Chapter 2
Numerical Methods for Stiff Ordinary Differential Equations

2.1 Stiff Ordinary Differential Equations

Remark 2.1. Stiffness. It was observed in Curtiss & Hirschfelder (1952) that explicit methods failed for the numerical solution of initial value problems for ordinary differential equations that model certain chemical reactions. They introduced the notation stiffness for such chemical reactions where the fast reacting components arrive in a very short time in their equilibrium and the slowly changing components are more or less fixed, i.e., stiff. In 1963, Dahlquist found out that the reason for the failure of explicit Runge–Kutta methods is their bad stability, see Section 2.3. It should be emphasized that the stability properties of the equations themselves are good, it is in fact a problem of the explicit methods.

There is no unique definition of stiffness in the literature. However, essential properties of stiff systems are as follows:

- There exist, for certain initial conditions, solutions that change slowly.
- Solutions in a neighborhood of these smooth solutions converge quickly to them.

A definition of stiffness can be found in (Strehmel & Weiner, 1995, p. 202), (Strehmel *et al.*, 2012, p. 208). This definition involves a certain norm that depends on the equation and it might be complicated to evaluate this norm. If the solution of (1.1) is sought in the interval $[x_0, x_e]$ and if the right-hand side of (1.1) is Lipschitz continuous in the second argument with Lipschitz constant L, then an approximation of this definition is as follows. A system of ordinary differential equations is called stiff if

$$L\,(x_e - x_0) \gg 1. \tag{2.1}$$

The term on the left-hand side corresponds to the term in the exponential of the error bound (1.7) for the global error. Thus, the first factor in the error bound is very large.

Another definition of stiffness will be given in Definition 2.28. □

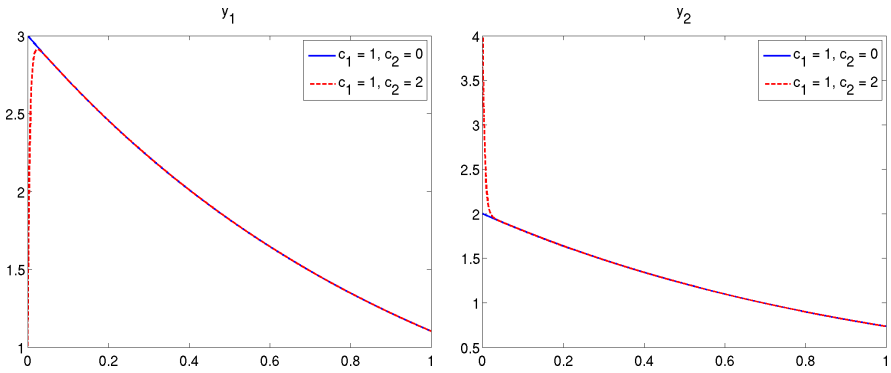

Fig. 2.1 Solutions of Example 2.2, left: first component, right: second component.

Example 2.2. Stiff system of ordinary differential equations. Consider the system

$$\begin{aligned} y_1' &= -80.6y_1 + 119.4y_2, \\ y_2' &= 79.6y_1 - 120.4y_2, \end{aligned}$$

in $(0, 1)$. This system is a linear system of ordinary differential equations that can be written in the form

$$\boldsymbol{y}' = \begin{pmatrix} -80.6 & 119.4 \\ 79.6 & -120.4 \end{pmatrix} \boldsymbol{y}.$$

Taking as Lipschitz constant, e.g., the l_1 norm of the system matrix (column sums), one gets $L = 239.8$ and condition (2.1) is satisfied. The general solution of this system is, compare Appendix A.2.3,

$$\boldsymbol{y}(x) = c_1 \begin{pmatrix} 3 \\ 2 \end{pmatrix} e^{-x} + c_2 \begin{pmatrix} -1 \\ 1 \end{pmatrix} e^{-200x}.$$

The first component is the slowly changing one and the second component the quickly (close to $x = 0$) changing one. The constants are determined by the initial condition. If the initial condition is such that $c_2 = 0$, then the solution is smooth for all $x > 0$. Otherwise, if $c_2 \neq 0$, then the solutions changes rapidly for small x while approaching the smooth solution, see Figure 2.1 □

2.2 Implicit Runge–Kutta Schemes

Remark 2.3. Motivation. If the upper triangular part of the matrix of a Runge–Kutta method, see Definition 1.22, is not identical to zero, the Runge–

Kutta method is called implicit. That means, there are increments that depend not only on previously computed increments but also on not yet computed increments. Thus, one has to solve a nonlinear problem for computing these increments. Consequently, the implementation of implicit Runge–Kutta methods is much more involved compared with the implementation of explicit Runge–Kutta methods. Generally, performing one step of an implicit method is much more time-consuming than for an explicit method. However, the great advantage of implicit methods is that they can be used for the numerical simulation of stiff systems, see the stability theory in Section 2.3. □

Remark 2.4. Derivation of implicit Runge–Kutta methods. Implicit Runge–Kutta schemes can be derived from the integral representation (1.8) of the initial value problem. One can show that for each implicit Runge–Kutta scheme with the weights b_j and the nodes $x_k + c_j h$ there is a corresponding quadrature rule with the same weights and the same nodes, see the section on Gaussian quadrature in Numerical Mathematics I. □

Example 2.5. Gauss–Legendre quadrature. Consider the interval $[x_k, x_k+h] = [x_k, x_{k+1}]$. Let $c_1, \ldots, c_s$ be the roots of the Legendre polynomial $P_s(t)$ of degree s with the arguments

$$t = \frac{2}{h}(x - x_k) - 1 \quad \Longrightarrow \quad t \in [-1, 1].$$

There are s mutually distinct real roots in $(-1, 1)$. After having computed $c_1, \ldots, c_s$, one can determine the coefficients a_{ij}, b_j such that one obtains a method of order $2s$, see Example 2.8. □

Remark 2.6. Simplifying order conditions. The order conditions for an implicit Runge–Kutta scheme with s stages are the same as given in Theorems 1.26, 1.27, and Remark 1.28. These conditions lead to a nonlinear system of equations for computing the parameters of the scheme. These computations are generally quite complicated.

A useful tool for solving this problem are the so-called simplifying order conditions, introduced in Butcher (1964):

$$\begin{aligned}
B(p) &: \sum_{i=1}^{s} b_i c_i^{k-1} = \frac{1}{k}, \quad k = 1, \ldots, p, \\
C(l) &: \sum_{j=1}^{s} a_{ij} c_j^{k-1} = \frac{1}{k} c_i^k, \quad i = 1, \ldots, s, \ k = 1, \ldots, l, \\
D(m) &: \sum_{i=1}^{s} b_i c_i^{k-1} a_{ij} = \frac{1}{k} b_j \left(1 - c_j^k\right), \quad j = 1, \ldots, s, \ k = 1, \ldots, m,
\end{aligned} \tag{2.2}$$

with $0^0 = 1$.

