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

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

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. □

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

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

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,

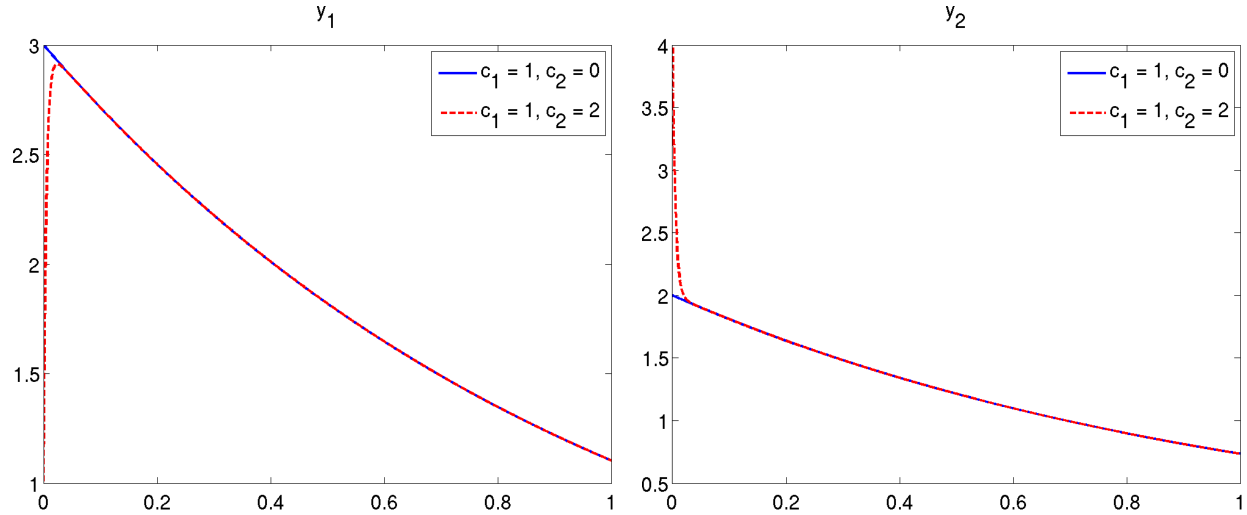

Fig. 2.1 Solutions of Example 2.2, left: first component, right: second component.

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

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. □