x)$. Since $k+1$ points are

interpolated, the approximation exhibits an error

$$y(x) - p(x) = \mathcal{O}(|h|^{k+1}) \qquad \text{for } x \in [x_{i-k+1}, x_{i+1}],$$

where $|h|$ denotes the maximum of all involved step sizes. The derivative of the polynomial yields an approximation satisfying

$$y'(x) - p'(x) = \mathcal{O}(|h|^{k}) \qquad \text{for } x \in [x_{i-k+1}, x_{i+1}].$$

It follows

$$p'(x_{i+1}) = y'(x_{i+1}) + \mathcal{O}(|h|^k) = f(x_{i+1}, y(x_{i+1})) + \mathcal{O}(|h|^k)$$

and thus the local error becomes

$$\tau = \left[\sum_{j=0}^{k} y(x_{i+1-j}) L_j'(x_{i+1})\right] - f(x_{i+1}, y(x_{i+1})) = \mathcal{O}(|h|^k).$$

This property represents the consistency of order k. □

Hence the k-step BDF method is convergent of order k provided that $k \leq 6$.

NDF methods

The formula (4.29) of the BDF method can be modified to

$$\sum_{l=0}^{k} \alpha_l y_{i+1-l} = h f(x_{i+1}, y_{i+1}) + \kappa \gamma_k \left(y_{i+1} - y_{i+1}^{(0)}\right) \tag{4.30}$$

with an arbitrary constant $\kappa \in \mathbb{R}$ and

$$\gamma_k = \sum_{j=1}^{k} \frac{1}{j}.$$

The starting value $y_{i+1}^{(0)}$ is obtained by interpolating the $k+1$ old values $y_{i-k}, \ldots, y_i$ and evaluating the polynomial at $x = x_{i+1}$. It follows

$$y(x_{i+1}) - y_{i+1}^{(0)} = \mathcal{O}(h^{k+1})$$

and thus the method (4.30) exhibits the order of consistency k. Techniques of the form (4.30) are called *numerical differentiation formulas* (*NDF*). The first-order NDF method is

$$y_{i+1} - y_i - \kappa(y_{i+1} - 2y_i + y_{i-1}) = hf(x_{i+1}, y_{i+1}),$$

which is already a two-step technique. Likewise, the kth-order NDF scheme is a $(k+1)$-step method.

In the formulas (4.30), the parameter κ can be chosen such that the leading term in the local error becomes minimal, while still good stability properties are preserved with respect to stiff ODEs. The optimal value is $\kappa = -\frac{1}{9}$ in case of $k = 2$. It follows that for the same accuracy the step sizes can be selected about 26% larger than in the corresponding BDF2 method. The methods are stable for $k \leq 5$. However, the stability properties become slightly worse than in case of the BDF methods.

4.6 Predictor-Corrector-Methods

We consider IVPs of systems of ODEs $y' = f(x, y)$, $y(x_0) = y_0$. In this subsection, we discuss the solution of nonlinear systems of algebraic equations, which result from implicit multistep methods. A linear k-step method reads

$$y_{i+1} - h\beta_0 f(x_{i+1}, y_{i+1}) = h\sum_{l=1}^{k} \beta_l f(x_{i+1-l}, y_{i+1-l}) - \sum_{l=1}^{k} \alpha_l y_{i+1-l}. \quad (4.31)$$

The equations (4.31) represent a system of n algebraic equations for the unknown values $y_{i+1} \in \mathbb{R}^n$. The right-hand side

$$w_i := h\sum_{l=1}^{k} \beta_l f(x_{i+1-l}, y_{i+1-l}) - \sum_{l=1}^{k} \alpha_l y_{i+1-l}$$

is given.

The nonlinear system

$$y_{i+1} - h\beta_0 f(x_{i+1}, y_{i+1}) - w_i = 0$$

can be solved numerically by the Newton method. We define the matrices $A^{(\nu)} \in \mathbb{R}^{n\times n}$

$$A^{(\nu)} := I - h\beta_0(\mathrm{D}f)(x_{i+1}, y_{i+1}^{(\nu)})$$

with the identity matrix $I \in \mathbb{R}^{n\times n}$ and the Jacobian matrix $\mathrm{D}f \in \mathbb{R}^{n\times n}$. The iteration reads

$$\begin{aligned} A^{(\nu)}\Delta y_{i+1}^{(\nu)} &= y_{i+1}^{(\nu)} - h\beta_0 f(x_{i+1}, y_{i+1}^{(\nu)}) - w_i \\ y_{i+1}^{(\nu+1)} &= y_{i+1}^{(\nu)} - \Delta y_{i+1}^{(\nu)} \end{aligned}$$

for $\nu = 0, 1, 2, \ldots$ with some starting value $y_{i+1}^{(0)} \in \mathbb{R}^n$. Thus we obtain a sequence of linear systems. In this situation, an appropriate starting value is $y_{i+1}^{(0)} = y_i$. We apply the simplified Newton method to save computational effort. The iteration becomes

$$\begin{aligned} A^{(0)}\Delta y_{i+1}^{(\nu)} &= y_{i+1}^{(\nu)} - h\beta_0 f(x_{i+1}, y_{i+1}^{(\nu)}) - w_i \\ y_{i+1}^{(\nu+1)} &= y_{i+1}^{(\nu)} - \Delta y_{i+1}^{(\nu)} \end{aligned} \tag{4.32}$$

for $\nu = 0, 1, 2, \ldots$. The speed of convergence is linear. The computational work of this simplified Newton iteration can be characterised as follows:

Start-up phase:

1. Compute the Jacobian matrix $\mathrm{D}f$ at $x = x_{i+1}$, $y = y_{i+1}^{(0)}$. If numerical differentiation is used, then n additional function evaluations of f are required.
2. Decompose $A^{(0)} = L \cdot U$ into lower triangular matrix L and upper triangular matrix U. The computational effort is $\sim n^3$.

In each step:

1. Evaluate f at $x = x_{i+1}$, $y = y_{i+1}^{(\nu)}$.
2. Solve the linear system in (4.32) using the LU-decomposition. The work for each forward and backward substitution is $\sim n^2$.

If step size control is used and the Newton iteration does not converge, then the step size $h_i = x_{i+1} - x_i$ is reduced. For example, the iteration is restarted with the new grid point $x_{i+1} = x_i + \frac{h_i}{2}$, since the available starting value $y_{i+1}^{(0)} = y_i$ becomes a better approximation due to the continuity of the exact solution.

We can apply an alternative strategy, which saves much more computational effort. The nonlinear system (4.31) can be written as a fixed point problem

$$y_{i+1} = \Phi(y_{i+1})$$

with the function

$$\Phi(y_{i+1}) = h\beta_0 f(x_{i+1}, y_{i+1}) + w_i.$$

Following Banach's theorem, the according fixed point iteration

$$y_{i+1}^{(\nu+1)} = \Phi(y_{i+1}^{(\nu)}) \qquad \text{for } \nu = 0, 1, 2, \ldots \tag{4.33}$$

is convergent, if the mapping Φ is contractive. In an arbitrary vector norm, it follows

$$\begin{aligned} \|\Phi(y) - \Phi(z)\| &= \|h\beta_0 f(x_{i+1}, y) + w_i - (h\beta_0 f(x_{i+1}, z) + w_i)\| \\ &= h \cdot |\beta_0| \cdot \|f(x_{i+1}, y) - f(x_{i+1}, z)\| \\ &\leq h \cdot |\beta_0| \cdot L \cdot \|y - z\| \end{aligned}$$

provided that the right-hand side satisfies the Lipschitz-condition (2.3) with constant $L > 0$. Consequently, the mapping Φ is contractive for

$$h \cdot |\beta_0| \cdot L < 1 \qquad \Leftrightarrow \qquad h < \frac{1}{|\beta_0| \cdot L}. \tag{4.34}$$

Thus we achieve a convergent fixed point iteration for sufficiently small step sizes. The speed of convergence is linear with constant $h|\beta_0|L$. The computational effort of each step (4.33) consists just in one evaluation of the right-hand side f. In particular, no linear systems have to be solved.

However, the contractivity condition (4.34) restricts the step size significantly in case of large constants L. Huge Lipschitz-constants L appear

in stiff systems of ODEs, which represent mathematical models in many applications. In these cases, the fixed point iteration (4.33) becomes useless, since a huge number of integration steps is required. In contrast, the Newton method still yields according approximations for much larger step sizes h.

Now we consider the implicit multistep method (4.31) for moderate constants L. The determination of the unknown y_{i+1} can be done by a *predictor-corrector-method*. The technique consists of two parts:

- *Predictor method*: A scheme that yields a good starting value.
- *Corrector method*: An iteration scheme converging to the a priori unknown value, where a constant number of iteration steps is done.

As an example, we consider the Adams-Moulton methods. The k-step (implicit) Adams-Moulton method (4.4) exhibits the order $k+1$, whereas the k-step (explicit) Adams-Bashforth method (4.3) is of order k. We choose the fixed point iteration (4.33) as corrector step. The k-step Adams-Bashforth method is used as predictor.

We denote the application of the predictor by P, a step of the corrector by C and a required evaluation of the right-hand side f by E (since the computational effort is specified by the number of function evaluations). Let $f_i := f(x_i, y_i)$. It follows a P(EC)^{m}E-method for a constant integer m. Table 4 specifies the algorithm. Usually just $m = 1$ or $m = 2$ is used, since more corrector steps do not increase the accuracy significantly.

In practice, the P(EC)^{m}E-method is used with variable step sizes, where the coefficients have to be recomputed in each step by divided differences (Newton interpolation). The difference

$$y_{i+1}^{(m)} - y_{i+1}^{(0)} = \mathcal{O}(h^{k+1})$$

yields the error estimate in the step size control, since $y_{i+1}^{(0)}$ is an approximation of order k and $y_{i+1}^{(m)}$ is an approximation of order $k+1$, see Sect. 3.7. Moreover, variable orders are applied corresponding to an order control.

Algorithm: $\mathbf{P}(\mathbf{EC})^m\mathbf{E}$ method

P: $y_{i+1}^{(0)} := y_i + h(\beta_1 f_i + \beta_2 f_{i-1} + \cdots + \beta_k f_{i-k+1})$ (Adams-Bashforth)

for $\nu = 0, 1, \ldots, m-1$

E: $f_{i+1}^{(\nu)} := f(x_{i+1}, y_{i+1}^{(\nu)})$

C: $y_{i+1}^{(\nu+1)} := y_i + h(\beta_0^* f_{i+1}^{(\nu)} + \beta_1^* f_i + \beta_2^* f_{i-1} + \cdots + \beta_k^* f_{i-k+1})$
(fixed point iteration for Adams-Moulton)

E: $f_{i+1} := f(x_{i+1}, y_{i+1}^{(m)})$ (required for the next integration step)

Table 4: Algorithm of predictor-corrector method for one integration step.

4.7 Order control

The step size control estimates the largest step sizes such that the local error is below a given bound, see Sect. 3.7. The aim is to keep the number of required steps low in the integration. The number of steps can be reduced further by an order control. Assume that several methods with the order of convergence $p = 1, 2, \ldots, p_{\max}$ are available ($p_{\max} = 5 - 15$ in practice). The idea is to choose the method, which exhibits the largest step size prediction in the next step.

Assume that the step size h is selected and an order p is suggested. Then we compute the approximations from the methods $p-1, p, p+1$. Each method implies a corresponding estimate of an optimal step size h_{p-1}, h_p, h_{p+1}. If one of the step sizes is above h, then the step is accepted. Furthermore let w_p be a quantification of the computational effort for one step using the method of order p. (For example, the number of function evaluations of the right-hand side.) Now each method implies an estimate

$$\sigma_{p-1} := \frac{w_{p-1}}{h_{p-1}}, \qquad \sigma_p := \frac{w_p}{h_p}, \qquad \sigma_{p+1} := \frac{w_{p+1}}{h_{p+1}}$$

of the computational work per unit step size. We apply the order $\hat{p}$ with the

lowest value $\sigma_{\hat{p}}$ as suggestion for the optimal order in the next step. The step size $h_{\hat{p}}$ is used in the next step.

Algorithms of linear multistep methods usually apply order control, for example, based on the Adams methods or the BDF methods. The reason is that the effort w_p is nearly independent of the value p for these methods. Remark that just $m + 1$ additional function evaluations are necessary in each step of the P(EC)^{m}E method for arbitrary order, since the other function evaluations are available from the previous steps. In contrast, explicit Runge-Kutta methods exhibit roughly $w_p \approx Cp$ with a constant C, since $p \approx s$ with the number of stages s.

Another class of techniques with a naturally variable order are the extrapolation methods. These techniques can be based on one-step methods or multistep methods.

Remark that each implementation of order control includes many sophisticated specifics in dependence on the underlying methods.

Outlook: General linear methods

It is obvious to ask for a combination of the concepts for Runge-Kutta methods (see Sect. 3.5) and for linear multistep methods (see Sect. 4.1). The resulting techniques include several given approximations from previous grid points as well as a priori unknown intermediate values. The corresponding schemes belong to the class of *general linear methods*. More details can be found in: Hairer, Nørsett, Wanner: Solving Ordinary Differential Equations I. (2nd Ed.) Springer. (Sect. III.8)

Chapter 5

Integration of Stiff Systems

Stiff systems of ordinary differential equations appear in many applications like chemical reactions, mechanical engineering and electric circuit simulation, for example. In principle, these systems can be solved by each convergent method introduced in the previous two chapters. However, explicit methods must not be used, since they are not efficient for stiff problems. This motivates the need for implicit methods.

5.1 Examples

To illustrate the phenomenon of stiffness, we consider two examples of systems of ODEs: the Van-der-Pol oscillator and a particular linear system of ODEs.

Van-der-Pol oscillator

The second-order ODE describing the Van-der-Pol oscillator reads

$$z''(t) + \mu(z(t)^2 - 1)z'(t) + z(t) = 0$$

with the scalar parameter $\mu > 0$. To achieve a frequency (nearly) independent of the parameter, we apply the scaling $x = \frac{t}{\mu}$. It follows for $y(x) = z(\mu x)$

$$\tfrac{1}{\mu^2} y''(x) + (y(x)^2 - 1)y'(x) + y(x) = 0.$$

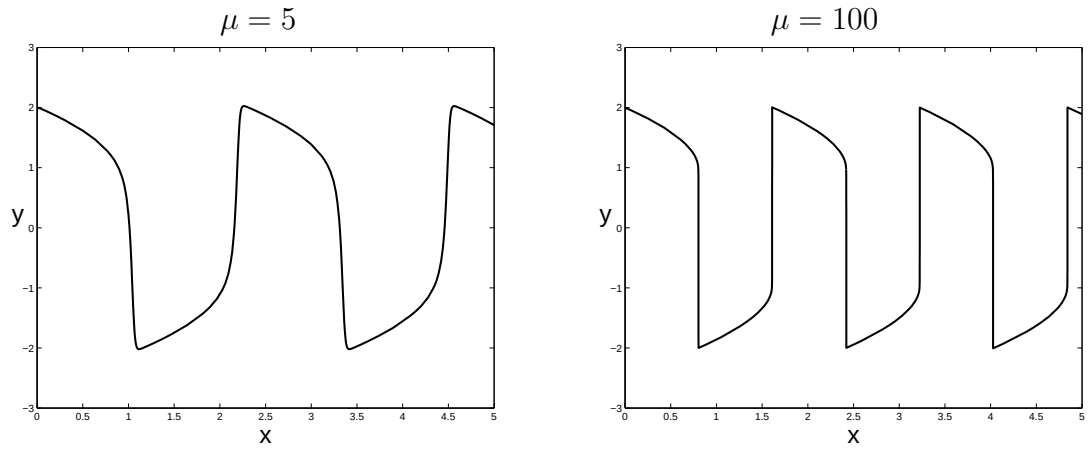

Figure 13: Solutions of Van-der-Pol oscillator.

Initial values $y(0) = 2$ and $y'(0) = 0$ are imposed. We apply the equivalent system of first order

$$\begin{aligned} y_1'(x) &= y_2(x), \\ y_2'(x) &= -\mu^2((y_1(x)^2 - 1)y_2(x) + y_1(x)). \end{aligned}$$

Fig. 13 illustrates solutions for two different parameters μ.

We solve the system with two methods: an explicit Runge-Kutta method of order 2(3) and the (implicit) trapezoidal rule (order 2). In both integrators, a local error control is used with the tolerances rtol $= 10^{-2}$ and atol $= 10^{-4}$. The simulations are done in the interval $x \in [0, 5]$. Table 5 illustrates the number of required steps in the integration for different parameters μ. Remark that the computational effort is proportional to the number of steps in each method. We recognise that the explicit method requires more and more steps for increasing parameter μ. If the step size is enlarged in the explicit scheme, then the integration fails. In contrast, the number of steps increases just slightly in the implicit method. Thus implicit integrators become superior. The behaviour of the system of ODEs for large parameters μ is called stiff.

	explicit method	implicit method
$\mu = 5$	145	201
$\mu = 10$	434	294
$\mu = 50$	9017	483
$\mu = 100$	36.067	542
$\mu = 200$	144.453	616
$\mu = 1000$	3.616.397	624

Table 5: Number of steps in simulation of Van-der-Pol oscillator.

Linear System of ODEs

We discuss a particular linear system of ODEs, namely

$$\begin{pmatrix} y_1'(x) \\ y_2'(x) \end{pmatrix} = \begin{pmatrix} -298 & 99 \\ -594 & 197 \end{pmatrix} \begin{pmatrix} y_1(x) \\ y_2(x) \end{pmatrix}. \tag{5.1}$$

The matrix exhibits the eigenvalues $\lambda_1 = -1$ and $\lambda_2 = -100$ with corresponding eigenvectors $v_1 = (1, 3)^\top$ and $v_2 = (1, 2)^\top$. Hence the general solution of the system (5.1) reads

$$y(x) = C_1 \mathrm{e}^{-x} \begin{pmatrix} 1 \\ 3 \end{pmatrix} + C_2 \mathrm{e}^{-100x} \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

with arbitrary constants $C_1, C_2 \in \mathbb{R}$. All solutions satisfy

$$\lim_{x \to \infty} y(x) = 0.$$

However, one term (w.r.t. λ_2) decreases rapidly, whereas the other term (w.r.t. λ_1) decreases relatively slowly.

We consider the initial value problem $y_1(0) = -\frac{1}{2}$ and $y_2(0) = \frac{1}{2}$. Fig. 14 (left) illustrates the corresponding solution. We apply an explicit Runge-Kutta method of order 2(3) and the trapezoidal rule with step size control (rtol $= 10^{-3}$, atol $= 10^{-6}$) again. In the interval $x \in [0, 10]$, the explicit scheme requires 413 steps and the implicit scheme needs just 94 steps. Fig. 14 (right) shows that the explicit method also chooses small step sizes at the end of the interval, where the solution is nearly constant. If the step size is increased in the explicit method, then the corresponding approximations become completely wrong. We want to understand this different performance of the integration techniques.

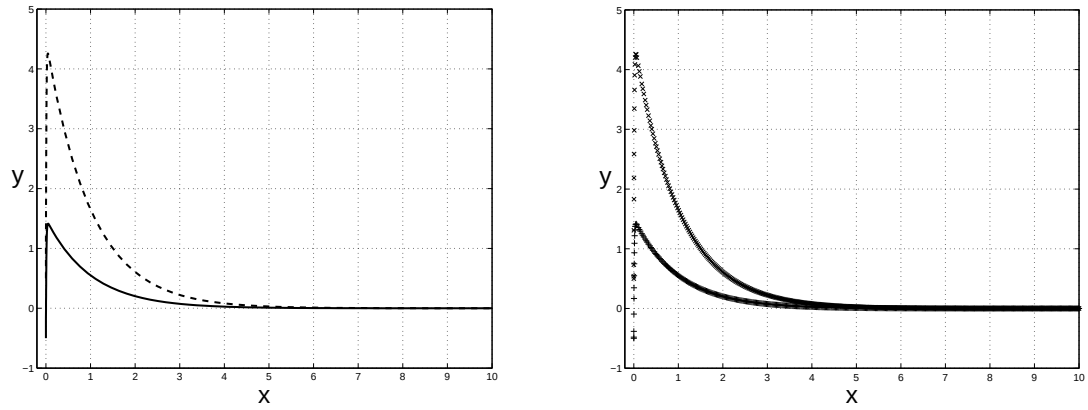

Figure 14: Stiff linear system: exact solution (left) – y_1 (solid line) and y_2 (dashed line) – as well as numerical approximations from explicit method with step size control (right).

According to this example, the stiff behaviour can be characterised as follows: The solutions of initial value problems tend rapidly to solutions, which vary just slowly. However, in a neighbourhood of the slowly changing solutions, there exist fastly changing solutions.

5.2 Test equations

We analyse the previous linear example in a general context now. Given a linear system of ODEs

$$y'(x) = Ay(x), \quad y : \mathbb{R} \to \mathbb{R}^n, \quad A \in \mathbb{R}^{n \times n}, \tag{5.2}$$

we assume that the involved matrix is diagonalisable, i.e.,

$$A = T^{-1}DT, \quad T \in \mathbb{C}^{n \times n}, \quad D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n).$$

The eigenvalues $\lambda_1, \ldots, \lambda_n \in \mathbb{C}$ may be complex numbers also in case of a real matrix A. Using the transformation $z(x) = Ty(x)$, the system decouples into the scalar linear ODEs

$$z_j'(x) = \lambda_j z_j(x) \qquad \text{for } \; j = 1, \ldots, n. \tag{5.3}$$

According initial values are transformed via $z(x_0) = Ty(x_0)$.

Dahlquist test equation

Due to the decoupled ODEs (5.3), we discuss the scalar linear ODE

$$y'(x) = \lambda y(x), \quad y : \mathbb{R} \to \mathbb{C}, \quad \lambda \in \mathbb{C}. \tag{5.4}$$

The ODE (5.4) is called Dahlquist test equation (1963). Given an initial value $y(0) = y_0$, the exact solution reads

$$y(x) = y_0 \mathrm{e}^{\lambda x} = y_0 \mathrm{e}^{\mathrm{Re}(\lambda)x} \cdot \mathrm{e}^{\mathrm{i}\cdot \mathrm{Im}(\lambda)x}.$$

It follows

$$|y(x)| = |y_0| \cdot \mathrm{e}^{\mathrm{Re}(\lambda)x}.$$

If $\mathrm{Re}(\lambda) < 0$ holds, then the solution decreases monotonically.

We will apply the explicit Euler method and the implicit Euler method to the test problem. Fig. 15 illustrates numerical solutions for $\lambda = -10$ and initial value $y_0 = 1$. We recognise that the implicit technique reproduces the qualitiative behaviour of the exact solution correctly for all step sizes. In contrast, the explicit method is qualitatively adequate only for small step sizes.

It is not difficult to explain the performance of the Euler methods:

(i) *Explicit Euler method*
Applied to Dahlquist's test equation (5.4), the scheme reads

$$y_1 = y_0 + h\lambda y_0 = (1 + h\lambda) y_0.$$

It follows successively (y_j is approximation of $y(jh)$)

$$y_j = (1 + h\lambda)^j \, y_0.$$

Hence it holds $|y_j| \leq |y_{j-1}|$ if and only if

$$|1 + h\lambda| \leq 1.$$

For $\lambda \in \mathbb{R}$ and $\lambda < 0$ (and of course $h > 0$), we obtain a step size restriction

$$h \leq \tfrac{2}{|\lambda|}.$$

Only for step sizes satisfying this condition, the approximations do not increase. For large $|\lambda|$, the step size h has to be small.

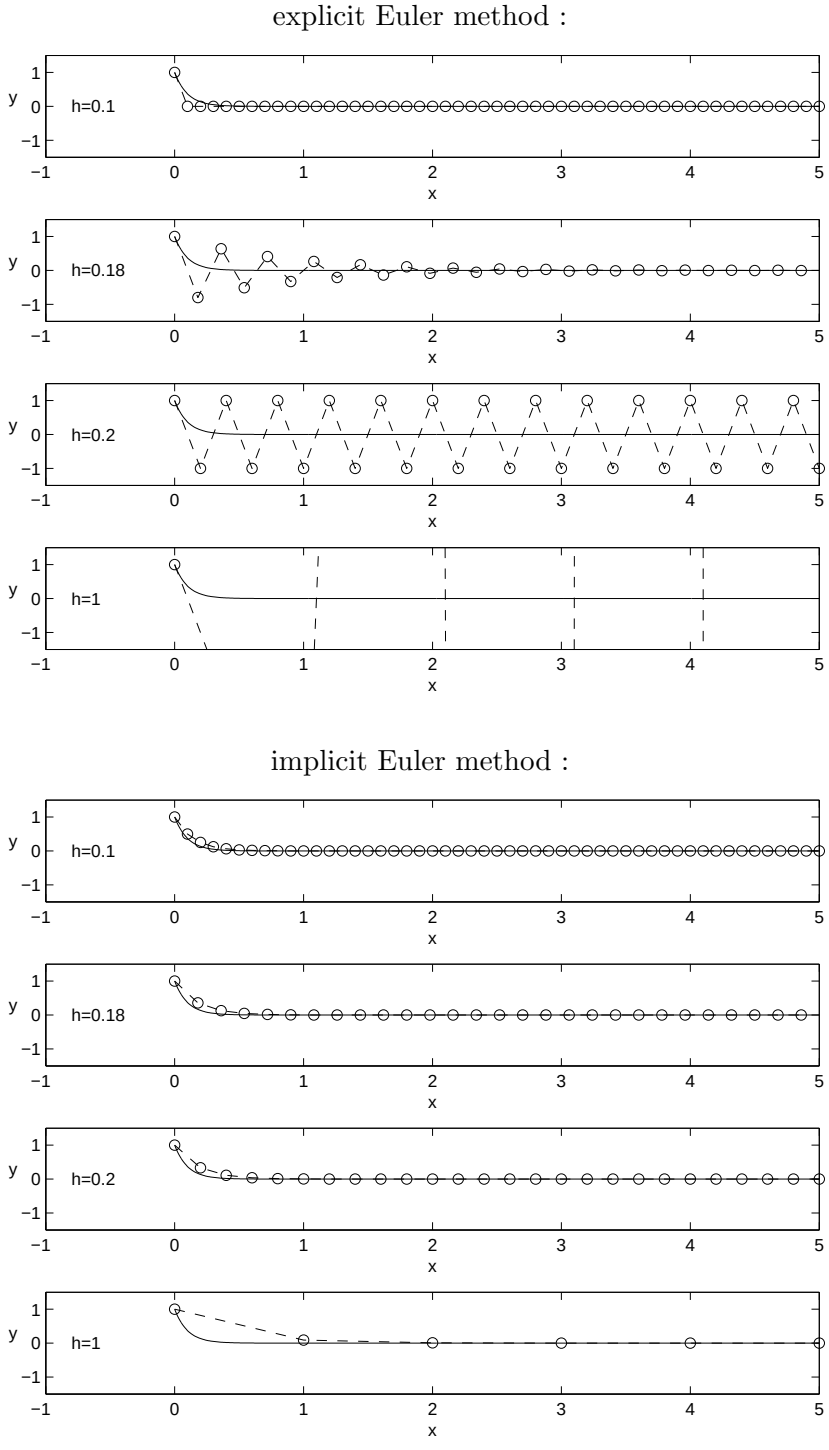

Figure 15: Solutions of Dahlquist's test equation with $\lambda = -10$: exact solution (solid line) and approximations (circles).