One can show that for sufficiently large values l and m, the conditions $C(l)$ and $D(m)$ can be reduced to $B(p)$ with appropriate p. □

Remark 2.7. Interpretation of $B(p)$ and $C(l)$. Consider the initial value problem

$$y'(x) = f(x), \quad y(x_0) = 0.$$

With the fundamental theorem of differential calculus, one sees that this problem has the solution

$$y(x_0 + h) = \int_{x_0}^{x_0+h} f(\xi)\ d\xi = h \int_0^1 f\left(x_0 + h\theta\right)\ d\theta.$$

A Runge–Kutta method with s stages gives

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

Consider in particular the case that $f(x)$ is a polynomial $f(x) = (x - x_0)^{k-1}$, $k \in \mathbb{N} \setminus \{0\}$. Then, the analytical solution has the form

$$y(x_0 + h) = h \int_0^1 (h\theta)^{k-1}\, d\theta = \left.\frac{(h\theta)^k}{k}\right|_{\theta=0}^{\theta=1} = \frac{h^k}{k}. \tag{2.3}$$

The Runge–Kutta scheme yields

$$y_1 = h \sum_{i=1}^{s} b_i (c_i h)^{k-1} = h^k \sum_{i=1}^{s} b_i c_i^{k-1}. \tag{2.4}$$

Comparing (2.3) and (2.4), one can observe that condition $B(p)$ means that the quadrature rule that is the basis of the Runge–Kutta method is exact for polynomials of degree $(p-1)$.

Condition $C(1)$ is (1.14) with the upper limit s

$$c_i = \sum_{j=1}^{s} a_{ij}, \quad i = 1, \ldots, s. \tag{2.5}$$

□

Example 2.8. Classes of implicit Runge–Kutta schemes.

- *Gauss–Legendre schemes.* The nodes of the Gauss–Legendre quadrature are used. A method with s stages possesses the maximal possible order $2s$, where all nodes are in the interior of the intervals. To get the optimal order, one has to show that $B(2s)$, $C(s)$, $D(s)$ are satisfied, see (Strehmel *et al.*, 2012, Section 8.1.2), i.e.,

$$\begin{aligned}
\sum_{i=1}^{s} b_i c_i^{k-1} &= \frac{1}{k}, \quad k = 1, \ldots, 2s, \\
\sum_{j=1}^{s} a_{ij} c_j^{k-1} &= \frac{1}{k} c_i^k, \quad i = 1, \ldots, s,\ k = 1, \ldots, s, \\
\sum_{i=1}^{s} b_i c_i^{k-1} a_{ij} &= \frac{1}{k} b_j \left(1 - c_j^k\right), \quad j = 1, \ldots, s,\ k = 1, \ldots, s.
\end{aligned} \tag{2.6}$$

An example is the implicit mid point rule, whose coefficients can be derived by setting $s = 1$ in (2.6). One obtains the following conditions

$$b_1 = 1,\ b_1 c_1 = \frac{1}{2},\ a_{11} = c_1,\ b_1 a_{11} = b_1 \left(1 - c_1\right).$$

Consequently, the implicit mid point rule is given by

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

- *Gauss–Radau*[1] *methods.* These methods are characterized by the feature that one of the end points of the interval $[x_k, x_{k+1}]$ belongs to the nodes. A method of this class with s stages has at most order $2s - 1$.
  Examples ($s = 1$):
  - $\begin{array}{c|c} 0 & 1 \\ \hline & 1 \end{array}$ $s = 1,\ p = 1,$
  - $\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$ $s = 1,\ p = 1,$ implicit Euler scheme.

  The first scheme does not satisfy condition (2.5).
- *Gauss–Lobatto*[2] *methods.* In these methods, both end points of the interval $[x_k, x_{k+1}]$ are nodes. A method of this kind with s stages cannot be of higher order than $(2s - 2)$.
  Examples:
  - trapezoidal rule, Crank[3]–Nicolson[4] scheme

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 1/2 & 1/2 \\ \hline & 1/2 & 1/2 \end{array} \quad s = p = 2.$$

[1] Rodolphe Radau (1835 – 1911)
[2] Rehuel Lobatto (1797 – 1866)
[3] John Crank (1916 – 2006)
[4] Phyllis Nicolson (1917 – 1968)

∘ other scheme

$$\begin{array}{c|cc} 0 & 1/2 & 0 \\ 1 & 1/2 & 0 \\ \hline & 1/2 & 1/2 \end{array} \qquad s = 2,\ p = 2.$$

The second scheme does not satisfy condition (2.5).

□

Remark 2.9. Diagonally implicit Runge–Kutta methods (DIRK methods). For an implicit Runge–Kutta method with s stages and a full matrix A, one has to solve a coupled nonlinear system for the increments $K_1(x,y),\ldots,K_s(x,y)$. This step is expensive for a large number of stages s. A compromise is the use of so-called diagonally implicit Runge–Kutta (DIRK) methods

$$\begin{array}{c|ccccc} c_1 & a_{11} & 0 & 0 & \cdots & 0 \\ c_2 & a_{21} & a_{22} & 0 & \cdots & 0 \\ c_3 & a_{31} & a_{32} & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & a_{s2} & \cdots & & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array} .$$

In DIRK methods, one has to solve s independent nonlinear equations for the increments. In the equation for $K_i(x,y)$, only the stages $K_1(x,y),\ldots,K_i(x,y)$ appear, where $K_1(x,y),\ldots,K_{i-1}(x,y)$ were already computed. □

2.3 Linear Stability Theory

Remark 2.10. On the stability theory. The stability theory studies numerical methods for solving the linear initial value problem

$$y'(x) = \lambda y(x), \quad y(0) = 1,\ \lambda \in \mathbb{C}. \tag{2.7}$$

It will turn out the even at the simple initial value problem (2.7) the most important stability properties of numerical methods can be explored. The solution of (2.7) is

$$y(x) = e^{\lambda x}.$$

If the initial condition will be slightly perturbed to be $1+\delta_0$, then the solution of the perturbed initial value problem is

$$\tilde{y}(x) = (1+\delta_0)e^{\lambda x} = e^{\lambda x} + \delta_0 e^{\lambda x}.$$

If $\lambda = a + ib$ with $a = \mathrm{Re}(\lambda) > 0$, then the difference

$$|y(x) - \tilde{y}(x)| = \left|\delta_0 e^{\lambda x}\right| = |\delta_0|\,|e^{ax}|\left|e^{ibx}\right| = |\delta_0|\,|e^{ax}|$$

becomes for each $\delta_0 \neq 0$ arbitrarily large if x is sufficiently large. That means, the initial value problem (2.7) is not stable in this case. In this situation, one cannot expect that any numerical method is stable. Hence, this situation is not of interest for numerical simulations.

In contrast, if $\mathrm{Re}(\lambda) < 0$, then the difference $|y(x) - \tilde{y}(x)|$ becomes arbitrarily small and the initial value problem is stable, i.e., small changes of the data result only in small changes of the solution. For $\mathrm{Re}(\lambda) = 0$, the difference $|y(x) - \tilde{y}(x)|$ is at least bounded. These cases, in particular the first one, are of interest for the stability theory of methods for solving ordinary differential equations.

This section considers one-step methods with equidistant meshes with step size h. The solution of (2.7) in the node $x_{k+1} = (k+1)h$ is

$$y(x_{k+1}) = e^{\lambda x_{k+1}} = e^{\lambda(x_k+h)} = e^{\lambda h}e^{\lambda x_k} = e^{\lambda h}y(x_k) =: e^z y(x_k),$$

with $z := \lambda h \in \mathbb{C}$, $\mathrm{Re}(z) \leq 0$. Now, it will be studied how the step from x_k to x_{k+1} looks like for different one-step methods. In particular, large steps are of interest, i.e., $|z| \to \infty$. □

Example 2.11. Behavior of different one-step methods for one step of the model problem (2.7).

