Chapter 1

Explicit One-Step Methods

Remark 1.1 *Contents.* This class presents methods for the numerical solution of explicit systems of initial value problems for ordinary differential equations of first order

$$\mathbf{y}'(x) = \mathbf{f}(x, \mathbf{y}(x)), \quad \mathbf{y}(x_0) = \mathbf{y}_0.$$

For the most part, only initial value problems for scalar ordinary differential equations of first order

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0, \tag{1.1}$$

considered, for simplicity of presentation. The extension of the results and the methods to systems is generally straightforward. □

1.1 Consistency and Convergence

Definition 1.2 Grid, step size. A grid is a decomposition I_h of the interval $I = [x_0, x_e]$

$$I_h = \{x_0, x_1, \ldots, x_N = x_e\}$$

with $x_0 < x_1 < \ldots < x_N$. The differences between neighboring grid points $h_k = x_{k+1} - x_k$ are called step sizes. □

Definition 1.3 Explicit and implicit methods. Let $y(x_k)$ denote the solution of (1.1) in the node x_k and y_k a numerical approximation of $y(x_k)$. A numerical method for the solution of (1.1) on a grid I_h is called explicit, if an approximation y_{k+1} in x_{k+1} can be calculated directly from already computed values y_i, $i < k$. Otherwise, the method is called implicit method. Implicit methods require in each step the solution of a generally nonlinear equation for computing y_{k+1}. □

Definition 1.4 One-step method, incremental function. A one-step method for the computation of an approximation y_{k+1} of the solution of (1.1) on a grid I_h has the form

$$y_{k+1} = y_k + h_k \Phi\left(x, y, h_k\right), \; k = 0, 1, \ldots, \quad y_0 = y(x_0). \tag{1.2}$$

Here, $\Phi(\cdot, \cdot, \cdot)$ is called incremental function of the one-step method. □

Example 1.5 *One-step methods, incremental functions.* The explicit or forward Euler method

$$y_{k+1} = y_k + h_k f\left(x_k, y_k\right), \ k = 0, 1, 2, \ldots, \quad y_0 = y(x_0),$$

is an explicit one-step method with the incremental function

$$\Phi\left(x, y, h_k\right) = f\left(x_k, y_k\right).$$

The computation of y_{k+1} requires only the substitution of already computed values into the function $f(x, y)$.

The implicit or backward Euler method

$$y_{k+1} = y_k + h_k f\left(x_{k+1}, y_{k+1}\right), \ k = 0, 1, 2, \ldots, \quad y_0 = y(x_0),$$

is an implicit one-step method with the incremental function

$$\Phi\left(x, y, h_k\right) = f\left(x_{k+1}, y_{k+1}\right).$$

One has to solve an equation for computing y_{k+1}. The complexity of this step depends on $f(x, y)$. □

Remark 1.6 *Representation of implicit one-step methods.* Explicit one-step methods have an incremental function of the form $\Phi\left(x_k, y_k, h_k\right)$. For the considerations in this section, one can adopt the point of view that also implicit one-step methods can be written as explicit one-step methods, but generally one does not know the concrete form of the incremental function. □

Example 1.7 *Incremental function of the implicit Euler method.* The incremental function of the implicit Euler method can be written in the form

$$\Phi\left(x, y, h\right) = f\left(x + h, y + h\Phi\left(x, y, h\right)\right),$$

which allows formally the representation of this method as explicit one-step scheme. □

Definition 1.8 Local error. Let $\hat{y}_{k+1}$ be the result of one step of an explicit one-step method (1.2) with the initial value at the solution $y(x)$, i.e.,

$$\hat{y}_{k+1} = y(x_k) + h_k \Phi\left(x_k, y(x_k), h_k\right).$$

Then,

$$\mathrm{le}\left(x_{k+1}\right) = \mathrm{le}_{k+1} = y\left(x_{k+1}\right) - \hat{y}_{k+1}$$

is called local error, see Figure 1.1 □

Remark 1.9 *The local error.* In the literature, sometimes

$$\frac{y\left(x_{k+1}\right) - y(x_k)}{h_k} - \Phi\left(x_k, y(x_k), h_k\right)$$

is defined to be the local error.

One should require for a reasonable method that the local error is small in an appropriate sense. □

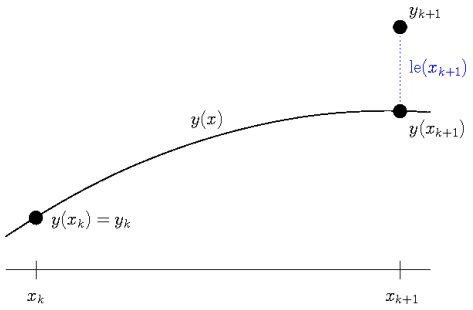

Figure 1.1: The local error.

Definition 1.10 Consistent method. Let $y(x)$ be the solution of the initial value problem (1.1), $h_{\max} = \max_k h_k$ and

$$S := \{(x, y) \ : \ x \in [x_0, x_e], \ y \in \mathbb{R}\}.$$

The one-step method (1.2) is said to be consistent, if for all $f \in C(S)$, that satisfy in S a Lipschitz condition with respect to y, it holds

$$\lim_{h_{\max} \to 0} \left(\max_{x_k \in I_h} \frac{|\mathrm{le}\,(x_{k+1})|}{h_k} \right) = 0$$

or

$$\lim_{h_{\max} \to 0} \left(\max_{x_k \in I_h} |f(x_k, y(x_k)) - \Phi\,(x_k, y(x_k), h_k)| \right) = 0.$$

Both conditions are equivalent. *problem for exercises* □

Remark 1.11 *Approximation of the derivative with the incremental function.* For bounded incremental functions it is obvious that the local error converges to zero if $h_{\max} \to 0$, because in this case it holds $h_K \to 0$ and $\hat{y}_{k+1} \to y(x_k)$. Consistency requires more, namely that the incremental function approximates the derivative of the solution sufficiently well

$$\frac{\mathrm{le}\,(x_{k+1})}{h_k} = \frac{y\,(x_{k+1}) - y(x_k)}{h_k} - \Phi\,(x_k, y(x_k), h_k) \approx y'(x_k) - \Phi\,(x_k, y(x_k), h_k)\,.$$

□

Example 1.12 *Consistency of the explicit Euler method.* For the explicit Euler method is $\Phi\,(x_k, y(x_k), h_k) = f\,(x_k, y(x_k))$. Hence, the second condition from Definition 1.10 is satisfied and the method is consistent. □

