# Numerical Analysis II.

Áron Erdélyi

2021.09.22.

# Contents

# 1 Quadrature rules for numerical integration

## 1.1 Simple quadrature rules

Polynomials are a good for approximating smooth functions, and can easily be integrated exactly. A simple quadrature rule is based on the approximation

$$\int_a^b f(x)dx \approx \int_a^b p(x)dx,$$

where $x_1, \dots, x_n \in [a, b]$ are distinct points and $p$ is the Lagrange interpolation polynomial of degree $n-1$ of $f$. That is, $p$ is of degree $n - a$ and

$$p(x_k) = f(x_k), \quad k = 1, \dots, n.$$

The Lagrange Interpolation Theorem guarantees the existence and uniqueness of such a polynomial, which can be written as

$$p(x) = \sum_{i=1}^{n} f(x_i)L_i(x),$$

where the Lagrange polynomials with respect to the points $x_1, \dots, x_n$ can be defined by setting

$$L_i(x) = \prod_{\substack{j=1 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \text{ if } n > 1; \quad L_1(x) = 1, \text{ if } n = 1.$$

These polynomials have degree $n-1$, have roots at $x_j$ for $j \neq i$, and take the value 1 at $x_i$. This important property can be written compactly as

$$L_i(x_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}, \quad 1 \leq i, j \leq n.$$

Then we obtain the quadrature weights $w_i$ as the integral of the Lagrange polynomial $L_i$:

$$\int_a^b f(x)dx \approx \int_a^b p(x)dx = \sum_{i=1}^{n} f(x_i) \int_a^b L_i(x)dx = \sum_{i=1}^{n} f(x_i)w_i.$$

**Example 1.1** (Midpoint rule)**.** *A polynomial of degree zero is constant, ans is determined by ts value at one point. If we take this point to be the midpoint of the interval $[a, b]$, we get a midpoint rule. Here the weight is equal to the interval length.*

$$\textit{points:} \quad x_1 = \frac{a+b}{2}$$

$$\textit{weights:} \quad w_1 = b - a.$$

**Example 1.2** (Trapezoidal rule)**.** *If we take the quadrature points to be the endpoints of the interval, we get the trapezoidal rule. The Lagrange polynomials are then $L_1 = \frac{b-x}{b-a}$ and $L_2 = \frac{x-a}{b-a}$, resulting in weights equaling half the interval length.*

$$\textit{points:} \quad x_1 = a, \quad x_2 = b$$

$$\textit{weights:} \quad w_1 = w_2 = \frac{b-a}{2}.$$

**Example 1.3** (Simpson rule)**.** *For Simpsons rule we have three quadrature points, the two end points and the midpoint. The weights are determined the same way as befoe.*

$$\textit{points:} \quad x_1 = a, \quad x_2 = \frac{a+b}{2}, \quad x_3 = b$$

$$\textit{weights:} \quad w_1 = w_3 = \frac{b-a}{6}, \quad w_2 = \frac{2}{3}(b-a).$$

**Theorem 1.1** (Taylor"s Theorem)**.** *Let $f : [a,b] \to \mathbb{R}$ and $n \in \mathbb{N}$ with $n > 0$. Assume that $f^{(n)}$ exists on $(a,b)$ and that $f^{(n-1)}$ is coninuous on $[a,b]$. Let $\alpha$ and $\beta$ be distinct points of $[a,b]$ and define the Taylor polynomial of $f$ of degree $(n-1)$ centered at $\alpha$ by*

$$P_\alpha^{n-1}(x) := \sum_{k=0}^{n-1} \frac{f^{(k)}(\alpha)}{k!}(x-\alpha)^k = f(\alpha) + f'(\alpha)(x-\alpha) + \frac{f''(\alpha)}{2!}(x-\alpha)^2 + \cdots + \frac{f^{(n-1)}(\alpha)}{(n-1)!}(x-\alpha)^{n-1}.$$

*Then, there is a point $\eta$ between $\alpha$ and $\beta$ such that*

$$f(\beta) = P_\alpha^{(n-1)}(\beta) + \frac{f^{(n)}(\eta)}{n!}(\beta-\alpha)^n.$$

Let

$$R_\alpha^{n-1}(x) := f(x) - P_\alpha^{n-1}(x).$$

**Corollary 1.1.** *Let $f : [a,b] \to \mathbb{R}$ and $n \in \mathbb{N}$ with $n > 0$. Assume that $f^{(n)}$ exists on $(a,b)$ and that $f^{(n-1)}$ is continuous on $[a,b]$. Then, for all $x \in [a,b]$,*

$$|R_\alpha^{n-1}(x)| \le \frac{1}{n!}|x-\alpha|^n \sup_{\eta\in(a,b)} |f^{(n)}(\eta)|.$$

**Theorem 1.2** (Error estimate for the midpoint rule)**.** *Let $f \in C^2[a,b]$. Then*

$$\left| I(f) - (b-a) f\left(\frac{a+b}{2}\right)\right| \le \frac{(b-a)^3}{24} \max_{x\in[a,b]} |f''(x)|.$$

*Proof.* We write

$$f(x) = P^1_{\frac{a+b}{2}}(x) + R^1_{\frac{a+b}{2}}(x) = f\left(\frac{a+b}{2}\right) + f'\left(\frac{a+b}{2}\right)\left(x - \frac{a+b}{2}\right) + R^1_{\frac{a+b}{2}}(x).$$

Then, by integrating both sides, we get

$$I(f) = (b-a) f\left(\frac{a+b}{2}\right) + f'\left(\frac{a+b}{2}\right)\int_a^b \left(x - \frac{a+b}{2}\right) dx + \int_a^b R^1_{\frac{a+b}{2}}(x)dx =$$

$$= (b-a) f\left(\frac{a+b}{2}\right) + 0 + \int_a^b R^1_{\frac{a+b}{2}}(x)dx.$$

Therefore

$$\left| I(f) - (b-a) f\left(\frac{a+b}{2}\right)\right| = \left|\int_a^b R^1_{\frac{a+b}{2}}(x)dx\right| \le \int_a^b \left|R^1_{\frac{a+b}{2}}(x)\right| dx \le$$

$$\le \frac{1}{2!} \max_{\eta\in[a,b]} |f^{(2)}(\eta)| \int_a^b \left(x - \frac{a+b}{2}\right)^2 = \frac{(b-a)^3}{24} \max_{\eta\in[a,b]} |f''(\eta)|.$$

□

## 1.2 Composite quadrature rules

Composite quadrature rules can be constructed from simple ones by dividing the interval $[a,b]$ into a number of subintervals, and applying a simple quadrature rule to each subinterval. This way one easily obtains a more accurate quadrature rule by increasing he number of quadrature points. For example, we can divide the interval into $N$ subintervals $[a_k, b_k]$, $k = 1, \ldots, N$, and apply the trapezoidal rule to each to obtain

$$\int_a^b f(x)dx = \sum_{k=1}^N \int_{a_k}^{b_k} f(x)dx \approx \sum_{k=1}^N \left(f(a_k) + f(b_k)\right) \frac{b_k - a_k}{2}.$$

If we now assume that all the subintervals have the same length, $b_k - a_k = h = \frac{b-a}{N}$, and let $x_k = a + (k-1)h$, $k = 1, \ldots, N+1$, then we get

$$\int_a^b f(x)dx \approx \sum_{k=1}^N \left(f(x_k) + f(x_{k+1})\right) \frac{h}{2} = f(x_1)\frac{h}{2} + \sum_{k=2}^N f(x_k)h + f(x_{N+1})\frac{h}{2}.$$

Composite versions of other simple quadrature rules can be derived the same way, with some presented below. For simplicity, the interval is chosen to be $[0,1]$, and $h = N^{-1}$.