1. *Explicit Euler method.* The general form of this method is

$$y_{k+1} = y_k + hf\left(x_k, y_k\right).$$

   In particular, one obtains for (2.7)

$$y_{k+1} = y_k + h\lambda y_k = (1+z)\,y_k =: R(z)y_k.$$

   It holds, independently of $\mathrm{Re}(z)$, that $\lim_{|z|\to\infty}|R(z)| = \infty$.
2. *Implicit Euler method.* This method has the form

$$y_{k+1} = y_k + hf\left(x_{k+1}, y_{k+1}\right).$$

   For applying it to (2.7), one can rewrite it as follows

$$\begin{aligned} y_{k+1} &= y_k + h\lambda y_{k+1} && \Longleftrightarrow \\ (1-z)y_{k+1} &= y_k && \Longleftrightarrow \\ y_{k+1} &= \frac{1}{1-z}y_k = \left(1 + \frac{z}{1-z}\right)y_k =: R(z)y_k. \end{aligned}$$

   For this method, one has, independently of $\mathrm{Re}(z)$, that $\lim_{|z|\to\infty}|R(z)| = 0$.
3. *Trapezoidal rule.* The general form of this method is

$$y_{k+1} = y_k + \frac{h}{2}\left(f(x_k, y_k) + f(x_{k+1}, y_{k+1})\right),$$

which can be derived from the Butcher tableau given in Example 2.8. For the linear differential equation (2.7), one gets

$$\begin{aligned} y_{k+1} &= y_k + \frac{h}{2}\left(\lambda y_k + \lambda y_{k+1}\right) &&\iff \\ \left(1 - \frac{z}{2}\right) y_{k+1} &= \left(1 + \frac{z}{2}\right) y_k &&\iff \\ y_{k+1} &= \frac{1 + z/2}{1 - z/2} y_k = \left(1 + \frac{z}{1 - z/2}\right) y_k =: R(z) y_k. \end{aligned}$$

Let $z = 2r\left(\cos(\phi) + i\sin(\phi)\right)$. Inserting this expression gives

$$\begin{aligned} \lim_{|z|\to\infty} \left|\frac{1 + z/2}{1 - z/2}\right| &= \lim_{r\to\infty} \left|\frac{1 + r\left(\cos(\phi) + i\sin(\phi)\right)}{1 - r\left(\cos(\phi) + i\sin(\phi)\right)}\right| \\ &= \lim_{r\to\infty} \left|\frac{1/r + \left(\cos(\phi) + i\sin(\phi)\right)}{1/r - \left(\cos(\phi) + i\sin(\phi)\right)}\right| \\ &= \frac{\left|\left(\cos(\phi) + i\sin(\phi)\right)\right|}{\left|-\left(\cos(\phi) + i\sin(\phi)\right)\right|} = \frac{1}{1} = 1. \end{aligned}$$

Hence, one has that $\lim_{|z|\to\infty} |R(z)| = 1$ for the trapezoidal rule, independently of ϕ, and with that independently of $\mathrm{Re}(z)$.

The function $R(z)$ describes for each method the step from x_k to x_{k+1}. Thus, this function is an approximation of e^z, which has for different methods different properties, e.g., the limit for $|z| \to \infty$. □

Definition 2.12. Stability function. Let $\mathbf{1} = (1, \ldots, 1)^T \in \mathbb{R}^s$, $\hat{\mathbb{C}} = \mathbb{C} \cup \infty$, where ∞ has to be understood as in function theory (Riemann sphere), and consider a Runge–Kutta method with s stages and with the parameters $(A, \boldsymbol{b}, \boldsymbol{c})$. Then, the function

$$R \;:\; \hat{\mathbb{C}} \to \hat{\mathbb{C}}, \quad z \mapsto 1 + z\boldsymbol{b}^T (I - zA)^{-1}\mathbf{1} \tag{2.8}$$

is called stability function of the Runge–Kutta method. □

Remark 2.13. Stability functions from Example 2.11. All stability functions from Example 2.11 can be written in the form (2.8). One obtains, e.g., for the trapezoidal rule

$$\boldsymbol{b} = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}, \quad I - zA = \begin{pmatrix} 1 & 0 \\ -\frac{z}{2} & 1 - \frac{z}{2} \end{pmatrix}, \quad (I - zA)^{-1} = \frac{1}{1 - \frac{z}{2}} \begin{pmatrix} 1 - \frac{z}{2} & 0 \\ \frac{z}{2} & 1 \end{pmatrix},$$

from what follows that

$$1 + z\boldsymbol{b}^T(I - zA)^{-1}\mathbf{1} = 1 + \frac{z}{1 - z/2}\left(\frac{1}{2} - \frac{z}{4} + \frac{z}{4} + \frac{1}{2}\right) = 1 + \frac{z}{1 - z/2}.$$

□

Theorem 2.14. Form of the stability function of Runge–Kutta methods. *Given a Runge–Kutta scheme with s stages and with the parameters $(A, \boldsymbol{b}, \boldsymbol{c})$, then the stability function $R(z)$ is a rational function defined on $\hat{\mathbb{C}}$, whose polynomial order in the numerator and in the denominator is at most s. The poles of this functions might be only at values that correspond to the inverse of an eigenvalue of A. For an explicit Runge–Kutta scheme, $R(z)$ is a polynomial.*

Proof. Consider first an explicit Runge–Kutta scheme. In this case, the matrix A is a strictly lower triangular matrix. Hence, $I - zA$ is a triangular matrix with the values one at its main diagonal. This matrix is invertible and it is

$$(I - zA)^{-1} = I + zA + \ldots + z^{s-1}A^{s-1}, \tag{2.9}$$

which can be checked easily by multiplication with $(I - zA)$ and using that $A^s = 0$ since A is strictly lower triangular. It follows from (2.8) and (2.9) that $R(z)$ is a polynomial in z of degree at most s.

Now, the general case will be considered. The expression $(I - zA)^{-1}\mathbf{1}$ can be interpreted as the solution of the linear system of equations $(I - zA)\underline{\zeta} = \mathbf{1}$. Using the Cramer rule, one finds that the i-th component of the solution has the form

$$\zeta_i = \frac{\det A_i}{\det(I - zA)},$$

where A_i is the matrix that is obtained by replacing the i-th column of $(I - zA)$ by the right-hand side, i.e., by $\mathbf{1}$. The numerator of ζ_i is a polynomial in z of order at most $(s-1)$ since there is one column where z does not appear. The denominator is a polynomial of degree at most s. Multiplying with $z\boldsymbol{b}^T$ from the left-hand side gives just a rational function with polynomials of at most degree s both in the numerator and in the denominator.

There is only one case where this approach does not work, namely if

$$\det(I - zA) = \det(z(I/z - A)) = z^s \det(I/z - A) = 0,$$

i.e., if $1/z$ is an eigenvalue of A. ■

Theorem 2.15. Solution of the initial value problem (2.7) obtained with a Runge–Kutta scheme. *Consider a Runge–Kutta method with s stages and with the parameters $(A, \boldsymbol{b}, \boldsymbol{c})$. If $z^{-1} = (\lambda h)^{-1}$ is not an eigenvalue of A, then the Runge–Kutta scheme is well-defined for the initial value problem (2.7). In this case, it is*

$$y_k = (R(h\lambda))^k, \quad k = 0, 1, 2, \ldots.$$

Proof. The statement of the theorem follows directly if one writes the Runge–Kutta scheme for (2.7) and applies induction. *exercise* ■

Definition 2.16. Stability domain. The stability domain of a one-step method is the set

$$S := \{z \in \hat{\mathbb{C}} \ : \ |R(z)| \leq 1\}.$$

□

Remark 2.17. Desirable property for the stability domain. The stability domain of the initial value problem (2.7) is, see Remark 2.10,

$$S_{\text{anal}} = \mathbb{C}_0^- := \{z \in \mathbb{C} \ : \ \text{Re}(z) \leq 0\},$$

since $R(z) = e^z$. In this domain, the solution decreases (for $\text{Re}(z) < 0$) or its absolute value is constant (for $\text{Re}(z) = 0$). A desirable property of a numerical method is that it should be stable for all parameters where the initial value problem is stable, i.e., $\mathbb{C}_0^- \subseteq S$. □

Definition 2.18. A-stable method. If for the stability domain S of a one-step method, it holds that $\mathbb{C}_0^- \subseteq S$, then this one-step method is called A-stable. □

Lemma 2.19. Property of an A-stable method. *Consider an A-stable one-step method, then it is* $|R(\infty)| \leq 1$.