Remark 1.13 *Quality of the approximation of the incremental function.* For practical purposes, not only the consistency itself but the quality of the approximation of the derivative by the incremental function is essential. The quality allows a comparison of different one-step methods. For simplicity of presentation, let $h_k = h$ for all k. □

Definition 1.14 Order of consistency. An explicit one-step method (1.2) has the consistency order $p \in \mathbb{N}$ if p is the largest natural number such that for all functions $f \in C(S)$, which satisfy a Lipschitz condition with respect to y, it holds

$$|\mathrm{le}\,(x_k + h)| \leq c h^{p+1}$$

for all $x_k \in I_h$, for all I_h with $h \in (0, H]$, and where the constant $c > 0$ is independent of h. The constant c might depend on derivatives of $y(x)$, on $f(x, y)$, and on partial derivatives of $f(x, y)$. □

Example 1.15 *Order of consistency of the explicit Euler method.* Consider the explicit Euler method and assume that the function $y(x)$ is two times continously differentiable. Then it follows with Taylor series expansion and using the differential equation that

$$\begin{aligned} |\mathrm{le}\,(x_k+h)| &= |y(x_k+h)-\hat{y}_{k+1}| \\ &= |y(x_k)+hy'(x_k)+\frac{h^2}{2}y''(x_k+\theta h)-y(x_k)-h\underbrace{f\,(x_k,y(x_k))}_{=y'(x_k)}| \\ &= \frac{h^2}{2}\,|y''(x_k+\theta h)| \le \frac{h^2}{2}\,\|y\|_{C^2([x_0,x_e])}\,, \end{aligned}$$

with $\theta \in (0,1)$. Hence, the method has consistency order 1. □

Remark 1.16 *Consistency and convergence.* The consistency is a local property of a one-step method. For practical purposes it is important that the computed solution converges to the analytical solution if the grid becomes finer and finer. Of course, the order of convergence is of importance, too.

It will be shown that, under certain conditions, the convergence of a one-step method follows from its consistency and that the order of convergence equals the consistency order. □

Definition 1.17 Convergent method, order of convergence. A one-step method (1.2) converges for the initial value problem (1.1) on the interval $I=[x_0,x_e]$, if for each sequence of grids $\{I_h\}$ with $h_{\max}=\max_{h_k} h_k \to 0$ for the global error

$$e(x_k,h)=y(x_k)-y_k, \quad x_k \in I_h,$$

it follows that

$$\max_{x_k\in I_h} |e(x_k,h)| \to 0 \quad \text{for} \quad h_{\max}\to 0.$$

The one-step method has the order of convergence p^*, if p^* is the largest natural number such that for all step lengths $h_{\max}\in(0,H]$ it holds

$$|e(x_k,h)| \le c h_{\max}^{p^*} \quad \forall\ x_k \in I_h,$$

where $c>0$ is independent of $h_{\max}$. □

Lemma 1.18 Estimate for a sequence of real numbers. *Assume that for real numbers x_n, $n=0,1,\ldots$, the inequality*

$$|x_{n+1}| \le (1+\delta)\,|x_n|+\beta$$

holds with constants $\delta>0$, $\beta\ge 0$. Then, it follows that

$$|x_n| \le e^{n\delta}\,|x_0| + \frac{e^{n\delta}-1}{\delta}\beta, \quad n=0,1,\ldots$$

Proof: With induction, problem for exercises. ■

Theorem 1.19 Connection of consistency and convergence. *Let $y(x)$ be the solution of the initial value problem (1.1) with $f\in C(S)$. Let a Lipschitz condition hold for the second argument of the incremental function*

$$|\Phi(x,y_1,h)-\Phi(x,y_2,h)| \le M\,|y_1-y_2| \quad \forall\ (x,y)\in S,\ h\in(0,H].$$

Assume that for the local error the estimate

$$|\mathrm{le}\,(x_k+h)| \le ch^{p+1} \quad \forall\ x \in I_h, h \in (0,H]$$

is valid and assume that $y_0 = y(x_0)$.

Then, it follows for the global error that

$$|e(x_{k+1},h)| \le c\frac{e^{M(x_{k+1}-x_0)}-1}{M}h^p,$$

where c *is indepent of* h.

Proof: It holds

$$\begin{aligned} y_{k+1} &= y_k + h\Phi\,(x_k,y_k,h)\,, \\ y(x_{k+1}) &= y(x_k) + h\Phi\,(x_k,y(x_k),h) + \mathrm{le}\,(x_{k+1})\,, \quad k=0,1,\ldots. \end{aligned}$$

Then, it follows with the triangle inequality and the Lipschitz condition of the incremental function that

$$\begin{aligned} |e(x_{k+1},h)| &= |y(x_{k+1}) - y_{k+1}| \\ &= \big|y(x_k) - y_k + \mathrm{le}\,(x_{k+1}) + h\big(\Phi\,(x_k,y(x_k),h) - \Phi\,(x_k,y_k,h)\,\big)\big| \\ &= \big|e(x_k,h) + \mathrm{le}\,(x_{k+1}) + h\big(\Phi\,(x_k,y(x_k),h) - \Phi\,(x_k,y_k,h)\,\big)\big| \\ &\le |e(x_k,h)| + |\mathrm{le}\,(x_{k+1})| + h\,|\Phi\,(x_k,y(x_k),h) - \Phi\,(x_k,y_k,h)| \\ &\le |e(x_k,h)| + ch^{p+1} + hM\,|y(x_k) - y_k| \\ &= (1+hM)\,|e(x_k,h)| + ch^{p+1}. \end{aligned}$$

This sequence of inequalities has the form that was considered in Lemma 1.18. One obtains with $e(x_0) = 0$

$$|e(x_{k+1},h)| \le e^{(k+1)hM}\,|e(x_0)| + c\frac{e^{(k+1)hM}-1}{hM}h^{p+1} = c\frac{e^{M\left(x_{k+1}-x_0\right)}-1}{M}h^p.$$

■

Remark 1.20 *To Theorem 1.19.*

- The consideration of a constant step length is only for simplicity of presentation. The result of the theorems holds also for non-constant step lengths with $h = \max_k h_k$.
- The one-step methods computes an approximation y_k of the solution in the grid points x_k, $k = 0,1,\ldots,N$. To enable a better comparison with the analytical solution, one connects these points linearly form (x_k,y_k) to (x_{k+1},y_{k+1}). One obtains a piecewise linear approximation of the solution that is defined on $[x_0,x_e]$. This function is called $y^h(x)$. The considerations from above can be extended to the function $y^h(x)$.