**Example 1.4** (Composite midpoint rule)**.** *In this example*

$$\textit{points:} \quad x_i^{mid} = \left(i - \frac{1}{2}\right) h \quad i = 1, \ldots, N,$$

$$\textit{weights:} \quad w_i^{mid} = h, \quad i = 1, \ldots, N.$$

**Example 1.5** (Composite trapezoidal rule)**.** *In this example*

$$\textit{points:} \quad x_i^{trap} = (i-1)\, h \quad i = 1, \ldots, N+1,$$

$$\textit{weights:} \quad w_i^{trap} = \begin{cases} \frac{h}{2} & i = 1,\ i = N+1 \\ h & 1 < i < N+1 \end{cases}.$$

**Example 1.6** (Composite Simpsons rule)**.** *In this example*

$$\textit{points:} \quad x_i^{Sim} = (i-1)\, \frac{h}{2} \quad i = 1, \ldots, 2N+1,$$

$$\textit{weights:} \quad w_i^{Sim} = \begin{cases} \frac{h}{6} & i = 1,\ i = 2N+1 \\ \frac{4h}{6} & i = 2, 4, 6, \ldots, 2N \\ \frac{2h}{6} & i = 3, 5, 7, \ldots, 2N-1 \end{cases}.$$

**Theorem 1.3.** *Let* $[a, b] = [0, 1]$ *and let* $f \in C^2[0, 1]$*. Then*

$$\left| I(f) - \sum_{i=1}^{N} f(x_i^{mid}) w_i^{mid} \right| \leq \frac{h^2}{24} \max_{x \in [0,1]} |f''(x)|$$

*and*

$$\left| I(f) - \sum_{i=1}^{N+1} f(x_i^{trap}) w_i^{trap} \right| \leq \frac{h^2}{12 \max_x \in [0,1]} |f''(x)|.$$

*If* $f \in C^4[0, 1]$*, then*

$$\left| I(f) - \sum_{i=1}^{2N+1} f(x_i^{Sim}) w_i^{Sim} \right| \leq \frac{h^4}{2880} \max_{x \in [0,1]} |f^{(4)}(x)|.$$

*Proof.* We will only prove the statement about the composite midpoint rule, the rest of the statements follow in a similar fashion. We have that

$$\left| I(f) - \sum_{i=1}^{N} f(x_i^{\text{mid}}) w_i^{\text{mid}} \right| = \left| \sum_{i=1}^{N} \int_{x_i}^{x_{i+1}} f(x)dx - \sum_{i=1}^{N} f(x_i^{\text{mid}}) w_i^{\text{mid}} \right| =$$

$$= \left| \sum_{i=1}^{N} \left( \int_{x_i}^{x_{i+1}} f(x)dx - f(x_i^{\text{mid}}) w_i^{\text{mid}} \right) \right|.$$

We can use a previous theorem on $[x_i, x_{i+1}]$ to conclude that

$$\left| \int_{x_i}^{x_{i+1}} f(x)dx - f(x_i^{\text{mid}}) w_i^{\text{mid}} \right| \leq \frac{h^3}{24} \max_{x \in [x_i, x_{i+1}]} f''(x)|.$$

Thus

$$\left| I(f) - \sum_{i=1}^{N} f(x_i^{\text{mid}}) w_i^{\text{mid}} \right| = \left| \sum_{i=1}^{N} \left( \int_{x_i}^{x_{i+1}} f(x)dx - f(x_i^{\text{mid}}) w_i^{\text{mid}} \right) \right| \leq$$

$$\leq \sum_{i=1}^{N} \left| \int_{x_i}^{x_{i+1}} f(x)dx - f(x_i^{\text{mid}}) w_i^{\text{mid}} \right| \leq \sum_{i=1}^{N} \frac{h^3}{24} \max_{x \in [x_i, x_{i+1}]} |f''(x)| \leq \max_{x \in [0,1]} |f''(x)| \sum_{i=1}^{N} \frac{h^3}{24} =$$

$$= Nh \frac{h^3}{24} \max_{x \in [0,1]} |f''(x)| = \frac{h^2}{24} \max_{x \in [0,1]} |f''(x)|.$$

□

## 1.3 Gaussian quadratures

In this section we use orthogonal polynomials to construct quadrature rules of the form

$$\int_a^b f(x)dx \approx \sum_{j=1}^n w_j f(x_j)$$

with nodes $x_j$ and weights $w_j$ carefully chosen.

**Definition 1.1.** *A quadrature of the form above is of order $p \in \mathbb{N}_0$, if forall $f \in C^p[a,b]$ the estimate*

$$\left| \int_a^b f(x)dx - \sum_{j=1}^n w_j f(x_j) \right| \leq c \max_{x\in[a,b]} |f^{(p)}(x)|$$

*holds for some $c > 0$ independent of $f$.*

**Lemma 1.1.** *A quadrature of the form above is of order $p \geq 1$ iff it is exact for all $f \in \mathbb{P}_{p-1}$.*

*Proof.* If a quadrature of the form above is of order $p$, then it is exact for all $f \in \mathbb{P}_{p-1}$ as $f^{(p)} = 0$ for all $f \in \mathbb{P}_{p-1}$. Conversely, suppose that the approximation is exact for all $f \in \mathbb{P}_{p-1}$ and let $f \in C^p[a,b]$. Then, using Taylor's Theorem, we may write

$$f(x) = P_\alpha^{p-1}(x) + R_\alpha^{p-1}(x),$$

with $P_\alpha^{p-1} \in \mathbb{P}_{p-1}$ and $|R_\alpha^{p-1}(x)| \leq m \max_{x\in[a,b]} |f^{(p)(x)}|$. Therefore

$$\left| \int_a^b f(x)dx - \sum_{j=1}^n w_j f(x_j) \right| \leq \left| P_\alpha^{p-1}(x)dx - \sum_{j=1}^n P_\alpha^{p-1}(x_j) \right| + \left| \int_a^b R_\alpha^{p-1}(x)dx - \sum_{j=1}^n w_j R_\alpha^{p-1}(x_j) \right| =$$

$$= \left| \int_a^b R_\alpha^{p-1}(x)dx - \sum_{j=1}^n w_j R_\alpha^{p-1}(x_j) \right| \leq \int_a^b |R_\alpha^{p-1}(x)|dx + \sum_{j=1}^n |w_j||R_\alpha^{p-1}(x_j)| \leq$$

$$\leq m \max_{x\in[a,b]} |f^{(p)}(x)|(b-a) + m \max_{x\in[a,b]} |f^{(p)}(x)| \sum_{j=1}^n |w_j|.$$

Thus, the quadrature rule is of order $p$. □

**Lemma 1.2.** *Given a distinct set of nodes $x_1, \ldots, x_n$ it is possible to find a unique set of weights $w_1, \ldots, w_n$ such that the quadrature of the form above is of order $p \geq n$.*

**Definition 1.2.** *We say that $p_m \in \mathbb{P}_m$ is an $m$th orthogonal polynomial if $p_m \neq 0$ and $p_m \perp \hat{p}$ for all $\hat{p} \in \mathbb{P}_{m-1}$.*

**Definition 1.3.** *We say that a real polynomial $p$ is monic if the coefficient of the leading term of $p$ equals 1.*

**Lemma 1.3.** *For every $m \geq 0$, there exists a unique monic $m$th order orthogonal polynomial $p_m$. Moreover, any $p \in \mathbb{P}_m$ can be written as a unique linear combination of $p_0, \ldots, p_m$.*

*Proof.* We prove the statement by induction. Let $p_0 = 1$. Suppose that $p_0, \ldots, p_n$ has already been obtained with the desired property. Let $q(x) := x^{n+1}$ and set

$$p_{n+1}(x) := q(x) - \sum_{k=0}^n \frac{\langle q, p_k \rangle}{\langle p_k, p_k \rangle} p_k(x).$$

One can easily check that $p_{n+1} \in \mathbb{P}_{n+1} \backslash \mathbb{P}_n$ and that $p_{n+1} \perp p_i$ for all $i = 0, \ldots, n$ using the induction hypothesis. Hence, $p_{n+1} \perp p$ for all $p \in \mathbb{P}_n$ as $p = \sum_{i=0}^n c_i p_i$ also by the induction hypothesis. If $p \in \mathbb{P}_{n+1}$,then $p = d_{n+1}p_{n+1} + q$, with $q \in \mathbb{P}_n$ and thus $p = \sum_{i=0}^{n+1} d_i p_i$, using again the induction hypothesis. Note that we must have $d_i = \langle p, p_i \rangle$ and hence the expansion is unique. This finishes the induction.

We finally prove uniqueness of the monic $m$th orthogonal polynomial. Suppose, by the way of contradiction, that $p_m$ and $\tilde{p}_m$ are two monic $m$th orthogonal polynomials such that $p_m \neq \tilde{p}_m$. Then $p_m - \tilde{p}_m \in \mathbb{P}_{m-1}$ and thus

$$\int_a^b |p_n(x) - \tilde{p}_m(x)|^2 dx = \langle p_m - \tilde{p}_m, p_m - \tilde{p}_m \rangle = \langle p_m, pm - \tilde{p}_m \rangle - \langle \tilde{p}_m, p_m - \tilde{p}_m \rangle = 0,$$

whence $p_m = \tilde{p}_m$. □

**Corollary 1.2.** *If $p_m$ and $\tilde{p}_m$ are to $m$th orthogonal polynomials, then there is $c \neq 0$ sch that $p_m = c\tilde{p}_m$. Hence the zeros of $m$th orthogonal polynomials coincide.*

**Lemma 1.4.** *For $m \geq 1$, all zeros of an $m$th orthogonal polynomial $p_m$ are contained in $(a, b)$ and they ae simple.*

**Theorem 1.4.** *Let $x_1, \ldots, x_n$ be the zeros of an $n$th orthogonal polynomial $p_n$ and let $w_1, \ldots, w_n$ be the solution of the Vandermonde system. Then, the corresponding quadrature method, given by the form above s of order $2n$. Furthermore, no quadrature method of the form can exceed this order.*

**Theorem 1.5.** *Let $n \geq 1$ and suppose that $f \in C^{2n}[a, b]$. Let $x_1, \ldots, x_n$ be the zeros of an $n$th orthogonal polynomial $p_n$ and let $w_1, \ldots, w_n$ be the solution of the Vandermonde system. Then, there is a number $\eta \in (a, b)$ such that*

$$\int_a^b f(x) - \sum_{k=1}^{n} w_k f(x_k) = \frac{f^{(2n)}(\eta)}{(2n)!} \int_a^b [\pi_n(x)]^2 dx,$$

*where*

$$\pi_n(x) = \prod_{i=1}^{n} (x - x_i).$$

# 2 Numerical methods for ODEs

## 2.1 One-step methods

We aim to approximate the solution of

$$y'(t) = f(t, y(t)), \quad t > t_0, \quad y(t_0) = y_0, \tag{1}$$

where $f : [t_0, \infty) \times \mathbb{R}^d$, $y_0 \in \mathbb{R}^d$ is a given vector, and $y : [t_0, \infty) \to \mathbb{R}^d$ is a continuously differentiable function which we call the solution of the problem. This is called a Cauchy problem or initial value problem.