(ii) *Implicit Euler method*
Now Dahlquist's test equation (5.4) leads to the scheme

$$y_1 = y_0 + h\lambda y_1 \qquad \Rightarrow \qquad y_1 = \frac{1}{1-h\lambda} y_0.$$

We obtain the approximations

$$y_j = \left(\frac{1}{1-h\lambda}\right)^j y_0.$$

The property $|y_j| \leq |y_{j-1}|$ is valid if and only if

$$\left|\frac{1}{1-h\lambda}\right| \leq 1 \qquad \Leftrightarrow \qquad 1 \leq |1-h\lambda|$$

holds. For $\lambda \in \mathbb{R}$ and $\lambda < 0$, this requirement is satisfied for arbitrary step size $h > 0$. Thus there is no restriction on the step size.

We investigate Dahlquist's equation in case of parameters λ with large negative real part. Remark that the corresponding Lipschitz constant becomes large, since it follows for $f(x, y) = \lambda y$

$$|f(x, y) - f(x, z)| = |\lambda y - \lambda z| = |\lambda| \cdot |y - z|.$$

Hence the fixed point iteration in predictor-corrector methods, cf. Sect. 4.6, exhibits a significant step size restriction needed for convergence.

Prothero-Robinson test equation

Another scalar problem, which illustrates stiff behaviour, is the Prothero-Robinson test equation (1973)

$$y'(x) = \lambda(y(x) - \varphi(x)) + \varphi'(x), \qquad y(x_0) = y_0 \tag{5.5}$$

with solution $y : \mathbb{R} \to \mathbb{R}$ and a real parameter $\lambda < 0$. The smooth function $\varphi : \mathbb{R} \to \mathbb{R}$ is predetermined. The solutions of the initial value problem (5.5) read

$$y(x) = (y_0 - \varphi(x_0))\mathrm{e}^{\lambda(x-x_0)} + \varphi(x).$$

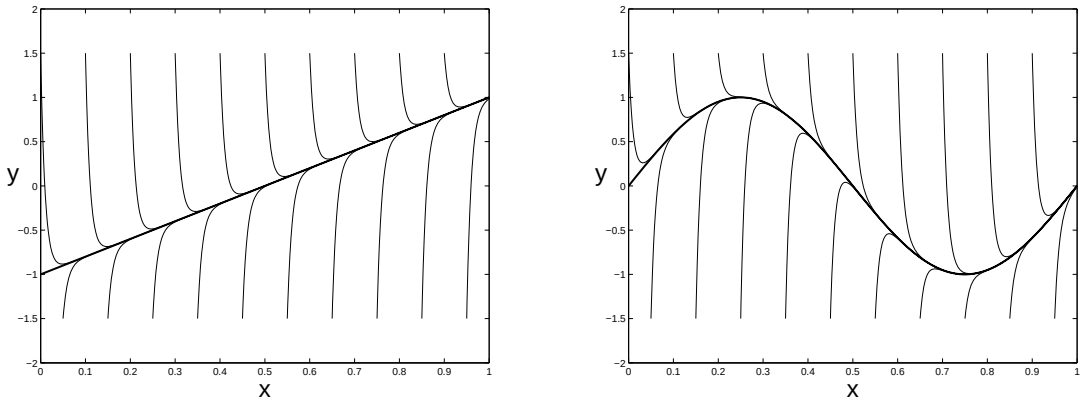

Figure 16: Solutions of several initial value problems corresponding to the Prothero-Robinson test equation with parameter $\lambda = -100$, function $\varphi(x) = 2x - 1$ (left) and $\varphi(x) = \sin(2\pi x)$ (right).

The particular solution $y \equiv \varphi$ represents an asymptotic phase, i.e. the other solutions tend rapidly to this function in case of large negative parameters λ. Fig. 16 illustrates two examples. Furthermore, the choice $\varphi \equiv 0$ yields Dahlquist's test equation (5.4).

Definition of stiff linear systems

We define the phenomenon of stiffness for linear systems now. Remark that it does not exist a precise definition of stiffness (for linear or nonlinear systems). One reason is that stiffness is not only a qualitative behaviour but also a quantitative behaviour. We formulate the two definitions:

- In the linear system $y' = Ax$, assume that all eigenvalues λ_j exhibit a negative real part. The system is stiff, if it exist eigenvalues with small negative real part as well as large negative real part, i.e., the ratio

$$\frac{\max\limits_{j=1,\ldots,n} |\mathrm{Re}(\lambda_j)|}{\min\limits_{j=1,\ldots,n} |\mathrm{Re}(\lambda_j)|} \tag{5.6}$$

  is very large. (If all eigenvalues exhibit a large negative real part with the same magnitute, i.e., the ratio (5.6) is small, then the stiff behaviour can be transformed out of the system.)

- The following characterisation due to Curtis and Hirschfelder (1952) was found from their observations in simulating chemical reaction kinetics (and holds also for nonlinear systems): "Stiff equations are equations, where certain implicit methods perform better – usually tremendously better – than explicit ones." In short form: Implicit is better than explicit.

5.3 A-stability for one-step methods

The performance of the Euler methods applied to Dahlquist's test equation (5.4) motivates the definition of a stability concept. Stability is seen as a necessary (not sufficient) condition to achieve suitable approximations. In this section, we consider just one-step methods.

Definition 10 (A-stability of one-step methods)
A one-step method is A-stable if the corresponding sequence of approximations $(y_j)_{j\in\mathbb{N}}$ *for Dahlquist's equation (5.4) with* $Re(\lambda) \leq 0$ *for any step size* $h > 0$ *is not increasing, i.e.,* $|y_{j+1}| \leq |y_j|$ *holds for all* j*.*

If a one-step method is A-stable, then it is suitable for solving stiff linear system of ODEs. Vice versa, a technique, which is not A-stable, should not be used for (linear or nonlinear) stiff problems.

We want to obtain a technique for verifying if a method is A-stable or not. We use the abbreviation $z := h\lambda \in \mathbb{C}$ in the following. On an equidistant grid $x_j = x_0 + jh$, the exact solution of Dahlquist's equation (5.4) satisfies

$$y(x_{j+1}) = \mathrm{e}^{h\lambda} y(x_j) = \mathrm{e}^{z} y(x_j).$$

Thus it holds $|y(x_{j+1})| \leq |y(x_j)|$ if and only if $\mathrm{Re}(\lambda) \leq 0$, which is equivalent to $\mathrm{Re}(z) \leq 0$. Applied to Dahlquist's test equation, the Euler methods can be written in the form

$$y_{j+1} = R(z) y_j$$

with

$$\text{expl. Euler}: \quad R(z) = 1 + z, \qquad \text{impl. Euler}: \quad R(z) = \frac{1}{1-z}.$$

We want that $|R(z)| \leq 1$ holds for each z with $\mathrm{Re}(z) \leq 0$. Each one-step method yields a formula $y_1 = R(z)y_0$. The mapping $R : \mathbb{C} \to \mathbb{C}$ is called the stability function of the method.

Definition 11 (stability domain of one-step methods)
The stability domain $S \subset \mathbb{C}$ of a one-step method $y_1 = R(z)y_0$ is the set

$$S := \{z \in \mathbb{C} : |R(z)| \leq 1\}.$$

Furthermore, we define $\mathbb{C}^- := \{z \in \mathbb{C} : \mathrm{Re}(z) \leq 0\}$. Hence A-stability is characterised as follows

$$\text{A-stable} \quad \Leftrightarrow \quad |R(z)| \leq 1 \text{ for all } z \in \mathbb{C}^- \quad \Leftrightarrow \quad \mathbb{C}^- \subseteq S.$$

For the Euler methods, the stability domains read

$$\begin{aligned} &\text{expl. Euler:} && S = \{z \in \mathbb{C} : |1 + z| \leq 1\}, \\ &\text{impl. Euler:} && S = \left\{z \in \mathbb{C} : \left|\tfrac{1}{1-z}\right| \leq 1\right\} = \{z \in \mathbb{C} : 1 \leq |1 - z|\}. \end{aligned}$$

These stability domains are the inside of a circle around $z = -1$ with radius 1 and the outside of a circle around $z = 1$ with radius 1, respectively, see Fig. 17. Hence $\mathbb{C}^- \subseteq S$ holds for the implicit Euler method but not for the explicit Euler method.

Example: Trapezoidal rule

The trapezoidal rule applied to Dahlquist's test equation (5.4) yields

$$y_1 = y_0 + \tfrac{h}{2}\left[\lambda y_0 + \lambda y_1\right].$$

It follows

$$y_1 = \frac{1 + \frac{1}{2}h\lambda}{1 - \frac{1}{2}h\lambda}\, y_0.$$

The stability function becomes

$$R(z) = \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z}.$$

A detailed analysis shows that $S = \mathbb{C}^-$ in this case. Thus the trapezoidal rule is A-stable.

Example: Explicit midpoint rule

The explicit midpoint rule (3.5) implies

$$y_1 = y_0 + h\lambda \left(y_0 + \tfrac{h}{2}\lambda y_0\right) = \left(1 + h\lambda + \tfrac{1}{2}h^2\lambda^2\right) y_0$$

when used for Dahlquist's equation (5.4). We obtain the stability function

$$R(z) = 1 + z + \tfrac{1}{2}z^2.$$

It follows that the explicit midpoint rule is not A-stable.

Fig. 17 illustrates the stability domains of the four elementary one-step methods discussed above, cf. Sect. 3.2.

General Runge-Kutta method

A general Runge-Kutta method with s stages for the ODE-IVP $y' = f(x, y)$, $y(x_0) = y_0$ reads

$$\begin{aligned} k_i &= f\left(x_0 + c_i h, y_0 + h\sum_{j=1}^{s} a_{ij} k_j\right) \qquad \text{for } \; i = 1, \ldots, s \;, \\ y_1 &= y_0 + h\sum_{i=1}^{s} b_i k_i. \end{aligned}$$

The method is uniquely determined by its coefficients

$$c = (c_i) \in \mathbb{R}^s, \qquad b = (b_i) \in \mathbb{R}^s, \qquad A = (a_{ij}) \in \mathbb{R}^{s\times s}.$$

In case of Dahlquist's test equation $y' = \lambda y$, a formula for the corresponding stability function can be derived. This formula is valid for both explicit and implicit Runge-Kutta methods.

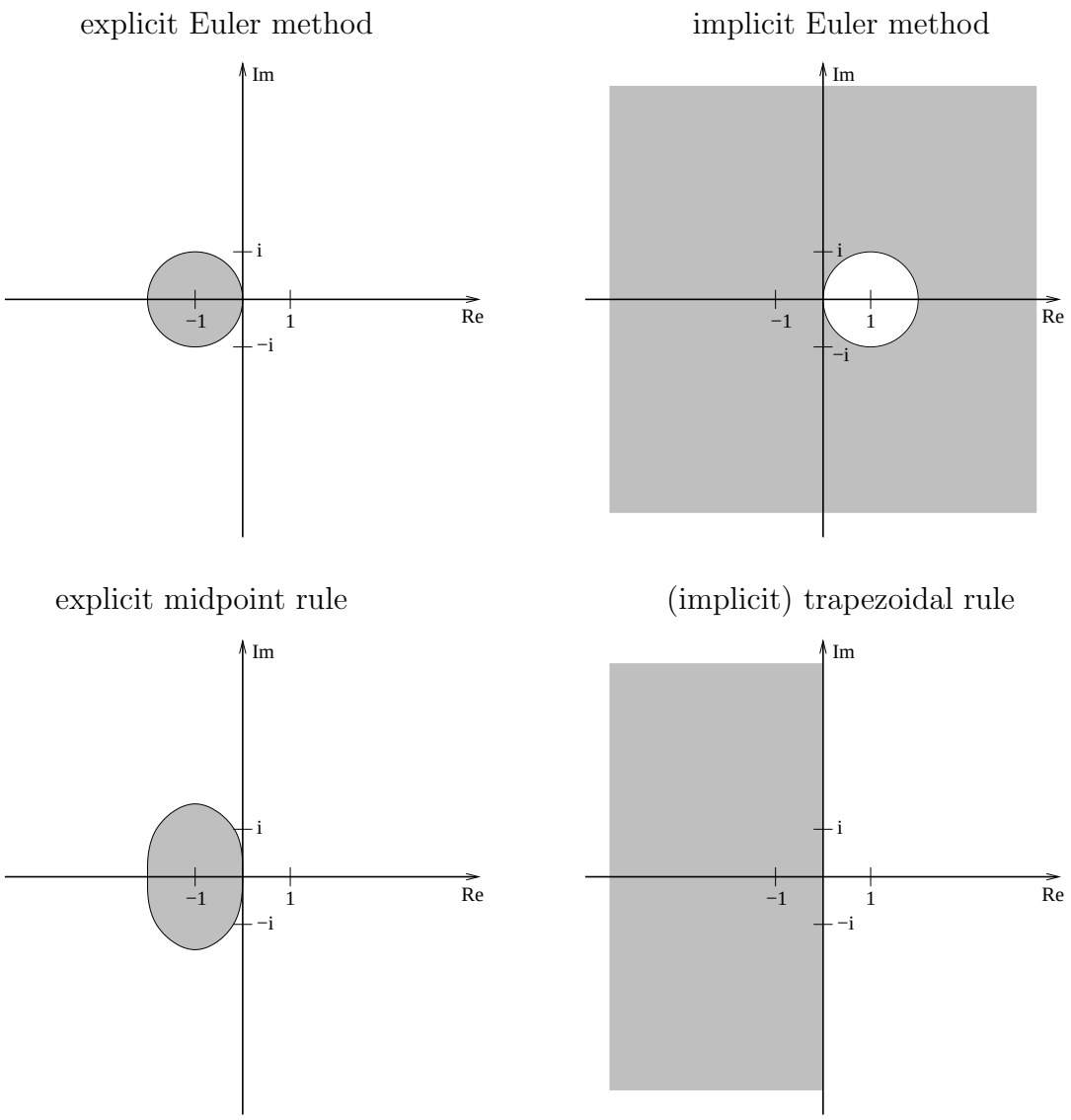

Figure 17: Stability domains of some important one-step methods.

Theorem 13 (stability function of Runge-Kutta method)
The stability function of a Runge-Kutta scheme is given by

$$R(z) = 1 + zb^\top (I - zA)^{-1}\mathbb{1} \tag{5.7}$$

with $\mathbb{1} := (1, \ldots, 1)^\top \in \mathbb{R}^s$ *and identity matrix* $I \in \mathbb{R}^{s\times s}$ *or, equivalently,*

$$R(z) = \frac{\det(I - zA + z\mathbb{1}b^\top)}{\det(I - zA)}.$$

Theorem 13 demonstrates that the stability function of a Runge-Kutta method is a rational function in the variable z. The stability function is not defined in case of $\det(I - zA) = 0$. Thus a finite number of poles may appear.

An explicit Runge-Kutta scheme corresponds to a strictly lower triangular matrix A. It follows $\det(I - zA) = 1$ for all $z \in \mathbb{C}$. The stability function of an explicit Runge-Kutta method is a polynomial

$$R(z) = \alpha_0 + \alpha_1 z + \alpha_2 z^2 + \cdots + \alpha_{s-1} z^{s-1} + \alpha_s z^s.$$

Consequently, it holds

$$|R(z)| \overset{\mathrm{Re}(z)\to-\infty}{\longrightarrow} +\infty.$$

Hence an explicit Runge-Kutta method is never A-stable.

L-stability

The concept of L-stability represents an improvement of the A-stability. Again the property is based on Dahlquist's test equation (5.4). The exact solution satisfies

$$y(h) = \mathrm{e}^z y(0) \qquad \text{with } \; z = h\lambda.$$

In the limit case of parameters λ with huge negative real part, it follows

$$\lim_{\mathrm{Re}(z)\to-\infty} y(h) = y(0) \lim_{\mathrm{Re}(z)\to-\infty} \mathrm{e}^z = 0.$$

We want that the numerical approximation

$$y_1 = R(z)y_0$$

of a one-step method inherits this property.

Definition 12 (L-stability) *A one-step method is called L-stable, if it is A-stable and in addition*

$$\lim_{z\to\infty} R(z) = 0.$$

Remark that $R(z)$ is a rational function in case of one-step methods. Thus it holds

$$\lim_{\mathrm{Re}(z)\to-\infty} R(z) = \lim_{|z|\to\infty} R(z) = \lim_{z\to\infty} R(z)$$

provided that the limit exists. A rational function $R(z)$ exhibits the form

$$R(z) = \frac{a_0 + a_1 z + \cdots + a_{n-1} z^{n-1} + a_n z^n}{b_0 + b_1 z + \cdots + b_{m-1} z^{m-1} + b_m z^m}$$

with $a_n, b_m \neq 0$. It follows

$$\lim_{z\to\infty} |R(z)| \begin{cases} = 0 & \text{for } n < m, \\ = \left|\frac{a_n}{b_n}\right| & \text{for } n = m, \\ \to \infty & \text{for } n > m. \end{cases}$$

It follows that the implicit Euler method is L-stable, since

$$\lim_{z\to\infty} R(z) = \lim_{z\to\infty} \frac{1}{1-z} = 0.$$

However, the trapezoidal rule yields for $\omega \in \mathbb{R}$

$$|R(\mathrm{i}\omega)|^2 = \frac{|1 + \frac{1}{2}\mathrm{i}\omega|^2}{|1 - \frac{1}{2}\mathrm{i}\omega|^2} = \frac{1 + \frac{1}{4}\omega^2}{1 + \frac{1}{4}\omega^2} = 1.$$

Since $R(z)$ is a rational function, we obtain

$$\lim_{z\to\infty} R(z) = 1$$

and thus the trapezoidal rule is not L-stable. Consequently, the trapezoidal rule is not appropriate for extremely stiff linear problems.

Padé-approximation

We reconsider Dahlquist's test equation (5.4). Let $y(h) = \mathrm{e}^z y_0$ be the exact solution and $y_1 = R(z)y_0$ the approximation from a method. It follows

$$y(h) - y_1 = \left(\mathrm{e}^z - R(z)\right) y_0.$$

j/k	0	1	2	$\cdots$
0	$\frac{1}{1}$	$\frac{1+z}{1}$	$\frac{1+z+\frac{1}{2}z^2}{1}$	$\cdots$
1	$\frac{1}{1-z}$	$\frac{1+\frac{1}{2}z}{1-\frac{1}{2}z}$	$\frac{1+\frac{2}{3}z+\frac{1}{6}z^2}{1-\frac{1}{3}z}$	
2	$\frac{1}{1-z+\frac{1}{2}z^2}$	$\frac{1+\frac{1}{3}z}{1-\frac{2}{3}z+\frac{1}{6}z^2}$	$\frac{1+\frac{1}{2}z+\frac{1}{12}z^2}{1-\frac{1}{2}z+\frac{1}{12}z^2}$	
$\vdots$	$\vdots$			$\ddots$

Table 6: Padé-approximations of e^z.

Thus we expect good approximations if the stability function approximates the exponential function appropriately. The correct approximation of the limit case $\mathrm{Re}(z) \to -\infty$ corresponds to the L-stability. The behaviour for $z \to 0$ leads to the general approach of the Padé-approximation.

Definition 13 (Padé-approximation) *Let $g : \mathbb{C} \to \mathbb{C}$ be analytic in a neighbourhood of $z = 0$. The rational function*

$$R_{jk}(z) = \frac{P_{jk}(z)}{Q_{jk}(z)}$$

with $P_{jk}(z) = a_0 + a_1 z + \cdots + a_k z^k$ and $Q_{jk}(z) = 1 + b_1 z + \cdots + b_j z^j$ is called the Padé-approximation of g with index (j, k), if it holds

$$R_{jk}^{(l)}(0) = g^{(l)}(0) \qquad \textit{for } l = 0, 1, \ldots, j + k.$$

If the Padé approximation exists, then it is unique. It follows

$$R_{jk}(z) = g(z) + \mathcal{O}(z^{j+k+1}) \qquad \text{for } z \to 0.$$

All Padé-approximations of the exponential function $g(z) = \mathrm{e}^z$ exist.

Table 6 shows some Padé-approximations of the exponential functions. We recognise the stability functions of the explicit Euler method (0,1), the implicit Euler method (1,0) and the trapezoidal rule (1,1). An approximation

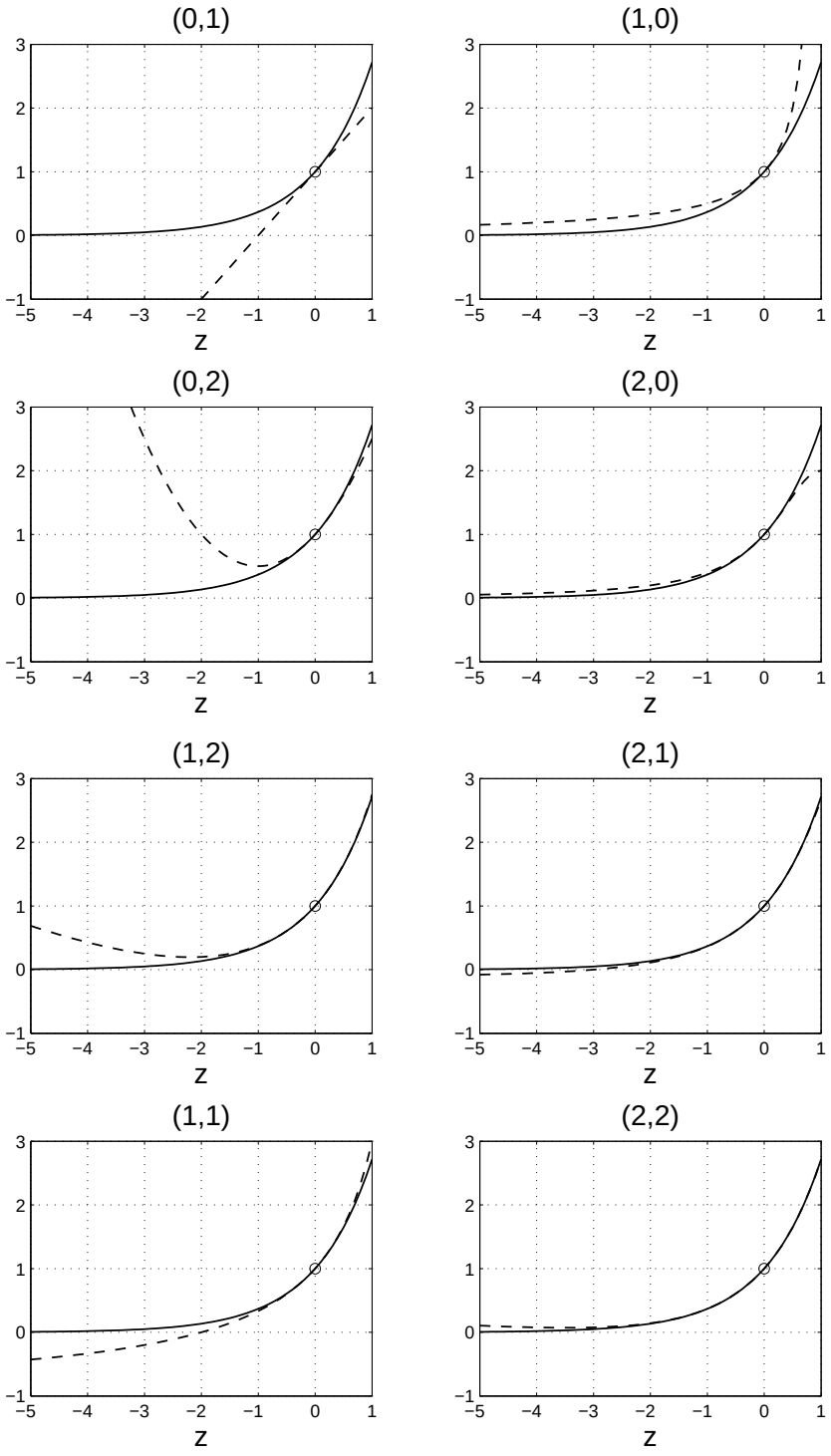

Figure 18: Exponential function e^z (solid line) and some Padé-approximations $R_{jk}(z)$ of index (j, k) (dashed lines) for real $z \in [-5, 1]$.

for $k > j$ cannot correspond to an A-stable method. The Gauss-Runge-Kutta methods yield the stability functions for the diagonal $j = k$ (implicit midpoint rule (3.12) owns the same stability function as trapezoidal rule) and are A-stable. However, since $|a_j| = |b_k|$ holds for $j = k$, these methods are not L-stable. Alternatively, Runge-Kutta methods exist corresponding to each $j = k + 1$ (the subdiagonal), which are A-stable as well as L-stable.

Remark that an explicit Runge-Kutta method exhibits a polynomial as stability function. A polynomial represents a bad approximation for a decreasing exponential function. In contrast, a rational function can approximate a decaying exponential function much better. Only implicit methods yield rational stability functions. Fig. 18 depicts some Padé-approximations of the exponential function for z from a real interval, which confirm these statements.

5.4 Implicit Runge-Kutta methods

Since an explicit Runge-Kutta method is never A-stable, we have to apply implicit Runge-Kutta (IRK) schemes to solve stiff problems.

Collocation methods

An important class of implicit Runge-Kutta methods are collocation methods. These schemes are successful in solving stiff systems of ODEs. Given the ODE-IVP $y' = f(x, y)$, $y(x_0) = y_0$, we want to construct an approximation $y_1 \doteq y(x_0 + h)$. The idea of collocation methods is simple: Choose nodes $0 \leq c_1 < c_2 < \cdots < c_s \leq 1$. Determine the polynomial $w \in \mathbb{P}_s$ via the conditions

$$\begin{aligned} w(x_0) &= y_0, \\ w'(x_0 + c_i h) &= f(x_0 + c_i h, w(x_0 + c_i h)) \quad \text{for } i = 1, \ldots, s. \end{aligned}$$

Hence we demand that the ODE is satisfied by the polynomial in the collocation points $x_i := x_0 + c_i h$. The approximation becomes $y_1 := w(x_0 + h)$.

We obtain a Runge-Kutta scheme. Using the Lagrange polynomials

$$L_j : \mathbb{R} \to \mathbb{R}, \quad L_j(x_i) = \begin{cases} 1 & \text{if } j = i, \\ 0 & \text{if } j \neq i, \end{cases} \qquad \text{for } j, i = 1, \ldots, s \ ,$$

the coefficients of the Butcher tableau read

$$b_i = \int_0^1 L_i(x_0 + uh) \, \mathrm{d}u, \qquad a_{ij} = \int_0^{c_i} L_j(x_0 + uh) \, \mathrm{d}u \tag{5.8}$$

for $i, j = 1, \ldots, s$. Each element a_{ij} is non-zero in general, which implies an implicit technique. The method is uniquely determined by the nodes $c_1, \ldots, c_s$. We choose the nodes from a quadrature scheme like Gaussian quadrature, for example. Concerning the order of consistency, the following relation holds.

Theorem 14 (order of collocation methods)
A Runge-Kutta method resulting from a collocation approach is consistent of order p if and only if the quadrature formula given by the collocation points c_i and the weights b_i is of order p.