□

1.2 Explicit Runge–Kutta Schemes

Remark 1.21 *Idea.* The Euler methods are only of first order. The idea of Runge[1]–Kutta[2] methods consists in using an incremental function $\Phi(x,y,h)$ that is a linear combination of values of $f(x,y)$ in discrete points. With this approach, one obtains methods of higher order for the cost of evaluating more values of $f(x,y)$.

[1] Carle David Tolmé Runge (1856 – 1927)
[2] Martin Kutta (1867 – 1944)

This approach can be illustrated well at the integral equation that is equivalent to the initial value problem (1.1). For simplicity, let the right hand side of (1.1) depend only on x. Then, the integral equation has the form

$$y(x) = y_0 + \int_{x_0}^{x} f(t)\ dt. \tag{1.3}$$

The idea of the Runge–Kutta methods consists in approximating the right hand side by a quadrature rule, e.g., in the interval $[x_k, x_{k+1}]$ by

$$\int_{x_k}^{x_{k+1}} f(t)\ dt \approx h_k \sum_{j=1}^{s} b_j f\left(x_k + c_j h_k\right)$$

with the weights b_j and the nodes $x_k + c_j h$.

In the following, only $h_k = h$ for all k will be considered for the reason of simplicity. □

Definition 1.22 Runge–Kutta methods, increments, stages. A Runge–Kutta method has the form

$$y_{k+1} = y_k + h\Phi(x, y, h),\ k = 0, 1, \ldots, \quad y_0 = y(x_0),$$

where the incremental function is defined with the help of

$$K_i(x, y, h) = f\left(x_k + c_i h, y_k + h \sum_{j=1}^{s} a_{ij} K_j(x, y, h)\right)$$

by

$$\Phi(x, y, h) = \sum_{i=1}^{s} b_i K_i(x, y, h),$$

with $c_1, \ldots, c_s, b_1, \ldots, b_s, a_{ij} \in \mathbb{R}$, $i, j = 1, \ldots, s$. The quantities $K_i(x, y, h)$, $i = 1, \ldots, s$, are called increments. The natural number $s \in \mathbb{N}$ is the number of stages of the method.

An equivalent definition is as follows

$$\begin{aligned} y_{k+1}^{(i)} &= y_k + h \sum_{j=1}^{s} a_{ij} f\left(x_k + c_j h, y_{k+1}^{(j)}\right), \\ \Phi(x, y, h) &= \sum_{i=1}^{s} b_i f\left(x_k + c_i h, y_{k+1}^{(i)}\right). \end{aligned}$$

The intermediate values $y_{k+1}^{(i)}$ are called stages. □

Remark 1.23 *Butcher*[3] *tableau.* For the reason of clearity, one writes a Runge–Kutta scheme in general in form of a tableau, the so-called Butcher tableau

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ c_3 & a_{31} & a_{32} & \cdots & a_{3s} \\ \vdots & \vdots & \vdots & & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ \hline & b_1 & b_2 & \cdots & b_s \end{array} \quad \Longrightarrow \quad \begin{array}{c|c} \mathbf{c} & A \\ \hline & \mathbf{b}^T \end{array}. \tag{1.4}$$

Here, $\mathbf{c}$ are the nodes, A is the matrix of the method, and $\mathbf{b}$ are the weights. □

[3] John C. Butcher, born. 1933

Remark 1.24 *Increments and Butcher tableau.* For explicit Runge–Kutta schemes, the increments can be computed one after another

$$\begin{aligned} K_1(x,y,h) &= f(x_k, y_k), \\ K_2(x,y,h) &= f\left(x_k + c_2 h, y_k + h a_{21} K_1(x,y,h)\right), \\ &\vdots \\ K_s(x,y,h) &= f\left(x_k + c_s h, y_k + h \sum_{j=1}^{s-1} a_{sj} K_j(x,y,h)\right). \end{aligned}$$

The Butcher tableau has the form

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

□

Example 1.25 *Explicit Euler scheme.* The explicit Euler scheme is an explicit Runge–Kutta scheme with the Butcher tableau

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

In the integral equation, the approximation

$$\int_{x_k}^{x_{k+1}} f(t, y(t))\, dt \approx h f\left(x_k, y_k(x_k)\right)$$

is used, see the proof of the Theorem of Peano, lectures notes of Numerical Mathematics I. □

Theorem 1.26 Consistency of explicit Runge–Kutta schemes. *Let $f \in C(S)$, see Definition 1.10. An explicit Runge–Kutta scheme is consistent if and only if*

$$\sum_{i=1}^{s} b_i = 1. \tag{1.5}$$

Proof: From the continuity of $f(x,y)$ it follows that

$$\lim_{h\to 0} K_i(x,y,h) = f(x_k, y(x_k)), \quad \forall\ (x,y) \in S,\ i = 1, \ldots, s,$$

for the case that the initial value of this step is $y_k = y(x_k)$. The continuity of the absolute value function gives

$$\begin{aligned} \lim_{h\to 0} \left| f(x_k, y(x_k)) - \Phi\left(x_k, y(x_k), h_k\right) \right| &= \lim_{h\to 0} \left| f(x_k, y(x_k)) - \sum_{i=1}^{s} b_i K_i(x,y,h) \right| \\ &= \left| f(x_k, y(x_k)) - \sum_{i=1}^{s} b_i \lim_{h\to 0} K_i(x,y,h) \right| \\ &= \left| f(x_k, y(x_k)) \left(1 - \sum_{i=1}^{s} b_i\right) \right| = 0 \end{aligned}$$

if and only if $\sum_{i=1}^{s} b_i = 1$. Hence, the second condition in Definition 1.10 is satisfied. ■

Theorem 1.27 Interpretation of the increments. *Let for the solution of* (1.1) *hold* $y \in C^2([x_0, x_e])$*, let* $f \in C(S)$*, and let* f *be Lipschitz continuous in the second argument. If* $y_k = y(x_k)$ *and*

$$c_i = \sum_{j=1}^{i-1} a_{ij}, \quad i \geq 2, \tag{1.6}$$

holds, then $K_i(x, y, h)$ *is an approximation of at least first order to* $y'(x + c_i h)$*, i.e.,*

$$y'(x_k + c_i h) - K_i(x, y, h) = \mathcal{O}\left(h^2\right).$$

Proof: The proof follows by induction.

