Mathematical modelling

Lecture 11, April 26th, 2022

Faculty of Computer and Information Science
University of Ljubljana

2021/22

Runge-Kutta methods

The idea of those methods is to approximate the derivative on the interval $[x_n, x_{n+1}]$ not only based on the derivative in the point x_n, but using a weighted average of more different derivatives on the interval $[x_n, x_{n+1}]$.

Example (Runge-Kutta of order 2 (RK2))

We approximate the derivate using the derivatives in the points x_n and $x_n + ch \in [x_n, x_{n+1}]$, where $h = x_{n+1} - x_n$ and $c \in [0, 1]$. The approximation y_{n+1} is computed using the weighted average of linear approximations in the points x_n and $x_n + ch$:

$$y_{n+1} = y_n + \underbrace{b_1}_{\text{weight}} \cdot \underbrace{(h \cdot f(x_n, y_n))}_{\substack{\text{move along} \\ \text{the tangent in } x_n}} + \underbrace{b_2}_{\text{weight}} \cdot \underbrace{(h \cdot f(x_n + ch, y(x_n + ch)))}_{\substack{\text{move along} \\ \text{the tangent in } x_n+ch}} \quad (1)$$

We use a linear approximation

$$y(x_n + ch) \approx y_n + chy'(x_n) = y_n + chf(x_n, y_n) \approx y_n + ahf(x_n, y_n), \quad (2)$$

where a is a new parameter.

Using (2) in (1) we obtain

$$y_{n+1} = y_n + b_1 \cdot \underbrace{(h \cdot f(x_n, y_n))}_{k_1} + b_2 \cdot \underbrace{(h \cdot f(x_n + ch, y_n + a \cdot k_1))}_{k_2}. \qquad (3)$$

Using Taylor series' of $y(x_n + h)$, $f(x_n + ch, y_n + ak_1)$ and comparing the coefficients at h and h^2 in (3) we get a system of equations

$$\begin{aligned} 1 &= b_1 + b_2, \\ \frac{1}{2}(f_x + f_y f)_n &= b_2 c (f_x)_n + b_2 a (f f_y)_n, \end{aligned} \qquad (4)$$

where f_n, $(f_x)_n$, $(f_y)_n$ stands for $f(x_n, y_n), f_x(x_n, y_n), f_y(x_n, y_n)$. The system (4) has many different solutions, e.g.:

- $b_1 = b_2 = \frac{1}{2}$ and $c = a = 1$. RK method is:

$$\begin{aligned} y_{n+1} &= y_n + \frac{1}{2}(k_1 + k_2), \\ k_1 &= hf(x_n, y_n), \\ k_2 &= hf(x_n + h, y_n + k_1). \end{aligned}$$

- $b_1 = 1$, $b_2 = 0$ in $c = a = \frac{1}{2}$. RK method is:

$$\begin{aligned} y_{n+1} &= y_n + k_2, \\ k_1 &= hf(x_n, y_n), \\ k_2 &= hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1). \end{aligned}$$

A general RK method is of the form

$$\begin{aligned} y_{n+1} &= y_n + b_1 k_1 + b_2 k_2 + \ldots + b_s k_s, \\ k_1 &= hf(x_n, y_n), \\ k_2 &= hf(x_n + c_2 h, y_n + a_{2,1} k_1), \\ k_3 &= hf(x_n + c_3 h, y_n + a_{3,1} k_1 + a_{3,2} k_2), \\ &\vdots, \\ k_s &= hf(x_n + c_s h, y_n + a_{s,1} k_1 + \ldots + a_{s,s-1} k_{s-1}). \end{aligned} \tag{5}$$

Butcher tableau

In a compact form the RK method (5) is given in the form of a Butcher tableau:

$$
\begin{array}{c|cccccc}
0 & 0 & & & & & \\
c_2 & a_{2,1} & 0 & & & & \\
c_3 & a_{3,1} & a_{3,2} & 0 & & & \\
\vdots & \vdots & & & & & \\
c_s & a_{s_1} & a_{s_2} & a_{s,3} & \cdots & a_{s,s-1} & 0 \\
\hline
 & b_1 & b_2 & b_3 & \cdots & b_{s-1} & b_s,
\end{array}
$$

where

$$
\begin{aligned}
c_2 &= a_{2,1}, \\
c_3 &= a_{3,1} + a_{3,2}, \\
&\vdots \\
c_s &= a_{s,1} + a_{s,2} + \ldots + a_{s,s-1}.
\end{aligned}
$$

Runge-Kutta mehod of order 4

Butcher tableau:

