Appendix B

Basic Time Discretisation Methods

A standard numerical approach for solving partial differential equations is the method of lines. There the problem is first discretised with respect to the space variable(s), leading to a system of ordinary differential equations. These ODEs are then solved using time discretisation methods. Basic space discretisations have already been introduced in Appendix A. Here we collect some facts about time discretisations. Our basic reference is the textbook *Solving ordinary differential equations I: Nonstiff problems*, by E. Hairer, S.P. Nørsett, and G. Wanner.[1]

Throughout the lectures t will denote the time variable. Let $f : [t_0, t_{\max}] \times \mathbb{R}^m \to \mathbb{R}^m$ be a continuous function, $y_0 \in \mathbb{R}^m$. We consider the following initial value problem on the time interval $[t_0, t_{\max}]$:

$$\begin{cases} y'(t) = f\big(t, y(t)\big) \\ y(t_0) = y_0, \end{cases} \tag{B.1}$$

where $y : [t_0, t_{\max}] \to \mathbb{R}^m$ is the unknown function to be determined.

In order to solve problem (B.1) numerically, we divide the time interval $[t_0, t_{\max}]$ into N pieces:

$$t_0 < t_1 < t_2 < \cdots < t_N = t_{\max}$$

with **variable time steps** $h_n = t_{n+1} - t_n$, $n = 0, ..., N-1$. Such a sub-divison of the time interval is often called a **time grid**. In the case of an equidistant grid with $h_0 = h_1 = \cdots = h_{N-1} =: h$, the lenght $h = \frac{t_{\max} - t_0}{N}$ is usually referred to as **constant time step**. The numerical scheme consists in approximating the exact solution $y(t_n)$ at time levels t_n for $n = 0, ..., N$. This approximation is called a **numerical solution** to problem (B.1) and is denoted by y_n. We note that for an equidistant time grid, the time levels are computed as $t_n = t_0 + nh$ (especially, $t_n = nh$ with $t_0 = 0$).

B.1 Euler's method

The simplest time discretisation method was introduced by L. Euler[2] in 1768. His idea was to replace the derivative y' by an approximation to the tangent, i.e., quotient by the difference quotient. Since we have

$$f(t, y) = y' = \lim_{h \to 0} \frac{y(t+h) - y(t)}{h}$$

by the initial value problem (B.1), we obtain **Euler's method** as

$$f(t_n, y_n) = \frac{y_{n+1} - y_n}{t_{n+1} - t_n},$$

[1] E. Hairer, S.P. Nørsett, G. Wanner, Solving ordinary differential equations I: Nonstiff problems. Springer Verlag, 2008.

[2] L. Euler, Institutionum Calculi Integralis. Volumen Primum, Opera Omnia **XI**, 1768.

usually written as

$$y_{n+1} = y_n + h_n f(t_n, y_n). \tag{B.2}$$

In order to demonstrate a drawback of Euler's method for problems we have in mind, and to motivate a modification, let us consider the problem (1.1)(1.2) appearing in Lecture 1:

$$\begin{cases} y'(t) = -k^2 y(t), \\ y(t_0) = y_0, \end{cases} \tag{B.3}$$

for $n \in \mathbb{N}$ with $t_0 = 0$ and $y_0 = 1$. Note that the exact solution $y(t) = \mathrm{e}^{-k^2 t}$ is positive for all t and bounded by 1. After the first step with equidistant step size h, application of Euler's method (B.2) yields

$$y_1 = y_0 + h(-k^2 y_0) = (1 - hk^2)y_0 = 1 - hk^2.$$

Note that the choice $h > \frac{1}{k^2}$ leads to $y_1 < 0$. The second step yields

$$y_2 = y_1 + h(-k^2 y_1) = (1 - hk^2) + h(-k^2)(1 - hk^2) = (1 - hk^2)^2 > 0.$$

It is easy to see that the numerical solution has the following form after n steps:

$$y_n = (1 - hk^2)^n.$$

For $h > \frac{1}{k^2}$ the value of y_n changes sign at each step, i.e., (y_n) is an alternating sequence for $n \in \mathbb{N}$. Moreover, in contrast to the boundedness of the exact solution $y(t)$, the sequence $\big(|y_n|\big)$ is unbounded for this particular choice of h. These phenomena suggest us (i) to choose the time step carefully, or (ii) to modify Euler's method.