$i = 2$. For $i = 2$ follows with (1.1), the Lipschitz continuity, and Taylor series expansion that

$$\begin{aligned}
&\left|y'(x_k + c_2 h) - K_2(x, y, h)\right| \\
&\quad = \left|f\left(x_k + c_2 h, y(x_k + c_2 h)\right) - f\left(x_k + c_2 h, y(x_k) + h a_{21} f(x_k, y(x_k))\right)\right| \\
&\quad \leq L\left|y(x_k + c_2 h) - y(x_k) - h a_{21} f(x_k, y(x_k))\right| \\
&\quad = L\left|y(x_k) + c_2 h y'(x_k) + \mathcal{O}(h^2) - y(x_k) - h a_{21} y'(x_k)\right| \\
&\quad = L\left|(c_2 - a_{21}) h y'(x_k) + \mathcal{O}(h^2)\right|.
\end{aligned}$$

Hence, in the case $c_2 = a_{21}$ the error is of order $\mathcal{O}(h^2)$.

$i > 2$. Let the asymptotic order of the errors be proved for all indices $2, \ldots, i-1$. Then, one gets in the same way as for $i = 2$

$$\begin{aligned}
&\left|y'(x_k + c_i h) - K_i(x, y, h)\right| \\
&\quad = \left|f\left(x_k + c_i h, y(x_k + c_i h)\right) - f\left(x_k + c_i h, y(x_k) + h \sum_{j=i}^{i-1} a_{ij} K_j(x, y, h)\right)\right| \\
&\quad \leq L\left|y(x_k + c_i h) - y(x_k) - h \sum_{j=i}^{i-1} a_{ij} K_j(x, y, h)\right| \\
&\quad = L\left|y(x_k) + c_i h y'(x_k) + \mathcal{O}(h^2) - y(x_k) - h \sum_{j=i}^{i-1} \left(a_{ij}\left(y'(x_k + c_j h) + \mathcal{O}(h^2)\right)\right)\right| \\
&\quad = L\left|c_i h y'(x_k) + \mathcal{O}(h^2) - h \sum_{j=i}^{i-1} \left(a_{ij}\left(y'(x_k) + \mathcal{O}(h)\right)\right)\right| \\
&\quad = L\left|h\left(c_i - \sum_{j=i}^{i-1} a_{ij}\right) y'(x_k) + \mathcal{O}(h^2)\right|.
\end{aligned}$$

The order of the error $\mathcal{O}(h^2)$ is given, if $c_i = \sum_{j=i}^{i-1} a_{ij}$. ■

Remark 1.28 *Conditions on the coefficients for certain orders of convergence.* The conditions from Theorems 1.26 and 1.27 are satisfied for many explicit Runge–Kutta schemes. The goal consists in determining the coefficients $b_1, \ldots, b_s$, and a_{ij} in such a way that one obtains an order of consistency as high as possible. The consistency order of a Runge–Kutta scheme with s stages can be derived from the Taylor series expansion of the local error. Let (1.5) be valid, then one obtains, e.g.,

- A Runge–Kutta scheme with the parameters $(A, \mathbf{b}, \mathbf{c})$ has at least consistency order $p = 2$ if

$$\sum_{j=1}^{s} b_j c_j = \frac{1}{2}. \tag{1.7}$$

This condition will be shown in Example 1.29 for $s = 2$.

- If in addition

$$\sum_{j=1}^{s} b_j c_j^2 = \frac{1}{3} \quad \text{and} \quad \sum_{j=1}^{s} b_j \sum_{k=1}^{s} a_{jk} c_k = \frac{1}{6}$$

  hold, then the order of consistency is at least $p = 3$.

The proof of the last statement and conditions for even higher order consistency can be found in the literature, e.g. in (Strehmel and Weiner, 1995; Strehmel et al., 2012, Section 2.4.2). □

Example 1.29 *Runge–Kutta methods with* 2 *stages.* For the investigation of 2-stage Runge–Kutta schemes one considers for simplicity the so-called autonomous initial value problem

$$y'(x) = f(y(x)), \quad y(x_0) = y_0.$$

One has for the increments

$$\begin{aligned} K_1(y,h) &= f(y_k), \\ K_2(y,h) &= f\left(y_k + ha_{21}K_1(y_k,h)\right) = f\left(y_k + ha_{21}f(y_k)\right) \\ &= f(y_k) + ha_{21}f(y_k)f_y(y_k) + \mathcal{O}(h^2). \end{aligned}$$

If the initial value is exact, it follows for the incremental function that

$$\begin{aligned} \Phi(y(x_k)) &= b_1K_1(y,h) + b_2K_2(y,h) \\ &= (b_1+b_2)f(y(x_k)) + hb_2a_{21}f(y(x_k))f_y(y(x_k)) + \mathcal{O}(h^2). \end{aligned}$$

The Taylor series expansion of the solution has the form

$$y(x_k+h) = y(x_k) + h \underbrace{y'(x_k)}_{=f(y(x_k))} + \frac{h^2}{2}y''(x_k) + \mathcal{O}\left(h^3\right).$$

One obtains with the chain rule

$$y''(x) = \frac{d}{dx}y'(x) = \frac{d}{dx}f(y(x)) = f_y(y)y'(x) = f_y(y)f(y(x)).$$

Now, it follows for the local error that

$$\begin{aligned} \mathrm{le}(x_k+h) &= y(x_k+h) - y(x_k) - h\Phi(y(x_k)) \\ &= y(x_k) + hf(y(x_k)) + \frac{h^2}{2}\left(f_y(y(x_k))f(y(x_k))\right) + \mathcal{O}\left(h^3\right) - y(x_k) \\ &\quad -h\Big((b_1+b_2)f(y(x_k)) + hb_2a_{21}f(y(x_k))f_y(y(x_k)) + \mathcal{O}(h^2)\Big) \\ &= h\big(1-(b_1+b_2)\big)f(y(x_k)) + h^2\left(\frac{1}{2} - b_2a_{21}\right)f(y(x_k))f_y(y(x_k)) \\ &\quad +\mathcal{O}\left(h^3\right). \end{aligned}$$

To achieve an order of consistency as large as possible, the first two terms have to vanish. One obtains with the condition $c_2 = a_{21}$ that

$$b_1 + b_2 = 1, \quad b_2a_{21} = \frac{1}{2} \iff b_2c_2 = \frac{1}{2}.$$

The first equation is the general condition for consistency (1.5) and the second condition is exactly (1.7) for $s = 2$. These two conditions characterize all 2-stage explicit Runge–Kutta methods that possess consistency and convergence order 2

$$\begin{array}{c|cc} c_2 & c_2 & \\ \hline & 1-\frac{1}{2c_2} & \frac{1}{2c_2} \end{array}, \quad \text{with } c_2 \neq 0.$$

In the case $c_2 = 1/2$, one obtains the method of Runge (1895)

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

This method corresponds with respect to the approximation of the integral in (1.3) to the application of the mid point rule.

