Numerical Methods in Engineering and Applied Science

Lecture 7. Runge–Kutta methods.

The *model problem* that we consider here is the following **initial value problem**: *For a function* $\boldsymbol{f} : [0,T]\times\mathbb{R}^N \to \mathbb{R}^N$ *and* $\boldsymbol{u}_0 \in \mathbb{R}^N$ *find a differentiable function* $\boldsymbol{u}(t)$ *such that*

$$\begin{cases} \dfrac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = \boldsymbol{f}(t, \boldsymbol{u}(t)), \\ \boldsymbol{u}(0) = \boldsymbol{u}_0. \end{cases} \tag{1}$$

Most commonly it describes the **time evolution** of some quantity. The numerical methods for it are called **time stepping** or **time marching** methods. These methods evaluate the numerical solution $\boldsymbol{u}^{n+1}$ at a time instant t_{n+1} using the information from previous time instants t_n, t_{n-1} etc.

We consider three groups of such methods:

- **Taylor series methods**;
- **Multistage (Runge-Kutta) methods**;
- **Multistep methods**.

We consider time points

$$t_0 = 0 < t_1 < t_2 < ... < t_{N-1} < t_N = T. \tag{2}$$

If the points are equidistant, we have

$$t_n = nh, \quad \forall n \in \{0, 1, ..., N\}, \tag{3}$$

where h is the discretization (or integration) step,

$$h = T/N. \tag{4}$$

Our objective is to find values u_n that would approximate $u(t_n)$ such that

$$||\boldsymbol{u}_n - \boldsymbol{u}(t_n)|| \to 0 \quad \text{as} \quad h \to 0. \tag{5}$$

Runge–Kutta methods: s-stage one-step methods.

Explicit Euler scheme

$$u_{n+1} = u_n + hf(t_n, u_n) \tag{6}$$

is a *first-order* method.

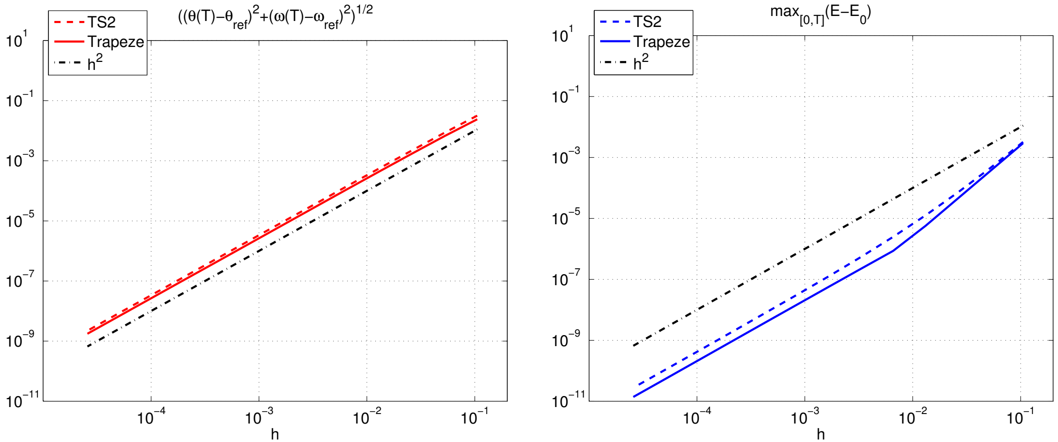
Improved Euler method (Heun method)

- Stage 1 : $\tilde{u}_{n+1} = u_n + hf(t_n, u_n)$
- Stage 2 : $u_{n+1} = u_n + h\dfrac{f(t_n, u_n) + f(t_{n+1}, \tilde{u}_{n+1})}{2}$

is a *second-order* method.

Note that these scheme have the same form for scalar and for vector ODEs.

Example. Nonlinear pendulum. Absolute error at $t = 4P$.