We will always assume, that $f$ is continuous and that it is Lipschitz continuous in the second variable, uniformly in the first:

$$||f(t, x) - f(t, y)|| \leq \lambda ||x - y||, \quad x, y \in \mathbb{R}^d, \quad t \in [t_0, \infty),$$

where $\lambda > 0$ and $|| \cdot ||$ denotes any norm in $\mathbb{R}^d$.

### 2.1.1 The Euler method

In order to derive a simple numerical method for the Cauchy problem we let $h > 0$ (time step), integrate (1) from $t_0$ to $t_0 + h$ and approximate the integral to get

$$y(t_0 + h) = y(t_0) + \int_{t_0}^{t_0+h} f(\tau, y(\tau))d\tau \approx y_0 + hf(t_0, y_0),$$

with $y_0 = y(t_0)$. Motivated by this, given a sequence of equidistant time–instances $t_0, t_1 = t_0 + h, t_2 = t_0 + 2h, \ldots$ we define the approximation

$$y_1 = y_0 + hf(t_0, y_0),$$

or more generally,

$$y_{n+1} = y_n + hf(t_n, y_n), \quad n = 0, 1, \ldots \tag{2}$$

This is called the explicit Euler method.

**Convergence of the Euler method.** For simplicity, we take the time–step $h = h_n$ to be constant and consider a time interval $[t_0, t_0 + t^*]$, with $t^* > 0$. o define what we mean by convergence we consider a sequence of approximations

$$y_n := y_{n,h}, \quad n = 0, 1, \ldots, \left\lfloor \frac{t^*}{h} \right\rfloor, h > 0, \tag{3}$$

of $y(t_n)$.

**Definition 2.1.** *A method given by (3) is said to be converent if*

$$\lim_{h \to 0+} \max_{n=0,1,\ldots,\left\lfloor \frac{t*}{h} \right\rfloor} ||y_{n,h} - y(t_n)|| = 0.$$

**Theorem 2.1.** *Suppose that $f : [t_0, \infty) \times \mathbb{R}^d \to \mathbb{R}^d$ is continuously differentiable and that $f$ is Lipschitz continuous in the second variable, uniformly in the first. Then, the Euler method given by (2) is convergent.*

*Proof.* We take $d = 1$, for simplicity. Let $h > 0$ and $e_i := y_i - y(t_i)$, $i = 0, 1, \ldots, \left\lfloor \frac{t^*}{h} \right\rfloor$ (here we supress the $h$-dependence in the notation). Then, using also that $y_i = e_i + y(t_i)$ we get

$$\begin{aligned} e_{i+1} - e_i &= y_{i+1} - y_i - (y(t_{i+1}) - y(t_i)) = hf(t_i, y_i) - (y(t_{i+1}) - y(t_i)) \\ &= hf(t_i, y(t_i)) - (y(t_{i+1}) - y(t_i)) + h(f(t_i, e_i + y(t_i)) - f(t_i, y(t_i))). \end{aligned} \tag{4}$$

Let

$$g_i := hf(t_i, y(t_i)) - (y(t_{i+1}) - y(t_i))$$

$$\psi_i = f(t_i, e_i + y(t_i)) - f(t_i, y(t_i)).$$

The term $g_i$ is called the local approximation error. Then we may rewrite (4) as

$$e_{i+1} - e_i = g_i + h\psi_i, \quad i = 0, 1, \dots, \left\lfloor \frac{t^*}{h} \right\rfloor.$$

We use the Lipschitz condition to obtain

$$|\psi_i| = |f(t_i, e_i + y(t_i)) - f(t_i, y(t_i))| \le \lambda |e_i|.$$

Therefore,

$$|e_{i+1}| \le (1 + \lambda h)|e_i| + |g_i|, \quad i = 0, 1, \dots, \left\lfloor \frac{t^*}{h} \right\rfloor.$$

Then, one can easily show, by induction, that the estimate

$$|e_n| \le (1 + \lambda h)^n |e_0| + \sum_{j=0}^{n-1} (1 + \lambda h)^j |g_{n-1-j}|$$

holds for $n = 0, 1, \dots, \left\lfloor \frac{t^*}{h} \right\rfloor$, where we define $\sum_{j=0}^{-1}(\dots) = 0$. Using estimates

$$(1 + \lambda h)^n \le e^{\lambda n h} \le e^{\lambda t^*}, \quad n = 0, 1, \dots, \left\lfloor \frac{t^*}{h} \right\rfloor,$$

and

$$(1 + \lambda h) \le (1 + \lambda h)^{n-1} \le e^{\lambda (n-1) h} \le e^{\lambda t^*}, \quad n = 0, 1, \dots, \left\lfloor \frac{t^*}{h} \right\rfloor, \quad j = 0, 1, \dots, n-1,$$

we arrive at the inequality

$$|e_n| \le e^{\lambda t^*} \left( |e_0| + \sum_{j=0}^{n-1} |g_{n-1-j}| \right). \tag{5}$$

Next, we estimate $|g_k|$. Note that since $f$ is $[t_0, \infty) \times \mathbb{R}^d$ continuously differentiable it follows that $y \in C^2[t_0, t_0 + t^*]$. Therefore we may use Taylor's theorem to write

$$y(t_{k+1}) - y(t_l) = h y'(t_k) + \frac{h^2}{2} y''(\xi_k) = h f(t_k, y(t_k)) + \frac{h^2}{2} y''(\xi_k),$$

where $\xi_k \in (t_k, t_{k+1})$. Thus,

$$|g_k| \le \frac{h}{2} |y''(\xi_k)| \le \frac{h^2}{2} \max_{s \in [t_0, t_0 + t^*]} |y''(s)| := \frac{h^2}{2} M, \quad k = 0, 1, \dots, n-1.$$

Inserting this into (5) we get

$$|e_n| \le e^{\lambda t^*} \left( |e_0| + \sum_{j=0}^{n-1} \frac{h^2}{2} M \right) = e^{\lambda t^*} \left( |e_0| + n h \frac{h}{2} M \right) \le e^{\lambda t^*} \left( |e_0| + \frac{t^* M}{2} h \right), \quad n = 0, 1, \dots, \left\lfloor \frac{t^*}{h} \right\rfloor.$$

As $e_0 = 0$ for the Euler method, we get

$$\max_{n = 0, 1, \dots, \left\lfloor \frac{t^*}{h} \right\rfloor} |e_n| \le e^{\lambda t^*} \frac{t^* M}{2} h$$

and hence

$$\lim_{h \to 0+} \max_{n = 0, 1, \dots, \left\lfloor \frac{t^*}{h} \right\rfloor} |e_n| = 0.$$

□

### 2.1.2 Crank–Nicolson method and the theta method

A more general approximation procedure can be derived in a similar fashion as in the case of the Euler method. Let $\theta \in [0,1]$. Again, integrate (1) from $t_0$ to $t_0+h$ and approximate the integral to get

$$y(t_0+h) = y(t_0) + \int_{t_0}^{t_0+h} f(\tau, y(\tau))d\tau \approx y_0 + hf(t_0,y_0)\theta + hf(t_0+h, y(t_0+h))(1-\theta),$$

with $y_0 = y(t_0)$. Motivated by this, given a sequence of equidistant time–instances $t_0, t_0 = t_0+h, t_2 = t_0+2h, \dots$ we define the approximation

$$y_0 = y_0 + \theta h f(t_0,y_0) + (0-\theta)hf(t_1,y_1),$$

or more generally

$$y_{n+1} = y_n + \theta h f(t_n,y_n) + (1-\theta)hf(t_{n+1},y_{n+1}), \quad n = 0,1,2,\dots \tag{6}$$

The family of methods defined by (6) is called the theta method. We highlight some special cases:

1. When $\theta = 0$, we get
$$y_{n+1} = y_n + hf(t_n,y_n), \quad n = 0,1,2,\dots$$
which is the Euler method.

2. When $\theta = 0$, we get
$$y_{n+1} = y_n + hf(t_{n+1},y_{n+1}), \quad n = 0,1,2,\dots$$
which is called the implicit Euler or backward Euler method.