Note that we have the freedom to choose the time level appearing in the argument of the function f. Taking t_{n+1} instead of t_n (i.e., we use the tangent at t_{n+1}), we obtain the **implicit Euler method**:

$$y_{n+1} = y_n + h_n f(t_{n+1}, y_{n+1}). \tag{B.4}$$

For the implicit Euler method, the numerical solution of problem (B.3) after one step reads

$$y_1 = y_0 + h(-k^2 y_1) \quad \Longrightarrow \quad (1 + hk^2)y_1 = y_0 \quad \Longrightarrow \quad y_1 = \frac{1}{1 + hk^2} \cdot y_0 = \frac{1}{1 + hk^2}.$$

The second step yields

$$y_2 = y_1 + h(-k^2 y_2) \quad \Longrightarrow \quad (1 + hk^2)y_2 = y_1 \quad \Longrightarrow \quad y_2 = \frac{1}{1 + hk^2} \cdot y_1 = \frac{1}{(1 + hk^2)^2}.$$

Hence, the numerical solution after n steps,

$$y_n = \frac{1}{(1 + hk^2)^n}$$

results in positive values for all choices of the step size h. Moreoverm it is bounded by 1.

The method is called implicit, because we have to solve a nonlinear system of equations to compute y_{n+1}. The question arises whether one can determine y_{n+1} from (B.4). It is, however, garanteed by the implicit function theorem. We note that explicit and implicit Euler methods are often called forward and backward Euler methods, respectively.

B.2 Runge's method

In order to derive the method introduced by C. Runge[3] in 1905, let us consider the problem (B.1) again, and integrate it between t_n and $t_{n+1} = t_n + h_n$:

$$\int_{t_n}^{t_n+h_n} y'(t)\,\mathrm{d}t = \int_{t_n}^{t_n+h_n} f\big(t, y(t)\big)\,\mathrm{d}t$$

$$y(t_n + h_n) = y(t_n) + \int_{t_n}^{t_n+h_n} f\big(t, y(t)\big)\,\mathrm{d}t. \tag{B.5}$$

The question is now how to approximate the above integral. The first two possibilities

$$\int_{t_n}^{t_n+h_n} f(t, y(t))\,\mathrm{d}t = h_n f(t_n, y_n) + \mathcal{O}(h_n^2)$$

or

$$= h_n f(t_{n+1}, y_{n+1}) + \mathcal{O}(h_n^2)$$

yield the explicit and implicit Euler methods, respectively. Instead of using only the left or right grid point, the function f can also be evaluated at the midpoint $t_n + \frac{h_n}{2}$:

$$\int_{t_n}^{t_n+h_n} f\big(t, y(t)\big)\,\mathrm{d}t = h_n f\big(t_n + \tfrac{h_n}{2}, y(t_n + \tfrac{h_n}{2})\big) + \mathcal{O}(h_n^3),$$

which is already of order 2. To approximate the unknown term $y(t_n + \frac{h_n}{2})$, we use Taylor's formula:

$$y\left(t_n + \tfrac{h_n}{2}\right) = y(t_n) + \frac{h_n}{2} f(t_n, y_n) + \mathcal{O}(h_n^2). \tag{B.6}$$

We note that this corresponds to an explicit Euler step. **Runge's method** for computing y_{n+1} is then defined by

$$y_{n+1} = y_n + h_n f\big(t_n + \tfrac{h_n}{2}, y_n + \tfrac{h_n}{2} f(t_n, y_n)\big). \tag{B.7}$$

B.3 Runge–Kutta methods

Following the idea of the derivation of Runge's method, one can generalise it by applying certain quadrature rules, being common in numerical integration, to approximate the integral appearing in formula (B.5). As already seen, application of left or right Riemann sums lead to explicit or implict Euler method, respectively. The midpoint rule combined with (B.6) yields Runge's method. For the trapezoidal rule we obtain the **Crank–Nicolson method**

$$y_{n+1} = y_n + \tfrac{h_n}{2}\big(f(t_n, y_n) + f(t_{n+1}, y_{n+1})\big) \tag{B.8}$$

being of great importance in practice.