The general form of any s-stage Runge–Kutta method can be written as

$$u_{n+1} = u_n + h\Phi(t_n, y_n, h), \quad \text{where } \Phi(t_n, u_n, h) = \sum_{i=1}^{s} b_i k_i, \tag{7}$$

$$k_i = f(t_n + c_i h,\ u_n + h \sum_{j=1}^{s} a_{i,j} k_j), \quad i = 1, ..., s \tag{8}$$

The coefficients can be written in a tabular form (*Butcher tableau*)

$$\begin{array}{c|cccc}
c_1 & a_{1,1} & a_{1,2} & \cdots & a_{1,s} \\
c_2 & a_{2,1} & a_{2,2} & \cdots & a_{2,s} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
c_s & a_{s,1} & a_{s,2} & \cdots & a_{s,s} \\
\hline
 & b_1 & b_2 & \cdots & b_s
\end{array}$$

In general, to evaluate $\{k_i\}$, it is necessary to solve a system of s non-linear equations (8). However, if $a_{i,j} = 0$ for all $j \geq i$, the unknowns k_i can be evaluated without solving non-linear equations. If this is the case, the method is called *explicit* :

$$
\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{2,1} & & & & \\
c_3 & a_{3,1} & a_{3,2} & & & \\
\vdots & \vdots & \vdots & \ddots & & \\
c_s & a_{s,1} & a_{s,2} & \cdots & a_{s,s-1} & \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}
$$

Otherwise, the methode is called *implicit*.

Definition. A Runge–Kutta is *consistent to order* p if its local truncation error (given that $u_n = u(t_n)$) is such that

$$||u(t_{n+1}) - u_{n+1}|| \leq Ch^{p+1}, \tag{9}$$

i.e., the first p terms of the Taylor series expansion of the exact solution $u(t_{n+1})$ and numerical solution u_{n+1} coincide. We assumed that the partial derivatives of $f(t, u)$ exist and that they are continuous to order p.

Theorem. Suppose that (9) is satisfied for all $t \in [0, T]$, i.e.,

$$||u(t + h) - u(t) - h\Phi(t, u(t), h)|| \leq Ch^{p+1} \tag{10}$$

and suppose that there exist a neighborhood of the solution where Φ satisfies

$$||\Phi(t, v, h) - \Phi(t, u, h)|| \leq \Lambda||v - u||. \tag{11}$$

Then the global discretization error is bounded by

$$||u(t_N) - u_N|| \leq h^p \frac{C}{\Lambda} \left(\exp(\Lambda T) - 1\right). \tag{12}$$

The Euler method is the only first-order explicit 1-stage Runge–Kutta (RK) method.

The order conditions for 2-stage methods are as follows.

The general form of explicit 2-stage RK methods is

$$\begin{cases} k_1 = f(t_n, u_n) \\ k_2 = f(t_n + c_2 h, u_n + a_{2,1} k_1 h) \\ u_{n+1} = u_n + h(b_1 k_1 + b_2 k_2) \end{cases}$$

$$\begin{array}{c|cc} 0 & 0 & 0 \\ c_2 & a_{2,1} & 0 \\ \hline & b_1 & b_2 \end{array}$$

We use Taylor expansion

$$f(t + \alpha h, u + \beta h) = f(t, u) + h\left(\alpha f_t(t, u) + \beta f_u(t, u)\right) + \mathcal{O}(h^2)$$

to obtain

$$k_2 = \left(f + h(c_2 f_t + a_{2,1} f f_u)\right)|_{t=t_n, u=u_n} + \mathcal{O}(h^2).$$

Consequently,

$$
\begin{aligned}
u_{n+1} &= u_n + h(b_1 k_1 + b_2 k_2) \\
&= u_n + hb_1 + hb_2 \left[(f + h(c_2 f_t + a_{2,1} f f_u)) \left.\right|_{t=t_n, u=u_n} + \mathcal{O}(h^2) \right] \\
&= u_n + h(b_1 + b_2) f(t_n, u_n) + h^2 b_2 (c_2 f_t + a_{2,1} f f_u)|_{t=t_n, u=u_n} + \mathcal{O}(h^3).
\end{aligned}
$$