Outline of the proof:

Let the quadrature formula be of order p, i.e., it holds

$$\mathrm{err}(g) := \left| \int_{x_0}^{x_0+h} g(s) \, \mathrm{d}s - h \sum_{i=1}^{s} b_i g(x_0 + c_i h) \right| \leq K h^{p+1} \max_{u \in [0,1]} \left| g^{(p)}(x_0 + uh) \right|$$

for a function $g \in C^p$ with a constant $K > 0$. For simplicity, we investigate the scalar case and a smooth function f. The exact solution corresponds to $y'(x) = f(x, y(x))$. The polynomial w satisfies

$$w'(x) = f(x, w(x)) + r(x) \qquad \text{with} \quad r(x) := w'(x) - f(x, w(x)).$$

Due to the construction of w, the function r exhibits s zeros. We consider r as a perturbation of the right-hand side f in the original ODE. Since $y(x_0) = w(x_0)$ holds, the Alekseev/Gröbner formula yields

$$w(x) - y(x) = \int_{x_0}^{x} \frac{\partial y}{\partial y_0}(x, s, w(s)) \cdot r(s) \, \mathrm{d}s,$$

where $y(x, x_0, y_0)$ denotes the solution of $y' = f(x, y),\ y(x_0) = y_0$. It follows

$$|y(x_0 + h) - w(x_0 + h)| = \left| \int_{x_0}^{x_0+h} \frac{\partial y}{\partial y_0}(x_0 + h, s, w(s)) \cdot r(s)\ \mathrm{d}s \right|.$$

We apply the quadrature rule to this integral. Since $r(x_0 + c_i h) = 0$ holds for all $i = 1, \ldots, s$, the quadrature rule yields the approximation zero. It follows

$$\left| \int_{x_0}^{x_0+h} \frac{\partial y}{\partial y_0}(x_0 + h, s, w(s)) \cdot r(s)\ \mathrm{d}s \right| = \mathrm{err}(g)$$

with

$$g(s) := \frac{\partial y}{\partial y_0}(x_0 + h, s, w(s)) \cdot r(s).$$

We obtain $\mathrm{err}(g) = \mathcal{O}(h^{p+1})$ owing to the above assumption. For this conclusion, it remains to show that the derivatives $g^{(p)}$ are bounded in a neighbourhood of x_0. Due to the construction $y_1 := w(x_0 + h)$, it follows

$$|y(x_0 + h) - y_1| = \mathcal{O}(h^{p+1}),$$

i.e., the consistency of order p is confirmed.

Vice versa, let the Runge-Kutta method be consistent of order p. The particular ODE-IVP $y'(x) = g(x),\ y(x_0) = 0$ exhibits the solution

$$y(x_0 + h) = \int_{x_0}^{x_0+h} g(s)\ \mathrm{d}s.$$

The Runge-Kutta scheme yields the increments $k_i = g(x_0 + c_i h)$ for each $i = 1, \ldots, s$ and thus the approximation

$$y_1 = h \sum_{i=1}^{s} b_i g(x_0 + c_i h).$$

The consistency of order p implies $y(x_0 + h) - y_1 = \mathcal{O}(h^{p+1})$. □

The Gaussian quadrature yields the optimal order of consistency $p = 2s$. The nodes are the s roots of the Legendre polynomials

$$P_s : [0, 1] \to \mathbb{R}, \qquad P_s(x) = \tfrac{\mathrm{d}^s}{\mathrm{d}x^s} \left[x^s (x - 1)^s \right].$$

The corresponding Gauss-Runge-Kutta methods have already been introduced in Sect. 3.5. These schemes correspond to the Padé-approximations of index (s, s).

Two types of collocation methods follow from the Radau quadrature formula. The nodes are the roots of the polynomials

$$\text{RadauI}: \quad P_s : [0,1] \to \mathbb{R}, \quad P_s(x) = \frac{\mathrm{d}^{s-1}}{\mathrm{d}x^{s-1}} \left[x^s (x-1)^{s-1} \right],$$

$$\text{RadauII}: \quad P_s : [0,1] \to \mathbb{R}, \quad P_s(x) = \frac{\mathrm{d}^{s-1}}{\mathrm{d}x^{s-1}} \left[x^{s-1} (x-1)^s \right].$$

The weights b_i are defined according to (5.8). The choice of the inner weights a_{ij} determines if a collocation method results or not. Four methods have been constructed:

method	collocation	A-stable	L-stable
RadauI (Butcher 1964)	yes	no	no
RadauIA (Ehle 1968)	no	yes	yes
RadauII (Butcher 1964)	no	no	no
RadauIIA (Ehle 1968)	yes	yes	yes

The resulting order of consistency is $p = 2s - 1$ in each scheme. The RadauIA methods have nodes $c_1 = 0$ and $c_s < 1$, whereas the RadauIIA methods yield nodes $0 < c_1$ and $c_s = 1$. This property and being a collocation method make the RadauIIA schemes more advantageous than RadauIA. The s-stage RadauIA and RadauIIA methods correspond to the Padé-approximations of index $(s, s-1)$. The RadauIIA method for $s = 1$ is just the implicit Euler method. The RadauIIA schemes for $s = 2$ and $s = 3$ exhibit the Butcher-tableaus:

$$\begin{array}{c|cc} \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{3}{4} & \frac{1}{4} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array} \qquad \begin{array}{c|ccc} \frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ 1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\ \hline & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{array}$$

Another type of collocation methods are the Lobatto schemes.

Solution of nonlinear systems

We discuss the efficient solution of nonlinear systems resulting from implicit Runge-Kutta methods now. For a single step applied to the nonlinear system $y' = f(x, y)$ with $y : \mathbb{R} \to \mathbb{R}^n$, the Runge-Kutta method reads

$$\begin{aligned} \tilde{y}_i &= y_0 + h\sum_{j=1}^{s} a_{ij} f(x_j, \tilde{y}_j) \qquad \text{for } i = 1, \ldots, s \\ y_1 &= y_0 + h\sum_{i=1}^{s} b_i f(x_i, \tilde{y}_i) \end{aligned}$$

with $x_i := x_0 + c_i h$. The transformation $z_i := \tilde{y}_i - y_0 \in \mathbb{R}^n$ yields the equivalent nonlinear system of algebraic equations

$$z_i - h\sum_{j=1}^{s} a_{ij} f(x_j, y_0 + z_j) = 0 \qquad \text{for } i = 1, \ldots, s.$$

It holds $z_i = \mathcal{O}(h)$ and thus the values z_i are smaller than the values $\tilde{y}_i$, which reduces the effects of roundoff errors. We use the abbreviations

$$Z := \begin{pmatrix} z_1 \\ \vdots \\ z_s \end{pmatrix}, \quad G(Z) := \begin{pmatrix} z_1 - h\sum_{j=1}^{s} a_{1j} f(x_j, y_0 + z_j) \\ \vdots \\ z_s - h\sum_{j=1}^{s} a_{sj} f(x_j, y_0 + z_j) \end{pmatrix}.$$

The nonlinear system $G(Z) = 0$ with sn equations has to be solved. Each evaluation of G demands s evaluations of the right-hand side f. We apply the simplified Newton iteration, cf. Sect. 4.6 in case of multistep methods. The iteration reads

$$\mathrm{D}G(Z^{(0)})\Delta Z^{(\nu)} = -G(Z^{(\nu)}), \qquad Z^{(\nu+1)} = Z^{(\nu)} + \Delta Z^{(\nu)}$$

for $\nu = 0, 1, 2, \ldots$ with the Jacobian matrix $\mathrm{D}G \in \mathbb{R}^{sn \times sn}$ of G. A suitable choice of the starting values is $z_i^{(0)} = 0$ for all i. The required Jacobian matrices of the right-hand side f are

$$J_i := \mathrm{D}f(x_i, y_0) \qquad \text{for } i = 1, \ldots, s.$$

We replace all Jacobian matrices by

$$J := \mathrm{D}f(x_0, y_0),$$

which represents just a slight simplification. Thus only one Jacobian matrix of f has to be evaluated like in an implicit multistep method. The simplified Jacobian matrix of G becomes

$$\hat{J} := I_{sn} - h \begin{pmatrix} a_{11}J & \cdots & a_{1s}J \\ \vdots & & \vdots \\ a_{s1}J & \cdots & a_{ss}J \end{pmatrix} = I_{sn} - h(A \otimes J)$$

using the notation of Kronecker products. The Newton iteration becomes

$$(I_{sn} - h(A \otimes J))\,\Delta Z^{(\nu)} = -G(Z^{(\nu)}), \qquad Z^{(\nu+1)} = Z^{(\nu)} + \Delta Z^{(\nu)}.$$

Since $\hat{J} \in \mathbb{R}^{sn \times sn}$ holds, an LU-decomposition demands a computational work proportional to $s^3 n^3$.

We can save a significant amount of computational effort by a transformation. We assume that the coefficient matrix $A = (a_{ij})$ is regular and that A can be diagonalised. It follows

$$T^{-1}A^{-1}T = \mathrm{diag}(\mu_1, \ldots, \mu_s) =: D$$

with a regular matrix $T \in \mathbb{R}^{s \times s}$. We apply the transformation

$$W := (T^{-1} \otimes I_n)Z, \qquad \Delta W := (T^{-1} \otimes I_n)\Delta Z.$$

The Newton iteteration reads (with $I_{sn} = I_s \otimes I_n$)

$$((I_s \otimes I_n) - h(A \otimes J))(T \otimes I_n)\Delta W^{(\nu)} = -G((T \otimes I_n)W^{(\nu)})$$

or, equivalently, due to the product rule $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$

$$((T \otimes I_n) - h((AT) \otimes J))\Delta W^{(\nu)} = -G((T \otimes I_n)W^{(\nu)}).$$

Multiplication with $(T^{-1}A^{-1}) \otimes I_n$ from the left produces

$$\begin{aligned} & ((T^{-1}A^{-1}T) \otimes I_n - h((T^{-1}A^{-1}AT) \otimes J))\Delta W^{(\nu)} \\ = \; & -((T^{-1}A^{-1}) \otimes I_n)G((T \otimes I_n)W^{(\nu)}). \end{aligned} \tag{5.9}$$

Thus the matrix in the left-hand side exhibits a block diagonal structure

$$(D \otimes I_n - h(I_s \otimes J)) = \begin{pmatrix} \mu_1 I_n - hJ & & & 0 \\ & \mu_2 I_n - hJ & & \\ & & \ddots & \\ 0 & & & \mu_s I_n - hJ \end{pmatrix}.$$

Thus the linear system of order sn is decoupled and just s linear systems of order n have to be solved. The computational effort for the involved LU-decompositions is $\sim sn^3$ in comparison to the effort $\sim s^3n^3$ for an LU-decompsition of the original matrix. Hence the amount of computational work is reduced by the factor $\frac{1}{s^2}$. Each evaluation of the right-hand side in (5.9) demands two transformations, which correspond to multiplications by $R \otimes I_n$ with some matrix $R \in \mathbb{R}^{s\times s}$. The computational effort of a matrix-vector multiplication becomes $\sim s^2n$ due to this sparse structure, which is small in comparison to n^3 for large n.

However, often not all eigenvalues of the matrix A are real. Pairs of complex conjugate eigenvalues may appear. We consider just one pair, for example. Let $\mu_1 = \alpha + \mathrm{i}\beta$, $\mu_2 = \alpha - \mathrm{i}\beta$. It follows

$$S^{-1}A^{-1}S = \begin{pmatrix} \alpha & -\beta & & & 0 \\ \beta & \alpha & & & \\ & & \mu_3 & & \\ & & & \ddots & \\ 0 & & & & \mu_s \end{pmatrix}$$

with a regular matrix $S \in \mathbb{R}^{s\times s}$. Solving the linear system

$$\begin{pmatrix} \alpha I_n - hJ & -\beta I_n \\ \beta I_n & \alpha I_n - hJ \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix} \tag{5.10}$$

demands $\sim (2n)^3 = 8n^3$ operations. Alternatively, we can apply a matrix $T \in \mathbb{C}^{s\times s}$ to transform the linear system in block diagonal form. Then $\sim 2n^3$ complex operations are necessary for the two LU-decompositions of order n. Since a complex multiplication demands four real multiplications, the computational work becomes $\sim 8n^3$ real operations again. We do not

safe effort, since we have not applied the specific structure of the linear system (5.10) yet. Alternatively, the complex-valued linear system

$$((\alpha + \mathrm{i}\beta)I_n - hJ)(u + \mathrm{i}v) = a + \mathrm{i}b$$

yields the solution of (5.10). The computational effort of the LU-decomposition becomes $\sim n^3$ complex operations, which corresponds to $\sim 4n^3$ real operations. Hence the computational work is just half times the effort of solving the system (5.10) directly.

The formula of an IRK can be written as a fixed point problem $Y = \Phi(Y)$ with the vector of unknowns $Y := (\tilde{y}_1, \ldots, \tilde{y}_s) \in \mathbb{R}^{sn}$. The function $\Phi : \mathbb{R}^{sn} \to \mathbb{R}^{sn}$ exhibits the contractivity condition

$$\|\Phi(Y) - \Phi(Z)\|_\infty \leq hLs \left(\max_{i,j=1,\ldots,s} |a_{ij}| \right) \|Y - Z\|_\infty$$

with the Lipschitz-constant L from (2.3). We obtain a step size restriction $h < C/L$ for the convergence of the fixed point iteration. For stiff problems, the constant L is large and thus tiny step sizes have to be applied. For example, a linear system $y' = Jy$ implies the constant $L = \|J\|$ in an arbitrary matrix norm. If eigenvalues with a large negative real part appear, then $\|J\|_2$ is large (and thus also the other matrix norms).

We motivate that such a step size restriction does not appear in the convergence of Newton iterations. For example, we consider the implicit Euler method $y_1 = y_0 + hf(x_1, y_1)$. The simplified Newton iteration reads

$$y_1^{(\nu+1)} = \Psi(y_1^{(\nu)}) := y_1^{(\nu)} - (I - hJ)^{-1} \left(y_1^{(\nu)} - y_0 - hf(x_1, y_1^{(\nu)}) \right)$$

with the Jacobian matrix $J := \mathrm{D}f(x_0, y_0)$. The Newton method represents a fixed point iteration with the function Ψ. Hence the iteration converges if $\|\mathrm{D}\Psi\| < 1$ holds in some domain. We obtain

$$\mathrm{D}\Psi(y) = I - (I - hJ)^{-1} (I - h\mathrm{D}f(x_1, y)) .$$

We decompose $f(x, y) = Jy + g(x, y)$ with $g = f - Jy$. We assume that the part Jy is stiff, whereas g is non-stiff. It follows

$$\begin{aligned} \|\mathrm{D}\Psi(y)\| &= \|I - (I - hJ)^{-1}(I - hJ - h\mathrm{D}g(x_1, y))\| = \|h(I - hJ)^{-1}\mathrm{D}g(x_1, y)\| \\ &\leq h \cdot \|(I - hJ)^{-1}\| \cdot \|\mathrm{D}g(x_1, y)\|. \end{aligned}$$

If λ is an eigenvalue of J, then $\hat{\lambda} := \frac{1}{1-h\lambda}$ is an eigenvalue of $(I - hJ)^{-1}$. In the stiff case, small and large eigenvalues λ appear. For large λ, $\hat{\lambda}$ will be small. For small λ, $\hat{\lambda}$ will be

around one. Thus we can assume $\|(I - hJ)^{-1}\| \leq 1 + c$, where the constant c does not depend on the stiffness of the system. We obtain

$$\|\mathrm{D}\Psi(y)\| < 1 \quad \text{for} \quad h < \frac{1}{(1+c)\|\mathrm{D}g\|}.$$

Thus there is no step size restriction by the stiffness of the system. The matrix $(I - hJ)^{-1}$ acts like a filter, which is filtering out the stiff part.

Diagonal implicit Runge-Kutta methods

We search for A-stable Runge-Kutta methods with a lower computational effort now. Since an explicit scheme cannot be A-stable, we still require implicit Runge-Kutta (IRK) methods. We consider the general form

$$\tilde{y}_i = y_0 + \sum_{j=1}^{s} a_{ij} f(x_j, \tilde{y}_j) \qquad \text{for } \; i = 1, \ldots, s$$

with $x_i := x_0 + c_i h$. Now let $A = (a_{ij})$ be a lower triangular matrix with non-zero entries on the diagonal, i.e., $a_{ij} = 0$ for $i < j$. The corresponding Butcher-tableau reads:

$$\begin{array}{c|cccc} c_1 & a_{11} & 0 & \cdots & 0 \\ \vdots & a_{21} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ c_s & a_{s1} & \cdots & a_{s,s-1} & a_{ss} \\ \hline & b_1 & \cdots & \cdots & b_s \end{array}$$

These schemes are called diagonal implicit Runge-Kutta (DIRK) methods. The formula for the unknown intermediate values $\tilde{y}_i$ becomes

$$\tilde{y}_i - h a_{ii} f(x_i, \tilde{y}_i) = y_0 + h \sum_{j=1}^{i-1} a_{ij} f(x_j, \tilde{y}_j)$$

for $i = 1, \ldots, s$. Hence the unknown intermediate values can be computed successively by solving s nonlinear systems of dimension n. In a Newton iteration, each nonlinear system can be written as

$$G_i :\equiv \tilde{y}_i - h a_{ii} f(x_i, \tilde{y}_i) - y_0 - h \sum_{j=1}^{i-1} a_{ij} f(x_j, \tilde{y}_j) = 0$$

for $i = 1, \ldots, s$. The involved Jacobian matrices become

$$\mathrm{D}G_i = I_n - ha_{ii}\mathrm{D}f(x_i, \tilde{y}_i).$$

We apply a simplified Newton iteration with the matrices

$$\widetilde{\mathrm{D}G_i} := I_n - ha_{ii}\mathrm{D}f(x_0, y_0) \qquad \text{for } i = 1, \ldots, s.$$

Now just one Jacobian matrix of f has to be evaluated in the complete integration step. Nevertheless, the LU-decompositions of s different matrices have to be computed. We can save more computational work in case of singly diagonal implicit Runge-Kutta (SDIRK) methods, which are characterised by the property

$$\gamma := a_{11} = a_{22} = \cdots = a_{ss}.$$

Thus we just need to compute the LU-decomposition of

$$(L \cdot U) := I - h\gamma\mathrm{D}f(x_0, y_0).$$

According SDIRK schemes exist, which are A-stable and thus appropriate for stiff problems. They exhibit a lower computational effort than IRK methods like Gauss schemes or Radau schemes. However, the maximum order of an SDIRK method is $p \leq s + 1$.

5.5 Rosenbrock-Wanner methods

Rosenbrock-Wanner (ROW) schemes are implicit integration techniques, which are similar to Runge-Kutta methods. The aim of Rosenbrock-Wanner methods is a further reduction of the computational effort in DIRK and SDIRK method.

In SDIRK methods, several nonlinear systems are resolved successively. The solution of nonlinear systems will be avoided now by the construction of according linear systems. For example, just a single step of the Newton iteration can be done for each nonlinear system in an SDIRK method. The resulting schemes are called linearly implicit Runge-Kutta methods.

Construction of ROW methods

The formulation of ROW methods is based on autonomous systems of ODEs $y' = f(y)$. The system can be written in the form

$$y'(x) = Jy(x) + (f(y(x)) - Jy(x)) \tag{5.11}$$

using the (constant) Jacobian matrix $J := \mathrm{D}f(y_0)$. The first term Jy (stiff part) is discretised by a diagonal implicit Runge-Kutta method, whereas the second term $f(y) - Jy$ (nonstiff part) is resolved by an explicit Runge-Kutta method. Based on the notation (3.14), we obtain

$$k_i = J\left(h\sum_{j=1}^{i} a_{ij}k_j\right) + f\left(y_0 + h\sum_{j=1}^{i-1}\alpha_{ij}k_j\right) - J\left(h\sum_{j=1}^{i-1}\alpha_{ij}k_j\right)$$

and thus

$$(I - ha_{ii}J)k_i = f\left(y_0 + h\sum_{j=1}^{i-1}\alpha_{ij}k_j\right) + hJ\sum_{j=1}^{i-1}(a_{ij} - \alpha_{ij})k_j$$

for $i = 1, \ldots, s$. We have achieved a linear system for each unknown increment k_i. We set $\gamma = a_{ii}$ for all i using some parameter $\gamma \in \mathbb{R}$ and define $\gamma_{ij} = a_{ij} - \alpha_{ij}$. Now the Rosenbrock-Wanner method reads

$$\begin{aligned} (I - h\gamma J)k_i &= f\left(y_0 + h\sum_{j=1}^{i-1}\alpha_{ij}k_j\right) + hJ\sum_{j=1}^{i-1}\gamma_{ij}k_j,\ i = 1, \ldots, s\ , \\ y_1 &= y_0 + h\sum_{i=1}^{s} b_i k_i. \end{aligned} \tag{5.12}$$

The Rosenbrock-Wanner scheme is determined by its coefficients (α_{ij}), (γ_{ij}) and γ. The matrix in the linear systems is the same for all i. Hence just a single LU-decomposition has to be computed in each step of the integration.

A non-autonomous system $y'(x) = f(x, y(x))$ can be written in the autonomous form

$$\tilde{y}'(\tau) = \begin{pmatrix} y'(\tau) \\ x'(\tau) \end{pmatrix} = \begin{pmatrix} f(x(\tau), y(\tau)) \\ 1 \end{pmatrix} = \tilde{f}(\tilde{y}(\tau)).$$

Application of the ROW scheme yields

$$(I - h\gamma\tilde{J})\tilde{k}_i = \begin{pmatrix} f\left(x_0 + h\sum_{j=1}^{i-1}\alpha_{ij}\cdot 1, y_0 + h\sum_{j=1}^{i-1}\alpha_{ij}k_j\right) \\ 1 \end{pmatrix} + h\tilde{J}\sum_{j=1}^{i-1}\gamma_{ij}\tilde{k}_j$$

with the Jacobian matrix

$$\tilde{J} = \begin{pmatrix} \mathrm{D}_y f & \mathrm{D}_x f \\ 0 & 0 \end{pmatrix}, \qquad \mathrm{D}_y f := \frac{\partial f}{\partial y}(x_0, y_0), \quad \mathrm{D}_x f := \frac{\partial f}{\partial x}(x_0, y_0).$$

The last equation implies that the $(n+1)$th component of $\tilde{k}_i$ is equal to one. It follows

$$(I - h\gamma\mathrm{D}_y f)k_i = f\left(x_0 + \alpha_i h, y_0 + h\sum_{j=1}^{i-1}\alpha_{ij}k_j\right) + h\gamma_i\mathrm{D}_x f + h\mathrm{D}_y f\sum_{j=1}^{i-1}\gamma_{ij}k_j$$

with the coefficients

$$\alpha_i := \sum_{j=1}^{i-1}\alpha_{ij}, \qquad \gamma_i := \gamma + \sum_{j=1}^{i-1}\gamma_{ij} \qquad \text{for } \; i = 1, \ldots, s.$$

In comparison to SDIRK methods, we do not have to solve several linear systems in each Newton iteration, where the computational effort is $\sim n^2$ for each linear system (since the LU-decomposition is already given). Moreover, just s evaluations of the right-hand side f are required in each integration step. However, the ROW method (5.12) includes a matrix-vector product on the right-hand side with computational work $\sim n^2$. The computation of this product can be omitted by a combination with the linear system.

It remains to investigate this class of methods with respect to consistency and A-stability.

Order conditions

As in the case of general Runge-Kutta methods, cf. Sect. 3.5, Taylor expansion can be used to determine the order of consistency for ROW methods.

The coefficients, which define the ROW scheme, have to satisfy certain order conditions. Remark that $\alpha_{ij} = 0$ for $j \geq i$, $\gamma_{ij} = 0$ for $j > i$ and $\gamma_{ii} = \gamma$. The conditions up to order 3 read:

$p = 1:$	$\sum_{i=1}^{s} b_i = 1$
$p = 2:$	$\sum_{i,j=1}^{s} b_i(\alpha_{ij} + \gamma_{ij}) = \frac{1}{2}$
$p = 3:$	$\sum_{i,j,k=1}^{s} b_i \alpha_{ij} \alpha_{ik} = \frac{1}{3}$
	$\sum_{i,j,k=1}^{s} b_i(\alpha_{ij} + \gamma_{ij})(\alpha_{jk} + \gamma_{jk}) = \frac{1}{6}$

If we replace $a_{ij} = \alpha_{ij} + \gamma_{ij}$, then the order conditions coincide with the conditions of the corresponding SDIRK method. ROW methods of each order p can be constructed by choosing a sufficiently large stage number s.

A-stability

The construction of the ROW method (5.12) is based on the decomposition (5.11) in a linear and a nonlinear part. It follows that the ROW method reduces to the underlying SDIRK method with coefficients $a_{ij} = \alpha_{ij} + \gamma_{ij}$ in case of linear systems $f(y) = Jy$. Since Dahlquist's test equation (5.4) is linear, the stability function of the ROW method coincides with the stability function of the SDIRK method. We use the formula (5.7) with the matrix $A = (a_{ij})$ and the stability function of the ROW method becomes

$$R(z) = 1 + zb^{\top}(I - zA)^{-1}\mathbb{1}.$$

The maximal order of consistency of an SDIRK method is $p = s + 1$. If we demand that the SDIRK method has the order $p = s$, then the stability function $R(z)$ depends on the parameter γ only. A-stability implies a condition $0 < \gamma_{\min} \leq \gamma \leq \gamma_{\max}$ for this parameter. L-stability yields a further condition for γ.

Example

The first implementations of ROW schemes, GRK4A and GRK4T (Generalised Runge-Kutta), are due to Kaps and Rentrop (1979). The ROW method of Shampine and Reichelt (1996) with $s = 2$ stages is defined by the coefficients

$$\gamma = \tfrac{1}{2+\sqrt{2}}, \quad \alpha_{21} = \tfrac{1}{2}, \quad \gamma_{21} = -\gamma, \quad b_1 = 0, \quad b_2 = 1.$$

The order of consistency is $p = 2$. For step size control, see Sect. 3.7, an additional stage is evaluated via

$$\begin{aligned}(I - h\gamma J)k_3 \;&=\; f(x_1, y_1) - (6+\sqrt{2})(k_2 - f(x_0 + \alpha_{21}h, y_0 + h\alpha_{21}k_1)) \\ &\quad - 2(k_1 - f(x_0, y_0)) + h\gamma \mathrm{D}_x f(x_0, y_0).\end{aligned}$$

Thereby, an FSAL (first same as last) strategy can be applied as in Runge-Kutta-Fehlberg methods. The local error is estimated by

$$\hat{y}_1 - y_1 = h\tfrac{1}{6}(k_1 - 2k_2 + k_3).$$

Both approximations y_1 and $\hat{y}_1$ are A-stable. However $\hat{y}_1$ is not L-stable, whereas y_1 is L-stable. Hence the lower-order approximation y_1 is used as the output of the method.

5.6 A-stability for multistep methods

Now we investigate linear multistep methods (4.6) with respect to stiff problems. The multistep method is (numerically) stable, if its characteristic polynomial satisfies the root condition.

We apply a linear k-step method (4.6) to Dahlquist's test equation (5.4). It follows a homogeneous linear difference equation

$$\sum_{l=0}^{k} \alpha_l y_{i+l} = h \sum_{l=0}^{k} \beta_l \lambda y_{i+l} \qquad \Rightarrow \qquad \sum_{l=0}^{k} (\alpha_l - h\lambda\beta_l)\, y_{i+l} = 0.$$

With $z := h\lambda$, we define the characteristic polynomials

$$q_z : \mathbb{C} \to \mathbb{C}, \qquad q_z(\xi) = \sum_{l=0}^{k} (\alpha_l - z\beta_l)\xi^l.$$

Due to Theorem 8, all solutions of the linear difference equation are bounded if and only if the polynomial satisfies the root condition. The roots $\xi_1, \ldots, \xi_k$ of q_z depend on z.

We consider $\mathrm{Re}(\lambda) \leq 0$ in Dahlquist's equation (5.4). It follows that the exact solutions do not increase. In particular, the exact solutions are bounded. The numerical solution of a multistep method may increase slightly. Thus we just demand that the numerical solution is bounded. The root condition leads to the following definition.