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

The method is

$$
\begin{aligned}
y_{n+1} &= y_n + \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4, \\
k_1 &= hf(x_n, y_n), \\
k_2 &= hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1), \\
k_3 &= hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2), \\
k_4 &= hf(x_n + h, y_n + k_3).
\end{aligned}
$$

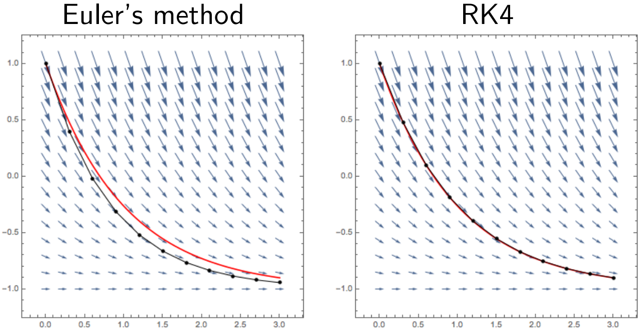
The error at each step is of order $\mathcal{O}(h^5)$. The cumulative error is of order $\mathcal{O}(h^4)$.

Euler vs RK4

Below is a comparison of Euler's and Rk4 methods for the DE

$$y' = -y - 1, \quad y(0) = 1 \quad \text{with step size } h = 0.3:$$

The red curve is the exact solution $y = 2e^{-x} - 1$.



Algorithms and example:

https://zalara.github.io/Algoritmi/euler_eng.m

https://zalara.github.io/Algoritmi/RK4_eng.m

https://zalara.github.io/Algoritmi/Euler_vs_RK4.m

Adaptive Runge Kutta methods

Let M_1, M_2 be two RK methods with the same matrices of coefficients $a_{i,j}$ (and hence also c_i), but different vectors of weights b_i and b_i^*. Let M_1 be of order p (global error $\mathcal{O}(h^p)$), while the other of order $p+1$ (global error $\mathcal{O}(h^{p+1})$).

Example: We use the adaptive method for the Butcher tableaus:

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

The first is Euler's method and has order 1, while the other is RK method of order 2:

$$y_{n+1} = y_n + k_1,$$
$$y^*_{n+1} = y_n + \frac{1}{2}(k_1 + k_2).$$

The approximation of the local error:

$$\ell_{n+1} \approx y^*_{n+1} - y_{n+1} = (-k_1 + k_2)/2.$$

If ℓ_{n+1} is small enough (we choose what this means in our problem), we accept y_{n+1} and continue, otherwise we decrease the step size and repeat the computations.

DOPRI5, Fehlberg, Cash-Karp

Very useful methods for practical computations are DOPRI5 (1980, authors Dormand in Prince), Fehlberg (1969), Cash-Karp, which are adaptive methods combining two RK methods, one of order 4 and one of order 5.

- https://en.wikipedia.org/wiki/Dormand%E2%80%93Prince_method
- https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta%E2%80%93Fehlberg_method
- https://en.wikipedia.org/wiki/Cash%E2%80%93Karp_method

Algorithm:
https://zalara.github.io/Algoritmi/DOPRI5_eng.m
https://zalara.github.io/Algoritmi/DOPRI5_example.m

Systems of first order ODE's

Let

$$f := (f_1, \dots, f_n) : \mathbb{R}^{n+1} \to \mathbb{R}^n,$$

$$f(x_1, \dots, x_{n+1}) = (f_1(x_1, \dots, x_{n+1}), \dots, f_n(x_1, \dots, x_{n+1})).$$

be a vector function. A system of first order DE's is an equation

$$\dot{x}(t) = f(x(t), t), \tag{6}$$

where

$$x(t) := (x_1(t), \dots, x_n(t)) : I \to \mathbb{R}^n$$

is an unknown vector function and $I \subset \mathbb{R}$ is some interval. Coordinate-wise the system (6) is equal to

$$\begin{aligned} \dot{x}_1(t) &= f_1(x_1(t), \dots, x_n(t), t), \\ &\vdots \\ \dot{x}_n(t) &= f_n(x_1(t), \dots, x_n(t), t). \end{aligned}$$

Solution of the system of DE's

For every $(x, t) \in \mathbb{R}^{n+1}$ in the domain of f, the value $f(x, t)$ is the tangent vector $\dot{x}(t)$ to the solution $x(t)$ at the given t.

The general solution is a family of parametric curves

$$x(t, C_1, \ldots, C_n),$$

where $C_1, C_2, \ldots, C_n \in \mathbb{R}$ are parameters, with the given tangent vectors.

An initial condition

$$x(t_0) = x_0 \in \mathbb{R}^n$$

gives a particular solution, that is, a specific parametric curve from the general solution that goes through the point x_0 at time t_0.