Proof. By the assumption $\mathbb{C}_0^- \subseteq S$, the absolute value of the stability function is bounded from above by 1 for all $|z| \to \infty$ with $\text{Re}(z) \leq 0$. From Theorem 2.14, it follows that the stability function has to be a rational function where the polynomial degree of the numerator is not larger than the polynomial degree of the denominator, since otherwise the function is unbounded for $|z| \to \infty$. It is known from function theory that such rational functions are continuous in ∞. Hence, it is $|R(\infty)| \leq 1$. ■

Remark 2.20. On A-stable methods. The behavior of the stability function for $|z| \to \infty$, $z \in \mathbb{C}_0^-$, is of utmost interest, since it describes the length of the steps that is admissible for given λ such that the method is still stable. However, from the property $|R(\infty)| \leq 1$, it does not follow that the step length can be chosen arbitrarily large without loosing the stability of the method. □

Definition 2.21. Strongly A-stable method, L-stable method. An A-stable one-step method is called strongly A-stable, if it satisfies in addition $|R(\infty)| < 1$. It is called L-stable (left stable), if even it holds that $|R(\infty)| = 0$. □

Example 2.22. Stability of some one-step methods. The types of stability defined in Definitions 2.18 and 2.21 are of utmost importance for the quality of a numerical method.

1. *Explicit Euler method.* It is $R(z) = 1 + z$, i.e., the stability domain is the closed circle with radius 1 and center $(-1, 0)$, see Figure 2.2. This method is not A-stable. For $|\lambda|$ large, one has to use very small steps in order to get stable simulations.
   The smallness of the step lengths for stable simulations of stiff problems is the basic problem of all explicit methods.

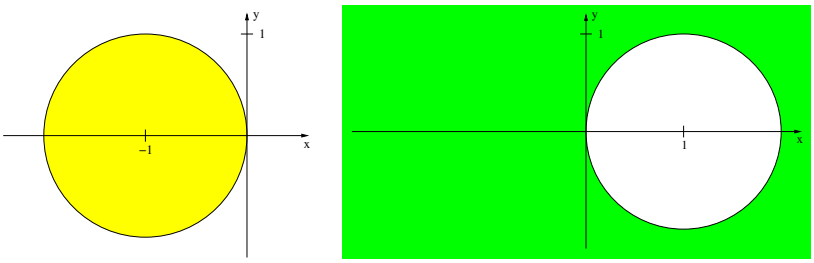

Fig. 2.2 Stability domain of the explicit Euler method (left) and the implicit Euler method (right).

2. *Implicit Euler method.* One has for this method $R(z) = 1/(1-z)$. The stability domain is the complete complex plane without the open circle with radius 1 and center $(1, 0)$, see Figure 2.2. Hence, the method is A-stable. From Example 2.11, it is already known that $|R(\infty)| = 0$ such that the implicit Euler method is even L-stable. A smallness condition on the step lengths does not arise for this method, at least for the model problem (2.7).
   In general, one can apply with the implicit Euler method much larger steps than, e.g., with the explicit Euler method. Step size restrictions arise, e.g., from the physics of the problem and from the required accuracy of the simulations. However, one has to solve in general in each node a nonlinear equation, like for each implicit scheme. Thus, the numerical costs and the computing time per step are usually much larger than for explicit schemes.
3. *Trapezoidal rule.* For the trapezoidal rule, one gets with $z = a + ib$, $a, b \in \mathbb{R}$,
$$|R(z)|^2 = \left|\frac{1+z/2}{1-z/2}\right|^2 = \left|\frac{1+a/2+ib/2}{1-a/2-ib/2}\right|^2 = \frac{(2+a)^2+b^2}{(2-a)^2+b^2}.$$
   Thus, $|R(z)| \leq 1$ if $|2+a| \leq |2-a|$, compare Figure 2.3, i.e.,
$$R(z) \begin{cases} < 1 \text{ for } a < 0 \iff \operatorname{Re}(z) < 0, \\ = 1 \text{ for } a = 0 \iff \operatorname{Re}(z) = 0, \\ = 1 \text{ for } z = \infty, \end{cases}$$
   see also Example 2.11. Hence, one obtains $S = \mathbb{C}_0^-$. This method is A-stable but not L-stable. However, in contrast to the implicit Euler method, which is a first order method, the trapezoidal rule is a second order method.

Summary: Already for the very simple model problem (2.7) it turns out that explicit methods need in more complicated situations, i.e., if $|z|$ is large, very (extremly) small steps for performing stable simulations. One cannot hope that this situation improves for more complex (stiff) problems. For such problems, usually implicit methods, which allow to use a reasonable

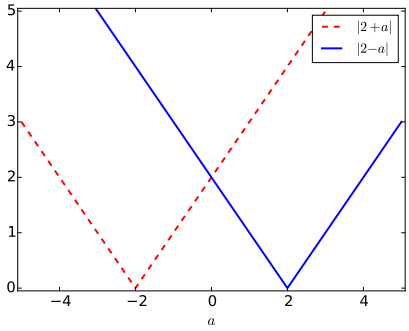

Fig. 2.3 Illustration to the trapezoidal rule, Example 2.22.

step length, are often the much more efficient approach, despite the necessity to solve a nonlinear problem in each step. □

Remark 2.23. Linear systems of ordinary differential equations. The goal of the reminder of this section consists in introducing at least one definition of stiffness. To this end, consider an initial value problem with a linear system of ordinary differential equations with constant coefficients

$$\boldsymbol{y}'(x) = A\boldsymbol{y}(x), \quad \boldsymbol{y}(0) = \boldsymbol{y}_0, \ A \in \mathbb{R}^{n\times n}, \ \boldsymbol{y}_0 \in \mathbb{R}^n. \tag{2.10}$$

The solution of (2.10) has the form

$$\boldsymbol{y}(x) = e^{Ax}\boldsymbol{y}_0,$$

where Ax is defined component-wise, as a multiplication of a scalar with a matrix. The first factor on the right-hand side is the matrix exponential. □

Definition 2.24. Matrix exponential. Let $A \in \mathbb{R}^{n\times n}$ and

$$A^0 := I, \ A^1 := A, \ A^2 := AA, \ \ldots, \ A^k := A^{k-1}A.$$

The matrix exponential is defined by

$$e^A := \exp(A) \ : \ \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}, \quad A \mapsto \sum_{k=0}^{\infty} \frac{A^k}{k!}.$$

□

Lemma 2.25. Properties of the matrix exponential. *The matrix exponential has the following properties:*

i) The series

$$\sum_{k=0}^{\infty} \frac{A^k}{k!}$$

converges absolutely for all $A \in \mathbb{R}^{n\times n}$*, like in the real case* $n = 1$*.*

ii) If the matrices $A, B \in \mathbb{R}^{n\times n}$ *are commuting, i.e., if* $AB = BA$ *holds, then it follows that*

$$e^A e^B = e^{A+B}.$$

iii) The matrix $\left(e^A\right)^{-1} \in \mathbb{R}^{n\times n}$ *exists for all* $A \in \mathbb{R}^{n\times n}$ *and it holds*

$$\left(e^A\right)^{-1} = e^{-A}.$$

This property corresponds to $e^x \neq 0$ *for the scalar case.*

iv) It holds $\operatorname{rank}\left(e^A\right) = n$, $\det\left(e^A\right) \neq 0$.

v) The matrix-valued function $\mathbb{R} \to \mathbb{R}^{n\times n}$, $x \mapsto e^{Ax}$*, where* Ax *is defined component-wise, is continuously differentiable with respect to* x *with*

$$\frac{d}{dx} e^{Ax} = Ae^{Ax}.$$

The derivative of the exponential is the first factor in this matrix product. The formula looks the same as in the scalar case.