3. When $\theta = \frac{1}{2}$ we get
$$y_{n+1} = y_n + \frac{1}{2}hf(t_n,y_n) + \frac{1}{2}hf(t_{n+1},y_{n+1}), \quad n = 0,1,2,\dots$$
which is called the Crank–Nicolson method or the trapezoidal rule.

## 2.2 Multistep methods

In the previous section we considered one step methods. One of the key features of these methods is that while marching forward in time one discards earlier approximation values and only uses one approximation value namely the preceding one. In this section we introduce a new class of numerical methods which might use approximation values from several earlier time-steps and thereby allowing for higher order approximations. As in the previous section we aim to approximate the solution of (1).

### 2.2.1 The Adams–Bashford method

Let $h > 0$ and denote $y_n$ the numerical approximation of $y(t_n)$, where $t_n = t_0 + nh$. Let $s \geq 1$ be an integer and suppose that we have already obtained the first $s$ approximations $y_m$ of $y(t_m)$, $m = 0,1,\dots,s-1$. We wish to advance the solution from $t_{n+s-1}$ to $t_{n+s}$, $n = 0,1,\dots$ Therefore, we integrate (1) from $t_{n+s-1}$ to $t_{n+s}$ to get

$$y(t_{n+s}) = y(t_{n+s-1}) + \int_{t_{n+s-1}}^{t_{n+s}} f(\tau, y(\tau))d\tau. \tag{7}$$

To derive an example of an algorithm that uses the bast $s$ approximation values we use Lagrange interpolation. We consider the Lagrange interpolation of the function $t \to f(t,y(t))$ based on the interval $[t_n, t_{n+s-1}] with respect to s$ points $t_m$, $m = n, n+1, \dots, n+s-1$:

$$f(t,y(t)) \approx p(t) = \sum_{m=1}^{s-1} L_m(t) f(t_{n+m}, y(t_{n+m})), \quad t \in [t_n, t_{n+s-1}] \tag{8}$$

where $L_m$ denotes the Lagrange polynomial

$$L_m(t) = \prod_{\substack{l=0 \\ l\neq m}}^{s-1} \frac{t - t_{n+l}}{t_{n+m} - t_{n+l}}.$$

Next, if we assume that $y$ is sufficiently smooth there is a good chance that (8) still provides a good approximation on the interval $[t_{n+s-1}, t_{n+s}]$ which we then insert to (7) to obtain the approximation

$$y(t_{n+s}) \approx y(t_{n+s-1}) + \sum_{m=0}^{s-1} f(t_{n+m}, y(t_{n+m})) \int_{t_{n+s-1}}^{t_{n+s}} L_m(\tau) d\tau.$$

Let

$$b_m = \frac{1}{h} \int_{t_{n+s-1}}^{t_{n+s}} L_m(\tau) d\tau = \frac{1}{h} \int_0^h L_m(t_{n+s-1} + \tau) d\tau, \quad m = 0, 1, \ldots, s-1.$$

We therefore arrive at the method defined by

$$y_{n+s} = y_{n+s-1} + h \sum_{m=0}^{s-1} b_m f(t_{n+m}, y_{n+m}), \quad n = 0, 1, \ldots$$

This scheme is referred to as the $s$–step Adams–Bashford Method.

### 2.2.2 General linear multistep methods

Let $h > 0$, and consider the general form of an $s$–step multistep method of the form

$$\sum_{m=0}^{s} a_m y_{n+m} = h \sum_{m=0}^{s} b_m f(t_{n+m}, y_{n+m}), \quad n = 0, 1, \ldots \tag{9}$$

where $a_m, b_m$ are give constants, independent of $n$ and $h$. We always take $a_s = 1$. When $b_s = 0$, then the method is called explicit, otherwise it is called implicit.

**Definition 2.2.** *We write that $f(x) = \mathcal{O}(g(x))$ as $x \to \alpha$ if there is a $C > 0$ such that $||f(x)|| \leq |g(x)|$ when $x$ is near $\alpha$. If $\alpha = \infty$ then the inequality has to hold for $x$ large.*

*Similar definitions can be stated in case $x \to \alpha+$, $x \to \alpha-$ and $x \to -\infty$.*

We consider the local approximation error of the scheme; that is, how well the solution of (1) satisfies the algorithm:

$$\sum_{m=0}^{s} a_m y(t + mh) - h \sum_{m=0}^{s} b_m f(t + mh, y(t + mh)) = \sum_{n=0}^{s} a_m y(t + mh) - h \sum_{m=0}^{s} b_m y'(t + mh). \tag{10}$$

**Definition 2.3.** *We say that the order of a method given by (9) is $p \geq 1$, if*

$$\psi(t, y) := \sum_{m=0}^{s} a_m y(t + mh) - h \sum_{m=0}^{s} b_m y'(t + mh) = \mathcal{O}(h^{p+1}) \text{ as } h \to 0$$

*for all sufficient smooth function $y$ and there exists at least one function $y$ for which the rate cannot be improved.*

In order to analyse the order of method given by (9) we introduce the so–called first characteristic polynomial

$$\rho(w) = \sum_{m=0}^{s} a_m w^m \tag{11}$$

and second characteristic polynomial

$$\sigma(w) = \sum_{m=0}^{s} b_m w^m. \tag{12}$$

**Theorem 2.2.** *The $s$–step multistep method given by (9) is of order $p \geq 1$ iff there exists $c \neq 0$ such that*

$$\rho(w) - \sigma(w) \ln w = c(w-1)^{p+1} + \mathcal{O}(|w-1|^{p+2}) \text{ as } w \to 1.$$

*Proof.* We suppose that $y$ is an analytic function with a radius of convergence of its Taylor series being larger than $sh$. In (10) we expand $y(t+mh)$ and $y'(t+mh)$ into Taylor series around $t$ and interchange sums (which is allowed as there are finitely many, convergent infinite series involved):

$$\psi(t,y) = \sum_{m=0}^{s} a_m \sum_{k=1}^{\infty} \frac{1}{k!} y^{(k)}(t) m^k h^k - h \sum_{m=0}^{s} b_m \sum_{k=0}^{\infty} \frac{1}{k!} y^{(k+1)}(t) m^k h^k$$

$$= \sum_{m=0}^{s} a_m y(t) + \sum_{k=1}^{\infty} \frac{1}{k!} y^{(k)}(t) h^k \sum_{m=0}^{s} a_m m^k - h \sum_{k=1}^{\infty} \frac{1}{(k-1)!} y^{(k)}(t) h^{k-1} \sum_{m=0}^{s} b_m m^{k-1}$$

$$= \sum_{m=0}^{s} a_m y(t) + \sum_{k=1}^{\infty} \frac{1}{k!} y^{(k)}(t) h^k \left( \sum_{m=0}^{s} a_m m^k - k \sum_{m=0}^{s} b_m m^{k-1} \right).$$

Thus, the method is of order $p$ iff the following conditions hold

$$\begin{aligned} \sum_{m=0}^{s} a_m &= 0; \\ \sum_{m=0}^{s} a_m m^k &= k \sum_{m=0}^{s} b_m m^{k-1}, \quad k = 1, 2, \ldots, p; \\ \sum_{m=0}^{s} a_m m^{p+1} &\neq (p+1) \sum_{m=0}^{s} b_m m^p. \end{aligned} \tag{13}$$

Let now $w = e^z$. Then $w \to 1$ iff $z \to 0$. Then, by a similar calculation as above, using Taylor series, we get

$$\rho(w) - \sigma(w) \ln w = \rho(e^z) - z\sigma(e^z) = \sum_{m=0}^{s} a_m e^{mz} - z \sum_{m=0}^{s} b_m e^{mz}$$

$$= \sum_{m=0}^{s} a_m \left( \sum_{k=0}^{\infty} \frac{1}{k!} m^k z^k \right) - z \sum_{m=0}^{s} b_m \left( \sum_{k=0}^{\infty} \frac{1}{k!} m^k z^k \right)$$

$$= \sum_{m=0}^{s} a_m + \sum_{k=1}^{\infty} \frac{1}{k!} z^k \left( \sum_{m=0}^{s} a_m m^k - k \sum_{m=0}^{s} b_m m^{k-1} \right).$$

Therefore,

$$\rho(e^z) - z\sigma(e^z) = cz^{p+1} + \mathcal{O}(|z|^{p+2}) \text{ as } z \to 0, \tag{14}$$

for some $c \neq 0$ iff the conditions is (13) hold. Finally (14) holds iff

$$\rho(w) - \sigma(w) \ln w = c(\ln w)^{p+1} + \mathcal{O}(|\ln w|^{p+2}) \text{ as } w \to 1,$$

which is equivalent to the equation in the theorem as

$$\ln w = w - 1 + \mathcal{O}(|w-1|^2) \text{ as } w \to 1,$$

as the Taylor series of $w \to \ln w$ around 1 confirms. □

**Definition 2.4.** *A polynomial obeys the root condition if all its zeros are contained in the closed unit disc of the complex plane and its zeros of modulo 1 are simple.*

**Theorem 2.3** (he Dahlquist equivalence theorem)**.** *Suppose that the starting values $y_1, \ldots, y_{s-1}$ of (9) converge to $y_0$ as $h \to 0+$. Then (9) converges iff it is of order $p \geq 1$ and its first characteristic polynomial $\rho$ obeys the root condition.*

**Definition 2.5.** *For a general linear multistep method to converge it has to be at least of order 1. This property is called consistency and it can be characterized using (13) as*

$$\sum_{m=0}^{s} a_m = 0 \text{ and } \sum_{m=0}^{s} a_m m = \sum_{m=0}^{s} b_m.$$

*In terms of the characteristic polynomials, this is equivalent to*

$$\rho(1) = 0 \text{ and } rho'(1) = \sigma(1).$$

**Definition 2.6.** *A liner multistep method is called strongly stable if 1 is the only zero of its first characteristic polynomial $\rho$ of modulus 1. If the method is consistent, then 1 is always a 0 of $\rho$.*

**Theorem 2.4.** *he maximal order of a convergent $s$–step linear multistep method given by (9) is $\left\lfloor \frac{s+2}{2} \right\rfloor$ for implicit methods and $s$ for explicit methods.*

### 2.2.3 Backward differentiation formulae

These are especially useful for stiff problems.

