# 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. □

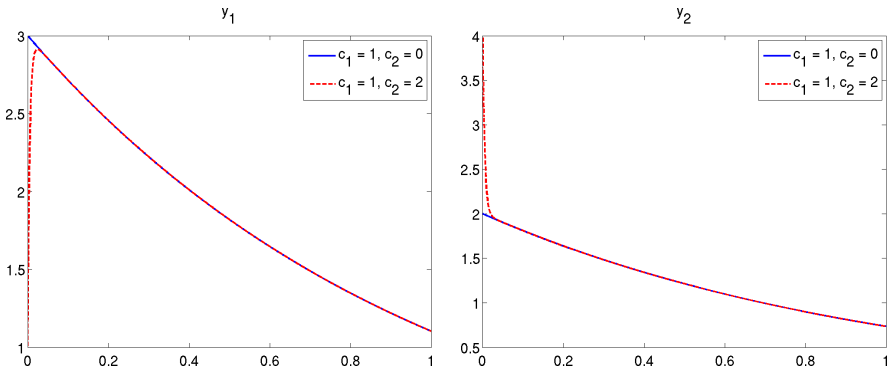


**Fig. 2.1** Solutions of Example 2.2, left: first component, right: second component.

*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 $\phi$, 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 $\infty$ 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 $\zeta_i$ 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 $\infty$. 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 $\lambda$ 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.