The local truncation error is

$$
\begin{aligned}
R_n &= u(t_{n+1}) - u_{n+1} \\
&= \left(u_n + hf + \frac{h^2}{2}(f_t + f f_u) + \mathcal{O}(h^3) \right)_{t=t_n, u=u_n} - u_{n+1} \\
&= h(1 - b_1 - b_2) f(t_n, u_n) \\
&\quad + h^2 \left[\left(\tfrac{1}{2} - b_2 c_2 \right) f_t(t_n, u_n) + \left(\tfrac{1}{2} - b_2 a_{2,1} \right) f(t_n, u_n) f_u(t_n, u_n) \right] + \mathcal{O}(h^3).
\end{aligned}
$$

In general, the scheme is not consistent: $|R_n| = \mathcal{O}(h)$, i.e., $p = 0$. To gain consistency and obtain a scheme of order $p > 0$, the following conditions should be satisfied:

- $p = 1 :\ b_1 + b_2 = 1 \ (\forall a_{2,1}, c_2)$
- $p = 2 :\ b_1 + b_2 = 1,\ c_2 = a_{2,1}$ and $b_2 a_{2,1} = 1/2$

We define $\theta := b_2$ ($\theta \neq 0$) and we obtain the coefficients of the 2-stage second-order RK schemes

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

Two popular choices are

- $\theta = 1/2$: trapezoidal rule
- $\theta = 1$: modified Euler method

Note that no choice of θ can ensure $p = 3$.

Order conditions for 3-stage RK methods

General form

$$\begin{cases} k_1 = f(t_n, u_n) \\ k_2 = f(t_n + c_2 h, u_n + a_{2,1} k_1 h) \\ k_3 = f(t_n + c_3 h, u_n + a_{3,1} k_1 h + a_{3,2} k_2 h) \\ u_{n+1} = u_n + h(b_1 k_1 + b_2 k_2 + b_3 k_3) \end{cases}$$

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ c_2 & a_{2,1} & 0 & 0 \\ c_3 & a_{3,1} & a_{3,2} & 0 \\ \hline & b_1 & b_2 & b_3 \end{array}$$

We require

$$c_i = \sum_{j=1}^{s} a_{i,j}, \quad i = 1, ..., s \quad (\text{here, } s = 3)$$

and we Taylor expand up to order $O(h^4)$.

We obtain the following conditions:

- $p = 1 :\ b_1 + b_2 + b_3 = 1$
- $p = 2 :\ b_1 + b_2 + b_3 = 1,\ b_2c_2 + b_3c_3 = \frac{1}{2}$
- $p = 3 :\ b_1 + b_2 + b_3 = 1,\ b_2c_2 + b_3c_3 = \frac{1}{2},\ b_2c_2^2 + b_3c_3^2 = \frac{1}{3},\ c_2a_{3,2}b_3 = \frac{1}{6}$

Some examples of *third-order* RK methods:

Third-order Heun scheme

$$\begin{array}{c|ccc} 0 & 0 & & \\ \frac{1}{3} & \frac{1}{3} & 0 & \\ \frac{2}{3} & 0 & \frac{2}{3} & 0 \\ \hline & \frac{1}{4} & 0 & \frac{3}{4} \end{array}$$

Third-order Kutta scheme

$$\begin{array}{c|ccc} 0 & 0 & & \\ \frac{1}{2} & \frac{1}{2} & 0 & \\ 1 & -1 & 2 & 0 \\ \hline & \frac{1}{6} & \frac{4}{3} & \frac{1}{6} \end{array}$$

Shu-Osher SSP (strong stability preserving) scheme

$$
\begin{aligned}
\tilde{u} &= u_n + hf(u_n) \\
\hat{u} &= \tfrac{3}{4}u_n + \tfrac{1}{4}\tilde{u} + \tfrac{1}{4}hf(t+h,\tilde{u}) \\
u_{n+1} &= \tfrac{1}{3}u_n + \tfrac{2}{3}\hat{u} + \tfrac{2}{3}hf(t+\tfrac{h}{2},\hat{u})
\end{aligned}
$$

$$
\begin{array}{c|ccc}
0 & 0 & & \\
1 & 1 & 0 & \\
\frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 \\
\hline
 & \frac{1}{6} & \frac{1}{6} & \frac{2}{3}
\end{array}
$$

Since the choice of parameters $a_{i,j}$ is not unique, one can set additional constraints to reduce the memory consumption, improve the stability, etc.