[3] C. Runge, "Über die numerische Auflösung totaler Differentialgleichungen", Göttinger Nachr. (1905), 252–257.

In case of general *quadrature rules*, for $s \in \mathbb{N}$ we define $c_1, ..., c_s$ distinct real numbers between 0 and 1. Then we define the corresponding collocation polynomial u of degree s whose derivative coincides with the function f at the collocation points $t_n + c_i h_n$ for $i = 1, ..., s$, that is,

$$u'(t_n + c_j h_n) = f\big(t_n + c_j h_n, u(t_n + c_j h_n)\big), \quad \text{for } j = 1, ..., s \tag{B.9}$$

with $$u(t_n) = y_n.$$

The numerical solution y_{n+1} of problem (B.1) at time level $t_{n+1} = t_n + h_n$ is then defined as

$$y_{n+1} := u(t_n + h_n). \tag{B.10}$$

In order to compute $u(t_n + h_n)$, let us denote $k_j := u'(t_n + c_j h_n)$ for $j = 1, ..., s$. Note that the Lagrange interpolation polynomials

$$l_i(\tau) = \prod_{\substack{m=1 \\ i \neq m}}^{s} \frac{\tau - c_m}{c_i - c_m}$$

satisfy $l_i(c_j) = \delta_{ij}$, where δ_{ij} denotes the Kronecker delta. Therefore, $\sum_{i=1}^{s} k_i l_i(c_j) = k_j$, which further equals to $u'(t_n + c_j h_n)$. By the Lagrange interpolation form, we obtain

$$u'(t_n + \tau h_n) = \sum_{i=1}^{s} k_i l_i(\tau). \tag{B.11}$$

Due to relation (B.10), the numerical solution y_{n+1} can be computed by integrating between 0 and 1 both sides of equation (B.11) above:

$$\int_0^1 u'(t_n + \tau h_n)\, \mathrm{d}\tau = \int_0^1 \sum_{i=1}^{s} k_i l_i(\tau)\, \mathrm{d}\tau$$

$$\tfrac{1}{h_n}\big(u(t_n + \tau h_n)\big)\big|_{\tau=0}^{\tau=1} = \sum_{i=1}^{s} k_i \underbrace{\int_0^1 l_i(\tau)\, \mathrm{d}\tau}_{=:b_i}$$

$$\tfrac{1}{h_n}\big(u(t_n + h_n) - u(t_n)\big) = \sum_{i=1}^{s} k_i b_i.$$

Then we obtain

$$u(t_n + h_n) = u(t_n) + h_n \sum_{i=1}^{s} b_i k_i.$$

Since $u(t_n) = y_n$ by construction, the numerical solution defined by (B.10) has the form

$$y_{n+1} = y_n + h_n \sum_{i=1}^{s} b_i k_i. \tag{B.12}$$

In order to determine the values of $k_i = f\big(u(t_n+c_ih_n, u(t_n+c_ih_n)\big)$, $i = 1, ..., s$, we have to compute $u(t_n + c_ih_n)$. To this end, let us integrate (B.11) again, but this time between 0 and c_i:

$$\begin{aligned}
\int_0^{c_i} u'(t_n + \tau h_n)\,\mathrm{d}\tau &= \int_0^{c_i} \sum_{j=1}^{s} k_j l_j(\tau)\,\mathrm{d}\tau \\
\tfrac{1}{h_n}\big(u(t_n + \tau h_n)\big)\big|_{\tau=0}^{\tau=c_i} &= \sum_{j=1}^{s} k_j \underbrace{\int_0^{c_i} l_j(\tau)\,\mathrm{d}\tau}_{=:a_{ij}} \\
\tfrac{1}{h_n}\big(u(t_n + c_ih_n) - u(t_n)\big) &= \sum_{i=j}^{s} k_j a_{ij}.
\end{aligned}$$