For $c_2 = 1$ one gets the method of Heun[4] (1900)

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

which corresponds to the use of the trapezoidal rule for the numerical quadrature in (1.3). □

Remark 1.30 *Autonomous ordinary differential equations.* Every explicit first order ordinary differential equation

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

can be transformed into an autonomous form

$$\tilde{\mathbf{y}}'(x) = \tilde{\mathbf{f}}\,(\tilde{\mathbf{y}}(x)) = \begin{pmatrix} \mathbf{f}(x, \mathbf{y}(x)) \\ 1 \end{pmatrix}$$

by introducing the function

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

□

Theorem 1.31 Consistency and convergence of explicit Runge–Kutta methods. *Let $y(x)$ be the solution of the initial value problem* (1.1) *with $f \in C(S)$ and let $f(x, y)$ satisfy a Lipschitz condition in the second argument. Then, an explicit Runge–Kutta scheme that is consistent of order p converges also with order p.*

Proof: The incremental function of an explicit Runge–Kutta scheme is a linear combination of values of the right hand side $f(x, y)$. Thus, the assumptions of Theorem 1.19 are satisfied since the Lipschitz condition in this theorem follows from the required Lipschitz condition on the right hand side of the differential equation. The statement of the theorem follows now directly from Theorem 1.19. ■

Remark 1.32 *Explicit Runge–Kutta methods of higher order.* Analogously to 2-stage methods, it is possible to derive conditions on the coefficients of an explicit Runge–Kutta scheme in order to construct methods of higher order. An important question is the minimal number of stages that is necessary to be able to reach a certain order. Some answers to this question are from Butcher (1963, 1965, 1985)

p	1	2	3	4	5	6	7	8
$\min s$	1	2	3	4	6	7	9	11

□

[4] Karl Heun (1859 – 1929)

Example 1.33 *Classical Runge–Kutta scheme (1901).* The so-called classical Runge–Kutta scheme has four stages and the Butcher tableau

$$\begin{array}{c|cccc} 0 & & & & \\ 1/2 & 1/2 & & & \\ 1/2 & 0 & 1/2 & & \\ 1 & 0 & 0 & 1 & \\ \hline & 1/6 & 1/3 & 1/3 & 1/6 \end{array} .$$

It is based on the Simpson[5] rule. The center node of the Simpson rule is used twice, $c_2 = c_3$, but with a different second argument for the computation of the increments. This method is of fourth order. □

1.3 Step Length Control

Remark 1.34 *Motivation.* The considerations so far did not provide a way for estimating a good step length for solving a given initial value problem with prescribed accuracy and with as little work as possible. A good step length depends certainly on the concrete problem and generally it will change within the considered interval. For these reasons, the step length has to be controlled during the numerical solution of the initial value problem.

A typical approach consists in computing two approximations of the solution in a node with different methods and to draw conclusions on the size of the local error based on the difference of these approximations. Of course, the consideration of the global error would be better. However, Theorem 1.19 shows that on the one hand the global error is influenced by problem-dependent terms, like the length of the interval $[0, T]$ or the Lipschitz constant. On the other hand, the global error can be small only if the local errors are small. □

1.3.1 The Richardson Method

Remark 1.35 *Idea.* Given a numerical method for solving an initial value problem and given a step length h. The Richardson[6] method consists of the following steps, see also Figure 1.2:

1. Starting from a node (x_0, y_0) and using a step length of $2h$, an approximation y_{2h} at the node $x_0 + 2h$ will be computed.
2. Two approximations y_h and $y_{2\times h}$ in $x_0 + h$ and $x_0 + 2h$ are computed with two steps of length h.
3. The step length will be controlled by comparing y_{2h} and $y_{2\times h}$.

In general, the more accurate approximation will be $y_{2\times h}$. In addition, it will be demonstrated that it is possible to improve the accuracy of $y_{2\times h}$ with the information obtained by this method. □

Example 1.36 *Richardson method for an explicit 2-stage Runge–Kutta method.* Consider an explicit 2-stage Runge–Kutta scheme. One obtains in the first step of the Richardson method

$$\begin{aligned} y_{2h}^{(1)} &= y_0, \\ y_{2h}^{(2)} &= y_0 + 2ha_{21}f(x_0, y_0), \\ y_{2h} &= y_0 + 2h\left[b_1K_1(x, y) + b_2K_2(x, y)\right] \\ &= y_0 + 2h\left[b_1 f(x_0, y_{2h}^{(1)}) + b_2 f\left(x_0 + 2c_2h, y_{2h}^{(2)}\right)\right], \end{aligned}$$

[5] Thomas Simpson (1710 – 1761)
[6] Lewis Fry Richardson (1881 – 1953)

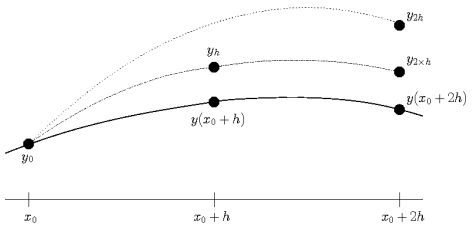

Figure 1.2: Sketch of the Richardson method.

or written as Butcher tableau

$$\begin{array}{c|cc} 0 & & \\ 2c_2 & 2a_{21} & \\ \hline & 2b_1 & 2b_2 \end{array}.$$

The second step of the Richardson method yields

$$\begin{aligned} y_{2\times h}^{(1)} &= y_0, \\ y_{2\times h}^{(2)} &= y_0 + ha_{21} f(x_0, y_0), \\ y_{2\times h}^{(3)} = y_h &= y_0 + h\left[b_1 f\left(x_0, y_{2\times h}^{(1)}\right) + b_2 f\left(x_0 + c_2 h, y_{2\times h}^{(2)}\right)\right], \\ y_{2\times h}^{(4)} &= y_h + ha_{21} f(x_0 + h, y_h), \\ y_{2\times h} &= y_h + h\left[b_1 f(x_0 + h, y_h) + b_2 f\left(x_0 + h + c_2 h, y_{2\times h}^{(4)}\right)\right]. \end{aligned}$$

The Butcher tableau of this method is

$$\begin{array}{c|cccc} 0 & & & & \\ c_2 & a_{21} & & & \\ 1 & b_1 & b_2 & & \\ 1+c_2 & b_1 & b_2 & a_{21} & \\ \hline & b_1 & b_2 & b_1 & b_2 \end{array}.$$

That means, the computation of $y_{2\times h}$ is equivalent to the computation of an approximation with the help of an explicit 4-stage Runge–Kutta scheme.