**Definition 2.7.** *An $s$–order $s$–step method is called a backward differentiation formula (BDF) if its second characteristic polynomial $\sigma$ is of the form $\sigma(w) = \beta w^2$, for some $\beta \in \mathbb{R}\backslash\{0\}$.*

**Theorem 2.5.** *For BDF we must have*

$$\beta = \left( \sum_{m=1}^{s} \frac{1}{m} \right)^{-1} \quad \text{and } \rho(w) = \beta \sum_{m=1}^{s} \frac{1}{m} w^{s-m}(w-1)^m.$$

**Theorem 2.6.** *The $s$–step BDF of order $s$ is convergent iff $1 \leq s \leq 6$.*

# 3 Runge–Kutta methods

## 3.1 Introductin to explicit Runge–Kutta methods

A in previous sections we aim to approximate the solution of

$$y'(t) = f(t, y(t)), \quad t > t_0; \quad y(t_0) = y_0, \tag{15}$$

where $f : [t_0, \infty) \times \mathbb{R}^d \to \mathbb{R}^d$, $y_0 \in \mathbb{R}^d$ is a given vector. Similarly to multistep methods we wish to use a quadrature to approximate the integrated version of (15) but this time on $[t_n, t_{n+1}]$:

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(\tau, y(\tau))d\tau = y(t_n) + h\int_0^1 f(t_n + h\tau, y(t_n + h\tau))d\tau$$

$$\approx y(t_n) + h\sum_{j=1}^{\nu} b_j f(t_n + c_j h, y(t_n + c_j h)), \quad n = 0, 1, \dots$$

This suggests the "method"

$$y_{n+1} = y_n + h\sum_{j=1}^{\nu} b_j f(t_n + c_j h, y(t_n + c_j h)), \quad n = 0, 1, \dots$$

However we do not have access to $y(t_n + c_j h)$ and hence we will further approximate them by $\xi_j$ defined as follows. We first set $c_1 = 0$ and let $\xi_1 = y_n$. Then we define

$$\begin{array}{c} \xi_1 = y_n \\ \xi_2 = y_n + ha_{2,1}f(t_n, \xi_1) \\ \xi_3 = y_n + ha_{3,1}f(t_n, \xi_1) + ha_{3,2}f(t_n + c_2 h, \xi_2) \\ \vdots \\ \xi_\nu = y_n + h\sum_{i=1}^{\nu-1} a_{\nu,i}f(t_n + c_i h, \xi_i) \end{array}, \tag{16}$$

and we finally set

$$y_{n+1} = y_n + h\sum_{j=1}^{\nu} b_j f(t_n + c_j h, \xi_j).$$

The matrix $A = (a_{j,i})_{j,i=1}^{\nu}$, where the missing elements are set to 0, is called the Runge–Kutta matrix, while the vectors $b = (b_1, \dots, b_\nu)^T$ and $c = (c_1, \dots, c_\nu)^T$ are called Runge–Kutta weights and Runge–Kutta nodes, respectively. We usually write the coefficients in the form

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

This s called a Butcher tableau of the Runge–Kutta method, where we do not list the 0 elements of $A$.

**Definition 3.1.** *Let*

$$\tilde{y}_{n+1} = y(t_n) + h\sum_{j=1}^{\nu} b_j f(t_n + c_j h, \xi_j),$$

*where $\xi_j$ are defined as in (16) with $y_n$ replaced by $y(t_n)$. The order of the Runge–Kutta method $p \geq 1$ if $y(t_{n+1}) - \tilde{y}_{n+1} = \mathcal{O}(h^{p+1})$ as $h \to 0$ for every sufficiently smooth $f$ and there is $f$ where this rate cannot be improved.*

**Corollary 3.1.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be twice differentiable in a ball $B$ around $a = (a_1, \dots, a_d)$ and let $x = (x_1, \dots, x_d) \in B$. Then, there is $b$ on the line segment joining $a$ and $x$ such that*

$$f(x) = f(a) + \sum_{k=1}^{d} \frac{\partial f}{\partial x_k}(x_k - a_k) + \frac{1}{2}\sum_{j,k=1}^{d} \frac{\partial^2 f}{\partial x_j \partial x_k}(b)(x_j - a_j)(x_k - a_k).$$

## 3.2 Consistency of Runge–Kutta methods

**Definition 3.2.** *A Runge–Kutta method with Butcher Tableau*

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

*is consistent iff* $\sum_{j=1}^{\nu} b_j = 1$.

*Proof.* First note, that the continuity of $f$ implies that $\xi_i \to y(t_n)$ as $h \to 0$. Also note that

$$\frac{y(t_{n+1}) - y(t_n)}{h} \to y'(t_n) = f(t_n, y(t_n)), \text{ as } h \to 0.$$

Therefore

$$\frac{y(t_{n+1}) - \tilde{y}_{n+1}}{h} = \frac{y(t_{n+1} - y(t_n))}{h} - \sum_{j=1}^{\nu} b_j f(t_n + c_j h, \xi_j)$$

$$\to f(t_n, y(t_n)) - \sum_{j=1}^{\nu} b_j f(t_n, y(t_n)) = 0, \text{ as } h \to 0$$

iff $\sum_{j=1}^{n} b_j = 1$. □

**Theorem 3.1.** *If* $y_n = y(t_n)$*, then*

$$c_j = \sum_{i=1}^{j-1} a_{j,i}, \quad 2 \le j \le \nu,$$

*iff*

$$y(t_n + c_j h) - \xi_j = \mathcal{O}(h^2), \quad 2 \le j \le \nu, \text{ as } h \to 0,$$

*for all ODE with* $f : [t_0, \infty) \times \mathbb{R}^d \to \mathbb{R}^d$ *is continuously differentiable and satisfying (13).*

**Corollary 3.2.** *Let* $f : [t_0, \infty) \times \mathbb{R}^d \to \mathbb{R}^d$ *e continuously differentiable and satisfy (13). An explicit Runge–Kutta method with Butcher Tableau*

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array}$$

*is of order one (at least) if*

$$\sum_{j=1}^{\nu} b_j = 1 \text{ and } c_j = \sum_{i=1}^{j-1} a_{j,i}, \quad 2 \le j \le \nu.$$

*Proof.* First nte the smoothness assumption on $f$ implies that $y \in C^2([t_n, t_{n+1}]; \mathbb{R}^d)$. Hence, by Taylor's theorem

$$y(t_{n+1}) - y(t_n) = y'(t_n)h + \mathcal{O}(h^2), \text{ as } h \to 0, \tag{17}$$

and also

$$y'(t_n + c_j h) = y'(t_n) + \mathcal{O}(h), \quad j = 1, \dots, \nu \text{ as } h \to 0. \tag{18}$$

We then have

$$y(t_{n+1}) - \tilde{y}_{n+1} = y(t_{n+1}) - y(t_n) - h\sum_{j=1}^{\nu} b_j f(t_n + c_j h, \xi_j)$$

$$= y(t_{n+1}) - y(t_n) - h\sum_{j=1}^{\nu} b_j (f(t_n + c_j h, \xi_j) - f(t_n + c_j h, y(t_n + c_i h))) - h\sum_{j=1}^{\nu} b_j f(t_n + c_j h, y(t_n + c_i h))$$

$$= y(t_{n+1}) - y(t_n) - h\sum_{j=1}^{\nu} b_j (f(t_n + c_j h, \xi_j) - f(t_n + c_j h, y(t_n + c_i h))) - h\sum_{j=1}^{\nu} b_j y'(t_n + c_j h). \tag{19}$$