Definition 14 (stability domain of multistep methods)
The stability domain $S \subset \mathbb{C}$ of a linear multistep method is

$$S := \{z \in \mathbb{C} : \textit{all roots of } q_z \textit{ fulfill } |\xi_l| \leq 1 \textit{ and } |\xi_l| < 1 \textit{ for multiple roots}\}.$$

Now we characterise A-stability for multistep methods like for one-step schemes.

Definition 15 (A-stability of multistep methods) *A linear multistep method is called A-stable if its stability domain satisfies $\mathbb{C}^- \subseteq S$.*

We show that this definition coincides with the A-stability of one-step methods in the intersection of both classes of methods. A linear multistep method with $k = 1$ steps reads

$$\alpha_1 y_1 + \alpha_0 y_0 = h\left[\beta_1 f(x_1, y_1) + \beta_0 f(x_0, y_0)\right].$$

On the one hand, the linear polynomial

$$q_z(\xi) = (\alpha_1 - z\beta_1)\xi + (\alpha_0 - z\beta_0)$$

has just the single root

$$\xi_1(z) = \frac{-\alpha_0 + z\beta_0}{\alpha_1 - z\beta_1}.$$

On the other hand, the application to Dahlquist's test equation yields

$$\alpha_1 y_1 + \alpha_0 y_0 = h\left[\beta_1 \lambda y_1 + \beta_0 \lambda y_0\right] \quad \Rightarrow \quad y_1 = \frac{-\alpha_0 + z\beta_0}{\alpha_1 - z\beta_1}\, y_0.$$

It follows that the stability function $R(z)$ of this one-step method coincides with the root $\xi_1(z)$ of the polynomial. Consequently, the conditions of A-stability $|R(z)| \leq 1$ and $|\xi_1(z)| \leq 1$ for all z with $\mathrm{Re}(z) \leq 0$ are equivalent.

Nevertheless, the concept of A-stability for multistep methods is slightly weaker than the definition of A-stability for one-step methods in case of $k > 1$ steps. The specific case $z = 0$ corresponds to the numerical stability of the multistep methods, cf. Sect. 4.3.

Again explicit multistep methods are not appropriate for stiff problems.

Theorem 15 *A convergent explicit linear multistep method is not A-stable.*

Proof:

An explicit linear multistep method (4.6) has the property $\beta_k = 0$. Without loss of generality, we assume $\alpha_k = 1$. The characteristic polynomial from the application to Dahlquist's test equation (5.4) reads

$$q_z(\xi) = \xi^k + (\alpha_{k-1} - z\beta_{k-1})\xi^{k-1} + \cdots + (\alpha_1 - z\beta_1)\xi + (\alpha_0 - z\beta_0).$$

The polynomial can be written in the form

$$q_z(\xi) = (\xi - \xi_1(z))(\xi - \xi_2(z)) \cdots (\xi - \xi_k(z))$$

with the roots $\xi_1, \ldots, \xi_k \in \mathbb{C}$ depending on z. Since the method is convergent, at least one coefficient $\beta_l \neq 0$ appears. It follows

$$|\alpha_l - z\beta_l| \overset{|z|\to\infty}{\longrightarrow} \infty.$$

Thus one coefficient of q_z becomes unbounded. Vieta's theorem implies that at least one root $\xi_j(z)$ must be unbounded in this case. Consequently, this root violates the condition in the Definition 14 of the stability domain S for $\mathrm{Re}(z) \to -\infty$. Hence it does not hold $\mathbb{C}^- \subseteq S$. □

In case of an implicit linear multistep method, it holds $\beta_k \neq 0$ and the characteristic polynomial becomes

$$q_z(\xi) = (\alpha_k - z\beta_k)\xi^k + (\alpha_{k-1} - z\beta_{k-1})\xi^{k-1} + \cdots + (\alpha_1 - z\beta_1)\xi + (\alpha_0 - z\beta_0).$$

The roots of this polynomial are the same as for

$$\tilde{q}_z(\xi) = \xi^k + \frac{\alpha_{k-1} - z\beta_{k-1}}{\alpha_k - z\beta_k}\xi^{k-1} + \cdots + \frac{\alpha_1 - z\beta_1}{\alpha_k - z\beta_k}\xi + \frac{\alpha_0 - z\beta_0}{\alpha_k - z\beta_k}$$

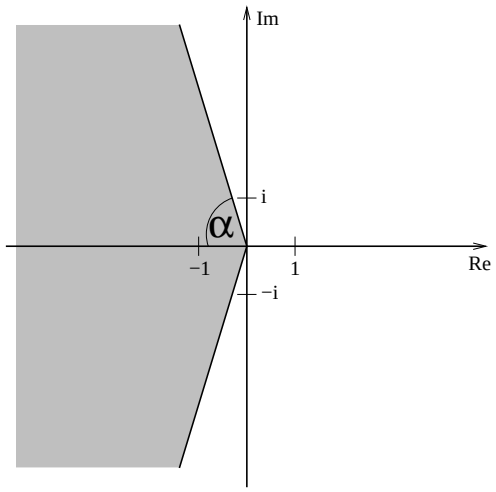

Figure 19: Domain $\mathbb{C}_\alpha$ for A(α)-stability.

provided that $\alpha_k - z\beta_k \neq 0$. Now the coefficients are rational functions in the variable z. The coefficient are bounded for $|z| \to \infty$. A-stable multistep methods are a subset of the implicit schemes. However, concerning A-stability of implicit multistep methods, a significant restriction appears.

Theorem 16 (second Dahlquist barrier) *A linear multistep method, which is convergent of order $p > 2$, cannot be A-stable.*

For a multistep method with k steps, we like to have an order of convergence $p \geq k$ (for example: Adams methods, BDF methods). Hence we do not achieve A-stable methods with $k > 2$ steps and order $p \geq k$.

The BDF methods for $k = 1$ and $k = 2$ are A-stable, whereas BDF schemes for $k \geq 3$ are not A-stable. Nevertheless, the BDF methods exhibit a good performance in solving stiff problems. The form of their stability domains suggests a modification of the concept of A-stability. For $0 \leq \alpha \leq \frac{\pi}{2}$, we define the domain

$$\mathbb{C}_\alpha := \left\{ z = |z| \cdot \mathrm{e}^{\mathrm{i}\varphi} \in \mathbb{C} : |\pi - \varphi| \leq \alpha \right\}.$$

Fig. 19 illustrates this domain.

Definition 16 (A(α)-stability) *A (one-step or multistep) method is called A(α)-stable if its stability domain S satisfies $\mathbb{C}_\alpha \subseteq S$.*

Of course, we characterise a method by the largest α, which still implies A(α)-stability. The specific case $\alpha = \frac{\pi}{2}$ corresponds to the ordinary A-stability due to $\mathbb{C}_{\pi/2} = \mathbb{C}^-$. If α is close to $\frac{\pi}{2}$, then the method is also suitable for stiff linear problems.

The k-step BDF methods feature the following maximum angles:

k	1	2	3	4	5	6
α	90°	90°	86.03°	73.35°	51.84°	17.84°

The BDF methods for $k \geq 7$ are not A(α)-stable for any angle $\alpha \geq 0$.

5.7 B-stability

The concept of A-stability concerns stiff linear systems of ODEs. Now we consider the nonlinear case. We investigate systems of ODEs $y' = f(x, y)$. If the Jacobian matrix $\mathrm{D}f$ exhibits large negative eigenvalues, then the system often behaves stiff. However, the solutions of a nonlinear system can exhibit a stiff behaviour, although the eigenvalues of the Jacobian matrix are all small. Thus another characterisation is necessary.

We assume that the system of ODEs satisfies the one-sided Lipschitz condition

$$\langle f(x, y) - f(x, z), y - z \rangle \leq \nu \|y - z\|^2 \tag{5.13}$$

with the constant $\nu \in \mathbb{R}$ and the Euclidean norm. It follows that corresponding solutions of initial value problems exhibit the contractivity property

$$\|y(x) - z(x)\| \leq \|y(x_0) - z(x_0)\| \cdot \mathrm{e}^{\nu(x - x_0)}.$$

If $\nu \leq 0$ holds, then the system is called dissipative. In this case, it follows

$$\|y(x) - z(x)\| \leq \|y(x_0) - z(x_0)\|. \tag{5.14}$$

The system exhibits a stiff behaviour for large negative ν.

In case of a linear system with $f(x, y) = Ay$, the condition (5.13) is equivalent to

$$\frac{\langle Av, v\rangle}{\langle v, v\rangle} \leq \nu \quad \text{for all } \; v \in \mathbb{R}^n \backslash \{0\}.$$

If the matrix A is symmetric, it follows

$$\nu = \max_{v \neq 0} \frac{v^\top A v}{v^\top v} = \lambda_{\max}(A).$$

If all eigenvalues are non-positive, then the linear system is dissipative.

The concept of B-stability demands that a numerical solution of a method inherits the property (5.14) of the exact solution without a step size restriction in case of dissipative systems.

Definition 17 (B-stability) *A one-step method is called B-stable if for all step sizes $h > 0$ and two initial values y_0, z_0 the approximation satisfies*

$$\|y_1 - z_1\| \leq \|y_0 - z_0\|$$

for all systems with $\langle f(x, y) - f(x, z), y - z\rangle \leq 0$.

A B-stable method is also A-stable: For $y' = \lambda y$ with $\lambda \in \mathbb{R}$, $\lambda \leq 0$, a B-stable one-step method implies

$$|R(h\lambda)| \cdot |y_0 - z_0| = |R(h\lambda)(y_0 - z_0)| = |y_1 - z_1| \leq |y_0 - z_0|$$

and thus $|R(h\lambda)| \leq 1$. In case of $\lambda = \alpha + \mathrm{i}\beta$ with $\alpha \leq 0$, we apply the real-valued system of ODEs

$$u' = Au \qquad \text{with} \quad u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \quad A = \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix},$$

which is dissipative due to $u^\top A u = \alpha(u_1^2 + u_2^2) \leq 0$. The conclusion follows from $u_1 = \mathrm{Re}(y)$ and $u_2 = \mathrm{Im}(y)$ for the solution of Dahlquist's equation, since $\|u\| = |y|$ holds.

Vice versa, an A-stable method is not necessarily B-stable. For example, linear implicit one-step methods (e.g. ROW methods) are never B-stable. B-stability can be found in the class of implicit Runge-Kutta methods only.

Theorem 17 *The Gauss-Runge-Kutta methods are B-stable.*

Proof:

Let w and $\tilde{w}$ be the polynomials of the collocation approach with initial values y_0 and z_0, respectively. We define $m(x) := \|w(x) - \tilde{w}(x)\|^2$. It holds

$$m'(x) = 2\langle w'(x) - \tilde{w}'(x), w(x) - \tilde{w}(x)\rangle$$

and thus in the collocation points $x_i := x_0 + c_i h$

$$m'(x_i) = 2\langle f(x_i, w'(x_i)) - f(x_i, \tilde{w}'(x_i)), w(x_i) - \tilde{w}(x_i)\rangle \leq 0$$

for $i = 1, \ldots, s$. It follows

$$\begin{aligned} m(x_0 + h) &= m(x_0) + \int_{x_0}^{x_0+h} m'(x)\, \mathrm{d}x \\ &= m(x_0) + h \sum_{i=1}^{s} b_i m'(x_i) \quad \leq \quad m(x_0), \end{aligned}$$

because the weights b_i are positive and m' is a polynomial of degree $2s-1$, which is approximated exactly by Gaussian quadrature. We obtain

$$\|y_1 - z_1\|^2 = m(x_0 + h) \leq m(x_0) = \|y_0 - z_0\|^2$$

and thus B-stability is valid. □

Moreover, the RadauIA and RadauIIA methods are B-stable.

For multistep methods, the concept of G-stability is defined. However, no linear multistep method of order $p > 2$ exists, which is G-stable.

Remark: The concept of B-stability is based on dissipative systems as test problems. However, not all dissipative systems behave stiff and not all stiff ODEs are dissipative.

5.8 Comparison of methods

In this section, we discuss the general properties of one-step methods in comparison to multistep methods. Each type has its own advantages and disadvantages.

First, we characterise the computational effort per integration step in the next table.

	Runge-Kutta method s stages	linear multistep method k steps
expl.	s evaluations of f	one evaluation of f
impl.	one Jacobian matrix of f LU-decomp.: $> sn^3$ operations ($\sim n^3$ op. for SDIRK, ROW) per Newton step: s evaluations of f s linear systems of dim. n	one Jacobian matrix of f LU-decomp.: $\sim n^3$ operations per Newton step: one evaluation of f one linear system of dim. n

In conclusion, one step of a Runge-Kutta methods is more expensive than one step of a multistep scheme. However, the accuracy of the methods has also to be considered.

The following table illustrates some advantages and disadvantages of the one-step methods versus the multistep methods.

	Runge-Kutta method		linear multistep method
$\ominus$	relatively large computational effort per step (depending on s)	$\oplus$	relatively small computational effort per step (independent of k)
$\oplus$	many coefficients ($s^2 + s$) (additional conditions can be fulfilled)	$\ominus$	just $2k + 1$ coefficients (low number of degrees of freedom)
$\oplus$	always (numerically) stable (no reduction of degrees of freedom)	$\ominus$	root condition required for stability (reduces the degrees of freedom, first Dahlquist barrier)
$\oplus$	high-order methods for stiff problems (A-stability and B-stability)	$\ominus$	only low-order methods are A-stable (second Dahlquist barrier), only A(α)-stable high-order methods
$\oplus$	robust step size selection	$\ominus$	stability condition demands small changes in step sizes (for example in BDF methods)
$\ominus$	no (efficient) order control	$\oplus$	efficient order control (straightforward to implement)

We cannot conclude that one-step schemes or multistep methods are better in general. It always depends on the system of ODEs, which is resolved, if some method is better than another technique.

Integrators in MATLAB

In the software package MATLAB (MATrix LABoratory), version 7.5.0 (R2007b), there are seven built-in functions for solving initial value problems of systems of ODEs $y' = f(x, y),\ y(x_0) = y_0$. The following table lists these algorithms. The type of the used method is specified. All involved methods have been introduced in the previous chapters. The schemes apply an automatic step size selection to control the local discretisation error. The table indicates the used strategy for estimating the local error. Furthermore, two methods apply order control.

code	method	step size control	order control
ode23	explicit Runge-Kutta	embedded scheme	no
ode45	explicit Runge-Kutta	embedded scheme	no
ode113	predictor-corrector Adams methods	free interpolants	yes order 1-13
ode23t	trapezoidal rule	free interpolants	no
ode23s	Rosenbrock-Wanner	embedded scheme	no
ode15s	NDF method optional: BDF	free interpolants	yes order 1-5
ode23tb	trapezoidal rule and BDF2 (alternating)	free interpolants	no

The table below gives an information, which methods can be used for stiff problems. Moreover, two algorithms can also solve differential algebraic equations (DAEs) of index 1. (Remark that most of the implicit schemes can be generalised to DAEs of index 1 or 2.)

code	problem type	DAEs (index 1)
ode23	non-stiff	no
ode45	non-stiff	no
ode113	non-stiff	no
ode23t	moderately stiff	yes
ode23s	stiff	no
ode15s	stiff	yes
ode23tb	stiff	no

All methods are able to integrate implicit systems of ODEs, i.e., problems of the form $My' = f(x, y)$ with a constant mass matrix M and $\det(M) \neq 0$. Furthermore, there is the algorithm `ode15i`, which resolves fully implicit systems of ODEs or DAEs. Numerical methods for implicit systems of ODEs and systems of DAEs are discussed in the next chapter.

More details on some of the above integrators in MATLAB can be found in the article: L.F. Shampine, M.W. Reichelt: The MATLAB ODE suite. SIAM Journal on Scientific Computing 18 (1997) 1, pp. 1-22.

Chapter 6

Methods for Differential Algebraic Equations

We consider initial values problems of systems of differential algebraic equations (DAEs), i.e., a mixture of ordinary differential equations and algebraic equations. Such mathematical models are typically large in technical applications.

6.1 Implicit ODEs

We observe implicit systems of ordinary differential equations, since they represent a first step towards differential algebraic equations. Consider the initial value problem

$$My'(x) = f(x, y(x)), \quad y(x_0) = y_0 \tag{6.1}$$

with unknown solution $y : \mathbb{R} \to \mathbb{R}^n$ and right-hand side $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$. Let $M \in \mathbb{R}^{n \times n}$ be a constant matrix with $M \neq I$. Often M is called the mass matrix. If M is the identity matrix, then the system (6.1) represents explicit ODEs. We distinguish two cases:

M regular:	(6.1) is a system of implicit ordinary differential equations,
M singular:	(6.1) is a system of differential algebraic equations.

In this section, we assume the case of implicit ODEs. Consequently, we can transform the system (6.1) into the explicit system

$$y'(x) = M^{-1} f(x, y(x)). \tag{6.2}$$

Each evaluation of the new right-hand side demands the solution of a linear system with the matrix M now. For example, the explicit Euler method yields the formula

$$y_1 = y_0 + hM^{-1} f(x_0, y_0).$$

Thus a linear system with matrix M has to be solved in each step of the integration. A corresponding LU-decomposition has to be calculated just once. Using an explicit Runge-Kutta method, we obtain a sequence of linear systems, which have to be solved for each increment, i.e.,

$$Mk_i = f\left(x_0 + c_i h, y_0 + h \sum_{j=1}^{i-1} a_{ij} k_j\right) \quad \text{for } \; i = 1, \ldots, s.$$

However, implicit ODEs are often stiff. Hence implicit methods have to be used. For example, the implicit Euler method applied to the system (6.2) yields the nonlinear system

$$y_1 = y_0 + hM^{-1} f(x_1, y_1)$$

for the unknown value y_1. Considering the nonlinear system

$$y_1 - hM^{-1} f(x_1, y_1) - y_0 = 0,$$

the corresponding simplified Newton iteration reads

$$\begin{aligned} (I - hM^{-1}\mathrm{D}f(x_1, y_1^{(0)}))\Delta y_1^{(\nu)} &= -y_1^{(\nu)} + hM^{-1} f(x_1, y_1^{(\nu)}) + y_0, \\ y_1^{(\nu+1)} &= y_1^{(\nu)} + \Delta y_1^{(\nu)}, \end{aligned}$$

where $\mathrm{D}f = \frac{\partial f}{\partial y}$ denotes the Jacobian matrix of f. We multiply the equation of the Newton iteration with M and achieve the equivalent formulation

$$(M - h\mathrm{D}f(x_1, y_1^{(0)}))\Delta y_1^{(\nu)} = M(y_0 - y_1^{(\nu)}) + hf(x_1, y_1^{(\nu)}). \tag{6.3}$$

Thus one linear system has to be solved for both explicit and implicit ODEs in each step of the iteration. Just an additional matrix-vector multiplication is necessary on the right-hand side of (6.3).

Likewise, an implicit Runge-Kutta method applied to (6.1) or (6.2) exhibits the relations

$$Mk_i = f\left(x_0 + c_i h, y_0 + h\sum_{j=1}^{s} a_{ij}k_j\right) \quad \text{for } i = 1, \ldots, s. \tag{6.4}$$

Given a nonlinear function f, a nonlinear system of sn equations for the unknown increments has to be solved as for explicit ODEs.

Hence the computational effort for implicit ODEs is not significantly higher than for explicit ODEs in case of implicit methods. The situation becomes more complicated, if the matrix M is not constant but depends on the independent variable or the unknown solution.

We distinguish the following cases (with increasing complexity):

- linear-implicit system of ODEs with constant mass matrix:
  $My'(x) = f(x, y(x))$
- linear-implicit system of ODEs with non-constant mass matrix:
  $M(x)y'(x) = f(x, y(x))$
- quasilinear implicit system of ODEs:
  $M(y(x))y'(x) = f(x, y(x))$ or $M(x, y(x))y'(x) = f(x, y(x))$
- fully implicit system of ODEs:
  $F(y'(x), y(x), x) = 0$,
  $F : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$, $(z, y, x) \mapsto F(z, y, x)$, $\det\left(\frac{\partial F}{\partial z}\right) \neq 0$

For an example of an implicit system of ODEs, see the Colpitts oscillator introduced in Sect. 1.2. The involved mass matrix is constant and regular. The system of ODEs exhibits a strongly stiff behaviour.

6.2 Linear DAEs

In this section, we consider linear systems of differential algebraic equations

$$Ay'(x) + By(x) = s(x), \qquad y(x_0) = y_0 \tag{6.5}$$

with unknown solution $y : \mathbb{R} \to \mathbb{R}^n$ and given input signal $s : \mathbb{R} \to \mathbb{R}^n$. We assume that the matrices $A, B \in \mathbb{R}^{n\times n}$ are constant. For $\det(A) \neq 0$, we obtain implicit ODEs, whereas $\det(A) = 0$ implies DAEs.

For simplicity, we assume $\det(B) \neq 0$ in the following. Stationary solutions of the DAEs (6.5) with some constant input $s \equiv s_0$ are characterised by $y' \equiv 0$. Hence a unique stationary solution is given by $y_0 = B^{-1}s_0$ in case of $\det(B) \neq 0$. We transform the system (6.5) to the equivalent system

$$B^{-1}Ay'(x) + y(x) = B^{-1}s(x). \tag{6.6}$$

We use $B^{-1}A = T^{-1}JT$ with the Jordan form J and the regular transformation matrix $T \in \mathbb{R}^{n\times n}$. Thus the system (6.6) is transformed to

$$\begin{aligned} TB^{-1}Ay'(x) + Ty(x) &= TB^{-1}s(x) \\ TB^{-1}AT^{-1}Ty'(x) + Ty(x) &= TB^{-1}s(x) \\ J(Ty(x))' + Ty(x) &= TB^{-1}s(x). \end{aligned} \tag{6.7}$$

The Jordan matrix J can be ordered such that it exhibits the form

$$J = \begin{pmatrix} R & 0 \\ 0 & N \end{pmatrix}, \qquad \begin{matrix} R \in \mathbb{R}^{n_1\times n_1}, \\ N \in \mathbb{R}^{n_2\times n_2}, \end{matrix} \qquad n_1 + n_2 = n, \tag{6.8}$$

where R contains all eigenvalues not equal to zero ($\det(R) \neq 0$) and N includes the eigenvalues equal to zero ($\det(N) = 0$). More precisely, N is a stricly upper triangular matrix. Hence N is nilpotent, i.e.,

$$N^{k-1} \neq 0, \quad N^k = 0 \qquad \text{for some } \; k \leq n_2. \tag{6.9}$$

We call k the nilpotency index of the linear DAE system (6.5). Since $\det(A) = 0$ holds, it follows $n_2 \geq 1$ and $k \geq 1$. The corresponding partitioning of the solution and the right-hand side reads

$$Ty = \begin{pmatrix} u \\ v \end{pmatrix}, \qquad TB^{-1}s = \begin{pmatrix} p \\ q \end{pmatrix} \tag{6.10}$$

with $u, p : \mathbb{R} \to \mathbb{R}^{n_1}$ and $v, q : \mathbb{R} \to \mathbb{R}^{n_2}$. Hence the system (6.5) is decoupled in two parts

$$\begin{aligned} Ru'(x) + u(x) &= p(x), \\ Nv'(x) + v(x) &= q(x). \end{aligned} \tag{6.11}$$

Since $\det(R) \neq 0$ holds, the first part represents an implicit ODE for the part u, which is equivalent to the linear explicit ODE

$$u'(x) = -R^{-1}u(x) + R^{-1}p(x).$$

The second part can be written as

$$\begin{aligned} v(x) &= q(x) - Nv'(x), \\ v^{(l)}(x) &= q^{(l)}(x) - Nv^{(l+1)}(x). \end{aligned}$$

We obtain successively together with $N^k = 0$

$$\begin{aligned} v(x) &= q(x) - Nv'(x), \\ &= q(x) - Nq'(x) + N^2v''(x) \\ &= q(x) - Nq'(x) + N^2q''(x) - N^3v^{(3)}(x) \\ &= \cdots \\ &= q(x) - Nq'(x) + N^2q''(x) - \cdots + (-1)^k N^k v^{(k+1)}(x) \\ &= \sum_{i=0}^{k-1} (-1)^i N^i q^{(i)}(x). \end{aligned} \tag{6.12}$$

Thus we achieve an algebraic relation for the part v depending on the higher derivatives of the input. The special case $N = 0$ yields $v(x) = q(x)$. We call u and v the differential and algebraic part, respectively. In particular, the initial value of the algebraic part follows from the input via

$$v(x_0) = \sum_{i=0}^{k-1} (-1)^i N^i q^{(i)}(x_0). \tag{6.13}$$

In contrast, the initial value $u(x_0) \in \mathbb{R}^{n_1}$ of the differential part can be chosen arbitrarily.

Differentiating the relation (6.12) one more time yields

$$v'(x) = \sum_{i=0}^{k-1} (-1)^i N^i q^{(i+1)}(x). \tag{6.14}$$

Hence by differentiating the system (6.5) k times, we obtain a system of ODEs for the part v.

If the source term includes a perturbation, i.e., the right-hand side changes into $\hat{s}(x) = s(x) + \delta(x)$, then the algebraic part reads

$$\hat{v}(x) = \sum_{i=0}^{k-1} (-1)^i N^i q^{(i)}(x) + \sum_{i=0}^{k-1} (-1)^i N^i \tilde{\delta}^{(i)}(x)$$

with transformed perturbations $\tilde{\delta} : \mathbb{R} \to \mathbb{R}^{n_2}$ due to (6.10). Thus also higher derivates of the perturbation influence the solution of the linear DAE system in case of $k > 1$.

Conclusions:

- To guarantee the existence of solutions of the linear DAEs (6.5), the right-hand side s has to be sufficiently smooth, namely $s \in C^{k-1}$. The algebraic part v may be just continuous and not smooth.
- Derivatives of perturbations in the right-hand side influence the solution of a perturbed system in case of nilpotency index $k \geq 2$.
- The initial values $y(x_0) = y_0$ of the system (6.5) cannot be chosen arbitrarily. A consistent choice is necessary regarding (6.13).

If the matrix B is singular, existence and uniqueness of solutions can still be obtained in case of a regular matrix pencil, i.e., $\det(\lambda A + B) \not\equiv 0$ holds. Take a fixed $\lambda \in \mathbb{R}$ such that $\det(\lambda A + B) \neq 0$. Now we transform the system (6.5) into

$$\begin{aligned} A(y'(x) - \lambda y(x)) + (\lambda A + B)y(x) &= s(x), \\ (\lambda A + B)^{-1} A(y'(x) - \lambda y(x)) + y(x) &= (\lambda A + B)^{-1} s(x). \end{aligned} \tag{6.15}$$

We use the Jordan form $(\lambda A + B)^{-1}A = T^{-1}JT$ with the structure (6.8). The transformation is analogue to (6.10). Consequently, the DAE system (6.5) is decoupled into the two parts

$$\begin{aligned} R(u'(x) - \lambda u(x)) + u(x) &= p(x), \\ N(v'(x) - \lambda v(x)) + v(x) &= q(x). \end{aligned} \tag{6.16}$$

The first part is equivalent to an explicit system of ODEs again. The second part can be written in the form

$$v(x) = (I - \lambda N)^{-1}q(x) - (I - \lambda N)^{-1}Nv'(x) = \tilde{q}(x) - \tilde{N}v'(x)$$

with $\tilde{q} := (I - \lambda N)^{-1}q$ and $\tilde{N} := (I - \lambda N)^{-1}N$. We arrange a von Neumann series to represent the inverse matrix

$$(I - \lambda N)^{-1} = \sum_{j=0}^{\infty} \lambda^j N^j = \sum_{j=0}^{k-1} \lambda^j N^j,$$

since $N^j = 0$ holds for all $j \geq k$. It follows

$$\tilde{N} = (I - \lambda N)^{-1}N = \sum_{j=0}^{k-2} \lambda^j N^{j+1}$$

and thus $\tilde{N}^{k-1} \neq 0,\ \tilde{N}^k = 0$ with the same k as in (6.9). Accordingly, we obtain the same results as in the case $\det(B) \neq 0$. However, we have not shown that the definition of the index k is unique in this case, i.e., k is independent of the choice of λ.