Altogether, 5 function evaluations are needed:

$$\begin{aligned} &f(x_0, y_0),\ f\left(x_0 + 2c_2 h, y_{2h}^{(2)}\right), f\left(x_0 + c_2 h, y_{2\times h}^{(2)}\right), \\ &f(x_0 + h, y_h),\quad f\left(x_0 + h + c_2 h, y_{2\times h}^{(4)}\right). \end{aligned}$$

In the case of a s-stage Runge–Kutta method, $3s - 1$ function evaluations are required. This number is rather large and the high costs per time step are a disadvantage of the Richardson method. □

Remark 1.37 *Comparison of both approximations.* Consider a one-step method

$$y_{k+1} = y_k + h\Phi(x, y, h)$$

of order p. Let the initial value x_0 be exact, then it follows for the local error in $x_0 + 2h$ that

$$y(x_0 + 2h) - y_{2h} = c(x_0)(2h)^{p+1} + \mathcal{O}\left(h^{p+2}\right). \tag{1.8}$$

For estimating the local error in $y_{2\times h}$ it will be assumed that the incremental function $\Phi(x, y, h)$ is Lipschitz continuous in the second argument. This assumption is

always satisfied for explicit Runge–Kutta schemes if $f(x, y)$ possesses this property, see the proof of Theorem 1.31. It is

$$y_{2\times h} = y_h + h\Phi\left(x + h, y_h, h\right). \tag{1.9}$$

Let

$$\hat{y}_{2\times h} = y(x_0 + h) + h\Phi\left(x + h, y(x_0 + h), h\right) \tag{1.10}$$

the iterate which is computed with the exact starting value. Using the definition of the consistency order, one obtains with (1.9) and (1.10)

$$\begin{aligned} & y(x_0 + 2h) - y_{2\times h} \\ &= \left(y(x_0 + 2h) - \hat{y}_{2\times h}\right) + \left(\hat{y}_{2\times h} - y_{2\times h}\right) \\ &= \left[c(x_0 + h)h^{p+1} + \mathcal{O}\left(h^{p+2}\right)\right] + \Big[y(x_0 + h) + h\Phi\left(x + h, y(x_0 + h), h\right) \\ &\quad - y_h - h\Phi\left(x + h, y_h, h\right)\Big]. \end{aligned}$$

For the terms with the incremental function, one gets from the Lipschitz continuity and the consistency order

$$\left|h\Phi\left(x + h, y(x_0 + h), h\right) - h\Phi\left(x + h, y_h, h\right)\right| \leq hL\left|y(x_0 + h) - y_h\right| = \mathcal{O}\left(h^{p+2}\right).$$

It follows that

$$\begin{aligned} & y(x_0 + 2h) - y_{2\times h} \\ &= c(x_0 + h)h^{p+1} + y(x_0 + h) - y_h + \mathcal{O}\left(h^{p+2}\right) \\ &= c(x_0 + h)h^{p+1} + c(x_0)h^{p+1} + \mathcal{O}\left(h^{p+2}\right) + \mathcal{O}\left(h^{p+2}\right) \\ &= 2c(x_0)h^{p+1} + \mathcal{O}\left(h^{p+2}\right), \end{aligned} \tag{1.11}$$

where one assumes that $c(x_0 + h) = c(x_0) + \mathcal{O}(h)$, i.e., that the constants do not change too rapidly.

Neglecting in (1.8) and (1.11) the higher order terms allows to eliminate the constant. One obtains

$$c(x_0) = \frac{1}{2}\left(\frac{y_{2\times h} - y_{2h}}{2^p - 1}\right)\frac{1}{h^{p+1}}. \tag{1.12}$$

It follows for the local error of the more accurate method that

$$y(x_0 + 2h) - y_{2\times h} = \frac{y_{2\times h} - y_{2h}}{2^p - 1} + \mathcal{O}\left(h^{p+2}\right). \tag{1.13}$$

□

Remark 1.38 *Increasing the accuracy.* Rearranging terms in (1.13) gives

$$y(x_0 + 2h) - \left(y_{2\times h} + \frac{y_{2\times h} - y_{2h}}{2^p - 1}\right) = \mathcal{O}\left(h^{p+2}\right).$$

Then,

$$\overline{y}_{2\times h} = y_{2\times h} + \frac{y_{2\times h} - y_{2h}}{2^p - 1}$$

is an approximation of the solution of order $p + 1$. □

Remark 1.39 *Automatic step length control.* From (1.12) it follows that

$$\text{err} = \frac{\left|y_{2\times h} - y_{2h}\right|}{2^p - 1} \approx 2c(x_0)h^{p+1} \tag{1.14}$$

is a computable approximation of the local error. This approximation will be compared with a prescribed tolerance. Often, a so-called scaled tolerance sc is used, (Hairer et al., 1993, p. 167) or (Strehmel et al., 2012, p. 61). The scaled tolerance is a combination of an absolute tolerance atol and a relative tolerance rtol

$$\mathrm{sc} = \mathrm{atol} + \max\left\{|y_0|, |y_{2\times h}|\right\} \mathrm{rtol}.$$

Then, the scaled error

$$\mathrm{err}_{\mathrm{sc}} = \frac{|y_{2\times h} - y_{2h}|}{(2^p - 1)\mathrm{sc}}$$

is defined.