Using the Lipschitz condition on $f$, we get

$$||f(t_n + c_j h, \xi_j) - f(t_n + c_j h, y(t_n + c_j h))|| \le \lambda ||\xi_j - y(t_n + c_j h)|| \le Ch^2, \quad j = 1, \dots, \nu, \tag{20}$$

as $h \to 0$. Thus, using (17), (18) and (20) in (19) it follows that

$$y(t_{n+1}) - \tilde{y}_{n+1} = y'(t_n)h + \mathcal{O}(h^2) + \mathcal{O}(h^3) - hy'(t_n)\sum_{j=1}^{\nu} b_j + \mathcal{O}(h^2) = \mathcal{O}(h^2) \text{ as } h \to 0$$

and the proof is complete. □

## 3.3 Implicit Runge–Kutta methods

Suppose that $c_j \in [0,1]$, $j = 1, \dots, \nu$, are distinct. We look for a degree $\nu$ polynomial $u$ with $u(t_n) = y_n$ that satisfies the differential equation at exactly $\nu$ points:

$$u'(t_n + c_j h) = f(t_n + c_j h, u(t_n + c_j h)), \quad j = 1, \dots, \nu. \tag{21}$$

We then set $y_{n+1} = u(t_{n+1})$. We consider the points $x_i = t_n + c_j h \in [t_n, t_{n+1}]$, $i = 1, \dots, \nu$, and the corresponding Lagrange polynomials $L_i(t)$:

$$L_i(t) = \prod_{i \neq j} \frac{t - x_j}{x_i - x_j}.$$

We then have

$$u'(t) = \sum_{j=1}^{\nu} L_l(t) u'(t_n + c_l h) = \sum_{j=1}^{\nu} L_l(t) f(t_n + c_j h, u(t_n + c_j h)).$$

Therefore, integrating from $t_n$ to $t$ and using that $u(t_n) = y_n$, we get

$$\begin{aligned} u(t) &= y_n + \sum_{l=1}^{\nu} f(t_n + c_l h, u(t_n + c_l h)) \int_{t_n}^{t} L_l(s) ds \\ &= y_n + h \sum_{l=1}^{\nu} f(t_n + c_l h, u(t_n + c_l h)) \frac{1}{h} \int_{t_n}^{t} L_l(s) ds. \end{aligned} \tag{22}$$

We set

$$a_{j,l} = \frac{q}{h} \int_{t_n}^{t_n + c_j h} L_l(s) ds, \quad j, l = 1, \dots, \nu. \tag{23}$$

An easy calculation shows, using a change of variables, that $a_{j,l}$ does not append on $h$ and $n$. We define

$$\xi_j = u(t_n + c_j h), \quad j = 1, \dots, \nu.$$

Then (22) gives

$$\xi_j = y_n + h \sum_{l=1}^{\nu} f(t_n + c_l h, u(t_n + c_l h)) a_{j,l} = y_n + h \sum_{l=1}^{\nu} a_{j,l} f(t_n + c_l h, \xi_l), \quad k = 1, \dots, \nu. \tag{24}$$

His is an implicit system of equations for $\xi_j$, $j = 1, \dots, \nu$. We finally set

$$b_l = \frac{1}{h} \int_{t_n}^{t_n + h} L_l(s) ds = \frac{1}{h} \int_{t_n}^{t_{n+1}} L_l(s) ds, \quad l = 1, \dots, \nu. \tag{25}$$

Again, $b_l$ does not depend on $h$ and $n$ as a change of variables shows. By (22), we then have

$$y_{n+1} = u(t_{n+1}) = y_n + h \sum_{l=1}^{\nu} b_l f(t_n + c_l h, u(t_n + c_l h)) = y_n + h \sum_{l=1}^{\nu} b_l f(t_n + c_l h, \xi_l). \tag{26}$$

The collocation method defined by (23)-(26) is a particular instance of an implicit Runge–Kutta method. When $c_i$, $i = 1, \dots, \nu$ are zeros of a $\nu$th orthogonal polynomial on $[0,1]$, then the method is called a Gauss–Legendre implicit Runge–Kutte method.

## 3.4 Gauss–Legrende Runge–Kutta methods

**Theorem 3.2.** *Suppose that the solution of (15) is sufficiently smooth. If $c_j \in [0,1]$, $j = 1, \dots, \nu$, are the zeros of a $\nu$th orthogoal polynomial on $[0,1]$, then the IRK method is given by (23)-(26) is of order $2\nu$.*

# 4 Finite Element Method

## 4.1 The stationary heat equation in 1D

We will use SI units, for example [[$K$]] for temperature in Kelvin, [$m$] for length in meter, [$J$] for energy. he stationary temperature $u(x)$ [$K$] at cross section $x$ [$m$] of a plate filled with width $L$ [$m$],

$$D(-a(x)Du(x)) = f(x), \quad x \in I = (0, L). \tag{27}$$

Here

- $D = \frac{d}{dx}$ $\left[\frac{1}{m}\right]$ is the derivative;
- $f(x)$ $\left[\frac{J}{m^3 s}\right]$ is the heat flux density of the source;
- $u(x)$ [$K$] temperature;
- $a(x)$ $\left[\frac{J}{mKs}\right]$ is the thermal diffusivity;
- $j(x) = -a(x)Du(x)$ $\left[\frac{J}{m^2 s}\right]$ is the heat flux density the $x$–direction (Fourier's law).

We suppose that no quantity depends on the coordinates $y$ and $z$. The same equation also describes heat conduction in a small rod with length $L$. Dimension control: the units equal on both sides.

### 4.1.1 Boundary conditions

At $x = L$ the heat flux in the outward direction is proportional to the temperature difference:

$$j(L) = k_L(u(L) - u_L), \tag{28}$$

where

- $u_L$ [$K$] is the temperature of the environment;
- $u(L)$ [$K$] is the temperature of the plate at the right boundary section;
- $k_L$ $\left[\frac{J}{m^2 Ks}\right]$ is the heat transfer coefficient.

On the other hand the heat flux satisfies Fourier's law:

$$j(L) = -a(L)Du(L).$$

Therefore,

$$-a(L)Du(L) = k_L(u(L) - u_L),$$

and hence

$$aDu + k_L(u - u_l) = 0 \quad \text{for } x = L.$$

Similarly at $x = 0$ the heat flux in the outward direction is proportional to the temperature difference:

$$-j(0) = k_0(u(0) - u_0), \tag{29}$$

as $-j(0)$ is the heat flux in the $-x$–direction. Again by Fourier's law we have $j(0) = -a(0)Du(0)$. Thus,

$$a(0)Du(0) = k_0(u(0) - u_0).$$

In summary we write the boundary conditions in a compact form as:

$$aD_N u + k(u - u_A) = 0 \quad \text{for } x = 0, L. \tag{30}$$

Here $u_A$ is the ambient temperature; that is, $u_A = u_0$ respectively $u_A = u_L$, the coefficient is $k = k_0$ respectively $k = k_L$, and $D_N$ is the directional derivative in the outward direction; that is,

$$D_N = -\frac{d}{dx} \text{ at } x = 0, \quad D_N = \frac{d}{dx} \text{ at } x = L.$$

The coefficient $k$ depends on how well the plate is isolated at the boundary.

**Special case 1:** $k = \infty$, no isolation. We divide by $k$,

$$\frac{1}{k} aD_n u + u - u_A = 0,$$

and let $k \to \infty$ to get $0 + u - u_A = 0$. Thus

$$u = u_A \text{ at } x = 0, L.$$

his boundary condition holds at the boundary which is not isolated, that is at $x = 0$ or $x = L$. In this case the temperature on the non–isolated boundary condition is called Dirichlet boundary condition.

**Special case 2:** $k = 0$, perfect isolation. With $k = 0$ we get

$$aD_N u = 0,$$

that is, there is no heat flow at the isolated boundary.

As $a > 0$ one arrives at

$$D_N u = 0 \text{ at } x = 0, L.$$

This kind of boundary condition is called Neumann boundary condition.

**Boundary value problem.** Find $u = u(x)$ such that

$$\begin{array}{cc} -D(aDu) = f & x \in I = (0, L), \\ aD_N u + k(u - u_A) = g & x = 0, L. \end{array} \tag{31}$$

## 4.2 Weak (variational) formulation

Here we will rewite the boundary value problem (31) in the so called weak form which will lead to a new solution concept called weak or variational solution. First of all this is essential to set up the finite element method. Secondly, in many cases the boundary value problem might not have a classical solution. We multiply the differential equation

$$-D(aDu) = f$$

with a smooth function $v$ and integrate by parts on $I = (0, L)$:

$$\int_0^L fvdx = -\int_0^L D(aDu)vdx = -\left[aDuv\right]_0^L + \int_0^L aDuDvdx$$

$$= a(0)Du(0)v(0) - a(L)Du(L)v(L) + \int_0^L aDuDvdv.$$

Now we use the boundary conditions from (31):

$$a(0)Du(0) = k_0(u(0) - u_0) - g_0,$$

$$-a(L)Du(L) = k_L(u(L) - u_L)g_L.$$

Therefore

$$\int_0^L fvdx = (k_0(u(0) - u_0) - g_0)v(0) + (k_L(u(L) - u_L) - g_L)v(L) + \int_0^L aDuDvdx.$$

We collect the erms that involve the unknown function $u$ on the left hand side and arrive at

$$\int_0^L aDuDvdx + k_0u(0)v(0) + k_Lu(L)v(L) = \int_0^L fvdx + (k_0u_0 + g_0)v(0) + (k_Lu_L + g_L)v(L).$$

This equation must be fulfilled for every choice of a smooth function $v$.