If $\det(\lambda A + B) \equiv 0$ holds, then either existence or uniqueness of solutions to the linear DAE system (6.5) is violated.

6.3 Index Concepts

The index of a system of DAEs represents an integer number, which characterises the qualitative differences of the DAE system in comparison to a system of ODEs. We distinguish the two cases

$$\begin{aligned} \text{index } k = 0: &\quad \text{system of ODEs,} \\ \text{index } k \geq 1: &\quad \text{system of DAEs.} \end{aligned}$$

The higher the index, the more the system of DAEs behaves different form a system of ODEs.

Several concepts for defining the index exist. We discuss two important approaches, namely the differential index and the perturbation index.

To define the index, we consider a general nonlinear system of differential algebraic equations

$$F(y'(x), y(x), x) = 0, \qquad y(x_0) = y_0 \tag{6.17}$$

with unknown solution $y : \mathbb{R} \to \mathbb{R}^n$ and $F : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n$. The predetermined initial values have to be consistent.

Differential Index

The system (6.17) represents ordinary differential equations, if the Jacobian matrix $\frac{\partial F}{\partial y'}$ is regular. We consider the extended system

$$\begin{aligned} F(y'(x), y(x), x) &= 0 \\ \frac{\mathrm{d}}{\mathrm{d}x} F(y'(x), y(x), x) &= 0 \\ \frac{\mathrm{d}^2}{\mathrm{d}x^2} F(y'(x), y(x), x) &= 0 \\ &\vdots \\ \frac{\mathrm{d}^k}{\mathrm{d}x^k} F(y'(x), y(x), x) &= 0 \end{aligned} \tag{6.18}$$

with $(k+1)n$ equations, which is achieved by a subsequent differentiation. In most cases, an explicit system of ODEs for the unknown solution in the form

$$y'(x) = G(y(x), x)$$

can be constructed from a larger system (6.18) by algebraic manipulations.

Definition 18 *The differential index of the system of DAEs (6.17) is the smallest integer $k \geq 0$ such that an explicit system of ODEs for the solution y can be constructed by algebraic manipulations using the extended system (6.18)*

The special case $k = 0$ implies that the system (6.17) is equivalent to an explicit system of ODEs, i.e., it is not a DAE.

As example, we discuss a semi-explicit system of DAEs

$$\begin{aligned} y'(x) &= f(y(x), z(x)), \quad y : \mathbb{R} \to \mathbb{R}^{n_1}, \quad f : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^{n_1}, \\ 0 &= g(y(x), z(x)), \quad z : \mathbb{R} \to \mathbb{R}^{n_2}, \quad g : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^{n_2}. \end{aligned} \tag{6.19}$$

The differential index of this system is always $k \geq 1$ provided that $n_2 > 0$. Differentiating the second part of the system yields

$$0 = \frac{\partial g}{\partial y} \cdot y'(x) + \frac{\partial g}{\partial z} \cdot z'(x) = \frac{\partial g}{\partial y} \cdot f(y(x), z(x)) + \frac{\partial g}{\partial z} \cdot z'(x).$$

If the Jacobian matrix $\frac{\partial g}{\partial z} \in \mathbb{R}^{n_2 \times n_2}$ is regular, then we obtain

$$z'(x) = -\left(\frac{\partial g}{\partial z}\right)^{-1} \cdot \frac{\partial g}{\partial y} \cdot f(y(x), z(x)).$$

Thus we achieve an explicit ODE for the solution y, z and the differential index results to $k = 1$. If the Jacobian matrix $\frac{\partial g}{\partial z}$ is singular, then the differential index satisfies $k \geq 2$ and further examinations are necessary.

This example indicates that the differential index possibly does not depend on the underlying system of DAEs only but also on the considered solution. Thus the same system may exhibit two different solutions with according indexes.

Perturbation Index

We observe a system of ODEs and a corresponding perturbed system

$$\begin{aligned} y'(x) &= f(x, y(x)), & y(x_0) &= y_0, \\ \hat{y}'(x) &= f(x, \hat{y}(x)) + \delta(x), & \hat{y}(x_0) &= \hat{y}_0. \end{aligned} \tag{6.20}$$

Let the function f be Lipschitz-continuous. We perform a similar analysis as in Sect. 2.3. However, we do not apply Gronwall's lemma now. The equivalent integral equations of (6.20) read

$$y(x) = y_0 + \int_{x_0}^{x} f(s, y(s)) \, \mathrm{d}s, \quad \hat{y}(x) = \hat{y}_0 + \int_{x_0}^{x} f(s, \hat{y}(s)) + \delta(s) \, \mathrm{d}s.$$

We consider an interval $I := [x_0, x_{\text{end}}]$. Let $R := x_{\text{end}} - x_0$. Subtracting the integral equations yields the estimate in the maximum norm

$$\begin{aligned}
\|\hat{y}(x) - y(x)\| &= \left\| \hat{y}_0 - y_0 + \int_{x_0}^{x} f(s, \hat{y}(s)) - f(s, y(s)) + \delta(s)\, \mathrm{d}s \right\| \\
&\leq \|\hat{y}_0 - y_0\| + \int_{x_0}^{x} \|f(s, \hat{y}(s)) - f(s, y(s))\|\, \mathrm{d}s + \left\| \int_{x_0}^{x} \delta(s)\, \mathrm{d}s \right\| \\
&\leq \|\hat{y}_0 - y_0\| + L \int_{x_0}^{x} \|\hat{y}(s) - y(s)\|\, \mathrm{d}s + \left\| \int_{x_0}^{x} \delta(s)\, \mathrm{d}s \right\| \\
&\leq \|\hat{y}_0 - y_0\| + L(x - x_0) \max_{s \in I} \|\hat{y}(s) - y(s)\| + \left\| \int_{x_0}^{x} \delta(s)\, \mathrm{d}s \right\| \\
&\leq \|\hat{y}_0 - y_0\| + LR \max_{s \in I} \|\hat{y}(s) - y(s)\| + \max_{s \in I} \left\| \int_{x_0}^{s} \delta(u)\, \mathrm{d}u \right\|
\end{aligned}$$

for all $x \in I$. Taking the maximum over all $x \in I$ on the left-hand side yields (provided that $LR < 1$)

$$\max_{x \in I} \|\hat{y}(x) - y(x)\| \leq \frac{1}{1 - LR} \left(\|\hat{y}_0 - y_0\| + \max_{s \in I} \left\| \int_{x_0}^{s} \delta(u)\, \mathrm{d}u \right\| \right).$$

Hence just the difference in the initial values and the integral of the perturbation give a contribution to the discrepancy of the two solutions. Furthermore, it holds the estimate

$$\max_{x \in I} \|\hat{y}(x) - y(x)\| \leq \frac{1}{1 - LR} \left(\|\hat{y}_0 - y_0\| + R \max_{s \in I} \|\delta(s)\| \right).$$

Given a general nonlinear system of DAEs (6.17) and a corresponding solution y on $I := [x_0, x_{\text{end}}]$, we consider the perturbed system

$$F(\hat{y}'(x), \hat{y}(x), x) = \delta(x), \qquad \hat{y}(x_0) = \hat{y}_0 \tag{6.21}$$

with sufficiently smooth perturbation $\delta : I \to \mathbb{R}^n$.

Definition 19 *The perturbation index of the system (6.17) corresponding to the solution y on an interval I is the smallest integer $k \geq 1$ such that an estimate*

$$\|\hat{y}(x) - y(x)\| \leq C \left(\|\hat{y}_0 - y_0\| + \sum_{l=0}^{k-1} \max_{s \in I} \|\delta^{(l)}(s)\| \right)$$

exists with a constant $C > 0$ for sufficiently small right-hand side. The perturbation index is $k = 0$ if an estimate of the form

$$\|\hat{y}(x) - y(x)\| \leq C \left(\|\hat{y}_0 - y_0\| + \max_{s \in I} \left\| \int_{x_0}^{s} \delta(u)\ du \right\| \right)$$

holds.

It can be shown that the perturbation index is $k = 0$ if and only if the system (6.17) represents explicit or implicit ODEs.

Remark that a perturbation can be small itself but exhibit large derivatives. For example, we discuss the function

$$\begin{aligned} \delta(x) &= \varepsilon \sin(\omega x), \\ \delta'(x) &= \varepsilon\omega \cos(\omega x). \end{aligned}$$

It holds $|\delta(x)| \leq \varepsilon$ for arbitrary $\omega \in \mathbb{R}$. However, we obtain $|\delta'(x)| \leq \varepsilon\omega$, which becomes large in case of $\omega \gg 1$ even if $\varepsilon > 0$ is tiny.

In view of this property, the numerical simulation of DAE models becomes critical in case of perturbation index $k \geq 2$, since derivatives of perturbations are involved. DAE systems of index $k = 1$ are well-posed, whereas DAE systems of index $k \geq 2$ are (strictly speaking) ill-posed. The higher the perturbation index becomes, the more critical is this situation. However, modelling electric circuits can be done by DAEs with index $k \leq 2$. The models of mechanical systems exhibit DAEs with index $k \leq 3$. In practice, mathematical models based on DAE systems with perturbation index $k > 3$ are avoided.

From the numerical point of view, the perturbation index is more interesting than the differential index, since it characterises the expected problems in

numerical methods. The result of a numerical technique can be seen as the exact solution of a perturbed system of DAEs (backward analysis). It is often difficult to determine the perturbation index of a system of DAEs, whereas the differential index is easier to examine.

For linear systems (6.5), the differential index and the perturbation index coincide and are equal to the nilpotency index. For a general nonlinear system (6.17), the two index concepts can differ arbitrarily. However, the differential index is equal to the perturbation index in many technical applications.

Examples: Electric Circuits

We discuss the differential index of two systems of DAEs, which result from modelling an electric circuit by a network approach. The two circuits are shown in Fig. 20.

The first circuit is an electromagnetic oscillator, which has already been introduced in Sect. 1.2. It consists of a capacitance C, an inductance L and a linear resistor R in parallel. The unknowns are the three currents I_C, I_L, I_R through the basic elements and the node voltage U depending on time. Each basic element is modelled by a current-voltage relation. Furthermore, Kirchhoff's current law is added. We obtain a linear system of DAEs

$$\begin{aligned} CU' &= I_C \\ LI_L' &= U \\ 0 &= U - RI_R \\ 0 &= I_C + I_L + I_R. \end{aligned} \tag{6.22}$$

We can eliminate the unknowns I_C, I_R such that a linear system of ODEs is achieved

$$\begin{aligned} CU' &= -I_L - \tfrac{1}{R}U \\ LI_L' &= U. \end{aligned} \tag{6.23}$$

Systems of the form (6.22) are arranged automatically by tools of computer aided design (CAD). In contrast, the advantageous description by ODEs

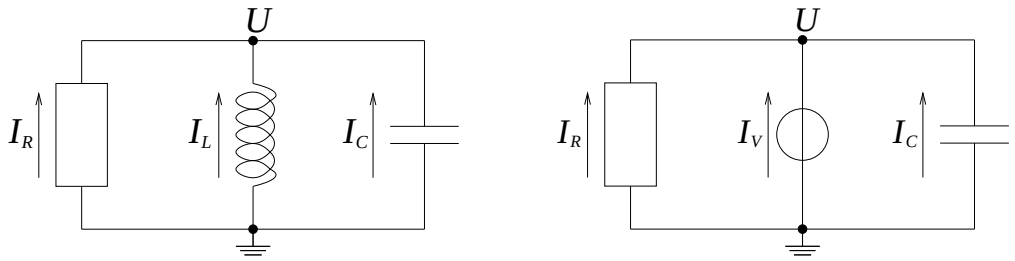

Figure 20: Example circuits.

like (6.23) has to be constructed by ourselves.

Differentiating the system (6.22) with respect to time yields

$$\begin{aligned} CU'' &= I_C' \\ LI_L'' &= U' \\ 0 &= U' - RI_R' \\ 0 &= I_C' + I_L' + I_R'. \end{aligned}$$

Hence we obtain an explicit system of ODEs for the unknowns

$$\begin{aligned} U' &= \tfrac{1}{C} I_C \\ I_L' &= \tfrac{1}{L} U \\ I_R' &= \tfrac{1}{R} U' = \tfrac{1}{RC} I_C \\ I_C' &= -I_L' - I_R' = -\tfrac{1}{L} U - \tfrac{1}{RC} I_C. \end{aligned}$$

Since just one differentiation is necessary to achieve this ODE system, the differential index of the DAE system (6.22) is $k = 1$.

Now we consider the second circuit, which consists of a capacitance C, an independent voltage source $V(t)$ and a linear resistor R. The corresponding DAE model reads

$$\begin{aligned} CU' &= I_C \\ 0 &= U - V(t) \\ 0 &= U - RI_R \\ 0 &= I_C + I_V + I_R. \end{aligned} \tag{6.24}$$

If the input voltage $V(t)$ is smooth, the solution can be calculated analytically

$$U = V(t), \quad I_R = \tfrac{1}{R}V(t), \quad I_C = CV'(t), \quad I_V = -CV'(t) - \tfrac{1}{R}V(t).$$

Furthermore, we arrange an explicit system of ODEs for the unknowns starting from the DAE system (6.24)

$$\begin{aligned} U' &= \tfrac{1}{C}I_C \\ I_R' &= \tfrac{1}{R}U' = \tfrac{1}{RC}I_C \\ I_C' &= CU'' = CV''(t) \\ I_V' &= -I_C' - I_R' = -CV''(t) - \tfrac{1}{RC}I_C. \end{aligned}$$

In this case, two differentiations of the system (6.24) with respect to time are required, since the relation $U'' = V''$ is used. Hence the differential index of the DAE system (6.24) is $k = 2$.

6.4 Methods for General Systems

In the next two sections, we outline the construction of numerical techniques for systems of DAEs. The numerical methods represent generalisations of corresponding schemes for systems of ODEs introduced in the previous chapters.

We consider initial value problems of fully implicit systems of DAEs (6.17), i.e., the most general form. The initial values have to be consistent with respect to the DAEs. We apply a grid $x_0 < x_1 < x_2 < \cdots < x_m$. Corresponding approximations $y_i \doteq y(x_i)$ of the solution are determined recursively by a numerical method.

Linear multistep methods

In case of systems of ODEs $y' = f(x, y)$, a linear multistep method is defined in (4.6) for equidistant step sizes. Since $y' = f$ holds, we can rewrite the

scheme as

$$\sum_{l=0}^{k} \alpha_l y_{i+l} = h \sum_{l=0}^{k} \beta_l z_{i+l}, \tag{6.25}$$

where $z_{i+l} = f(x_{i+l}, y_{i+l})$ represents an approximation of $y'(x_{i+l})$. In case of general DAE systems, this value is obtained by solving the nonlinear system (6.17). It follows the method

$$\begin{aligned} \sum_{l=0}^{k} \alpha_l y_{i+l} &= h \sum_{l=0}^{k} \beta_l z_{i+l} \\ F(z_{i+k}, y_{i+k}, x_{i+k}) &= 0 \end{aligned} \tag{6.26}$$

with the unknowns y_{i+k}, z_{i+k} in each step.

The BDF methods, see Sect. 4.5, are suitable for solving systems of DAEs. The k-step BDF scheme reads

$$\sum_{l=0}^{k} \alpha_l y_{i+l} = h z_{i+k}.$$

(Remark that the numbering of the coefficients is opposite to (4.29)). Hence we can replace z_{i+l} in $F(z_{i+k}, y_{i+k}, x_{i+k})$ by this formula. Consequently, the method (6.26) exhibits the simple form

$$F\left(\frac{1}{h}\sum_{l=0}^{k} \alpha_l y_{i+l}, y_{i+k}, x_{i+k}\right) = 0$$

with then unknown y_{i+k}. The BDF methods for fully implicit DAE systems (6.17) are implemented in the FORTRAN code DASSL (Differential Algebraic System SoLver) by Petzold (1982).

Although the trapezoidal rule represents a one-step method, we can write it in the form (6.25)

$$y_{i+1} - y_i = h\left[\tfrac{1}{2} z_i + \tfrac{1}{2} z_{i+1}\right] \quad \Rightarrow \quad z_{i+1} = -z_i + \tfrac{2}{h}(y_{i+1} - y_i).$$

Inserting z_{i+1} in $F(z_{i+1}, y_{i+1}, x_{i+1})$ yields the scheme

$$F\left(-z_i + \tfrac{2}{h}(y_{i+1} - y_i), y_{i+1}, x_{i+1}\right) = 0 \tag{6.27}$$

with the unknown y_{i+1}. The value z_i is known from the previous step.

Runge-Kutta Methods

We consider a Runge-Kutta method given in (3.14) for systems of ODEs. An approximation of the solution at the intermediate points is achieved via

$$y(x_0 + c_i h) \doteq y_0 + h \sum_{j=1}^{s} a_{ij} k_j \qquad \text{for } \; i = 1, \ldots, s.$$

Due to $y' = f$, the increments k_i represent approximations of the derivatives $y'(x_0 + c_i h)$, i.e.,

$$y'(x_0 + c_i h) \doteq k_i = f\left(x_0 + c_i h, y_0 + h \sum_{j=1}^{s} a_{ij} k_j\right) \qquad \text{for } \; i = 1, \ldots, s.$$

Now we solve general DAE systems. We apply the nonlinear system (6.17) for the determination of these derivatives again. It holds exactly

$$F\left(y'(x_0 + c_i h), y(x_0 + c_i h), x_0 + c_i h\right) = 0 \qquad \text{for } \; i = 1, \ldots, s.$$

It follows the numerical method

$$\begin{aligned} &F\left(k_i, y_0 + h\sum_{j=1}^{s} a_{ij} k_j, x_0 + c_i h\right) = 0 \quad \text{for } \; i = 1, \ldots, s\,, \\ &y_1 = y_0 + h\sum_{i=1}^{s} b_i k_i. \end{aligned} \tag{6.28}$$

In case of systems $My' = f(x, y)$, the technique (6.28) results just to (6.4).

As example, we consider the trapezoidal rule again. The coefficients of this Runge-Kutta scheme are $c_1 = 0$, $c_2 = 1$, $a_{11} = a_{12} = 0$, $a_{21} = a_{22} = b_1 = b_2 = \frac{1}{2}$. It follows

$$\begin{aligned} F(k_1, y_0, x_0) &= 0 \\ F(k_2, y_0 + h(\tfrac{1}{2}k_1 + \tfrac{1}{2}k_2), x_1) &= 0 \\ y_0 + h(\tfrac{1}{2}k_1 + \tfrac{1}{2}k_2) &= y_1. \end{aligned}$$

If we replace k_2 by the other values, then the second equation changes into

$$F\left(-k_1 + \frac{2}{h}(y_1 - y_0), y_1, x_1\right) = 0.$$

Hence the method coincides with (6.27).

6.5 Methods for Semi-Explicit Systems

In case of semi-explicit systems of DAEs (6.19), methods for systems of ODEs can be generalised immediately. Two approaches exist for this purpose.

Direct Approach (ε-embedding)

The semi-explicit system of DAEs (6.19) is embedded into a family of systems of ODEs

$$\begin{array}{rcl} y'(x) &=& f(y(x), z(x)), \\ \varepsilon z'(x) &=& g(y(x), z(x)), \end{array} \quad \Leftrightarrow \quad \begin{array}{rcl} y'(x) &=& f(y(x), z(x)), \\ z'(x) &=& \frac{1}{\varepsilon} g(y(x), z(x)). \end{array} \tag{6.29}$$

The original DAE is recovered for $\varepsilon \to 0$. Systems of the form (6.29) are also called singularly perturbed systems. Systems of DAEs can be seen as the limit case of stiff systems, where the amount of stiffness becomes infinite.

As an example, we consider the Van-der-Pol oscillator

$$y'' + \mu^2((y^2 - 1)y' + y) = 0 \quad \Leftrightarrow \quad \varepsilon y'' + (y^2 - 1)y' + y = 0$$

with parameter $\varepsilon = \frac{1}{\mu^2}$. The system becomes more and more stiff in case of $\varepsilon \to 0$. We investigate the corresponding system of first order

$$y' = z, \quad \varepsilon z' = -(y^2 - 1)z - y.$$

Setting $\varepsilon = 0$ implies the semi-explicit DAE system

$$y' = z, \quad 0 = -(y^2 - 1)z - y.$$

It follows

$$y' = z = \frac{y}{1 - y^2} \quad \text{for } y \neq \pm 1.$$

We can solve this ODE for y partly and achieve (with a constant $C \in \mathbb{R}$)

$$\ln|y(x)| - \tfrac{1}{2}y(x)^2 = x + C.$$

If a solution of the semi-explicit DAEs reaches a singularity $y = \pm 1$, then the existence of the solution is violated. In contrast, the solution of the ODE continues to exist and exhibits steep gradients at the singularity. This solution changes fastly from $y = 1$ to $y = -2$ and from $y = -1$ to $y = 2$. We apply the above relation to obtain an estimate of the period of the solution of the oscillator. Let $y(x_1) = 2$ and $y(x_2) = 1$, i.e., the solution changes slowly between x_1 and x_2. It follows

$$\ln 2 - 2 = x_1 + C, \quad \ln 1 - \tfrac{1}{2} = x_2 + C \quad \Rightarrow \quad x_2 - x_1 = -\ln 2 + \tfrac{3}{2}.$$

The period is $T \approx 2(x_2 - x_1) = 3 - 2\ln 2 \approx 1.6137$ in case of $\varepsilon \approx 0$. Numerical simulations confirm this estimate.

Now we can apply a numerical method for ODEs to the system (6.29). Implicit techniques typically have to be considered, since DAEs represent the limit of stiff systems of ODEs. Performing the limit $\varepsilon \to 0$ yields a method for the semi-explicit DAEs (6.19).

For example, the implicit Euler method implies

$$\begin{aligned} y_1 &= y_0 + hf(y_1, z_1), \\ z_1 &= z_0 + h\tfrac{1}{\varepsilon}g(y_1, z_1). \end{aligned}$$

The second equation is equivalent to

$$\varepsilon z_1 = \varepsilon z_0 + hg(y_1, z_1).$$

In the limit $\varepsilon \to 0$, we obtain the numerical method

$$\begin{aligned} y_1 &= y_0 + hf(y_1, z_1), \\ 0 &= g(y_1, z_1), \end{aligned} \tag{6.30}$$

which represents a nonlinear system for the unknown approximation y_1, z_1.

Indirect Approach (state space form)

For the semi-explicit DAEs (6.19), we consider the component z as the solution of a nonlinear system for given y, i.e.,

$$z(x) = \Phi(y(x)), \qquad g(y(x), \Phi(y(x))) = 0. \tag{6.31}$$

Due to the implicit function theorem, the regularity of the Jacobian matrix $\frac{\partial g}{\partial z}$ is sufficient for the existence and the local uniqueness of a continuous function $\Phi : U \to V$ with $U \subset \mathbb{R}^{n_1}, V \subset \mathbb{R}^{n_2}$. This condition corresponds to a semi-explicit DAE of differential index 1. Consequently, the differential part of the DAE depends only on y

$$y'(x) = f(y(x), \Phi(y(x))). \tag{6.32}$$

This system is called the state space form of the problem. Now we are able to use a method for ODEs directly to this system. In a numerical method, we have to evaluate the right-hand side of (6.32) for given values y. Each evaluation demands the solution of a nonlinear system (6.31).

As example, we apply the implicit Euler method again. It follows

$$\begin{aligned} y_1 &= y_0 + hf(y_1, \Phi(y_1)), \\ 0 &= g(y_1, \Phi(y_1)). \end{aligned} \tag{6.33}$$

Hence the resulting technique (6.33) is equivalent to the scheme (6.30) obtained by the direct approach in case of the implicit Euler method.

The direct and indirect approach represent just techniques to obtain a suggestion for a numerical method. The properties of the corresponding method for ODEs do not necessarily hold for the resulting scheme to solve DAEs. Hence an analysis of consistency and stability of the constructed numerical methods has still to be performed.

Runge-Kutta Methods

We investigate Runge-Kutta methods now, see Sect. 3.5. The indirect approach is straightforward to apply. We obtain the formula

$$\begin{aligned} Y_i &= y_0 + h\sum_{j=1}^{s} a_{ij} f(Y_j, Z_j) \\ 0 &= g(Y_i, Z_i) \qquad\qquad \text{for } i = 1, \ldots, s \\ y_1 &= y_0 + h\sum_{i=1}^{s} b_i f(Y_i, Z_i). \end{aligned}$$

The value z_1 can be computed by solving the nonlinear system $g(y_1, z_1) = 0$.

The direct approach yields

$$\begin{aligned} Y_i &= y_0 + h\sum_{j=1}^{s} a_{ij} f(Y_j, Z_j) \\ \varepsilon Z_i &= \varepsilon z_0 + h\sum_{j=1}^{s} a_{ij} g(Y_j, Z_j) \qquad \text{for } i = 1, \ldots, s \\ y_1 &= y_0 + h\sum_{i=1}^{s} b_i f(Y_i, Z_i) \\ \varepsilon z_1 &= \varepsilon z_0 + h\sum_{i=1}^{s} b_i g(Y_i, Z_i). \end{aligned}$$

We assume that the matrix $A = (a_{ij})$ is regular in the following. Let $A^{-1} = (\omega_{ij})$. We transform the second equation into

$$hg(Y_i, Z_i) = \varepsilon \sum_{j=1}^{s} \omega_{ij}(Z_i - z_0) \qquad \text{for } i = 1, \ldots, s.$$

Accordingly, the fourth equation becomes

$$\varepsilon z_1 = \varepsilon z_0 + \varepsilon \sum_{i=1}^{s} b_i \left(\sum_{j=1}^{s} \omega_{ij}(Z_i - z_0) \right).$$

The limit $\varepsilon \to 0$ yields the method

$$\begin{aligned} Y_i &= y_0 + h\sum_{j=1}^{s} a_{ij} f(Y_j, Z_j) \\ 0 &= g(Y_i, Z_i) \qquad \text{for } i = 1, \ldots, s \\ y_1 &= y_0 + h\sum_{i=1}^{s} b_i f(Y_i, Z_i) \\ z_1 &= \left(1 - \sum_{i,j=1}^{s} b_i \omega_{ij}\right) z_0 + \sum_{i,j=1}^{s} b_i \omega_{ij} Z_j. \end{aligned} \tag{6.34}$$

The scheme (6.34) of the direct approach coincides with the method (6.28) applied to semi-explicit DAEs in case of a regular coefficient matrix A.

In (6.34), the involved coefficient satisfies

$$1 - \sum_{i,j=1}^{s} b_i \omega_{ij} = \lim_{z \to \infty} R(z) =: R(\infty)$$

with the stability function $R(z) = 1 + zb^\top (I - zA)^{-1} \mathbb{1}$ of the Runge-Kutta method from (5.7).