The schemes with $s = 4$ stages have 10 parameters that must satisfy 8 conditions to make the scheme attain its maximum possible order $p = 4$.
Some examples of fourth-order, 4-stage RK schemes:

The classical RK scheme

$$\begin{cases} k_1 = f(t_n, u_n) \\ k_2 = f(t_n + \frac{1}{2}h, u_n + \frac{1}{2}k_1 h) \\ k_3 = f(t_n + \frac{1}{2}h, u_n + \frac{1}{2}k_2 h) \\ k_4 = f(t_n + h, u_n + k_3 h) \\ u_{n+1} = u_n + h(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4) \end{cases}$$

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

3/8 rule

$$\begin{cases} k_1 = f(t_n, u_n) \\ k_2 = f(t_n + \frac{1}{3}h, u_n + \frac{1}{3}k_1 h) \\ k_3 = f(t_n + \frac{2}{3}h, u_n - \frac{1}{3}k_1 h + k_2 h) \\ k_4 = f(t_n + h, u_n + k_1 h - k_2 h + k_3 h) \\ u_{n+1} = u_n + h(\frac{1}{8}k_1 + \frac{3}{8}k_2 + \frac{3}{8}k_3 + \frac{1}{8}k_4) \end{cases}$$

$$\begin{array}{c|cccc} 0 & 0 & & & \\ \frac{1}{3} & \frac{1}{3} & 0 & & \\ \frac{2}{3} & -\frac{1}{3} & 1 & 0 & \\ 1 & 1 & -1 & 1 & 0 \\ \hline & \frac{1}{8} & \frac{3}{8} & \frac{3}{8} & \frac{1}{8} \end{array}$$

Gill's scheme requires only 3 'registries': u, k and q

$u := initial\ value, \quad k := hf(u), \quad u := u + 0.5k, \quad q := k,$

$k := hf(u), \quad u := u + (1 - \sqrt{0.5})(k - q),$

$q := (2 - \sqrt{2})k + (-2 + 3\sqrt{0.5})q,$

$k := hf(u), \quad u := u + (1 + \sqrt{0.5})(k - q),$

$q := (2 + \sqrt{2})k + (-2 - 3\sqrt{0.5})q,$

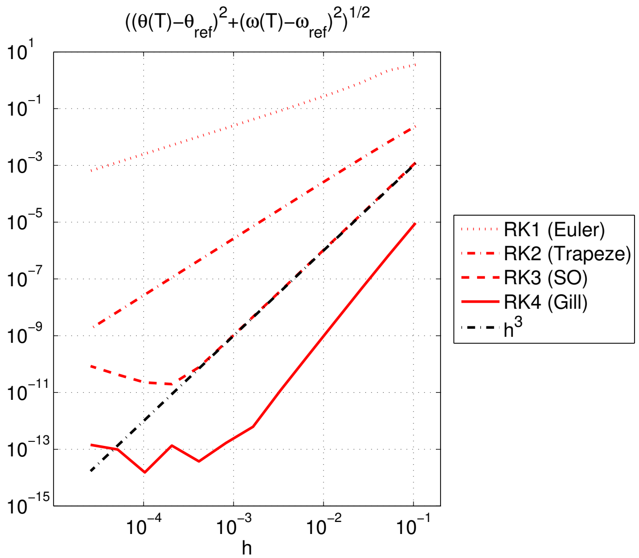
$k := hf(u), \quad u := u + k/6 - q/3, \quad \to$ next step

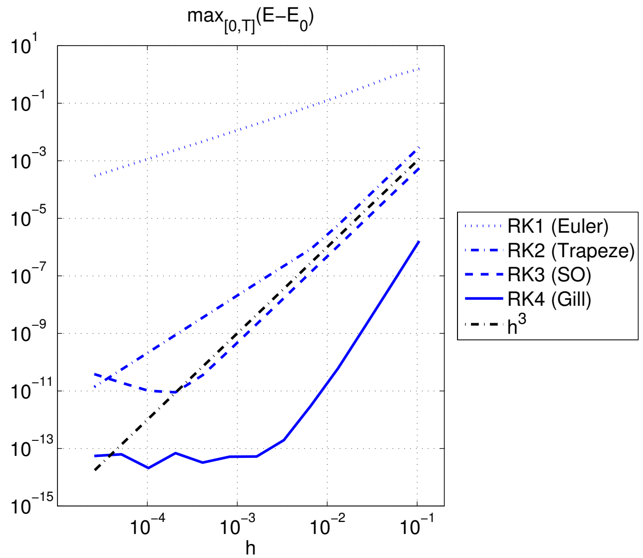
An explicit Runge–Kutta method of order p must have the number of stages greater or equal to s_{min}:

Order p	*Rank* s_{min}
$1, 2, 3, 4$	$1, 2, 3, 4 (resp.)$
5	6
6	7
7	9
8	11
9	12...17
10	13...17

For example, the parameters of a 9-th order method must satisfy 486 algebraic non-linear equations. To construct schemes of order higher than 3, some additional hypotheses are introduced to simplify the problem (see Hairer, Nørsett & Wanner, *Solving Ordinary Differential Equations I: Nonstiff problems.)*

Example. Solution de of the non-linear pendulum problem using Runge–Kutta methods. Absolute error at time $t = 4P$, where P is the period of oscillation.





Error control

We have considered so far only methods with the time step h fixed from the beginning. A value of h may be adequate in some regions and too large (or too small) in others. We can use the time step h to control the precision. There are combinations of Runge–Kutta schemes that are particularly suitable for error control.

Example: Runge-Kutta-Fehlberg method

$$
\begin{aligned}
k_1 &= f(t_n, u_n) \\
k_2 &= f(t_n + \tfrac{1}{4}h, u_n + \tfrac{1}{4}k_1 h) \\
k_3 &= f(t_n + \tfrac{3}{8}h, u_n + \tfrac{3}{32}k_1 h + \tfrac{9}{32}k_2 h) \\
k_4 &= f(t_n + \tfrac{12}{13}h, u_n + \tfrac{1932}{2197}k_1 h - \tfrac{7200}{2197}k_2 h + \tfrac{7296}{2197}k_3 h) \\
k_5 &= f(t_n + h, u_n + \tfrac{439}{216}k_1 h - 8k_2 h + \tfrac{3680}{513}k_3 h - \tfrac{845}{4104}k_4 h) \\
k_6 &= f(t_n + \tfrac{1}{2}, u_n - \tfrac{8}{27}k_1 h + 2k_2 h - \tfrac{3544}{2565}k_3 h + \tfrac{1859}{4104}k_4 h - \tfrac{11}{40}k_5 h)
\end{aligned}
$$

We construct a 4-th order method

$$u_{n+1} = u_n + h(\frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5) = u_n + h\Phi(t_n, u_n)$$

and a 5-th order method

$$\tilde{u}_{n+1} = u_n + h(\frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6) = u_n + h\tilde{\Phi}(t_n, u_n)$$

The local truncation errors are, respectively

$$u(t_{n+1}) - u_{n+1} = \mathcal{O}(h)^5 = Ah^5 + \mathcal{O}(h^6)$$

and

$$u(t_{n+1}) - \tilde{u}_{n+1} = \mathcal{O}(h^6).$$

We obtain

$$u(t_{n+1}) - u_{n+1} = (\tilde{u}_{n+1} - u_{n+1}) + \mathcal{O}(h^6).$$

We find that the error can be estimated as

$$E = (\tilde{u}_{n+1} - u_{n+1}) = h\left(\frac{1}{360}k_1 - \frac{128}{4275}k_3 - \frac{2197}{75240}k_4 + \frac{1}{50}k_5 + \frac{2}{55}k_6\right).$$

We also found earlier that $||E|| \leq Ch^5$. We can adjust the time step to ensure $Ch_{opt}^5 = tol$, where

$$tol = Atol + \max(|u_n|, |u_{n+1}|)Rtol.$$

Then the new time step is equal to

$$h_{opt} = h\left(\frac{tol}{E}\right)^{1/5}.$$

It may be useful to include limiters

$$h_{new} = h \min\left(csmax, \max\left(csmin, \left(cs\frac{tol}{E}\right)^{1/5}\right)\right).$$

with $cs = 0.8...0.9$, $csmax = 1...5$.