- If $\mathrm{err}_{\mathrm{sc}} \leq 1 \iff \mathrm{err} \leq \mathrm{sc}$, then the performed step will be accepted. Starting from $y_{2\times h}$ or $\overline{y}_{2\times h}$, the next step will be performed. If $\overline{y}_{2\times h}$ is used, this approach is also called local Richardson extrapolation.
  An important aspect is the choice of the step length h_{new} for the next step. The guideline is that the scaled error for the next step should be on the one hand still smaller than 1 but on the other hand as close as one as possible. Following (1.14), it should hold

$$\begin{aligned} 1 &= \frac{\mathrm{err}_{\mathrm{new}}}{\mathrm{sc}} = \frac{2c\left(x_0 + 2h_{\mathrm{new}}\right) h_{\mathrm{new}}^{p+1}}{\mathrm{sc}} \approx \frac{2c(x_0) h_{\mathrm{new}}^{p+1}}{\mathrm{sc}} \\ &= \frac{2c(x_0) h^{p+1}}{\mathrm{sc}} \left(\frac{h_{\mathrm{new}}}{h}\right)^{p+1} \approx \mathrm{err}_{\mathrm{sc}} \left(\frac{h_{\mathrm{new}}}{h}\right)^{p+1}, \end{aligned}$$

  i.e. h_{new} has to be chosen such that

$$h_{\mathrm{new}} \approx \left(\frac{1}{\mathrm{err}_{\mathrm{sc}}}\right)^{1/(p+1)} h. \tag{1.15}$$

- If $\mathrm{err}_{\mathrm{sc}} > 1$, then the performed step will be rejected. The Richardson is repeated from (x_0, y_0) with a step length $h_{\mathrm{new}} < h$.
  That means, the work which was spent for performing the step with step length h was wasted. One likes to avoid this situation.

□

Remark 1.40 *Issues of the practical implementation.* In practical simulations, one uses some modifications of (1.15) for stabilizing the algorithm.

- A safety factor $\alpha \in (0, 1)$ is introduced

$$h_{\mathrm{new}} = \alpha \left(\frac{1}{\mathrm{err}_{\mathrm{sc}}}\right)^{1/(p+1)} h,$$

  often $\alpha \in [0.8, 0.9]$.
- One likes to avoid large oscillations of the sizes of subsequent steps. For this reason, a factor for the maximal increase $\alpha_{\max}$ of the new step size with respect to the current step size and a factor for the maximal decrease $\alpha_{\min} < \alpha_{\max}$ are used. Then, one obtains

$$h_{\mathrm{new}} = h \min\left\{\alpha_{\max}, \max\left\{\alpha_{\min}, \alpha\left(\frac{1}{\mathrm{err}_{\mathrm{sc}}}\right)^{1/(p+1)}\right\}\right\}.$$

  If a very large step length is proposed

$$\alpha \left(\frac{1}{\mathrm{err}_{\mathrm{sc}}}\right)^{1/(p+1)} > \alpha_{\max},$$

  then the factor $\alpha_{\max}$ becomes effective and similarly for the case that a very small step length is proposed.

- Usually, one describes a minimal step length $h_{\min}$ and a maximal step length $h_{\max}$ and requires for all step lengths that $h_k \in [h_{\min}, h_{\max}]$.
- In the first step, one has to estimate h. Generally, this estimate has to be corrected. In practice, this correction is done very fast by algorithms for automatic step length control. An algorithm for determining a good initial step length can be found in (Hairer et al., 1993, p. 168).

□

1.3.2 Embedded Runge–Kutta Schemes

Remark 1.41 *Motivation, embedded Runge–Kutta schemes.* Richardson extrapolation is quite expensive in terms of evaluations of the incremental function. It is possible to construct a step length control that needs less evaluations, with so-called embedded Runge–Kutta schemes.

The idea of embedded Runge–Kutta schemes consists in computing numerical approximations of the solution at the next time with two one-step methods with different order. The methods are chosen in such a way that it is possible to use certain evaluations of the incremental function for both of them. That means, one has to construct a Runge–Kutta scheme of the form

$$\begin{array}{c|cccc} 0 & & & & \\ c_2 & a_{21} & & & \\ \vdots & & \ddots & & \\ c_s & a_{s1} & \cdots & a_{s,s-1} & \\ \hline & b_1 & \cdots & b_{s-1} & b_s \\ & \tilde{b}_1 & \cdots & \tilde{b}_{s-1} & \tilde{b}_s \end{array},$$

such that

$$y_1 = y_0 + h \sum_{i=1}^{s} b_i K_i(x, y)$$

is order of p and

$$\tilde{y}_1 = y_0 + h \sum_{i=1}^{s} \tilde{b}_i K_i(x, y)$$

is of order q, see Figure 1.3. In general, it is $q = p - 1$ or $q = p + 1$. □

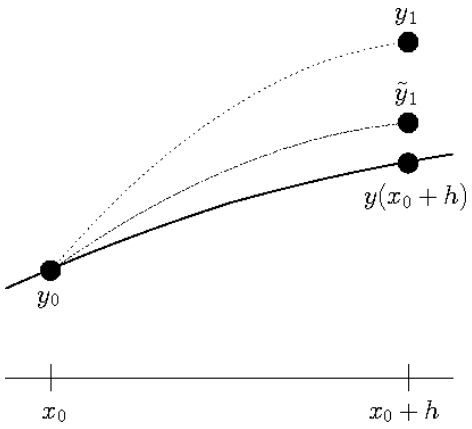

Figure 1.3: Sketch of embedded Runge–Kutta schemes.

Example 1.42 *Runge–Kutta–Fehlberg 2(3).* Consider explicit Runge–Kutta schemes with 3 stages

$$\begin{array}{c|ccc} 0 & & & \\ c_2 & a_{21} & & \\ c_3 & a_{31} & a_{32} & \\ \hline & b_1 & b_2 & b_3 \\ & \tilde{b}_1 & \tilde{b}_2 & \tilde{b}_3 \end{array} \quad \begin{array}{l} \\ \\ \\ \hline p=2 \\ q=3 \end{array}.$$

One of the schemes should be of order 2 and the other one of third order. There are 11 parameters to choose. From Theorem 1.26, Theorem 1.27 , and Remark 1.28 it follows that 8 equations have to be satisfied

$$\begin{aligned} c_2 &= a_{21}, \\ c_3 &= a_{31} + a_{32}, \\ b_1 + b_2 + b_3 &= 1, \\ b_2 c_2 + b_3 c_3 &= \frac{1}{2}, \\ \tilde{b}_1 + \tilde{b}_2 + \tilde{b}_3 &= 1, \\ \tilde{b}_2 c_2 + \tilde{b}_3 c_3 &= \frac{1}{2}, \\ \tilde{b}_2 c_2^2 + \tilde{b}_3 c_3^2 &= \frac{1}{3}, \\ \tilde{b}_3 a_{32} c_2 &= \frac{1}{6}. \end{aligned}$$