Linear systems of 1st order ODEs

A linear system of DEs is of the form

$$\begin{bmatrix} \dot{x}_1(t) \\ \vdots \\ \dot{x}_n(t) \end{bmatrix} = \begin{bmatrix} a_{11}(t) & \dots & a_{1n}(t) \\ \vdots & \ddots & \vdots \\ a_{n1}(t) & \dots & a_{nn}(t) \end{bmatrix} \begin{bmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{bmatrix} + \begin{bmatrix} g_1(t) \\ \vdots \\ g_n(t) \end{bmatrix}, \tag{7}$$

where

$$x_i : I \to \mathbb{R}, \quad a_{ij} : I \to \mathbb{R} \quad \text{and} \quad g_i : I \to \mathbb{R}$$

are functions of t and $I \subseteq \mathbb{R}$ is an interval. In a compact form (7) can be written as

$$\dot{x}(t) = A(t)x + g(t), \tag{8}$$

where

$$A(t) = [a_{ij}(t)]_{i,j=1}^{n}$$

is a $n \times n$ matrix function and

$$g(t) = \begin{bmatrix} g_1(t) & \dots & g_n(t) \end{bmatrix}^T$$

is a $n \times 1$ vector function.

The system (8)

- is homogeneous if for every t in the domain I we have $g(t) = \mathbf{0}$.
- has constant coefficients, if the matrix A is constant, i.e., independent of t.
- is autonomous, if it is homogeneous and has constant coefficients.

An autonomous linear system

$$\dot{x} = Ax \tag{9}$$

of 1st order DEs can be solved analytically, using methods from linear algebra. Recall that such a system can be written in coordinates as:

$$\begin{aligned}
\dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n, \\
\dot{x}_2 &= a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n, \\
&\vdots \\
\dot{x}_n &= a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n.
\end{aligned}$$

Autonomous system: diagonal matrix A

Assume first that the matrix A in (9) is diagonal. Then (9) is the following:

$$\begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} \lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}.$$

Or equivalently,

$$\dot{x}_1 = \lambda_1 x_1, \qquad \dot{x}_2 = \lambda_2 x_2, \qquad \dots \qquad , \qquad \dot{x}_n = \lambda_n x_n.$$

In this (simple) case the general solution is easily determined:

$$x(t) = \begin{bmatrix} C_1 e^{\lambda_1 t} \\ C_2 e^{\lambda_2 t} \\ \vdots \\ C_n e^{\lambda_n t} \end{bmatrix} = C_1 e^{\lambda_1 t} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + C_2 e^{\lambda_2 t} \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} + \cdots + C_n e^{\lambda_n t} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}.$$

Autonomous system: n linearly independent eigenvectors

Assume next, that A in (9) has n linearly independent eigenvectors $v_1, \dots v_n$ with the corresponding eigenvalues $\lambda_1, \dots, \lambda_n$.

- For every fixed t, the vector $x(t)$ can be expressed as a linear combination
$$x(t) = \varphi_1(t)v_1 + \cdots + \varphi_n(t)v_n.$$
- Hence, the coefficients
$$\varphi_i(t) : I \to \mathbb{R}, \qquad i = 1, \dots, n,$$
are functions of t.
- Since $v_1, \dots v_n$ are eigenvectors it follows from $\dot{x} = Ax$, that
$$\sum_{i=1}^{n} \dot{\varphi}_i(t)v_i = \sum_{i=1}^{n} \varphi_i(t)Av_i = \sum_{i=1}^{n} \varphi_i(t)\lambda_i v_i.$$
- Since $v_1, \dots v_n$ are linearly independent, it follows that for every i we have
$$\dot{\varphi}_i(t) = \lambda_i \varphi_i(t) \quad \Rightarrow \quad \varphi_i(t) = C_i e^{\lambda_i t}, \quad C_i \in \mathbb{R}.$$
- Hence the general solution of the system is
$$x(t) = C_1 e^{\lambda_1 t} v_1 + \cdots + C_n e^{\lambda_n t} v_n.$$

Example

Find the general solution of the system

$$\begin{aligned} \dot{x}_1 &= x_1 + x_2, \\ \dot{x}_2 &= 4x_1 - 2x_2. \end{aligned}$$

The matrix of the system is $A = \begin{bmatrix} 1 & 1 \\ 4 & -2 \end{bmatrix}$. Its eigenvalues are the solutions of

$$\det(A - \lambda I) = (1-\lambda)(-2-\lambda) - 4 = \lambda^2 + \lambda - 6 = 0,$$

so $\lambda_1 = -3$ and $\lambda_2 = 2$, and the corresponding eigenvectors are

$$v_1 = \begin{bmatrix} 1 & -4 \end{bmatrix}^T \qquad \text{and} \qquad v_2 = \begin{bmatrix} 1 & 1 \end{bmatrix}^T.$$