Proof. i) with comparison test with a majorizing series, using that the corresponding series with real argument converges for all real numbers, see literature,
ii) follows from i), exercise,
iii) follows from ii), exercise,
iv) follows from iii),
v) direct calculation with difference quotient, exercise.

■

Example 2.26. Matrix exponential. There are only few classes of matrices that allow an easy computation of the matrix exponential: diagonal matrices and nilpotent matrices.

1. Consider

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{pmatrix} \quad \Longrightarrow \quad A^k = \begin{pmatrix} 1^k & 0 & 0 \\ 0 & 2^k & 0 \\ 0 & 0 & 3^k \end{pmatrix}.$$

It follows that

$$e^{Ax} = \sum_{k=0}^{\infty} \frac{(Ax)^k}{k!} = \sum_{k=0}^{\infty} \frac{1}{k!} \begin{pmatrix} x^k & 0 & 0 \\ 0 & (2x)^k & 0 \\ 0 & 0 & (3x)^k \end{pmatrix}$$

$$= \begin{pmatrix} \sum_{k=0}^{\infty} \frac{x^k}{k!} & 0 & 0 \\ 0 & \sum_{k=0}^{\infty} \frac{(2x)^k}{k!} & 0 \\ 0 & 0 & \sum_{k=0}^{\infty} \frac{(3x)^k}{k!} \end{pmatrix} = \begin{pmatrix} e^x & 0 & 0 \\ 0 & e^{2x} & 0 \\ 0 & 0 & e^{3x} \end{pmatrix}.$$

2. This example illustrates property ii) of Lemma 2.25. For the matrices

$$A = \begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix},$$

it is possible to calculate the corresponding series easily, since B is a nilpotent matrix ($B^2 = 0$). One obtains

$$e^A = \begin{pmatrix} e^2 & 0 \\ 0 & e^3 \end{pmatrix}, \quad e^B = I + B = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

It holds $AB \neq BA$ and

$$e^A e^B = \begin{pmatrix} e^2 & e^2 \\ 0 & e^3 \end{pmatrix} \neq \begin{pmatrix} e^2 & e^3 \\ 0 & e^3 \end{pmatrix} = e^B e^A.$$

Assume that $e^A e^B = e^{A+B}$. Since $e^{A+B} = e^{B+A}$, it follows that then also $e^B e^A = e^{B+A} = e^{A+B} = e^A e^B$, which is a contradiction to the calculations from above.

□

Remark 2.27. Extension of the stability theory to linear systems. Consider system (2.10). Let the n eigenvalues of A be $\lambda_1, \ldots, \lambda_n \in \mathbb{C}$.

It will be assumed that this matrix can be diagonalized, i.e., there exists a matrix $Q \in \mathbb{R}^{n \times n}$ such that

$$\Lambda = Q^{-1} A Q, \quad \text{with } \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n).$$

This property is given, e.g., if all eigenvalues are mutually different. The columns $\boldsymbol{q}_i$ of Q are the eigenvectors of A. With the substitution

$$\boldsymbol{y}(x) = Q\boldsymbol{z}(x) \quad \Longrightarrow \quad \boldsymbol{y}'(x) = Q\boldsymbol{z}'(x),$$

one obtains the differential equation

$$Q\boldsymbol{z}'(x) = AQ\boldsymbol{z}(x) \quad \Longleftrightarrow \quad \boldsymbol{z}'(x) = Q^{-1}AQ\boldsymbol{z}(x) = \Lambda\boldsymbol{z}(x).$$

The equations of this system are decoupled. Its general solution is given by

$$\boldsymbol{z}(x) = e^{\Lambda x}\boldsymbol{c} = \left(c_i e^{\lambda_i x}\right)_{i=1,\ldots,n}.$$

It follows that the general solution of (2.10) has the form

$$\boldsymbol{y}(x) = Q\boldsymbol{z}(x) = \sum_{i=1}^{n} c_i e^{\lambda_i x} \boldsymbol{q}_i.$$

Inserting this expression in the initial condition yields

$$\boldsymbol{y}(0) = \sum_{i=1}^{n} c_i \boldsymbol{q}_i = Q\boldsymbol{c} = \boldsymbol{y}_0 \quad \Longrightarrow \quad \boldsymbol{c} = Q^{-1}\boldsymbol{y}_0.$$

Hence, one obtains the following solution of the initial value problem

$$\boldsymbol{y}(x) = \sum_{i=1}^{n} \left(Q^{-1}\boldsymbol{y}_0\right)_i e^{\lambda_i x} \boldsymbol{q}_i, \tag{2.11}$$

where $\left(Q^{-1}\boldsymbol{y}_0\right)_i$ is the i-th component of $Q^{-1}\boldsymbol{y}_0$. Now, one can easily see that the solution is stable (small changes of the initial data lead to small changes of the solution) only if all eigenvalues have a negative real part.

The study of numerical methods makes sense only in the case that the problem is well posed, i.e., all eigenvalues have a negative real part. Then, the most important term in (2.11) with respect to stability is the term with the eigenvalue of A with the largest absolute value of its real part, since for the stability, the absolute values of the product of the real parts of the eigenvalues and the step length are important. □

Definition 2.28. Stiff system of ordinary differential equations. The linear system of ordinary differential equations

$$\boldsymbol{y}'(x) = A\boldsymbol{y}(x), \quad A \in \mathbb{R}^{n\times n},$$

is called stiff, if all eigenvalues λ_i of A possess a negative real part and if

$$q := \frac{\max\{|\mathrm{Re}(\lambda_i)|\,,\ i = 1,\ldots,n\}}{\min\{|\mathrm{Re}(\lambda_i)|\,,\ i = 1,\ldots,n\}} \gg 1.$$

Sometimes, the system is called weakly stiff if $q \approx 10$ and stiff if $q > 10$. □

Remark 2.29. On Definition 2.28. Definition 2.28 has a disadvantage. The ratio becomes large also in the case that the eigenvalue with the smallest absolute value of the real part is close to zero. However, this eigenvalue is not important for the stability of numerical methods, only the eigenvalue with the largest absolute value of the real part. □

Remark 2.30. Local stiffness for general ordinary differential equations. The concept of stiffness can be extended in some sense from linear differential equations to general differential equations. The differential equation

$$\boldsymbol{y}'(x) = \boldsymbol{f}(x, \boldsymbol{y}(x))$$

can be transformed, by introducing the functions

$$\overline{y}(x) := x \quad \text{and} \quad \tilde{\boldsymbol{y}}(x) := \begin{pmatrix} \boldsymbol{y}(x) \\ \overline{y}(x) \end{pmatrix},$$

to the autonomous form

$$\tilde{\boldsymbol{y}}'(x) = \tilde{\boldsymbol{f}}\left(\tilde{\boldsymbol{y}}(x)\right) = \begin{pmatrix} \boldsymbol{f}(x, \boldsymbol{y}(x)) \\ 1 \end{pmatrix}.$$

By linearizing at the initial value $\tilde{\boldsymbol{y}}_0$, one obtains a differential equation of the form $\tilde{\boldsymbol{y}}'(x) = A\tilde{\boldsymbol{y}}(x)$. Applying some definition of stiffness to the linearized equation, it is possible to define a local stiffness for the general equation.