A Runge-Kutta method is called stiffly accurate, if it holds

$$a_{si} = b_i \qquad \text{for } i = 1, \ldots, s.$$

For example, the RadauIIa schemes are stiffly accurate ($s = 1$: implicit Euler). In this case, it follows $y_1 = Y_s$ and $z_1 = Z_s$, i.e., the direct approach coincides with the indirect approach.

Given a Runge-Kutta method with order of consistency p in case of ODEs, we are interested in the order of convergence in case of semi-explicit DAEs. Let q be the stage order of the method, i.e., $Y_i - y(x_0 + c_i h) = \mathcal{O}(h^{q+1})$ for all i in case of ODEs. We consider semi-explicit DAEs (6.19) with differential index 1. Using the indirect approach, the order of convergence is equal p for both differential part y and algebraic part z. The direct approach implies the global errors

$$y_N - y(x_{\text{end}}) = \mathcal{O}(h^p), \qquad z_N - z(x_{\text{end}}) = \mathcal{O}(h^r)$$

with

(i) $r = p$ for stiffly accurate methods ($R(\infty) = 0$),

(ii) $r = \min(p, q+1)$ for $-1 \leq R(\infty) < 1$,

(iii) $r = \min(p-1, q)$ for $R(\infty) = 1$,

(iv) divergence if $|R(\infty)| > 1$.

For methods, which are not stiffy accurate, an order reduction appears ($r < p$). The A-stability of a Runge-Kutta technique is sufficient (not necessary) for the convergence of the algebraic part.

Now we consider DAEs of index 2. Thereby, we analyse semi-explicit DAEs of the form

$$\begin{aligned} y' &= f(y,z), \\ 0 &= g(y). \end{aligned} \tag{6.35}$$

The system cannot have the differential index 1, since it holds $\frac{\partial g}{\partial z} \equiv 0$. The system (6.35) exhibits the differential index $k = 2$ if the matrix $\frac{\partial g}{\partial y}\frac{\partial f}{\partial z}$ is always regular. It can be shown that a system of the form (6.35) has the differential index $k = 2$ if and only if the perturbation index is $k = 2$.

The indirect approach cannot be applied to (6.35), since the function Φ from (6.31) is not defined. In contrast, the direct approach yields the same Runge-Kutta method (6.34) as in the case of index 1 (just replace $g(y,z)$ by the special case $g(y)$). The analysis of convergence becomes more complicated in case of differential index 2. We just cite the results for the Gauss and the Radau methods with s stages. The following table illustrates the orders of the local errors and the global errors.

	local error for ODEs	global error for ODEs	local error		global error	
			y	z	y	z
Gauss, s odd	$2s+1$	$2s$	$s+1$	s	$s+1$	$s-1$
Gauss, s even	$2s+1$	$2s$	$s+1$	s	s	$s-2$
RadauIA	$2s$	$2s-1$	s	$s-1$	s	$s-1$
RadauIIA	$2s$	$2s-1$	$2s$	s	$2s-1$	s

We recognise that the behaviour of the methods is much more complex than in the case of index 1. The RadauIIA schemes exhibits the best convergence properties within these examples, since these techniques are stiffly accurate.

For further reading on numerical methods for systems of DAEs, see
E. Hairer, G. Wanner: Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. (2nd Ed.) Springer, Berlin, 1996.

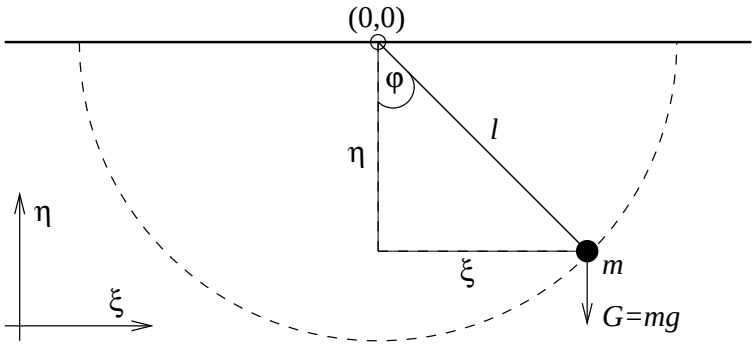

Figure 21: Mathematical Pendulum.

6.6 Illustrative Example: Mathematical Pendulum

Fig. 21 demonstrates the problem of the mathematical pendulum. We desire a mathematical model, which describes the positions ξ, η of the mass m with respect to time. On the one hand, Newton's law states $F = mx''$ for the force F acting on the mass m and for the space variables $x := (\xi, \eta)^\top$. On the other hand, the force F is the sum of the gravitational force $G = (0, mg)^\top$ with gravitation constant g and the force $F_r = -2\lambda x$ in direction of the rope, where λ represents a time-dependent scalar. The force F_r causes that the mass moves on a circle with radius l, since the constant l denotes the length of the rope. It follows

$$\begin{aligned} m\xi''(t) &= -2\lambda(t)\xi(t) \\ m\eta''(t) &= -2\lambda(t)\eta(t) - mg. \end{aligned}$$

A semi-explicit system of DAEs including five equations results

$$\begin{aligned} \xi'(t) &= u(t) \\ \eta'(t) &= v(t) \\ u'(t) &= -\tfrac{2}{m}\lambda(t)\xi(t) \\ v'(t) &= -\tfrac{2}{m}\lambda(t)\eta(t) - g \\ 0 &= \xi(t)^2 + \eta(t)^2 - l^2 \end{aligned} \tag{6.36}$$

with the unknowns ξ, η, u, v, λ. The components u, v are the components of the velocity of the mass, i.e., $x' = (u, v)^\top$. The last equation of the

system (6.36) represents the constraint that the mass moves on a circle with radius l only. The unknown λ characterises the magnitude of the force, which keeps the mass on this circle.

The most appropriate model of the mathematical pendulum results by considering the angle φ. It holds $\sin\varphi = \xi/l$ and $\cos\varphi = \eta/l$. Consequently, we achieve an ordinary differential equation of second order

$$\varphi''(t) = -\frac{g}{l}\sin(\varphi(t)), \qquad \varphi(t_0) = \varphi_0, \quad \varphi'(t_0) = \varphi_0'.$$

Hence the problem can be modelled by an explicit system of two ODEs of first order. In contrast, the system (6.36) represents a system of five DAEs. However, computer aided design is able to construct mathematical models based on DAEs automatically. A model for large technical problems involving just ODEs cannot be found by the usage of existing software codes.

Differentiating the algebraic constraint of the system (6.36) with respect to time yields the relation

$$2\xi(t)\xi'(t) + 2\eta(t)\eta'(t) = 0 \qquad \Leftrightarrow \qquad \xi(t)u(t) + \eta(t)v(t) = 0. \tag{6.37}$$

Thus we obtain an additional algebraic relation, which the exact solution of (6.36) satisfies. The equation (6.37) represents a hidden constraint, since it is not included directly in the system (6.36). A further differentiation in time shows the relation

$$u(t)^2 + \xi(t)u'(t) + v(t)^2 + \eta(t)v'(t) = 0. \tag{6.38}$$

Multiplying the third and fourth equation of (6.36) by ξ and η, respectively, it follows

$$\begin{aligned} \xi(t)u'(t) &= -\tfrac{2}{m}\lambda(t)\xi(t)^2 \\ \eta(t)v'(t) &= -\tfrac{2}{m}\lambda(t)\eta(t)^2 - g\eta(t). \end{aligned}$$

Summing up these two equations and using (6.38) implies an algebraic relation for the unknown λ

$$\lambda(t) = \frac{m}{2l^2}\left(u(t)^2 + v(t)^2 - g\eta(t)\right). \tag{6.39}$$

Differentiating this equation with respect to time results to

$$\lambda'(t) = \frac{m}{2l^2}\left(2u(t)u'(t) + 2v(t)v'(t) - gv(t)\right). \tag{6.40}$$

Inserting the ODEs (6.36) and using (6.37) yields

$$\lambda'(t) = -\frac{3mg}{2l^2}v(t). \tag{6.41}$$

If we replace the algebraic constraint in (6.36) by the equation (6.41), then we achieve a system of five ODEs for the five unknowns. Three differentiations of the original system (6.36) with respect to time are necessary to derive this ODE system. Thus the differential index of the DAE system (6.36) is $k = 3$. It can be shown that the perturbation index is also $k = 3$.

Now we perform a numerical simulation of the mathematical pendulum using the DAE model (6.36) as well as the regularised model with (6.41), which represents an ODE model. We apply the parameters $m = 1$, $l = 2$, $g = 9.81$. The initial values are

$$\xi(0) = \sqrt{2}, \quad \eta(0) = -\sqrt{2}, \quad u(0) = 0, \quad v(0) = 0$$

The initial value $\lambda(0)$ follows from (6.39). The numerical solutions are computed in the time interval $t \in [0, 20]$.

The BDF methods are damping the amplitude of oscillations in a numerical simulation. In contrast, trapezoidal rule conserves the energy of a system and thus the amplitude of oscillations is reproduced correctly. We solve the ODE model by trapezoidal rule with adaptive step size control. Thereby, two different demands of relative accuracy are applied, namely 10^{-3} and 10^{-6}, whereas the absolute accuracy is set to 10^{-6}. The number of necessary integration steps is 610 and 4778, respectively. Fig. 22 illustrates the solution of the coordinates ξ, η by phase diagrammes. We recognise that the solution leaves the circle significantly in case of the lower accuracy.

To analyse this effect more detailed, we compute the values of the circle condition (last equation of (6.36)) and of the hidden constraint (6.37). For the exact solution, these values are equal to zero, since the constraints are

satisfied. On the contrary, the numerical solution causes an error in these constraints. Fig. 23 shows the corresponding discrepancies. We see that the error increases in time for each accuracy demand. Thus the numerical solution will leave a given neighbourhood of the circle at a later time. The reason is that the simulated ODE system does not include the information of the circle explicitly. This phenomenon is called *drift off*: the numerical solution of the regularised DAE, i.e., the ODE, drifts away from the manifold, where the true solution is situated.

Alternatively, we simulate the DAE model (6.36) directly using the trapezoidal rule with constant step sizes. We apply 1000 integration steps in the interval $t \in [0, 20]$. In each integration step, we perform just one step of the Newton method to solve the involved nonlinear system of algebraic equations.

The resulting solutions for ξ, η as well as the corresponding errors in the constraints are illustrated in Fig. 24. Both the circle condition and the hidden constraint exhibit an oscillating error, whose amplitude remains constant in time. Since the system (6.36) includes the circle condition directly, the error in this constraint depends just on the accuracy demand in solving the nonlinear system in each integration step. Hence the DAE model generates a significantly better numerical approximation than the corresponding ODE formulation using (6.41).

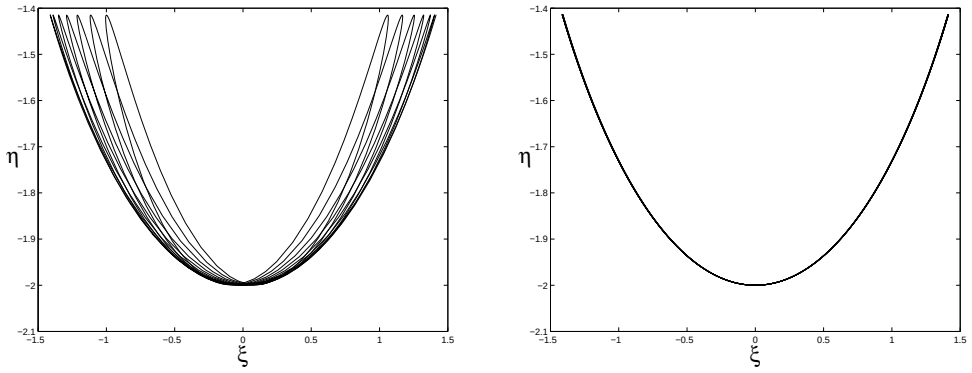

Figure 22: Phase diagramme of solution of ODE model for mathematical pendulum computed by trapezoidal rule with relative tolerance 10^{-3} (left) and 10^{-6} (right).

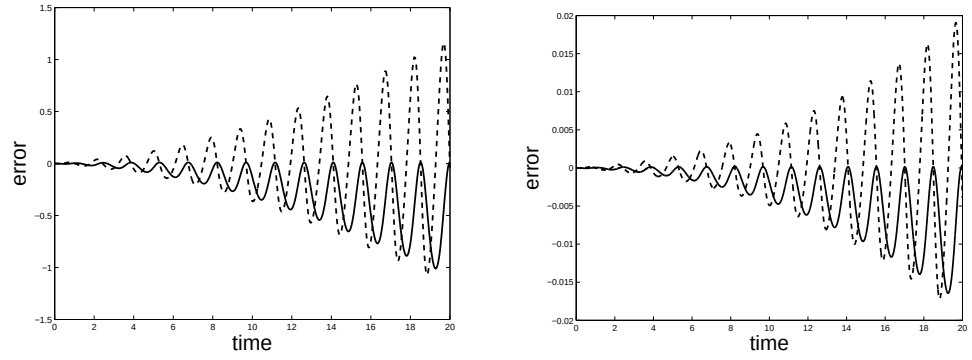

Figure 23: Error in circle condition (solid line) and hidden constraint (dashed line) for solution of ODEs corresponding to relative tolerance 10^{-3} (left) and 10^{-6} (right).

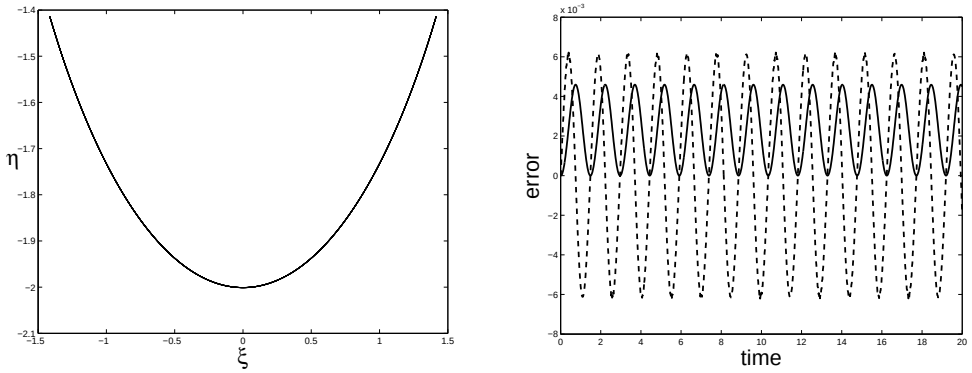

Figure 24: Phase diagramme of solution of DAE model for mathematical pendulum computed by trapezoidal rule (left) and corresponding errors (right) in circle condition (solid line) as well as hidden constraint (dashed line).

Chapter 7

Boundary Value Problems

In the previous chapters, we have discussed initial value problems (IVPs) of systems of ordinary differential equations (ODEs) or differential algebraic equations (DAEs). Now we consider boundary value problems (BVPs) of systems of ODEs.

7.1 Problem Definition

Let a system of ODEs

$$y'(x) = f(x, y(x)) \tag{7.1}$$

be given with $y : [a, b] \to \mathbb{R}^n$ and $f : [a, b] \times \mathbb{R}^n \to \mathbb{R}^n$ for $x \in [a, b]$. On the one hand, an IVP of the ODEs is specified by a condition $y(a) = y_0$, where y_0 is a prescribed value. Consequently, a solution of the system of ODEs in the interval $[a, b]$, which satisfies the initial condition, shall be determined. For $f \in C^1$, the IVP exhibits a unique solution for each $y_0 \in \mathbb{R}^n$ in some interval $[a, a + \varepsilon]$. Thus n parameters identify a specific solution.

On the other hand, a two-point BVP is defined by a relation

$$r(y(a), y(b)) = 0 \tag{7.2}$$

with a (general) function $r : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$. The function r depends on the initial values $y(a)$ and the final values $y(b)$, which are both unknown a

priori. We want to achieve a solution, where the initial values and the final values satisfy the condition (7.2). Since a solution of the system of ODEs exhibits n degrees of freedom, it is reasonable to impose n equations for obtaining a well-posed problem. However, the existence and uniqueness of a solution of a BVP is not guaranteed as in the case of IVPs. Each problem may have a unique solution, a finite number of solutions, an infinite family of solutions or no solution at all. In the following, we assume that the BVP exhibits a unique solution.

In most cases, linear boundary conditions appear, i.e.,

$$r(y(a), y(b)) \equiv Ay(a) + By(b) - c = 0 \tag{7.3}$$

with constant matrices $A, B \in \mathbb{R}^{n\times n}$ and a constant vector $c \in \mathbb{R}^n$. In contrast, the ODEs are often nonlinear.

Examples:

1. Consider the BVP for a scalar ODE of second order

$$u'' = f(x, u, u'), \qquad u(0) = \alpha, \;\; u(1) = \beta, \tag{7.4}$$

where $\alpha, \beta \in \mathbb{R}$ are predetermined. The equivalent system of first order with $y_1 := u, y_2 := u'$ reads

$$\begin{aligned} y_1' &= y_2, \\ y_2' &= f(x, y_1, y_2), \end{aligned} \qquad y_1(0) = \alpha, \;\; y_1(1) = \beta. \tag{7.5}$$

Thereby, the domain of dependence is standardised to $x \in [0, 1]$. The boundary conditions are linear, see (7.3), where it holds

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}, \quad c = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}.$$

A particular instance is the BVP

$$u''(x) = \lambda \sinh(u(x)), \qquad u(0) = 0, \quad u(1) = 1$$

with a real parameter $\lambda \geq 0$.

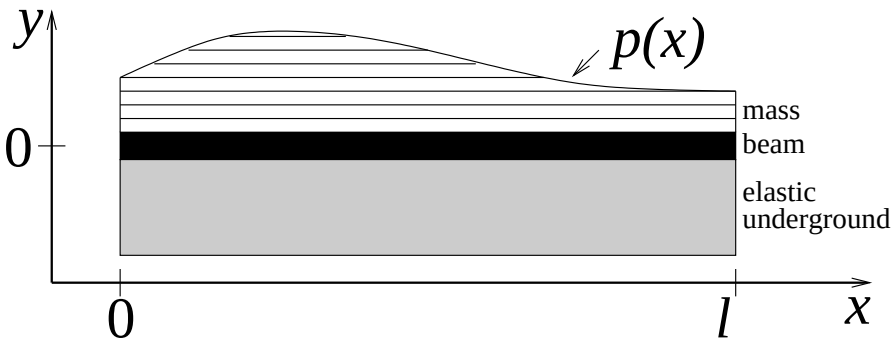

Figure 25: Beam on an elastic underground with mass on top.

2. We consider a beam of length l on an elastic underground, where some mass is put on top, see Fig. 25. Let $y(x)$ describe the positon of the beam for $x \in [0, l]$. It follows the ODE

$$-(a(x)y''(x))'' + k(x)y(x) + p(x) = 0,$$

where $a(x)$ characterises the stiffness of the beam, $k(x)$ corresponds to the spring constants of the elastic underground and $p(x)$ specifies the mass distribution. In case of $a(x) \equiv a_0$, we obtain the equivalent ODE

$$y^{(4)}(x) = \frac{k(x)}{a_0}y(x) + \frac{p(x)}{a_0}.$$

Since the corresponding system of first order consists of four equations, we have to specify four boundary conditions.

Several possibilities exist:

(i) At the boundaries, the beam rests upon supports from below:

$$y(0) = 0, \quad y(l) = 0, \quad y''(0) = 0, \quad y''(l) = 0.$$

(ii) At the boundaries, the beam is fixed in horizontal direction:

$$y(0) = 0, \quad y(l) = 0, \quad y'(0) = 0, \quad y'(l) = 0.$$

(iii) The beam is not supported or fixed at all:

$$y''(0) = 0, \quad y''(l) = 0, \quad y'''(0) = 0, \quad y'''(l) = 0.$$

Furthermore, we can have mixed types of boundary conditions, where the type at $x = 0$ is different from the type at $x = l$.

Separated boundary conditions

Often linear boundary conditions (7.3) are separated, i.e., they exhibit the form

$$\tilde{A}y(a) = c_1, \qquad \tilde{B}y(b) = c_2 \tag{7.6}$$

with matrices $\tilde{A} \in \mathbb{R}^{n_1 \times n}$, $\tilde{B} \in \mathbb{R}^{n_2 \times n}$ and vectors $c_1 \in \mathbb{R}^{n_1}$, $c_2 \in \mathbb{R}^{n_2}$ ($n_1 + n_2 = n$). For example, see the problem (7.5), where $\tilde{A} = (1, 0)$ and $\tilde{B} = (1, 0)$. Periodic BVPs are not separated. On the one hand, the special case $n_1 = n$ ($n_2 = 0$, $\tilde{B} = \emptyset$) in (7.6) just represents an IVP provided that $\det \tilde{A} \neq 0$. On the other hand, the case $n_2 = n$ ($n_1 = 0$, $\tilde{A} = \emptyset$) corresponds to a final value problem (or end value problem) provided that $\det \tilde{B} \neq 0$. The final value problem can be resolved as an IVP by an integration backwards from $x = b$ to $x = a$. A non-trivial BVP appears for $n_1, n_2 \geq 1$.

Periodic problems of non-autonomous systems

A periodic solution y of the system of ODEs is characterised by the property $y(x) = y(x+T)$ for all x, where $T > 0$ is the period. Therefore, we demand also $f(x, z) = f(x + T, z)$ for each constant $z \in \mathbb{R}^n$. The periodic solution with minimum period $T > 0$ is uniquely determined by the two-point BVP $y(0) = y(T)$ (provided that IVPs are uniquely solvable). If follows the periodic boundary value problem

$$r(y(0), y(T)) \equiv y(0) - y(T) = 0, \tag{7.7}$$

where the period $T > 0$ is given. The boundary conditions exhibit the linear form (7.3) with $A = I$, $B = -I$ and $c = 0$. Periodic BVPs already make sense in the scalar case ($n = 1$).

Periodic problems of autonomous systems

An autonomous system of ODEs reads

$$y'(x) = f(y(x)). \tag{7.8}$$

Given a solution $y : \mathbb{R} \to \mathbb{R}^n$, the shifted function $y_c(x) := y(x + c)$ with a constant $c \in \mathbb{R}$ also solves the system (7.8). A periodic solution satisfies $y(x) = y(x + T)$ for all x with some minimal period $T > 0$. The period is unknown a priori. It follows that a periodic solution is not unique. We require an additional condition to achieve an isolated solution. For example, we demand $y_1(0) = \eta$ with some $\eta \in \mathbb{R}$ for the first component.

We can formulate a two-point BVP via the linear transformation $x = sT$ ($x \in [0, T]$, $s \in [0, 1]$). Let $\tilde{y}(s) := y(sT)$. We arrange the system of ODEs

$$\begin{aligned} \tilde{y}'(s) &= T f(\tilde{y}(s)), \\ T'(s) &= 0, \end{aligned}$$

where a trivial ODE for the constant T is included. The corresponding boundary conditions read

$$\begin{aligned} \tilde{y}(0) - \tilde{y}(1) &= 0, \\ \tilde{y}_1(0) - \eta &= 0. \end{aligned}$$

Hence we achieve $n + 1$ ODEs and $n + 1$ boundary conditions for $n + 1$ unknown functions, where one function represents the unknown period. Accordingly, we achieve the standard formulation of a two-point BVP and common numerical methods are feasible.

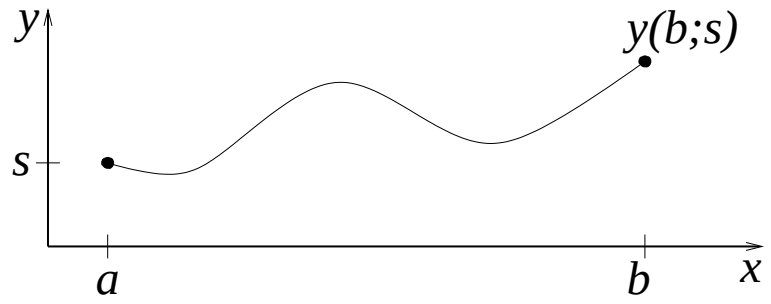

Figure 26: Dependence of solution on initial values.

7.2 Single Shooting Method

Now we will construct a numerical method for solving the BVP (7.2) of the system of ODEs. Since we have considered IVPs in the previous chapters, we want to apply IVPs for the solution of the BVP. For a smooth right-hand side f, an IVP exhibits a unique solution in general. The initial values determine the solution in the complete interval, especially in the final point $x = b$. We note the dependence, see also Fig. 26,

$$s := y(a) \quad \longrightarrow \quad y(b) = y(b; s),$$

where $s \in \mathbb{R}^n$ are free parameters. Thus we rewrite the boundary conditions as

$$r(y(a), y(b)) \equiv r(s, y(b; s)) = 0.$$

We consider a nonlinear system of algebraic equations

$$g(s) :\equiv r(s, y(b; s)) = 0 \qquad (g : \mathbb{R}^n \to \mathbb{R}^n), \tag{7.9}$$

where the initial values $s \in \mathbb{R}^n$ represent the unknowns. Consequently, we want to determine the initial values, which produce the solution of the BVP. Each evaluation of the nonlinear system (7.9) demands the solution of an IVP to obtain the value $y(b; s)$. The nonlinear system can be solved by methods of Newton type.

Example

We consider the BVP

$$u'' = \lambda \sin(2\pi u), \qquad u(0) = 0, \quad u(1) = 1 \tag{7.10}$$

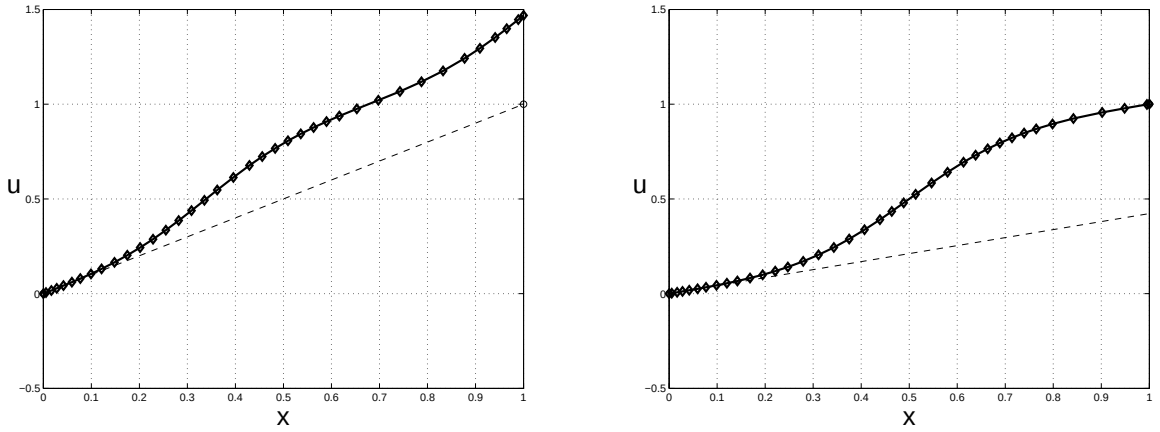

Figure 27: Shooting method – solutions of IVPs for $\tilde{s}^{(0)}$ (left) and $\tilde{s}^{(4)}$ (right).

with unknown solution $u \in C^2$ and a real parameter $\lambda > 0$. For $\lambda = 0$, the solution becomes just $u(x) \equiv x$. The equivalent system of first order ($y_1 := u, y_2 := u'$) reads

$$\begin{aligned} y_1' &= y_2, \\ y_2' &= \lambda \sin(2\pi y_1), \end{aligned} \qquad y_1(0) = 0, \;\; y_1(1) = 1.$$

Initial and boundary conditions yield

$$y(0) = s = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}, \qquad r(s, y(1;s)) \equiv \begin{pmatrix} s_1 - 0 \\ y_1(1; s_1, s_2) - 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

In this example, we can use the first equation to eliminate the unknown s_1 and obtain one equation for $\tilde{s} := s_2$ alone. It follows

$$y(0) = \begin{pmatrix} 0 \\ \tilde{s} \end{pmatrix}, \qquad g(\tilde{s}) \equiv y_1(1; 0, \tilde{s}) - 1 = 0.$$

This nonlinear equation can be solved via bisection. Alternatively, a Newton iteration yields an approximation now. We use $\lambda = 5$ and the starting value $\tilde{s}^{(0)} = 1$. Trapezoidal rule resolves the IVPs. Fig. 27 illustrates the solution of the IVPs for the starting value as well as after four iteration steps. We observe that the boundary condition is satisfied sufficiently accurate.