**The weak formulation.** Find a function $u = u(x)$ such hat the equation

$$\int_0^L aDuDvdx + k_0u(0)v(0) + k_Lu(L)v(L) = \int_0^L fvdx + (k_0u_0 + g_0)v(0) + (k_Lu_L + g_L)v(L) \tag{32}$$

holds for all test functions $v$.

## 4.3 The finite element method in 1D

We will compute an approximate solution $y = U(x)$ that is a piecewise linear function. Therefore we consider a mesh in the interval $I = (0, L)$:

$$0 = x_1 < x_2 < \cdots < x_i < \cdots < x_N = L.$$

We always consider $N$ points (also called nodes) $x_o$ and $N-1$ intervals $I_i = (x_i, x_{i+1})$ of length $h_i = x_{i+1} - x_i$.

A continuous piecewise function $y = U(x)$ is completely determined by its nodal values $U_i = U(x_i)$. To represent $U$ we will use basis functions $y = \phi_i(x)$, one for each node $x_i$.

The function $y = \phi_i(x)$ are given the following way: they are continuous, piecewise linear functions such that

$$\phi_i(x_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}.$$

A general, continuous, piecewise linear function $y = U(x)$ can be uniquely written as a linear combination of basis functions:

$$U(x) = \sum_{i=1}^{N} U_i \phi_i(x), \quad \text{with coefficients } U_i = U(x_i).$$

Note that both sides of the above equality are continuous, piecewise linear functions and their nodal values coencide:

$$U(x_j) = \sum_{i=1}^{N} U_i \phi_i(x_j) = U_j.$$

We now have a formula that expresses a general, continuous, piecewise linear function $y = U(x)$ using its nodal values $U_i$. We aim now to calculate the unknown nodal values $U_i$ so that the function $y = U(x)$ is an appropriate solution to the boundary value problem. To do so we will use the weak formulation

$$\int_0^L aDuDvdx + k_0 u(0)v(0) + k_L u(L)v(L) = \int_0^L fvdx + (k_0 u_0 + g_0)v(0) + (k_L u_L + g_L)v(L)$$

We replace the solution $u$ in the weak formulation with the ansatz $U(x) = \sum_{i=1}^{N} U_i \phi_i(x)$ and use test functions $v = \phi_j$. We then get

$$\sum_{i=1}^{N} U_i \int_0^L aD\phi_i D\phi_j dx + k_0 U_1 \phi_j(0) + k_L U_N \phi_j(L)$$

$$= \int_0^L f\phi_j dx + (k_0 u_0 + g_0)\phi_j(0) + (k_L u_L + g_L)\phi_j(L), \quad j = 1, \ldots, N.$$

Note that the basic functions $\phi_j$ are not smooth in the classical sense as they are not differentiable at some of the nodes. However, you may assign any value to $\phi_j'$ at these nodes as there are only finitely many of those and the value of the integral $\int_0^L aD\phi_i D\phi_j dx$ is going to be unaffected. Next, we use the notation

$$a_{ij} = a_{ji} = \int_0^L aD\phi_i D\phi_j dx, \quad b_j = \int_0^L f\phi_j dx,$$

and

$$r_{11} = k_0, \quad r_{NN} = k_L, \quad r_{ij} = 0, \quad s_1 = k_0 u_0 + g_0, \quad s_N = k_L u_L + g_L, \quad s_j = 0,$$

and arrive at

$$\sum_{i=1}^{N} (a_{ij} + r_{ij}) U_i = b_j + s_j, \quad j = 1, \ldots, N;$$

or in matrix form

$$(\mathcal{A} + \mathcal{R})U = b + s.$$

This is a linear system of equations of $N$ equations and $N$ unknowns. The matrix $\mathcal{R}$ and the vector $s$ are related to the boundary conditions.

The matrix $\mathcal{K} := \mathcal{A} + \mathcal{R}$ is called the stiffness matrix. The stiffness matrix is symmetric and tridiagonal, that is $k_{ij} = 0$, except for $j = i-1, i, i+1$. The vector $l := b + s$ is called the load vector. The interval $I_i(x_i, x_{i+1})$ together with its two basis functions $\phi_i, \phi_{i+1}$ is called a finite element.

## 4.4 Stiff ODE systems and linear stability analysis

### 4.4.1 Stiff systems

Consider the simple scalar ODE

$$\begin{cases} y'(t) = \lambda y(t) & t > 0 \\ y(0) = y_0 \end{cases},$$

where $\lambda \in \mathbb{R}$ is fixed. The unique solution is given by $y(t) = e^{\lambda t} y_0$. If $\lambda < 0$, then $\lim_{t\to\infty} y(t) = 0$ exponentially fast. Let $h > 0$. We solve the system with the explicit Euler method

$$y^E_{n+1} = y^E_n + h\lambda y^E_n = (1 + h\lambda) y^E_n, \quad n = 0, 1, \dots$$

yielding

$$y^E_n = (1 + h\lambda)^n y_0, \quad n = 1, 2, \dots$$

and the implicit Euler method

$$y^I_{n+1} = y^I_n + h\lambda y^I_{n+1}, \quad N = 0, 1, \dots$$

$$y^I_n = (1 - h\lambda)^{-n} y_0, \quad n = 1, 2, \dots$$

Since $\lambda < 0$ and $h > 0$ we have $y^I_n \to 0$ as $n \to \infty$ for any $h > 0$. For the explicit Euler method

$$|1 + h\lambda| < 1 \Leftrightarrow 0 < h|\lambda| < 2$$

and thus

$$y^E_n \to 0 \Leftrightarrow h < \frac{2}{|\lambda|}.$$

If $h > \frac{2}{|\lambda|}$, then $|y^E_n| \to +\infty$ in an oscillating fashion. This means that for the implicit Euler method there is no restriction on the stepsize in order for the method to exhibit the same qualitative behavior as the solution. This is in contrast to explicit Euler method, for which the stepsize restriction $h < \frac{2}{|\lambda|}$ is necessary to exhibit the same behaviour.

For $d \geq 2$ consider the linear system of equations

$$\begin{cases} y'(t) = A\lambda y(t) & t > 0 \\ y(0) = y_0 \in \mathbb{R}^d \end{cases},$$

where $A \in \mathbb{R}^{d\times d}$. The unique solution is given by

$$y(t) = e^{tA} y_0 = \left( \sum_{k=0}^{\infty} \frac{(tA)^k}{k!} \right) y_0 = \sum_{k=0}^{\infty} \frac{t^k A^k y_0}{k!},$$

where the first series converges in any induced matrix norm and the second series converges to any vector norm. Suppose that $A$ is diagonalisable. Notice that

$$A^2 = M^{-1} D M M^{-1} D M = M^{-1} D^2 M$$

and iterating this one sees that

$$A^n = M^{-1} D^n M, \quad n = 1, 2, \dots$$

Therefore

$$y(t) = e^{tA} y_0 = \sum_{k=0}^{\infty} \frac{t^k A^k y_0}{k!} = \sum_{k=0}^{\infty} \frac{t^k M^{-1} D^k M y_0}{k!} = \sum_{k=0}^{\infty} \frac{t^k M^{-1} D^k (M y_0)}{k!}$$

$$= M^{-1} \sum_{k=0}^{\infty} \frac{t^k D^k (M y_0)}{k!} = M^{-1} e^{tD} M y_0.$$

**Definition 4.1.** *We call a linear system stiff if all the eigenvalues of $A$ have negative real parts and the ratio of the largest of the real parts of its eigenvalues and smallest of the real parts of its eigenvalues is "large".*

### 4.4.2 Linear stability analysis

In this section we will investigate which methods are suitable for solving stiff systems.

**Definition 4.2.** *The linear stability domain $D$ of the underlying numerical method is the set of all complex numbers $z = h\lambda \in \mathbb{C}$ such that the numerical method with constant stepsize $h$ applied to the scalar equation*

$$\begin{cases} y'(t) = \lambda y(t) & t > 0 \\ y(0) = y_0 \end{cases},$$

*satisfies $y_n \to 0$ as $n \to \infty$, where $y_n$ denotes the approximation of $y(t_n)$.*

**Definition 4.3.** *A numerical method with linear stability domain $D$ is called $A$–stable if*

$$\mathbb{C}_{=}\{z : Re\ z < 0\} \subset D.$$

As an immediate consequence we have that if a method is $A$–stable then $y_n \to 0$ for all $h > 0$ and Re $\lambda < 0$; that is, there is no restriction on the stepsize. For stiff problems, one should generally use $A$–stable methods.