However, if one considers nonlinear problems, one has to be careful in the interpretation of the results. In general, the results are valid only locally, i.e., in a neighborhood of the point of linearization, and they do not describe the stability of a numerical method in the whole domain of definition of the nonlinear problem. □

2.4 Rosenbrock Methods

Remark 2.31. Goal. From the stability theory, it became obvious that one has to use implicit methods for stiff problems. However, implicit methods are computationally expensive, one has to solve in general nonlinear problems in each step. The goal consists in constructing implicit methods that have on the one hand a reduced computational complexity but on the other hand, they should be still accurate and stable. □

Remark 2.32. Linearly implicit Runge–Kutta methods. Consider, without loss of generality, the autonomous initial value problem in $\mathbb{R}^n$

$$\boldsymbol{y}'(x) = \boldsymbol{f}(\boldsymbol{y}), \quad \boldsymbol{y}(0) = \boldsymbol{y}_0,$$

compare Remark 1.30. DIRK methods, see Remark 2.9, have a Butcher tableau of the form

$$\begin{array}{c|ccccc} c_1 & a_{11} & 0 & 0 & \cdots & 0 \\ c_2 & a_{21} & a_{22} & 0 & \cdots & 0 \\ c_3 & a_{31} & a_{32} & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ c_s & a_{s1} & a_{s2} & \cdots & & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_{s-1} & b_s \end{array}.$$

One has to solve s decoupled nonlinear equations

$$\boldsymbol{K}_j = \boldsymbol{f}\left(\boldsymbol{y}_k + h\sum_{l=1}^{j-1} a_{jl}\boldsymbol{K}_l + ha_{jj}\boldsymbol{K}_j\right), \quad j = 1, \ldots, s. \tag{2.12}$$

This fixed-point equation can be solved with a fixed-point iteration. As a special fixed-point iteration, the quasi Newton method for solving the j-th equation leads to an iterative scheme of the form

$$\begin{aligned}\boldsymbol{K}_j^{(m+1)} &= \boldsymbol{K}_j^{(m)} \\ &\quad - \left(I - a_{jj}hJ\right)^{-1} \underbrace{\left[\boldsymbol{K}_j^{(m)} - \boldsymbol{f}\left(\boldsymbol{y}_k + h\sum_{l=1}^{j-1} a_{jl}\boldsymbol{K}_l + ha_{jj}\boldsymbol{K}_j^{(m)}\right)\right]}_{\text{residual}},\end{aligned} \tag{2.13}$$

$m = 0, 1, \ldots$. The derivative with respect to $\boldsymbol{K}_j$ of the corresponding nonlinear problem to (2.12) with right-hand side $\mathbf{0}$ is

$$I - \underbrace{a_{jj}h}_{\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{K}_j}}\partial_{\boldsymbol{y}}\boldsymbol{f}\left(\boldsymbol{y}_k + h\sum_{l=1}^{j-1} a_{jl}\boldsymbol{K}_l + ha_{jj}\boldsymbol{K}_j\right) \in \mathbb{R}^{n\times n}.$$

In (2.13), one uses usually the approximation of the derivative $J = \partial_{\boldsymbol{y}}\boldsymbol{f}(\boldsymbol{y}_k)$ instead of the derivative at the current iterate, hence it is a quasi Newton method. If the step length h is sufficiently small, then the matrix $\left(I - a_{jj}hJ\right)$ is non-singular, since then it is sufficiently close to the identity, and the linear systems of equations possess a unique solution.

Often, it turns out to be sufficient for reaching the required accuracy to perform just one step of the iteration. This statement holds in particular if the step length is sufficiently small and if a sufficiently accurate start value $\boldsymbol{K}_j^{(0)}$ is available. One utilizes the ansatz (linear combination of the already computed increments)

$$\boldsymbol{K}_j^{(0)} := \sum_{l=1}^{j-1} \frac{d_{jl}}{a_{jj}}\boldsymbol{K}_l,$$

where the coefficients d_{jl}, $l = 1, \ldots, j-1$, still need to be determined. Applying just one step in (2.13) with this ansatz, one obtains an implicit method with linear systems of equations of the form

$$\begin{aligned}\left(I - a_{jj}hJ\right)\boldsymbol{K}_j &= \boldsymbol{f}\left(\boldsymbol{y}_k + h\sum_{l=1}^{j-1}\left(a_{jl} + d_{jl}\right)\boldsymbol{K}_l\right) - hJ\sum_{l=1}^{j-1} d_{jl}\boldsymbol{K}_l, \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad j = 1, \ldots, s, \\ \boldsymbol{y}_{k+1} &= \boldsymbol{y}_k + h\sum_{j=1}^{s} b_j\boldsymbol{K}_j.\end{aligned} \tag{2.14}$$

This class of methods is called linearly implicit Runge–Kutta methods.

Linearly implicit Runge–Kutta methods are still implicit methods. One has to solve in each step only s linear systems of equations. That means, these methods are considerably less computationally complex than the original implicit methods and the first goal stated in Remark 2.31 is achieved. Now, one has to study which properties of the original methods are transferred to linearly implicit methods. In particular, stability is of importance. If stability will be lost, then linearly implicit methods are not suited for solving stiff differential equations. □

Theorem 2.33. Stability of linearly implicit Runge–Kutta methods. *Consider a Runge–Kutta method with the parameters $(A, \boldsymbol{b}, \boldsymbol{c})$, where $A \in \mathbb{R}^{s\times s}$ is a non-singular lower triangular matrix (which was used for the derivation of (2.14)). Then, the corresponding linearly implicit Runge–Kutta method (2.14) with $J = \partial_{\boldsymbol{y}} \boldsymbol{f}(\boldsymbol{y}_k)$ has the same stability function $R(z)$ as the original method, independently of the choice of $\{d_{jl}\}$.*

Proof. The linearly implicit method will be applied to the one-dimensional (to simplify notations) test problem

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

with $\mathrm{Re}(\lambda) < 0$. Since $f(y) = \lambda y$, one obtains $J = \lambda$. The j-th equation of (2.14) has the form

$$\begin{aligned}\left(1 - a_{jj}h\lambda\right) K_j &= \lambda \left(y_k + h \sum_{l=1}^{j-1} \left(a_{jl} + d_{jl}\right) K_l \right) - h\lambda \sum_{l=1}^{j-1} d_{jl} K_l \\ &= \lambda y_k + h\lambda \sum_{l=1}^{j-1} a_{jl} K_l, \quad j = 1, \ldots, s.\end{aligned}$$

Multiplication with h gives with $z = \lambda h$

$$K_j h - z \sum_{l=1}^{j} a_{jl} K_l h = z y_k, \quad j = 1, \ldots, s.$$

This equation is equivalent, using matrix-vector notation, to

$$(I - zA)\, \boldsymbol{K} h = z y_k \mathbf{1}, \quad \boldsymbol{K} = (K_1, \ldots, K_s)^T.$$

Let h be chosen in such a way that z^{-1} is not an eigenvalue of A. Then, one obtains by inserting this equation in the second equation of (2.14)

$$y_{k+1} = y_k + h\boldsymbol{b}^T \boldsymbol{K} = y_k + h\boldsymbol{b}^T (I - zA)^{-1} \mathbf{1} \frac{z}{h} y_k = \left(1 + z\boldsymbol{b}^T (I - zA)^{-1} \mathbf{1}\right) y_k = R(z) y_k.$$

Now one can see that in the parentheses there is the stability function $R(z)$ of the original Runge–Kutta method, see (2.8). ■

Remark 2.34. On the stability and consistency. Since the most important stability properties of a numerical method for solving initial value problems with ordinary differential equations depend only on the stability function, these

properties transfer from the original implicit Runge–Kutta method to the corresponding linearly implicit method if $J = \partial_{\boldsymbol{y}} \boldsymbol{f}(\boldsymbol{y}_k)$.