That means, one has to set three parameters. First, one can choose $c_2 = 1$, $b_3 = 0$. Then, it follows from the first equation that $a_{21} = 1$, from the fourth equation that $b_2 = 1/2$, and from the third equation that $b_1 = 1/2$. Now, one chooses $c_3 = 1/2$. From the sixth and seventh equation it follows that $\tilde{b}_2 = 1/6$ and $\tilde{b}_3 = 4/6$. Then, one gets from the fifth equation $\tilde{b}_1 = 1/6$ and from the eight equation $a_{32} = 1/4$. Finally, the second equation gives $a_{31} = 1/4$. The resulting methods have the form

$$\begin{array}{c|ccc} 0 & & & \\ 1 & 1 & & \\ 1/2 & 1/4 & 1/4 & \\ \hline & 1/2 & 1/2 & 0 \\ & 1/6 & 1/6 & 4/6 \end{array} \quad \begin{array}{l} \\ \\ \\ \hline p=2 \\ q=3 \end{array}.$$

The method with order $q = 3$ is the Simpson rule. The complete embedded approach is called Runge–Kutta–Fehlberg[7] 2(3) method (RKF 2(3)). □

Remark 1.43 *Error estimate, theoretical drawback.* It holds for the embedded scheme

$$y_1 = y(x_0 + h) + \mathcal{O}\left(h^{p+1}\right), \quad \tilde{y}_1 = y(x_0 + h) + \mathcal{O}\left(h^{q+1}\right).$$

It follows that

$$\begin{aligned} |\text{err}| &:= |\tilde{y}_1 - y_1| = \left|y(x_0 + h) + \mathcal{O}\left(h^{p+1}\right) - y(x_0 + h) + \mathcal{O}\left(h^{q+1}\right)\right| \\ &= \left|O\left(h^{p+1}\right) + \mathcal{O}\left(h^{q+1}\right)\right| \end{aligned} \tag{1.16}$$

is an estimate of the main error term of the Runge–Kutta scheme of order $q^* = \min\{p, q\}$. That means, one obtains only an estimate of the error of the lower order method. To obtain information only on the lower order method is the main theoretical drawback of the approach, since one is interested actually in the higher order method and one will continue the computation also from the higher order approximation. □

[7] Erwin Fehlberg (1911 – 1972)

Remark 1.44 *Automatic step length control, I Controller.* Let h be the step size that was used for computing y_1 of order p and $\tilde{y}_1$ of order q with $p < q$. From (1.16), one has

$$\text{err} = y_1 - \tilde{y}_1 = ch^{p+1}.$$

Given a tolerance tol for the maximal local error.

- One approach consists in controlling the error per step (EPS). Then, one requires that

  $$r_1 = |\text{err}| \leq \text{tol}. \tag{1.17}$$

  If this condition is satisfied, then one requires for the new step size that the local error is equal to the tolerance

  $$|c|\, h_{\text{new}}^{p+1} = \text{tol}.$$

  This requirement gives

  $$h_{\text{new}} = \left(\frac{\text{tol}}{|c|}\right)^{1/(p+1)} = \left(\frac{\text{tol}}{|c|\, h^{p+1}}\right)^{1/(p+1)} h$$

  Thus, the new step length is computed by

  $$h_{\text{new}} = \left(\frac{\alpha\ \text{tol}}{r_1}\right)^{1/k} h, \tag{1.18}$$

  where $k = p + 1$ and $\alpha \in (0, 1)$ is again a safety factor.
- Another way is the consideration of the error relative to the current step length, the so-called error per unit step (EPUS),

  $$r_1 = \frac{|\text{err}|}{h}.$$

  The satisfaction of a condition of form (1.17) leads to a new step of form (1.18) with $k = p$.
- If (1.17) is not satisfied, then the step is rejected and it will be repeated with a step length smaller than h.
- A generalization of this approach is the so-called I Controller. Replacing in (1.18) $1/k$ by k_I gives

  $$h_{\text{new}} = \left(\frac{\alpha\ \text{tol}}{r_1}\right)^{k_I} h.$$

  For obtaining a useful automatic step length control mechanism, the choice $kk_I = 1$ is not necessary. The following choices can be found in the literature

  $$\begin{array}{rcll} kk_I \in [0,2] & \Longleftrightarrow & k_I \in [0, 2/k] & \text{stable control,} \\ kk_I \in (1,2) & \Longleftrightarrow & k_I \in (1/k, 2/k) & \text{fast and oscillating control,} \\ kk_I \in (0,1) & \Longleftrightarrow & k_I \in (0, 1/k) & \text{slow and smooth control,} \\ kk_I = 1 & \Longleftrightarrow & k_I = 1/k & \text{standard I Controller.} \end{array}$$

  There are more sophisticated controllers that are used in practical simulations, see Söderlind (2002) for an overview.

□

Remark 1.45 *Methods used in practice.* In practice, one uses, e.g.,

- RKF 4(5), $s = 6$, Fehlberg (1964),
- RKF 7(8), $s = 13$, Fehlberg (1969),

- DOPRI 4(5) (or DOPRI 5(4) or DOPRI5), $s = 6$, Dormand[8], Prince[9]: Dormand and Prince (1980),
- DOPRI 7(8), $s = 13$, Prince and Dormand (1981).

The standard routine `ode45` from MATLAB uses DOPRI 4(5) and the routine `ode23` uses RKF 2(3), see Example 1.42. □

Remark 1.46 *Fehlberg trick.* The Fehlberg trick requires that

$$K_s = f\left(x_k + c_s h, y_k + h\sum_{i=1}^{s-1} a_{si}K_i\right) \stackrel{!}{=} f\left(x_k + h, y_k + h\sum_{i=1}^{s} b_i K_i\right),$$

i.e., the last evaluation of the incremental function of the old step can be used as first value of the incremental function in the new step. The conditions for applying this trick are

$$a_{si} = b_i,\ i = 1, \ldots, s-1, \quad b_s = 0, \quad c_s = 1.$$

It can be applied, e.g., in DOPRI 4(5). This trick works only if $h_{\text{old}} \approx h_{\text{new}}$. □

[8] John R. Dormand
[9] P. J. Prince