The general solution of the system is

$$x(t) = C_1 e^{-3t} \begin{bmatrix} 1 \\ -4 \end{bmatrix} + C_2 e^{2t} \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Example

Find the general solution of

$$\dot{x}_1 = x_2,$$
$$\dot{x}_2 = -4x_1.$$

The matrix of the system is $A = \begin{bmatrix} 0 & 1 \\ -4 & 0 \end{bmatrix}$. It has a conjugate pair of complex eigenvalues and a corresponding conjugate pair of eigenvectors:

$$\lambda_{1,2} = \pm 2i, \quad v_{1,2} = \begin{bmatrix} 1 & \pm 2i \end{bmatrix}^T.$$

The general solution is a family of complex valued functions

$$x(t) = C_1 e^{2it} \begin{bmatrix} 1 \\ 2i \end{bmatrix} + C_2 e^{-2it} \begin{bmatrix} 1 \\ -2i \end{bmatrix}$$

(which is not very useful in modelling real-valued phenomena).

Autonomous system: complex conjugate eigenvalues

Assume that the matrix of the system A has a complex pair of eigenvalues $\lambda_{1,2} = \alpha \pm i\beta$ and corresponding eigenvectors $v_{1,2} = u \pm iw$.

The real and imaginary parts of the two complex valued solutions are:

$$\begin{aligned} & e^{(\alpha \pm i\beta)t}(u \pm iw) \\ = \quad & e^{\alpha t}(\cos(\beta t) \pm i \sin(\beta t))(u \pm iw) \\ = \quad & e^{\alpha t}\left[\cos(\beta t)u - \sin(\beta t)w \pm i(\sin(\beta t)u + \cos(\beta t)w)\right]. \end{aligned}$$

Any linear combination (with coefficients $C_1, C_2 \in \mathbb{R}$) of these is a real-valued solution, so the real-valued general solution is

$$x(t) = e^{\alpha t}\left[C_1(\cos(\beta t)u - \sin(\beta t)w) + C_2(\sin(\beta t)u + \cos(\beta t)w)\right].$$

Autonomous system: complex conjugate eigenvalues

Example

In the case of the previous example, $\lambda_{1,2} = \pm 2i$, i.e. $\alpha = 0$ and $\beta = 2$, and

$$v_{1,2} = \begin{bmatrix} 1 \\ \pm 2i \end{bmatrix} \quad \Rightarrow \quad u = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad w = \begin{bmatrix} 0 \\ 2 \end{bmatrix}.$$

Hence, the general solution is

$$\begin{aligned} x(t) = C_1 \Big(\cos(2t) \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \sin(2t) \begin{bmatrix} 0 \\ 2 \end{bmatrix} \Big) \\ + C_2 \Big(\sin(2t) \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \cos(2t) \begin{bmatrix} 0 \\ 2 \end{bmatrix} \Big). \end{aligned}$$

Autonomous system: less than n eigenvectors

If A has less than n linearly independent eigenvectors, additional solutions can also be obtained (e.g., with the use of Jordan form of A), but we will not consider this case here.

The general solution of a system $\dot{x} = Ax$ of n equations is of the form

$$x(t) = C_1 x^{(1)}(t) + \ldots + C_n x^{(n)}(t),$$

where $x^{(1)}(t), \ldots, x^{(n)}(t)$ are specific, linearly independent solutions.

For every eigenvalue $\lambda \in \mathbb{R}$ or a pair of eigenvalues $\lambda = \alpha \pm i\beta$ we obtain as many solutions as there are corresponding linearly independent eigenvectors.

Adding initial conditions to an autonomous system

An initial condition $x(t_0) = x^{(0)}$ gives a nonsingular system (if the vectors $x_1(t_0), \ldots, x_n(t_0)$ are linearly independent) of n linear equations for the constants $C_1, \ldots, C_n$.

$$x^{(0)} = C_1 x_1(t_0) + \ldots + C_n x_n(t_0).$$

This implies that a problem

$$\dot{x} = Ax, \quad x(t_0) = x^{(0)}$$

has a unique solution for any $x^{(0)}$.

Example

The initial condition $x^{(0)} = x(0) = \begin{bmatrix} 0 & 5 \end{bmatrix}^T$ for the system in the first example above gives the following system of equations for C_1 and C_2:

$$C_1 + C_2 = 0, \quad -4C_1 + C_2 = 5,$$

so $C_1 = -1$ and $C_2 = 1$.