The choice of the coefficients $\{d_{jl}\}$ will influence the order of the linearly implicit method. For an inappropriate choice of these coefficients, the order of the linearly implicit method might be lower than the order of the original method. □

Example 2.35. Linearly implicit Euler method. The implicit Euler method has the Butcher tableau

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}.$$

With (2.14), it follows that the linearly implicit Euler method has the form

$$\left(I - h\partial_{\boldsymbol{y}} \boldsymbol{f}(\boldsymbol{y}_k)\right) \boldsymbol{K}_1 = \boldsymbol{f}\left(\boldsymbol{y}_k\right), \quad \boldsymbol{y}_{k+1} = \boldsymbol{y}_k + h\boldsymbol{K}_1.$$

The linearly implicit Euler method is L-stable, like the implicit Euler method, and one has to solve in each step only one linear system of equations. There are no coefficients $\{d_{jl}\}$ to be chosen in this method. □

Remark 2.36. Rosenbrock[5] *methods.* Another possibility for simplifying the use of linearly implicit methods and decreasing the numerical costs consists in using for all increments the same coefficient $a_{jj} = a$. In this case, all linear systems of equations in (2.14) possess the same system matrix $(I - ahJ)$. Then, one needs only one LU decomposition of this matrix and can solve all systems in (2.14) with this decomposition. This approach is called Rosenbrock methods or Rosenbrock–Wanner[6] methods (ROW methods)

$$\begin{aligned} (I - ahJ)\, \boldsymbol{K}_j &= \boldsymbol{f}\left(\boldsymbol{y}_k + h\sum_{l=1}^{j-1} \left(a_{jl} + d_{jl}\right) \boldsymbol{K}_l\right) - hJ\sum_{l=1}^{j-1} d_{jl}\boldsymbol{K}_l, \;\; j = 1, \ldots, s, \\ \boldsymbol{y}_{k+1} &= \boldsymbol{y}_k + h\sum_{j=1}^{s} b_j \boldsymbol{K}_j. \end{aligned} \tag{2.15}$$

In practice, it is often even possible to use the same approximation J of the Jacobian for some subsequent steps. This is true in particular, if the solution changes only slowly. In this way, one can save additional computational costs. □

Example 2.37. The method `ode23s`. In MATLAB, one can find for solving stiff initial value problems with ordinary differential equations the Rosenbrock method `ode23s`, see Shampine & Reichelt (1997). This method has the form

[5] Howard H. Rosenbrock (1920 – 2010)

[6] Gerhard Wanner, born 1942

$$
\begin{aligned}
(I - ahJ)\,\boldsymbol{K}_1 &= \boldsymbol{f}(\boldsymbol{y}_k), \quad a = \frac{1}{2+\sqrt{2}} \approx 0.2928932,\\
(I - ahJ)\,\boldsymbol{K}_2 &= \boldsymbol{f}\left(\boldsymbol{y}_k + \frac{1}{2}h\boldsymbol{K}_1\right) - ahJ\boldsymbol{K}_1,\\
\boldsymbol{y}_{k+1} &= \boldsymbol{y}_k + h\boldsymbol{K}_2.
\end{aligned}
\tag{2.16}
$$

From the equation for the second increment, it follows that $d_{21} = a$. Then, one obtains with (2.15) $a_{21} = 1/2 - d_{21} = 1/2 - a$. Using the condition that the nodes are the sums of the rows of the matrix, it follows that the corresponding Butcher tableau looks like

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

□

Theorem 2.38. Consistency order of `ode23s`. *The Rosenbrock method* `ode23s` *is of second order consistent if* $h \in (0, 1/(2a\,\|J\|_2))$.

Proof. Let $h \in (0, 1/(2a\,\|J\|_2))$, where $\|\cdot\|_2$ denotes the spectral norm of J, which is induced by the Euclidean vector norm $\|\cdot\|_2$. It can be shown, see class Computer Mathematics, that the matrix $(I - ahJ)$ is invertible if $\|ahJ\|_2 < 1$. This condition is satisfied for the choice of h from above.

Let $\boldsymbol{K}$ be the solution of

$$
(I - ahJ)\boldsymbol{K} = \boldsymbol{f}. \tag{2.17}
$$

Then, one obtains with the triangle inequality, with the compatibility of the Euclidean vector norm and the spectral matrix norm, and with the choice of h that

$$
\begin{aligned}
\|(I - ahJ)\boldsymbol{K}\|_2 &\geq \|\boldsymbol{K}\|_2 - ah\,\|J\boldsymbol{K}\|_2 \geq \|\boldsymbol{K}\|_2 - ah\,\|J\|_2\,\|\boldsymbol{K}\|_2\\
&\geq \|\boldsymbol{K}\|_2 - \frac{a\,\|J\|_2}{2a\,\|J\|_2}\,\|\boldsymbol{K}\|_2 = \frac{1}{2}\,\|\boldsymbol{K}\|_2\,.
\end{aligned}
$$

It follows with (2.17) that

$$
\frac{1}{2}\,\|\boldsymbol{K}\|_2 \leq \|(I - ahJ)\boldsymbol{K}\|_2 = \|\boldsymbol{f}\|_2 \quad \Longrightarrow \quad \|\boldsymbol{K}\|_2 \leq 2\,\|\boldsymbol{f}\|_2\,. \tag{2.18}
$$

Thus, the solution of the linear system of equations is bounded by the right-hand side, independently of h. This result will be applied to (2.16). For $\boldsymbol{K}_1$, the right-hand side does not depend on h. Also the right-hand side of $\boldsymbol{K}_2$ does not depend on negative powers of h, e.g., using the step length restriction, one obtains also for $\boldsymbol{K}_2$ a bound that is independent of negative powers of h

$$
\begin{aligned}
\left\|\boldsymbol{f}\left(\boldsymbol{y}_k + \frac{1}{2}h\boldsymbol{K}_1\right) - ahJ\boldsymbol{K}_1\right\|_2 &\leq \left\|\boldsymbol{f}\left(\boldsymbol{y}_k + \frac{1}{2}h\boldsymbol{K}_1\right)\right\|_2 + ah\,\|J\|_2\,\|\boldsymbol{K}_1\|_2\\
&\leq \left\|\boldsymbol{f}\left(\boldsymbol{y}_k + \frac{1}{2}h\boldsymbol{K}_1\right)\right\|_2 + \frac{1}{2}\,\|\boldsymbol{K}_1\|_2\,,
\end{aligned}
$$

and the first term on the right-hand side can be further estimated as in (2.20) below.