Hence, we obtain

$$u(t_n + c_ih_n) = u(t_n) + h_n \sum_{j=1}^{s} a_{ij}k_j, \quad \text{for } i = 1, ..., s.$$

This means, application of quadrature rules to approximate the integral in formula (B.5) leads to **collocation methods** defined by

$$y_{n+1} = y_n + h_n \sum_{i=1}^{s} b_i k_i \tag{B.13}$$

with
$$k_i = f\big(t_n + c_ih_n, y_n + h_n \sum_{j=1}^{s} a_{ij}k_j\big), \quad \text{for } i = 1, ..., s \tag{B.14}$$

where
$$b_i = \int_0^1 l_i(\tau)\,\mathrm{d}\tau \quad \text{and} \quad a_{ij} = \int_0^{c_i} l_j(\tau)\,\mathrm{d}\tau, \quad \text{for } i, j = 1, ..., s$$

and
$$l_i(\tau) = \prod_{\substack{i=1 \\ i\neq m}}^{s} \frac{\tau - c_m}{c_i - c_m} \quad \text{are the Lagrange interpolation polynomials.}$$

We note here that for special choice of coefficients c_i, $i = 1, ..., s$, the order of collocation methods can even equal $2s$.

One obtain a possible generalisation to collocation methods by choosing the coefficients a_{ij}, b_i, c_i arbitrary, instead of assigning them the special values above. For a fixed $s \in \mathbb{N}$ and some coefficients a_{ij}, b_i, c_i for $i, j = 1, ..., s$, time discretisation methods of the form (B.13), (B.14) are called **s-stage Runge–Kutta methods**. We note that if $a_{ij} = 0$ for $i \leq j$ we have explicit, otherwise we have implicit Runge–Kutta methods. Notice that all collocation methods are implicit Runge–Kutta methods.

It is common to collect the coefficients a_{ij}, b_i, c_i of a Runge–Kutta method in the Butcher tableau proposed by J. C. Butcher[4] in 1964:

[4] J. C. Butcher, "On Runge–Kutta processes of high order", J. Austral. Math. Soc. **IV** (1964), 179–194.

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

Example B.1. Examples for s-stage Runge–Kutta methods.

1. Explicit Euler method ($s = 1$):

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

2. Implicit Euler method ($s = 1$):

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

3. Runge–Kutta method based on Gaussian quadrature with one node ($s = 1$):

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

4. Runge's method ($s = 2$):

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

5. Crank–Nicolson method ($s = 2$):

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

6. Runge–Kutta method based on Radau II A quadrature with two nodes ($s = 2$):

$$\begin{array}{c|cc} 1/3 & 5/12 & -1/12 \\ 1 & 3/4 & 1/4 \\ \hline & 3/4 & 1/4 \end{array}$$

B.4 Exercises

1. Prove first- and second-order consistency of the explicit Euler method and the explicit method of Runge, respectively.

2. Construct Euler's number e with the help of Euler's method.
Hint: Consider the differential equation $y' = y$ *with the initial condition* $y(0) = 1$.

3. Derive the Runge–Kutta methods based on Gaussian and Radau II A rules. *Hint: For Gauß, the nodes are* $c_1 = \frac{1}{2}$ *for* $s = 1$ *and* $c_{1,2} = \frac{1}{2} \pm \frac{\sqrt{3}}{6}$ *for* $s = 2$. *For Radau II A, the nodes are* $c_1 = 1$ *for* $s = 1$ *and* $c_1 = \frac{1}{3}$, $c_2 = 1$ *for* $s = 2$.

4. Analyse the Runge–Kutta methods based on Gaussian and Radau II A rules for the problem $y' = -k^2 y$. Are there any conditions for the step size?

5. Derive the Butcher Tableau for the Crank–Nicolson scheme (B.8).