Computation of Jacobian matrix

Newton's method to solve the nonlinear system (7.9) yields the iteration

$$s^{(i+1)} = s^{(i)} - \Big(\mathrm{D}g(s^{(i)})\Big)^{-1} g(s^{(i)}) \qquad \text{for } i = 0, 1, 2, \dots . \tag{7.11}$$

Consequently, we have to compute (approximately) the Jacobian matrix $\mathrm{D}g \in \mathbb{R}^{n\times n}$. The chain rule of multidimensional differentiation implies

$$\mathrm{D}g(s) = \frac{\partial}{\partial s} r = \frac{\partial r}{\partial y(a)} + \frac{\partial r}{\partial y(b)} \cdot \frac{\partial y(b;s)}{\partial s}. \tag{7.12}$$

The matrices $\frac{\partial r}{\partial y(a)}, \frac{\partial r}{\partial y(b)}$ are often directly available. For example, linear boundary conditions (7.3) imply $\frac{\partial r}{\partial y(a)} = A, \frac{\partial r}{\partial y(b)} = B$.

The matrix $\frac{\partial y(b;s)}{\partial s}$ is called sensitivity matrix, since it describes the sensitivity of the solution with respect to the initial values. The matrix $\frac{\partial y(x;s)}{\partial s}$ is identical to the matrix $\Psi(x)$ introduced in Theorem 4, see Sect. 2.3.

Two possibilities exist to compute the sensitivity matrix $\Psi(b)$:

1. *Numerical differentiation*:
   Numerical differentiations yield the columns of the matrix $\Psi \in \mathbb{R}^{n\times n}$. Let $\Psi_j \in \mathbb{R}^n$ be the jth column. We obtain the approximation

   $$\Psi_j(b;s) \doteq \frac{1}{\delta_j}\left[y(b; s + \delta_j e_j) - y(b;s)\right] \qquad \text{for } j = 1, \dots, n\ ,$$

   where $e_j = (0, \dots, 0, 1, 0, \dots, 0)^\top \in \mathbb{R}^n$ represents the jth unit vector. The selection of the increment $\delta_j \in \mathbb{R}\backslash\{0\}$ depends on the jth component s_j of s. Given the machine precision ϵ, an appropriate choice is

   $$\delta_j := \begin{cases} s_j \cdot \sqrt{\epsilon} & \text{for } |s_j| > 1, \\ \operatorname{sign}(s_j) \cdot \sqrt{\epsilon} & \text{for } 0 < |s_j| \le 1, \\ \sqrt{\epsilon} & \text{for } s_j = 0. \end{cases}$$

   Hence n additional IVPs of the underlying system of ODEs have to be resolved with perturbed initial values $s + \delta_j e_j$ for $j = 1, \dots, n$.

2. *Solve matrix-valued ODEs*:
   As shown in Sect. 2.3, the matrix Ψ satisfies an IVP of the matrix-valued system (2.11). We write this system of ODEs in the form

$$\Psi'(x;s) = \mathrm{D}f(x, y(x;s)) \cdot \Psi(x;s), \qquad \Psi(a;s) = I, \tag{7.13}$$

   where $\mathrm{D}f = \frac{\partial f}{\partial y} \in \mathbb{R}^{n\times n}$ denotes the Jacobian matrix of f with respect to y. The identity matrix $I \in \mathbb{R}^{n\times n}$ provides the initial values. The system (7.13) is linear. Thus an implicit method demands just the solution of a linear system in each integration step. The ODE system (7.13) is also called the sensitivity equations.

   The a priori unknown function $y(x;s)$ appears in the matrix-valued ODEs (7.13). We can solve the ODEs for $y(x;s)$ in combination with the system (7.13). Alternatively, $y(x;s)$ is computed first, which yields approximations just in grid points $a = x_0 < x_1 < x_2 < \cdots < x_{\mathrm{end}} = b$. Then the matrix-valued ODEs (7.13) are solved by a numerical method, where the required values $y(x;s)$ are interpolated from the available data $y(x_i;s)$.

Often the second technique yields better approximations and is more robust than the first strategy. However, for non-stiff problems (using explicit methods for IVPs) the numerical differentiation is preferred, since evaluations of the Jacobian matrix do not appear. In case of stiff problems, often the matrix-valued ODEs are resolved, because the integration steps can be combined efficiently with the solution of the underlying system of ODEs by implicit methods for IVPs.

For solving nonlinear systems of algebraic equations, we applied the simplified Newton method in implicit integrators for IVPs, since good starting values are given by the solution of the previous step. In contrast, good starting values are often not available in case of BVPs. Hence we apply the (ordinary) Newton method.

7.3 Multiple Shooting Method

The single shooting method fails if one of the following two cases occur:

1. We always assume that a solution of the BVP exists. Let s^* be the corresponding initial value. The solution $y(x; s^*)$ exists for all $x \in [a, b]$. In the Newton iteration, we need some starting values $s^{(0)}$. However, the corresponding solution $y(x; s^{(0)})$ may exist just within an interval $x \in [a, a + \varepsilon)$ for $\varepsilon \leq b - a$. Thus the single shooting method fails completely.

   Example: We discuss the BVP

   $$y'' = -(y')^2, \qquad y(0) = 1, \quad y(1) = -4.$$

   The ODE is satisfied by the function

   $$y(x; s) = \ln(sx + 1) + 1 \qquad \text{for } \ s \in \mathbb{R}$$

   and it holds $y(0; s) = 1$ for all s. The boundary condition $y(1) = -4$ is fulfilled for

   $$s^* = -1 + \mathrm{e}^{-5} = -0.993\ldots \ .$$

   However, the logarithm is defined for $sx+1 > 0$ only. Assuming $s < 0$, this condition is equivalent to $x < \frac{1}{|s|}$. For the starting value $s^{(0)} = -1$, the solution exists just in $x \in [0, 1)$ and the shooting method breaks down. For $s^{(0)} < -1$, the existence is given in $x \in [0, \varepsilon)$ with $\varepsilon < 1$. The case $s^{(0)} > -1$ is feasible.

2. In some applications, the sensitivity of the solution with respect to the initial values is extremely high. Hence the condition of the IVPs is very bad. The correct solution cannot be reproduced by solving an IVP in $x \in [a, b]$, since small numerical errors are amplified.

   The estimate (2.10) from Sect. 2.3 yields

   $$\|y(b; s + \Delta s) - y(b; s)\| \leq \mathrm{e}^{L(b-a)} \cdot \|\Delta s\| \tag{7.14}$$

   with the Lipschitz-constant satisfying $\|f(x, y) - f(x, z)\| \leq L \cdot \|y - z\|$. Thus the difference between two IVPs is allowed to increase exponentially with respect to the length $b-a$ of the interval, which also happens sometimes.

Example: We consider the linear problem

$$\begin{pmatrix} y_1' \\ y_2' \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 110 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \qquad y_1(0) = 1, \quad y_1(10) = 1. \tag{7.15}$$

The general solution of this system of ODEs reads

$$\begin{pmatrix} y_1(x) \\ y_2(x) \end{pmatrix} = C_1 \mathrm{e}^{-10x} \begin{pmatrix} 1 \\ -10 \end{pmatrix} + C_2 \mathrm{e}^{11x} \begin{pmatrix} 1 \\ 11 \end{pmatrix}, \qquad C_1, C_2 \in \mathbb{R}.$$

Given the initial value $y(0) = (s_1, s_2)^\top$, the solution is

$$\begin{pmatrix} y_1(x) \\ y_2(x) \end{pmatrix} = \frac{11s_1 - s_2}{21} \mathrm{e}^{-10x} \begin{pmatrix} 1 \\ -10 \end{pmatrix} + \frac{10s_1 + s_2}{21} \mathrm{e}^{11x} \begin{pmatrix} 1 \\ 11 \end{pmatrix}.$$

The boundary conditions are satisfied for the initial values

$$s_1 = 1, \qquad s_2 = -10 + \frac{21(1 - \mathrm{e}^{-100})}{\mathrm{e}^{110} - \mathrm{e}^{-100}} \approx -10 + 10^{-47}.$$

At $x = 10$, it holds $\mathrm{e}^{-10x} \approx 0$ and $\mathrm{e}^{11x} \approx 10^{48}$. The corresponding sensitivity of the solution at $x = 10$ is approximately

$$\begin{pmatrix} \Delta y_1 \\ \Delta y_2 \end{pmatrix} \approx \frac{\mathrm{e}^{110}}{21} \left[10\Delta s_1 + \Delta s_2\right] \begin{pmatrix} 1 \\ 11 \end{pmatrix}.$$

If we know s up to machine precision ϵ, i.e., $\frac{|\Delta s_j|}{|s_j|} \approx \epsilon$, it follows

$$|\Delta y_j| \approx \frac{\mathrm{e}^{110}}{21} \epsilon \approx 10^{30}$$

for $\epsilon \approx 10^{-16}$. Hence we cannot reproduce the correct solution of the BVP. This problem can be solved by using a sufficiently small machine precision (longer mantissa). However, we want to avoid this, since a larger computational effort results.

Both scenarios are omitted by a multiple shooting method. The idea is to divide the interval $[a, b]$ into several subintervals and to consider an IVP in each subinterval. Fig. 28 sketches this technique. The subintervals are defined by the grid

$$a = x_1 < x_2 < \cdots < x_{m-1} < x_m = b. \tag{7.16}$$

The corresponding IVPs read

$$y'(x) = f(x, y(x)), \quad y(x_k) = s_k \qquad \text{with } \; x \in [x_k, x_{k+1}] \tag{7.17}$$

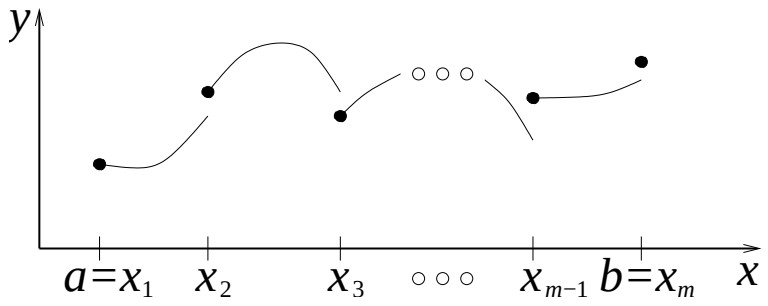

Figure 28: Strategy of multiple shooting method.

for $k = 1, 2, \ldots, m-1$. Let $y(x; x_k, s_k)$ be the solution of the kth problem (7.17) and $s_m := y(x_m; x_{m-1}, s_{m-1})$. The complete solution $y(x)$ for $x \in [a, b]$ is not continuous for general initial values. We demand the continuity together with the boundary conditions

$$\begin{aligned} y(x_{k+1}; x_k, s_k) &= s_{k+1} \qquad \text{for } k = 1, 2, \ldots, m-1 \\ r(s_1, s_m) &= 0. \end{aligned}$$

These conditions yield a nonlinear system

$$G(S) := \begin{pmatrix} y(x_2; x_1, s_1) - s_2 \\ y(x_3; x_2, s_2) - s_3 \\ \vdots \\ y(x_m; x_{m-1}, s_{m-1}) - s_m \\ r(s_1, s_m) \end{pmatrix} = 0, \qquad S = \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_{m-1} \\ s_m \end{pmatrix}$$

with $G : \mathbb{R}^{nm} \to \mathbb{R}^{nm}$ and the unknowns $S \in \mathbb{R}^{nm}$. We solve the nonlinear system by a Newton iteration again

$$S^{(i+1)} = S^{(i)} - \left(\mathrm{D}G(S^{(i)})\right)^{-1} G(S^{(i)}) \qquad \text{for } i = 0, 1, 2, \ldots .$$

The involved Jacobian matrix exhibits the block structure

$$\mathrm{D}G = \begin{pmatrix} C_1 & -I & & & \\ & C_2 & -I & & \\ & & \ddots & \ddots & \\ & & & \ddots & \ddots \\ & & & & C_{m-1} & -I \\ A & & & & & B \end{pmatrix} \tag{7.18}$$

with the identity I and the matrices

$$A := \frac{\partial r}{\partial s_1}, \quad B := \frac{\partial r}{\partial s_m}, \quad C_k := \frac{\partial y(x_{k+1}; x_k, s_k)}{\partial s_k}.$$

The matrices C_k represent sensitivity matrices, which can be computed as in a single shooting method, see Sect. 7.2. Although $m-1$ IVPs (7.17) are considered, the computational effort is independent of m, since the subdivisions always generate the same total interval $[a, b]$. Hence the computational work for evaluating G and DG is nearly the same as in a single shooting method.

Under some general assumptions, it can be shown that the Jacobian matrix (7.18) is regular. Let $\Delta s_k := s_k^{(i+1)} - s_k^{(i)}$ and $G = (g_1, \ldots, g_m)$. The linear system in each Newton step reads

$$\begin{aligned} C_1 \Delta s_1 - \Delta s_2 &= -g_1 \\ C_2 \Delta s_2 - \Delta s_3 &= -g_2 \\ &\vdots \\ C_{m-1} \Delta s_{m-1} - \Delta s_m &= -g_{m-1} \\ A \Delta s_1 + B \Delta s_m &= -g_m. \end{aligned}$$

We can eliminate $\Delta s_2, \ldots, \Delta s_m$ successively. Using

$$\Delta s_{k+1} = g_k + C_k \Delta s_k, \tag{7.19}$$

it follows the condensed system

$$(A + B C_{m-1} C_{m-2} \cdots C_2 C_1) \Delta s_1 = w \tag{7.20}$$

with

$$w := -(g_m + B g_{m-1} + B C_{m-1} g_{m-2} + \cdots + B C_{m-1} C_{m-2} \cdots C_2 g_1).$$

Gaussian elimination yields the solution Δs_1 of the linear system (7.20). We obtain the other increments Δs_{k+1} successively using (7.19). The computational effort for this linear algebra part becomes more expensive than in a single shooting method. However, the total effort is dominated by the evaluation of G and DG, which is not more costly in comparison to a single shooting technique.

In the multiple shooting method, we reduce the critical behaviour indicated by the estimate (7.14). The technique implies

$$\|y(x_{k+1}; x_k, s_k + \Delta s_k) - y(x_{k+1}; x_k, s_k)\| \leq \mathrm{e}^{L(x_{k+1}-x_k)} \cdot \|\Delta s_k\|$$

for some perturbation Δs_k. In case of $m-1$ subintervals with the same length, it follows

$$\mathrm{e}^{L(x_{k+1}-x_k)} = \mathrm{e}^{L\frac{b-a}{m-1}} = \sqrt[m-1]{\mathrm{e}^{L(b-a)}}.$$

For example, it holds

$$\mathrm{e}^{110} \approx 10^{48}, \quad \sqrt{\mathrm{e}^{110}} \approx 10^{24}, \quad \sqrt[3]{\mathrm{e}^{110}} \approx 10^{16}, \quad \sqrt[4]{\mathrm{e}^{110}} \approx 10^{12}.$$

Hence the multiple shooting method with four subintervals should be successful for the example (7.15) using the machine precision $\epsilon \approx 10^{-16}$.

On the one hand, let $M_k \subset \mathbb{R}^n$ be the set of all initial values s_k, where the solution of (7.17) exists in the interval $[x_k, x_{k+1}]$. On the other hand, let $N_k \subset \mathbb{R}^n$ be the set of all s_k, where the solution of (7.17) exists in the total interval $[a, b]$. A single shooting method using the unknown s_k is only feasible for $s_k \in N_k \subset M_k$. In contrast, the multiple shooting method is defined for initial values

$$S \in M := M_1 \times M_2 \times \cdots \times M_{m-1} \times \mathbb{R}^n.$$

Thus we avoid the other drawback of the single shooting technique.

We outline the determination of an appropriate subdivision (7.16). We assume that a starting trajectory $\eta : [a, b] \to \mathbb{R}^n$ is given, which fulfills the boundary conditions. (η is a guess for the unknown solution.) Let $x_1 := a$. If x_k is chosen, then solve the IVP $y' = f(x, y)$, $y(x_k) = \eta(x_k)$. We choose a new grid point x_{k+1} when the solution y becomes large in comparison to η. For example, x_{k+1} is the smallest value $\xi > x_k$ satisfying

$$\|y(\xi)\| \geq \gamma \cdot \|\eta(\xi)\|$$

using some vector norm and a threshold γ, say $\gamma = 2$. The function η also provides us starting values $s_k^{(0)} := \eta(x_k)$ for the Newton iteration.

7.4 Finite Difference Methods

Another idea to solve the boundary value problem (7.1),(7.2) is to discretise the derivatives in the ODEs by a difference formula first. This strategy yields a nonlinear system of algebraic equations for unknown values of the solution in grid points. We consider an equidistant grid

$$x_j = a + jh \quad \text{with} \quad h = \tfrac{b-a}{m+1} \quad \text{for} \quad j = 0, 1, \ldots, m, m+1. \tag{7.21}$$

with some $m \in \mathbb{N}$. We want to determine approximations $u_j \doteq y(x_j)$ of the unknown solution for each $j = 0, \ldots, m+1$. It holds $u_0 \doteq y(a)$ and $u_{m+1} \doteq y(b)$.

For example, we apply finite differences of first order, which correspond to the Euler scheme. The difference formula reads

$$\frac{y(x+h) - y(x)}{h} = y'(x) + \tfrac{1}{2} h y''(x + \vartheta h) = y'(x) + \mathcal{O}(h) \tag{7.22}$$

with $\vartheta \in (0, 1)$. It follows the system

$$\frac{u_{j+1} - u_j}{h} = f(x_j, u_j) \qquad \text{for} \quad j = 0, 1 \ldots, m.$$

Together with the boundary conditions, we obtain

$$\begin{aligned} u_{j+1} - u_j - h f(x_j, u_j) &= 0 \qquad \text{for} \quad j = 0, 1 \ldots, m \\ r(u_0, u_{m+1}) &= 0, \end{aligned}$$

which represents a nonlinear system of $(m+2)n$ equations for the $(m+2)n$ unknown values. Again, methods of Newton type can be applied to solve this system provided that $f, r \in C^1$ holds. The corresponding Jacobian matrix exhibits the block structure

$$\left(\begin{array}{c|c|c|c|c|c} -I - h\mathrm{D}f & I & & & & \\ \hline & -I - h\mathrm{D}f & I & & & \\ \hline & & \ddots & \ddots & & \\ \hline & & & \ddots & \ddots & \\ \hline & & & & -I - h\mathrm{D}f & I \\ \hline \frac{\partial r}{\partial y(a)} & & & & & \frac{\partial r}{\partial y(b)} \end{array}\right),$$

where each block has the size $n \times n$. ($\mathrm{D}f = \frac{\partial f}{\partial y}$ denotes the Jacobian matrix of f.) Therefore a band structure of the matrix appears (except for the block $\frac{\partial r}{\partial y(a)}$). The computational effort for an LU-decomposition of this matrix is approximately $\mathcal{O}(m^2 n^2)$ for $m \gg n$.

Alternatively, we use finite differences of second order to achieve a higher accuracy. This symmetric difference formula reads

$$\frac{y(x+h) - y(x-h)}{2h} = y'(x) + \mathcal{O}(h^2). \tag{7.23}$$

This approach corresponds to a midpoint rule. We set up the system

$$\frac{u_{j+1} - u_{j-1}}{2h} = f(x_j, u_j) \qquad \text{for } \; j = 1, \ldots, m.$$

Together with the boundary conditions, we obtain less equations than unknowns now. Hence for $j = 0$, the formula (7.22) of first order is added. We arrange the system

$$\begin{aligned} u_1 - u_0 - hf(x_0, u_0) &= 0 \\ u_{j+1} - u_{j-1} - 2hf(x_j, u_j) &= 0 \qquad \text{for } \; j = 1, \ldots, m \\ r(u_0, u_{m+1}) &= 0, \end{aligned}$$

i.e., $(m+2)n$ equations for $(m+2)n$ unknowns again. In a Newton method, the Jacobian matrix owns the structure

$$\left(\begin{array}{c|c|c|c|c|c} -I - h\mathrm{D}f & I & & & & \\ \hline -I & -2h\mathrm{D}f & I & & & \\ \hline & \ddots & \ddots & \ddots & & \\ \hline & & \ddots & \ddots & \ddots & \\ \hline & & & -I & -2h\mathrm{D}f & I \\ \hline \frac{\partial r}{\partial y(a)} & & & & & \frac{\partial r}{\partial y(b)} \end{array}\right).$$

The difference formula of first order for $j = 0$ can be replaced by an asymmetric formula of second order (similar to BDF2). Consequently, the complete finite difference method is consistent of second order.

An advantage of the single or multiple shooting method is that the involved nonlinear systems exhibits a much lower dimension than the nonlinear systems from finite difference methods. Nevertheless, finite difference methods are often more robust, i.e., the corresponding Newton methods feature better convergence properties. The two disadvantages of the single shooting method mentioned at the beginning of Sect. 7.3 are also omitted in a finite difference method, since no IVPs are resolved now.

Periodic problems

A periodic solution satisfies $y(x+T) = y(x)$ for all $x \in \mathbb{R}$ with the period $T > 0$. The periodic BVP reads $y(0) = y(T)$. In the finite difference method, it follows $u_0 = u_{m+1}$ due to $u_0 \doteq y(0)$, $u_{m+1} \doteq y(T)$, $a = 0$, $b = T$. We can use the relation $u_0 = u_{m+1}$ to eliminate unknowns of the system. For the difference formula (7.22) of first order, it follows a system of $(m+1)n$ equations for $(m+1)n$ unknowns

$$\begin{aligned} u_{j+1} - u_j - hf(x_j, u_j) &= 0 \qquad \text{for } j = 0, \ldots, m-1 \\ u_0 - u_m - hf(x_m, u_m) &= 0. \end{aligned}$$

Moreover, the periodicity condition $y(x+T) = y(x)$ for all x also implies the extended boundary conditions $u_i = u_{m+1+i}$ for possibly new grid points $x_i = ih$ and an integer $i \in \mathbb{Z}$. We can apply difference formulas (7.23) of second order everywhere by identifying $u_0 = u_{m+1}$, $u_{-1} = u_m$. The resulting nonlinear system reads

$$\begin{aligned} u_1 - u_m - 2hf(x_0, u_0) &= 0 \\ u_{j+1} - u_{j-1} - 2hf(x_j, u_j) &= 0 \qquad \text{for } j = 1, \ldots, m-1 \\ u_0 - u_{m-1} - 2hf(x_m, u_m) &= 0. \end{aligned}$$

The corresponding Jacobian matrix exhibits a block tridiagonal structure and additional blocks in the edges of the matrix.

ODE of second order

For a BVP of an ODE of second order

$$y'' = f(x, y), \qquad y(a) = \alpha, \quad y(b) = \beta,$$

we can construct a finite difference method without using the equivalent system of ODEs of first order. We apply the symmetric difference formula

$$\frac{y(x+h) - 2y(x) + y(x-h)}{h^2} = y''(x) + \frac{h^2}{12} y^{(4)}(x + \vartheta h) \tag{7.24}$$

with $\vartheta \in (-1, 1)$. It follows the system

$$\begin{aligned} u_0 - \alpha &= 0 \\ u_{j+1} - 2u_j + u_{j-1} - h^2 f(x_j, u_j) &= 0 \qquad \text{for } j = 1, \ldots, m \\ u_{m+1} - \beta &= 0. \end{aligned} \tag{7.25}$$

If the function f is nonlinear with respect to y, then we apply a Newton iteration to solve this large nonlinear system.

Example

We discuss again the BVP (7.10) with parameter $\lambda = 5$. The grid (7.21) is used for $a = 0, b = 1$ and $m + 1 = 50$. According to (7.25), the nonlinear system is

$$u_{j+1} - 2u_j + u_{j-1} = h^2 \lambda \sin(2\pi u_j) \qquad \text{for } j = 1, 2, \ldots, m. \tag{7.26}$$

The boundary conditions allow to eliminate the unknowns $u_0 = 0,\ u_{m+1} = 1$ directly. Hence the nonlinear system (7.26) consists of m equations for the unknowns $u_1, \ldots, u_m$. We use an ordinary Newton iteration to obtain an approximate solution. As starting values, we employ $u_j = jh$ for all j, which corresponds to the exact solution in the case $\lambda = 0$. Fig. 29 depicts the approximations, which appear after the first iteration step and after the fourth iteration step. We observe just a small difference in these approximations, which indicates a fast convergence of the iteration. The finite difference method yields the same solution as the single shooting method

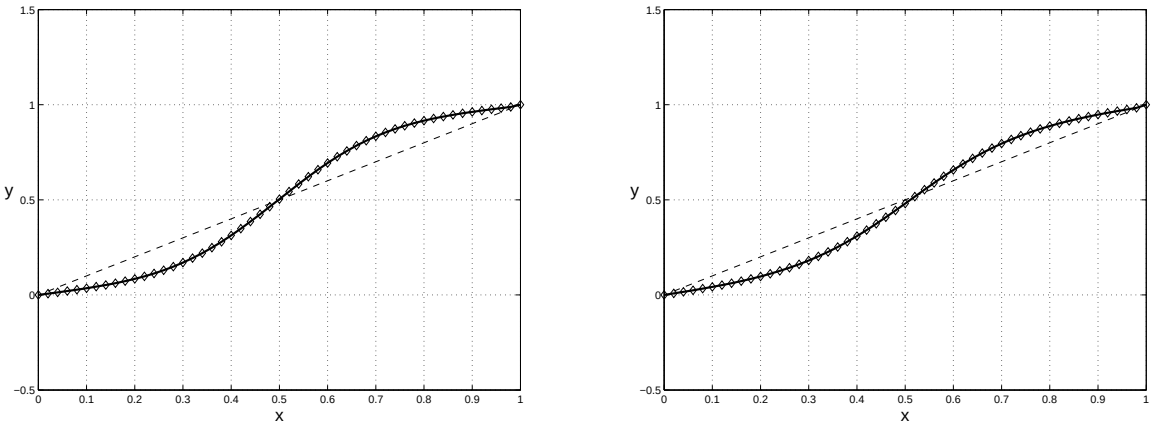

Figure 29: Approximations from finite difference method obtained by the first Newton step (left) and by the fourth Newton step (right).

except for numerical errors of the techniques, cf. Fig. 27. Moreover, this finite difference method succeeds in the case $\lambda = 10$ for the same starting values, whereas the single shooting method fails for this parameter value using comparable starting values.