One obtains for the first increment of `ode23s` by recursive insertion, using (2.16),

$$\begin{aligned}\boldsymbol{K}_1 &= \boldsymbol{f}(\boldsymbol{y}_k) + ahJ\boldsymbol{K}_1 = \boldsymbol{f}(\boldsymbol{y}_k) + ahJ\left(\boldsymbol{f}(\boldsymbol{y}_k) + ahJ\boldsymbol{K}_1\right)\\ &= \boldsymbol{f}(\boldsymbol{y}_k) + ahJ\boldsymbol{f}(\boldsymbol{y}_k) + h^2a^2J^2\boldsymbol{K}_1\\ &= \boldsymbol{f}(\boldsymbol{y}_k) + ahJ\boldsymbol{f}(\boldsymbol{y}_k) + \mathcal{O}\left(h^2\right). \end{aligned} \tag{2.19}$$

The last step is allowed since $\boldsymbol{K}_1$ is bounded by the data of the problem (the right-hand side $\boldsymbol{f}(\boldsymbol{y}_k)$) independently of h, see (2.18) where the constant in the estimate is 2. Using a Taylor series expansion and considering only first order terms explicitly, one obtains in a similar way for the second increment of `ode23s`

$$\begin{aligned}\boldsymbol{K}_2 &= \boldsymbol{f}\left(\boldsymbol{y}_k + \frac{1}{2}h\boldsymbol{K}_1\right) - ahJ\boldsymbol{K}_1 + ahJ\boldsymbol{K}_2\\ &= \boldsymbol{f}\left(\boldsymbol{y}_k\right) + \frac{1}{2}h\partial_{\boldsymbol{y}}\boldsymbol{f}\left(\boldsymbol{y}_k\right)\boldsymbol{K}_1 - ahJ\boldsymbol{K}_1 + ahJ\boldsymbol{K}_2 + \mathcal{O}\left(h^2\right) \qquad (2.20)\\ &\overset{(2.19)}{=} \boldsymbol{f}\left(\boldsymbol{y}_k\right) + \frac{1}{2}h\partial_{\boldsymbol{y}}\boldsymbol{f}\left(\boldsymbol{y}_k\right)\boldsymbol{f}\left(\boldsymbol{y}_k\right) - ahJ\boldsymbol{f}(\boldsymbol{y}_k) + ahJ\boldsymbol{K}_2 + \mathcal{O}\left(h^2\right)\\ &\overset{(2.20)}{=} \boldsymbol{f}\left(\boldsymbol{y}_k\right) + \frac{1}{2}h\partial_{\boldsymbol{y}}\boldsymbol{f}\left(\boldsymbol{y}_k\right)\boldsymbol{f}\left(\boldsymbol{y}_k\right) - ahJ\boldsymbol{f}(\boldsymbol{y}_k) + ahJ\boldsymbol{f}(\boldsymbol{y}_k) + \mathcal{O}\left(h^2\right)\\ &= \boldsymbol{f}\left(\boldsymbol{y}_k\right) + \frac{1}{2}h\partial_{\boldsymbol{y}}\boldsymbol{f}\left(\boldsymbol{y}_k\right)\boldsymbol{f}\left(\boldsymbol{y}_k\right) + \mathcal{O}\left(h^2\right). \end{aligned}$$

Inserting these results in (2.16) gives for one step of `ode23s`

$$\boldsymbol{y}_{k+1} = \boldsymbol{y}_k + h\boldsymbol{f}\left(\boldsymbol{y}_k\right) + \frac{1}{2}h^2\partial_{\boldsymbol{y}}\boldsymbol{f}\left(\boldsymbol{y}_k\right)\boldsymbol{f}\left(\boldsymbol{y}_k\right) + \mathcal{O}\left(h^3\right). \tag{2.21}$$

The Taylor series expansion of the solution $\boldsymbol{y}(x)$ of the system of differential equations in x_k has the form, using the differential equation and the chain rule,

$$\begin{aligned}\boldsymbol{y}(x_{k+1}) &= \boldsymbol{y}(x_k) + h\boldsymbol{y}'(x_k) + \frac{h^2}{2}\boldsymbol{y}''(x_k) + \mathcal{O}\left(h^3\right)\\ &= \boldsymbol{y}(x_k) + h\boldsymbol{f}\left(\boldsymbol{y}_k\right) + \frac{h^2}{2}\frac{\partial\boldsymbol{f}\left(\boldsymbol{y}\right)}{\partial x}(x_k) + \mathcal{O}\left(h^3\right)\\ &= \boldsymbol{y}(x_k) + h\boldsymbol{f}\left(\boldsymbol{y}_k\right) + \frac{h^2}{2}\partial_{\boldsymbol{y}}\boldsymbol{f}\left(\boldsymbol{y}_k\right)\boldsymbol{y}'(x_k) + \mathcal{O}\left(h^3\right)\\ &= \boldsymbol{y}(x_k) + h\boldsymbol{f}\left(\boldsymbol{y}_k\right) + \frac{h^2}{2}\partial_{\boldsymbol{y}}\boldsymbol{f}\left(\boldsymbol{y}_k\right)\boldsymbol{f}\left(\boldsymbol{y}_k\right) + \mathcal{O}\left(h^3\right). \end{aligned}$$

Starting with the exact value at x_k, then the first three terms of (2.21) correspond to the Taylor series expansion of the solution $\boldsymbol{y}(x)$ of the system of differential equations in x_k. Thus, it follows that the local error is of order $\mathcal{O}\left(h^3\right)$, from what follows that the consistency order of `ode23s` is two, see Definition 1.14. ■

Remark 2.39. To the proof of Theorem 2.38. Note that it is not needed in the proof of Theorem 2.38 that J is the exact derivative $\partial_{\boldsymbol{y}}\boldsymbol{f}\left(\boldsymbol{y}_k\right)$. The method `ode23s` remains a second order method if J is only an approximation of $\partial_{\boldsymbol{y}}\boldsymbol{f}\left(\boldsymbol{y}_k\right)$ and even if J is an arbitrary matrix. However, the transfer of the stability properties from the original method to `ode23s` is only guaranteed for the choice $J = \partial_{\boldsymbol{y}}\boldsymbol{f}\left(\boldsymbol{y}_k\right)$, see Theorem 2.33. □

Theorem 2.40. Stability function of `ode23s`. *Assume that $J = \partial_{\boldsymbol{y}}\boldsymbol{f}(\boldsymbol{y}_k)$, then the stability function of the Rosenbrock method* `ode23s` *has the form*

$$R(z) = \frac{1 + (1 - 2a)z}{(1 - az)^2}. \tag{2.22}$$

Proof. The statement of the theorem follows from applying the method to the usual test equation, *exercise*. ■

Corollary 2.41. Stability of `ode23s`. *If $J = \partial_{\boldsymbol{y}} \boldsymbol{f}(\boldsymbol{y}_k)$, then the Rosenbrock method* `ode23s` *is L-stable.*

Proof. The statement is obtained by applying the definition of L-stability to the stability function (2.22). ■

Remark 2.42. On the order of `ode23s`. It remains the question whether an appropriate choice of J might even increase the order of the method. However, for the model problem of the linear stability analysis, a series expansion of the stability function shows that the exponential function is reproduced exactly only up to the quadratic term. From this observation, it follows that one does not obtain a third order method even with exact Jacobian. In practice, there is no important reason from the point of view of accuracy to compute a new Jacobian in each step. Often, it is sufficient to update J every now and then. □