**Theorem 4.1** (Dahlquist's second barrier)**.** *No consistent explicit linear multistep method is $A$–stable. The highest order $A$–stable linear multistep method is of order 2.*

**Definition 4.4.** *A numerical method with linear stability domain $D$ is called $A(\alpha)$–stable if there is an $\alpha(0, \pi]$ such that the finite sector*

$$\sum_{\alpha} = \{z = \rho e^{-i\theta}, \rho > 0, \pi - \alpha < \theta < \pi + \alpha\}$$

*we have $\sum_\alpha \subset D$.*

## 4.5 Initial–boundary value problems and FEM in one spatial variable

### 4.5.1 Heat equation

**Initial–boundary value problem for the heat equation.** FInd $u = u(x, t)$ such that

$$D_t u(x,t) - D_x(a(x)D_x u(x,t)) = f(x,t), \quad x \in I = (0, L), \quad t > 0;$$

$$aD_N u + k(u - u_A(t)) = g(t), \quad x = 0, L;$$

$$u(x, 0) = w(x), \quad x \in I.$$

**The weak formulation of the heat equation.** Find a function $u = u(x, t)$ such that $u(x, 0) = w(x)$ and for all $t > 0$, the equation

$$\int_0^L D_t uv dx + \int_0^L aD_x uD_x v dx + k_0 u(0,t)v(0) + k_L u(L,t)v(L)$$

$$= \int_0^L fv dx + (k_0 u_0(t) + g_0(t))v(0) + (k_L u_L(t) + g_L(t))v(L)$$

holds for all test functions $v$.

**Finite element approximation** The FEM approximation is based on the weak formulation. We replace the solution $u$ in the weak formulation with the ansatz $U(x,t) = \sum_{i=1}^N U_i(t)\phi_i(x)$ and use test functions $v = \phi_j$. This yields

$$\sum_{i=1}^N \dot{U}_i(t) \int_0^L \phi_i \phi_j dx + \sum_{i=1}^N U_i(t) \int_0^L aD_x\phi_i D_j dx + k_0 U_1(t)\phi_j(0) + k_L U_N(t)\phi_j(L)$$

$$= \int_0^L f(x,t)\phi(x)dx + (k_0 u_0(t) + g_0(t))\phi_j(0) + (k_L u_L(t) + g_L(t))\phi_j(L), \quad j = 1, \ldots, N.$$

Using the notation

$$a_{ij} = a_{ji} = \int_0^L aD_x\phi_i D_x\phi_j dx, \quad m_{ij} = m_{ji} = \int_0^L \phi_i\phi_j dx, \quad b_j(t) = \int_0^L f(x,t)\phi_j(x)dx,$$

and

$$r_{11} = k_0,\ r_{NN} = k_L,\ r_{ij} = 0, \quad s_1(t) = k_0u_0(t) + g_0(t),\ s_N(t) = k_Lu_L(t) + g_L(t),\ s_j = 0.$$

We arrive at

$$\sum_{i=1}^N m_{ij}\dot{U}_i(t) + \sum_{i=1}^N (a_{ij} + r_{ij})U_i(t) = b_j(t) + s_j(t), \quad j = 1,\dots,N.$$

In the matrix form this reads as

$$\mathcal{M}\dot{U}(t) + (\mathcal{A} + \mathcal{R})U(t) = b(t) + s(t) = \mathcal{M}\dot{U}(t) + \mathcal{K}U(t) = I(t).$$

The matrix $\mathcal{M}$ is called the mass matrix. This is a linear, first order stiff differential equation system that would be solved by a time–stepping method suited for stiff problems, such as the backward Euler method. One needs to supplement this equation by an initial vector $U(0) = y$. This can be obtained in various ways. One possibility is to look for a continuous piecewise linear function that is the closest to $w$ in some sense. Let $U(0) = \sum_{i=1}^N y_i\phi_i$. We then require that

$$\int_0^L \left(w - \sum_{i=1}^N y_i\phi_i\right)\phi_j dx = 0, \quad j = 1,\dots,N.$$

This leads to

$$\mathcal{M}y = c,$$

where $y = (y_1,\dots,y_N)^T$ and $c = (c_1,\dots,c_N)^T$ with $c_j = \int_0^L w(x)\phi_j(x)dx$.

### 4.5.2 Wave equation

**Initial–boundary value problem for the wave equation.** Find $u = u(x,t)$ such that

$$D_t^2u(x,t) - a^2D_x^2u(x,t) = f(x,t), \quad x \in (0,L), \quad t > 0,$$

$$\tau D_N u + ku = 0, \quad x = 0, L,$$
$$u(x,0) = w(x), \quad x \in [0,L],$$
$$S_t u(x,0) = z(x), \quad x \in [0,L].$$

**The weak formulation of the wave equation.** Find a function $u = u(x,t)$ such that $u(x,0) = w(x)$, $D_tu(x,0) = z(x)$ and, for all $t > 0$, the equation

$$\int_0^L D_t^2uvdx + a^2\int_0^L D_xuD_xvdx + \frac{a^2k_0}{\tau}u(0,t)c(0) + \frac{a^2k_L}{\tau}u(L,t)v(L) = \int_0^L fvdx$$

holds for all test functions $v$.

**Finite element approximation** Like before we use the weak formulation with $U(x,t) = \sum_{i=1}^N U_i(t)\phi_i(x)$ and use test functions $v = \phi_j$. This yields

$$\sum_{i=1}^N \ddot{U}_i(t)\int_0^L \phi_i\phi_j dx + \sum_{i=1}^N U_i(t)a^2\int_0^L D_x\phi_iD_x\phi_j dx + \frac{a^2k_0}{\tau}U_1(t)\phi_j(0) + \frac{a^2k_L}{\tau}U_N(t)\phi_j(L)$$

$$= \int_0^L f(x,t)\phi_j(x)dx, \quad j = 1,\dots,N.$$

Using the notation

$$a_{ij} = a_{ji} = a^2\int_0^L D_x\phi_iD_x\phi_j dx, \quad m_{ij} = m_{ji} = \int_0^L \phi_i\phi_j dx, \quad \int_0^L f(x,t)\phi_j(x)dx,$$

and

$$r_{11} = \frac{a^2 k_0}{\tau},\ r_{NN} = \frac{a^2 k_L}{\tau},\ r_{ij} = 0,$$

we arrive at

$$\sum_{i=1}^{N} m_{ij}\ddot{U}_i(t) + \sum_{i=1}^{N}(a_{ij} + r_{ij})U_i(t) = b_j(t), \quad j = 1, \ldots, N.$$

In the matrix form this yields:

$$\mathcal{M}\ddot{U}(t) + (\mathcal{A} + \mathcal{R})U(t) = b(t),$$

with $\mathcal{K} = \mathcal{A} + \mathcal{R}$,

$$\mathcal{M}\ddot{U}(t) + \mathcal{K}U(t) = b(t).$$

$$\mathcal{M}y^i = c^i, \quad i = 1, 2,$$

where $c_j^1 = \int_0^L w(x)\phi_j(x)dx$ and $c_j^2 = \int_0^L z(x)\phi_j(x)dx$, $j = 1, \ldots, N$.

## 4.6 Error control: embedded Runge-Kutta methods

**The Milne device.** The Milne device is a heuristic way of controlling the timestep in a time–stepping procedure by assessing the local error. Let

$$\kappa = ||y_{n+1} - \overline{x}_{n+1}||$$

and our goal is that

$$||y_{n+1} - y(t_{n+1})|| \leq \delta$$

for some tolerance $\delta$. We suppose that the local errors accumulate at a constant rate and that the sum of the local errors approximately make up the global error.Therefore we require that $\kappa \leq h\delta$ in the stepsize selection. We proceed as follows:

1. With a given stepsize $h$ we calculate $y_{n+1}$, $\overline{x}_{n+1}$ and $\kappa$.
2. If $\kappa > h\delta$, we have $h$ and recalculate $y_{n+1}$, $\overline{x}_{n+1}$ and $\kappa$.
3. If $\kappa << h\delta$ we double $h$ and recalculate $y_{n+1}$, $\overline{x}_{n+1}$ and $\kappa$.
4. If $\kappa < h\delta$, accept $h$ and calculate $y_{n+1}$

Assume that we can choose $\hat{c} \in \mathbb{R}^{\tilde{\nu}-\nu}$ and $\hat{A} \in \mathbb{R}^{\tilde{\nu}-\nu \times \tilde{\nu}}$ such that the Butcher–tableau of the control method is of form

| | | |
|---|---|---|
| $c$ | $A$ | $0$ |
| $\hat{c}$ | $\hat{A}$ | |
| | $\hat{b}^T$ | |

and that $\hat{A}$ is strictly lower diagonal. In this case, we say that the first method is embedded in the second one and together they form a so–called embedded Runge-Kutta pair. We usually write such pair in a single Butcher-tableau as

| | |
|---|---|
| $\tilde{c}$ | $\tilde{A}$ |
| | $b^T$ |
| | $\tilde{b}^T$ |

It turns out that embedded Runge–Kutta pairs exists.