Linear ODE of second order

We focus on linear ODEs of second order now and discuss consistency, stability and convergence of a finite difference method in detail. We consider the specific BVP

$$-y''(x) + q(x)y(x) = g(x), \quad y(a) = \alpha, \quad y(b) = \beta \tag{7.27}$$

with predetermined functions $q, g \in C[a, b]$. The condition $q(x) \geq 0$ implies the existence of a unique solution. Moreover, we assume $y \in C^4[a, b]$. Using the symmetric difference formula (7.24), we define the local errors

$$\tau_j := y''(x_j) - \frac{y(x_{j+1}) - 2y(x_j) + y(x_{j-1})}{h^2} = -\frac{h^2}{12} y^{(4)}(x_j + \vartheta_j h). \tag{7.28}$$

for $j = 1, \ldots, m$. The discretisation is consistent, since it holds

$$\lim_{h \to 0} \tau_j = \lim_{h \to 0} -\frac{h^2}{12} y^{(4)}(x_j + \vartheta_j h) = 0$$

uniformly for each j provided that $y \in C^4[a,b]$. Note that $h = \frac{b-a}{m+1}$.

The exact solution of the BVP (7.27) satisfies

$$\begin{aligned} y(x_0) &= \alpha \\ -\tfrac{y(x_{j+1})-2y(x_j)+y(x_{j-1})}{h^2} + q(x_j)y(x_j) &= g(x_j) + \tau_j \qquad \text{for } j = 1, \ldots, m \\ y(x_{m+1}) &= \beta. \end{aligned}$$

We define vectors

$$\hat{y} := \begin{pmatrix} y(x_1) \\ y(x_2) \\ \vdots \\ y(x_{m-1}) \\ y(x_m) \end{pmatrix}, \quad \tau := \begin{pmatrix} \tau_1 \\ \tau_2 \\ \vdots \\ \tau_{m-1} \\ \tau_m \end{pmatrix}, \quad d := \begin{pmatrix} g(x_1) + \frac{\alpha}{h^2} \\ g(x_2) \\ \vdots \\ g(x_{m-1}) \\ g(x_m) + \frac{\beta}{h^2} \end{pmatrix}$$

and a symmetric tridiagonal matrix $A \in \mathbb{R}^{m \times m}$

$$A := \frac{1}{h^2} \begin{pmatrix} 2 + q(x_1)h^2 & -1 & & & & \\ -1 & 2 + q(x_2)h^2 & -1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & -1 & 2 + q(x_{m-1})h^2 & -1 & \\ & & & -1 & 2 + q(x_m)h^2 \end{pmatrix}.$$

It follows

$$A\hat{y} = d + \tau. \tag{7.29}$$

Since we do not know the local discretisation errors in τ, we solve the system

$$Au = d \tag{7.30}$$

to achieve an approximation $u = (u_1, \ldots, u_m)$.

To analyse the error $e := u - \hat{y} \in \mathbb{R}^m$, we require the following lemma. Thereby, we write $A \leq B$ for $A = (a_{ij}) \in \mathbb{R}^{m \times m}$ and $B = (b_{ij}) \in \mathbb{R}^{m \times m}$ if $a_{ij} \leq b_{ij}$ holds for all $i, j = 1, \ldots, m$.

Lemma 5 *If $q(x_j) \geq 0$ holds for all $j = 1, \ldots, m$, then the symmetric matrix A is positive definite and it holds $0 \leq A^{-1} \leq A_0^{-1}$ with the symmetric and positive definite matrix*

$$A_0 := \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}.$$

The proof can be found in the book of Stoer/Bulirsch.

In particular, it follows that the linear system (7.30) owns a unique solution. We can use the Cholesky decomposition to solve the system (7.30). However, an LU-decomposition (without pivoting) is more efficient in case of tridiagonal matrices.

The property shown by Lemma 5 corresponds to the stability of the method. Stability means the Lipschitz-continuous dependence of the numerical solution with respect to perturbations in the input data, where the Lipschitz-constants are independent of the step size of a discretisation.

Theorem 18 *We consider $Au = r$ and $A\tilde{u} = \tilde{r}$ with the matrix A from above and arbitrary right-hand sides $r, \tilde{r}$. If the property $q(x_j) \geq 0$ holds for all $j = 1, \ldots, m$, then it follows*

$$\max_{j=1,\ldots,m} |u_j - \tilde{u}_j| \leq \tfrac{1}{2}(b-a)^2 \max_{j=1,\ldots,m} |r_j - \tilde{r}_j|.$$

Proof:

Let $A^{-1} = (\bar{a}_{ij})$ and $A_0^{-1} = (\bar{a}_{ij}^0)$. Lemma 5 implies $0 \leq \bar{a}_{ij} \leq \bar{a}_{ij}^0$. We define

$$\rho := \max_{l=1,\ldots,m} |r_l - \tilde{r}_l|.$$

We have $A(u-\tilde{u}) = r-\tilde{r}$. It follows $u-\tilde{u} = A^{-1}(r-\tilde{r})$ and component-wise

$$\begin{aligned} |u_i - \tilde{u}_i| &= \left| \sum_{j=1}^{m} \bar{a}_{ij}(r_j - \tilde{r}_j) \right| \quad \leq \quad \sum_{j=1}^{m} |\bar{a}_{ij}| \cdot |r_j - \tilde{r}_j| \quad \leq \quad \rho \sum_{j=1}^{m} |\bar{a}_{ij}| \\ &= \rho \sum_{j=1}^{m} \bar{a}_{ij} \quad \leq \quad \rho \sum_{j=1}^{m} \bar{a}^0_{ij} \quad = \quad \rho \sum_{j=1}^{m} \bar{a}^0_{ij} \cdot 1. \end{aligned}$$

Using the notation

$$|u| := (|u_1|, \ldots, |u_m|)^\top \in \mathbb{R}^m,$$

it follows

$$|u - \tilde{u}| \leq \rho A_0^{-1} \mathbb{1}$$

with $\mathbb{1} := (1, \ldots, 1)^\top \in \mathbb{R}^m$.

Now we calculate $A_0^{-1}\mathbb{1}$ directly. Consider the auxiliary BVP

$$-y''(x) = 1, \quad y(a) = y(b) = 0. \tag{7.31}$$

The exact solution reads $y(x) = \frac{1}{2}(x-a)(b-x)$. Since $y^{(4)} \equiv 0$ holds, the local errors (7.28) are zero. The finite difference method applied to (7.31) yields the linear system $A_0\hat{y} = \mathbb{1}$. It follows $\hat{y} = A_0^{-1}\mathbb{1}$ and

$$\left(A_0^{-1}\mathbb{1}\right)_i = y(x_i) = \tfrac{1}{2}(x_i - a)(b - x_i).$$

We achieve the estimate

$$(A_0^{-1}\mathbb{1})_i \leq \tfrac{1}{2}(b-a)^2 \qquad \text{for } \; i = 1, \ldots, m \; ,$$

which confirms the statement of the theorem. □

We conclude that the property $q(x) \geq 0$ is sufficient for the stability of the finite difference method, since the Lipschitz-constant $\frac{1}{2}(b-a)^2$ in Theorem 18 is independent of h or, equivalently, m. The relation $u - \tilde{u} = A^{-1}(r - \tilde{r})$ implies directly

$$\|u - \tilde{u}\|_\infty = \|A^{-1}\|_\infty \cdot \|r - \tilde{r}\|_\infty.$$

However, A and thus A^{-1} depend on m. Theorem 18 yields the uniform bound

$$\|A^{-1}\|_\infty \leq \tfrac{1}{2}(b-a)^2 \quad \text{for all } m.$$

Note that the stability criterion of multistep methods for IVPs also represents a uniform bound, cf. Sect. 4.3.

Concerning the convergence of the finite difference method, we achieve the following theorem, which employs the consistency and the stability of the technique.

Theorem 19 *Assume that the linear BVP (7.27) exhibits a unique solution $y \in C^4[a,b]$ with $|y^{(4)}(x)| \leq M$ for all $x \in [a,b]$. Let $q(x) \geq 0$ for all x. Then the approximation u from (7.30) satisfies*

$$\max_{j=1,\ldots,m} |u_j - y(x_j)| \leq \frac{M}{24}(b-a)^2 h^2,$$

i.e., the finite difference method is convergent of order two.

Proof:

For the solution of the systems $A\hat{y} = d + \tau$ (7.29) and $Au = d$ (7.30), Theorem 18 implies ($\tilde{r} - r = \tau$)

$$\max_{j=1,\ldots,m} |u_j - y(x_j)| \leq \tfrac{1}{2}(b-a)^2 \max_{j=1,\ldots,m} |\tau_j|.$$

The local errors (7.28) satisfy the estimate

$$|\tau_j| \leq \frac{M}{12} h^2 \qquad \text{for all } j = 1, \ldots, m$$

due to $|y^{(4)}(x)| \leq M$ for all $x \in [a,b]$. Thus the convergence is verified. $\square$

We recognise that the order of convergence coincides with the order of consistency present in the discretisation.

7.5 Techniques with Trial Functions

A different approach to solve the BVP (7.1),(7.2) numerically uses trial functions (also: ansatz functions) to approximate the solution. For simplicity, we consider the scalar case $n = 1$, i.e., $y : [a, b] \to \mathbb{R}$. We specify a linear combination

$$u(x; \alpha_1, \ldots, \alpha_k) := v_0(x) + \sum_{l=1}^{k} \alpha_l v_l(x) \tag{7.32}$$

for $\alpha_1, \ldots, \alpha_k \in \mathbb{R}$ with predetermined linearly independent trial functions

$$v_0, v_1, \ldots, v_k : [a, b] \to \mathbb{R}, \qquad V_k := \operatorname{span}\{v_1, \ldots, v_k\}. \tag{7.33}$$

The trial functions shall be smooth, i.e., $v_l \in C^1[a, b]$ for all l. The scalar coefficients $\alpha_1, \ldots, \alpha_k \in \mathbb{R}$ are unknown. It holds $u \in v_0 + V_k$. The fixed function v_0 can be used to match the boundary condition. Otherwise select $v_0 \equiv 0$. We demand that each linear combination satisfies the boundary conditions, i.e.,

$$r(u(a; \alpha_1, \ldots, \alpha_k), u(b; \alpha_1, \ldots, \alpha_k)) = 0 \quad \text{for all } \alpha_1, \ldots, \alpha_k. \tag{7.34}$$

Furthermore, the trial functions shall be elementary such that the derivative

$$u'(x; \alpha_1, \ldots, \alpha_k) = v_0'(x) + \sum_{l=1}^{k} \alpha_l v_l'(x)$$

can be evaluated easily.

A special case represent IVPs $y' = f(x, y), y(a) = y_0$ for $x \in [a, b]$. We can choose polynomials as trial functions. It follows ($v_0 = y_0$, $v_l = (x - a)^l$)

$$u(x; \alpha_1, \ldots, \alpha_k) = y_0 + \sum_{l=1}^{k} \alpha_l (x - a)^l.$$

The initial condition is always fulfilled. In a collocation method, the coefficients are determined such that the ODE is satisfied at certain points $x_j \in [a, b]$ for $j = 1, \ldots, k$, cf. Sect. 5.4.

For periodic BVPs (7.7), trigonometric polynomials are applied as trial functions

$$u(x;\alpha_0,\alpha_1,\ldots,\alpha_k,\beta_1,\ldots,\beta_k) := \frac{\alpha_0}{2} + \sum_{l=1}^{k} \alpha_l \sin\left(\tfrac{2\pi}{T}x\right) + \beta_l \cos\left(\tfrac{2\pi}{T}x\right), \tag{7.35}$$

where a function space V with $\dim(V) = 2k+1$ appears ($v_0 \equiv 0$). Each trial function is periodic and thus the linear combination is also periodic.

Since u always satisfies the boundary conditions due to (7.34), we have to determine the coefficients such that a good approximation of the solution of the ODE-BVP is achieved. To quantify the accuracy, we define the residual $q : [a,b] \to \mathbb{R}$ of u via

$$q(x;\alpha_1,\ldots,\alpha_k) := u'(x;\alpha_1,\ldots,\alpha_k) - f(x,u(x;\alpha_1,\ldots,\alpha_k)). \tag{7.36}$$

For $v_0,\ldots,v_k \in C^1[a,b]$ and f continuous, the residual satisfies $q \in C[a,b]$. If f is nonlinear, then q depends nonlinearly on the coefficients $\alpha_1,\ldots,\alpha_k$. The residual of the exact solution is equal to zero. Consequently, we want that the residual of u becomes small in some sense. There are several possibilities to determine the according coefficients:

1. **Minimisation**:
   We apply the integral norm $\|\cdot\| : C[a,b] \to \mathbb{R}$ of the residual

   $$\|q(\cdot;\alpha_1,\ldots,\alpha_k)\|^2 := \int_a^b q(x;\alpha_1,\ldots,\alpha_k)^2 \,\mathrm{d}x. \tag{7.37}$$

   Now the coefficients $\hat{\alpha}_1,\ldots,\hat{\alpha}_k$ are determined such that the norm is minimised, i.e.,

   $$\|q(\cdot;\hat{\alpha}_1,\ldots,\hat{\alpha}_k)\| \le \|q(\cdot;\alpha_1,\ldots,\alpha_k)\| \quad \text{for all } \alpha_1,\ldots,\alpha_k.$$

   However, a minimisation procedure demands a relatively large computational effort. Thus this approach is usually not used in practice. We do not want an optimisation problem for the coefficients but a nonlinear system of algebraic equations for the coefficients.

2. **Method of weighted residuals / Galerkin approach**:
In this strategy, we apply the inner product corresponding to the integral norm (7.37). Let $\langle \cdot, \cdot \rangle : C[a,b] \times C[a,b] \to \mathbb{R}$ be defined by

$$\langle g, h \rangle := \int_a^b g(x) \cdot h(x) \, \mathrm{d}x \quad \text{for } \; g, h \in C[a,b].$$

Now the idea is to choose a second set of linearly independent functions

$$w_1, \ldots, w_k : [a,b] \to \mathbb{R}, \qquad W_k := \operatorname{span}\{w_1, \ldots, w_k\}, \tag{7.38}$$

which are called test functions. In contrast to the trial functions (7.33), test functions $w_l \in C[a,b]$ are also sufficient. Now we demand that the residual is orthogonal to the space W_k, i.e.,

$$\langle q(\cdot; \alpha_1, \ldots, \alpha_k), w_j(\cdot) \rangle = 0 \quad \text{for all } \; j = 1, \ldots, k. \tag{7.39}$$

Hence we obtain k nonlinear equations for the coefficients $\alpha_1, \ldots, \alpha_k$. If the space W_k contains good approximations of all possible residuals, then the condtion (7.39) produces a specific u with a residual, which is close to the minimum of all residuals. Thus the test functions should approximate all possible residuals, whereas the trial functions approximate the exact solution.

In the special case $v_0 \equiv 0, w_1 \equiv v_1, \ldots, w_k \equiv v_k$, the test functions (7.38) coincide with the trial functions (7.33), i.e., $V_k = W_k$. This approach is called Galerkin method. The advantage is that the construction of a second set of functions is not necessary. Moreover, symmetric and positive definite matrices result in the case of linear ODEs. For example, the Galerkin method is suitable for periodic BVPs (7.7) with trial functions (7.35). Using a large degree k, the trigonometric polynomials yield good approximations for a broad class of periodic functions. The exact solution of (7.7) as well as the residual for (7.35) are periodic. Thus it is reasonable to choose the same space for trial functions and test functions.

3. **Collocation method**:
   Another approach is to demand simply that the residual vanishes at k prescribed points $a \leq x_1 < x_2 < \cdots < x_k \leq b$. We achieve the nonlinear system

$$q(x_j; \alpha_1, \ldots, \alpha_k) = 0 \quad \text{for } \; j = 1, \ldots, k \tag{7.40}$$

   including the unknown coefficients.

Example: ODE of second order

We consider the BVP $y'' = f(x, y)$, $y(a) = \alpha$, $y(b) = \beta$ of an ODE of second order. Just replace the first by the second derivative in the above examples. The method of weighted residuals implies the nonlinear system

$$\int_a^b \left(v_0''(x) + \sum_{l=1}^{k} \alpha_l v_l''(x) - f\left(x, v_0(x) + \sum_{l=1}^{k} \alpha_l v_l(x)\right)\right) \cdot w_j(x) \, \mathrm{d}x = 0$$

for $j = 1, \ldots, k$ to determine the coefficients $\alpha_1, \ldots, \alpha_k$.

We arrange the trial functions

$$v_0(x) = a + \frac{\beta - \alpha}{b - a}(x - a), \qquad v_l(x) = x^{l-1}(x - a)(x - b) \quad \text{for } \; l \geq 1.$$

The function v_0 satisfies the demanded boundary conditions, whereas it holds $v_l(a) = v_l(b) = 0$ for $l \geq 1$. We recognise that $v_0 + V_k \subset \mathbb{P}_{k+1}$ and $v_0 + V_k \neq \mathbb{P}_{k+1}$. Although $\dim(\mathbb{P}_{k+1}) = k + 2$, the degrees of freedom reduce to k coefficients due to the two boundary conditions. As test functions, we choose polynomials $w_l(x) := x^{l-1}$ and thus

$$W_k = \operatorname{span}\left\{1, x, x^2, \ldots, x^{k-1}\right\} = \mathbb{P}_{k-1}.$$

For example, we arrange the linear BVP $y'' = \lambda y$, $y(0) = 0$, $y(1) = 1$ with $\lambda > 0$, say $\lambda = 25$. It follows $v_0 \equiv x$. The method of the weighted residuals yields

$$\int_0^1 \left(\sum_{l=1}^{k} \alpha_l v_l''(x) - \lambda \left(v_0(x) + \sum_{l=1}^{k} \alpha_l v_l(x)\right)\right) \cdot w_j(x) \, \mathrm{d}x = 0,$$

which is equivalent to

$$\sum_{l=1}^{k} \alpha_l \left(\int_0^1 (v_l''(x) - \lambda v_l(x)) w_j(x) \, \mathrm{d}x \right) = \lambda \int_0^1 v_0(x) w_j(x) \, \mathrm{d}x$$

for $j = 1, \ldots, k$. We obtain a linear system for $\alpha_1, \ldots, \alpha_k$. The corresponding matrix is dense in contrast to finite difference methods. Since all involved functions are polynomials, the required integrals can be calculated analytically. Fig. 30 illustrates the approximations of this technique for $k = 1, 2, 3, 4$. Futhermore, the corresponding maximum errors

$$\max_{x \in [0,1]} |u(x; \alpha_1, \ldots, \alpha_k) - y(x)| \quad \text{with} \quad y(x) = \frac{\sinh(5x)}{\sinh(5)}$$

are shown in Fig. 31. We observe that the error decreases exponentially for increasing dimension k in this academic example.

Galerkin method for periodic problems

We consider a scalar ODE $y' = f(x, y)$ with the periodic boundary conditions (7.7). The trial functions are the trigonometric polynomials (7.35). We apply the complex formulation

$$v_l(x) = \mathrm{e}^{\mathrm{i}\omega l x} \qquad \text{for } \; l = -k, \ldots, k \tag{7.41}$$

with the imaginary unit $\mathrm{i} = \sqrt{-1}$ and the frequency $\omega = \frac{2\pi}{T}$. The according inner product $\langle \cdot, \cdot \rangle : C[a, b] \times C[a, b] \to \mathbb{C}$ reads

$$\langle g, h \rangle := \frac{1}{T} \int_0^T g(x) \cdot \overline{h(x)} \, \mathrm{d}x.$$

The basis functions (7.41) are orthonormal, i.e.,

$$\langle v_l, v_j \rangle = \begin{cases} 1 & \text{for } \; l = j, \\ 0 & \text{for } \; l \neq j. \end{cases}$$

Furthermore, it holds

$$v_l'(x) = \mathrm{i}\omega l \mathrm{e}^{\mathrm{i}\omega l x} = \mathrm{i}\omega l v_l(x).$$

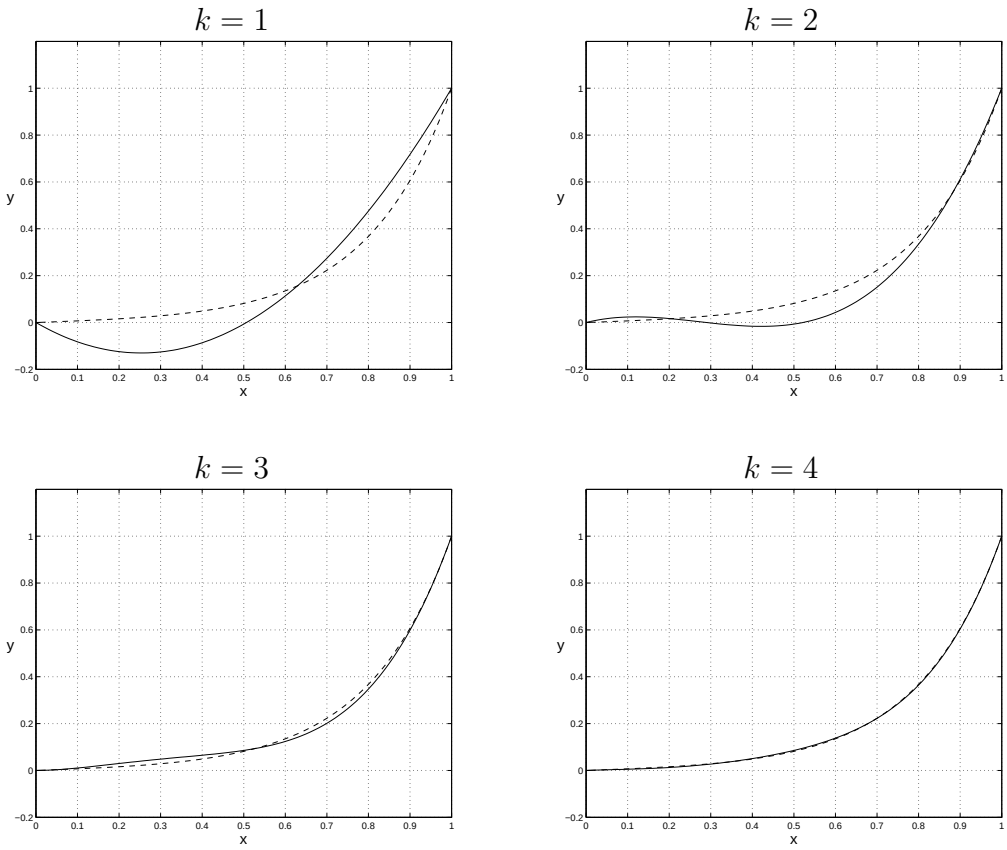

Figure 30: Approximations from method of weighted residuals using k trial functions (solid lines) and exact solution of ODE-BVP (dashed lines).

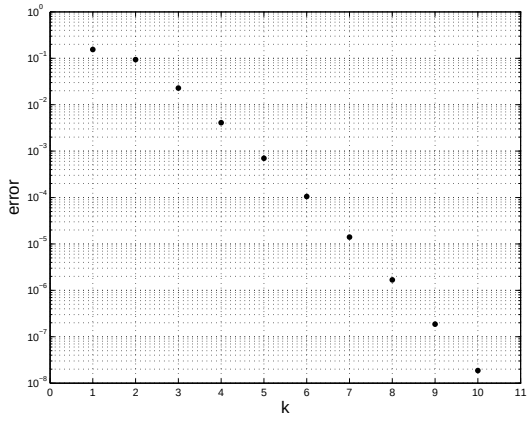

Figure 31: Maximum absolute errors in method of weighted residuals using k trial function (semi-logarithmic scale).

We apply the Galerkin method, i.e., the test functions $w_l = v_l$. The condition $\langle q, v_j \rangle = 0$ for the residual q yields

$$\left\langle \sum_{l=-k}^{k} \alpha_l v_l'(x), v_j(x) \right\rangle - \left\langle f\left(x, \sum_{l=-k}^{k} \alpha_l v_l(x)\right), v_j(x) \right\rangle = 0$$

for $j = -k, \ldots, k$. It follows

$$(\mathrm{i}\omega j) \cdot \alpha_j - \left\langle f\left(x, \sum_{l=-k}^{k} \alpha_l v_l(x)\right), v_j(x) \right\rangle = 0$$

for $j = -k, \ldots, k$. We apply the notation

$$Z := \begin{pmatrix} \alpha_{-k} \\ \vdots \\ \alpha_k \end{pmatrix}, \quad P := \begin{pmatrix} \langle f, v_{-k} \rangle \\ \vdots \\ \langle f, v_k \rangle \end{pmatrix}, \quad \Omega := \mathrm{i}\omega \begin{pmatrix} -k & & \\ & \ddots & \\ & & k \end{pmatrix}.$$

It follows the nonlinear system

$$G(Z) :\equiv \Omega Z - P(Z) = 0.$$

However, we cannot evaluate the integrals of the inner products exactly for a general nonlinear function f. Hence we apply a quadrature scheme. For periodic integrands, trapezoidal rule with $2k+1$ equidistant nodes $x_j = jh$ is the most efficient method. Let

$$y_j := \sum_{l=-k}^{k} \alpha_l v_l(x_j) \tag{7.42}$$

and $Y := (y_0, \ldots, y_{2k})^\top$. The trapezoidal rule yields

$$\langle f, v_j \rangle \doteq \frac{h}{T} \sum_{l=0}^{2k} f(x_l, y_l) \cdot \overline{v_j(x_l)} = \frac{h}{T} \sum_{l=0}^{2k} f(x_l, y_l) \cdot v_{-j}(x_l). \tag{7.43}$$

Let $E(Y) := (f(x_0, y_0), \ldots, f(x_{2k}, y_{2k}))^\top \in \mathbb{C}^{2k+1}$.

The step (7.42) corresponds to an inverse (discrete) Fourier transformation, since it represents the evaluation of trigonometric polynomials. Thus it

holds $Y = \mathcal{F}^{-1}Z$ with a matrix $\mathcal{F}^{-1}$. The evaluations (7.43) can be written as a (discrete) Fourier transformation, i.e., $P \doteq \mathcal{F}E(Y)$ with a matrix $\mathcal{F}$. It holds $\mathcal{F}\mathcal{F}^{-1} = \sigma I$. Hence the Galerkin method can be summerised by

$$G(Z) \equiv \Omega Z - \mathcal{F}E(\mathcal{F}^{-1}(Z)) = 0$$

with a function $G : \mathbb{C}^{2k+1} \to \mathbb{C}^{2k+1}$. We obtain a nonlinear system for the coefficients Z, since the mapping $E : \mathbb{C}^{2k+1} \to \mathbb{C}^{2k+1}$ is nonlinear. The corresponding Jacobian matrix reads

$$\mathrm{D}G = \Omega - \mathcal{F} \cdot \frac{\partial E}{\partial Y} \cdot \mathcal{F}^{-1}.$$

The matrix $\frac{\partial E}{\partial Y}$ consists of Jacobian matrices $\mathrm{D}f$. All steps, where transformations with the matrices $\mathcal{F}, \mathcal{F}^{-1}$ are involved, can be done efficiently by fast Fourier transformation (FFT). Furthermore, an equivalent Galerkin method with real numbers only can be constructed using (7.35).

This particular Galerkin approach represents a method in frequency domain, since the unknowns are the Fourier coefficients Z. For simulating electric circuit, this method is called harmonic balance. In contrast, a finite difference method is a technique in time domain (x often represents the time). The above Galerkin approach is more efficient than a finite difference method, if the number k of basis functions required for a sufficiently accurate approximation is relatively low.

Outlook: Variational Methods

Another class of numerical techniques for solving boundary value problems of systems of ODEs are the variational methods. These techniques apply to some important types of problems, where the solutions possess certain minimality properties. For further reading, we refer to the book of Stoer/Bulirsch (Section 7.5).