Runge-Kutta Methods

For Solving First Order Ordinary Differential Equations

Mohammad Nourbakhsh Marvast

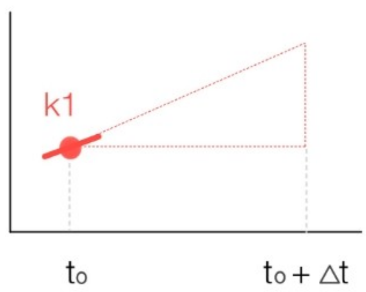
mo.nourbakhsh@Mail.sbu.ac.ir

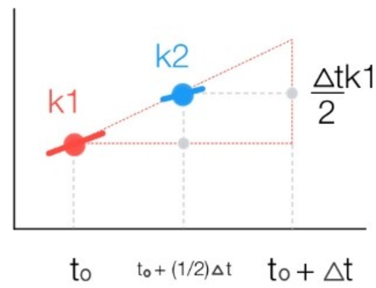
November 1, 2021

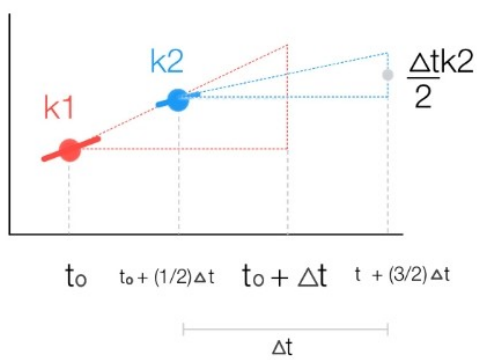
Numerical Differential Equations,
Department of Applied Mathematics,
Shahid Beheshti University

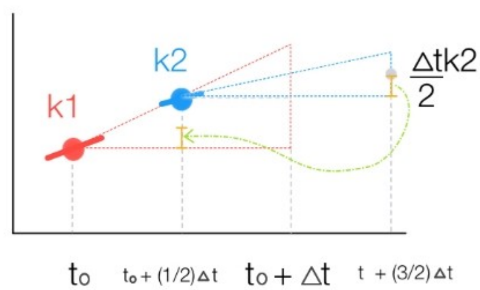
Contents

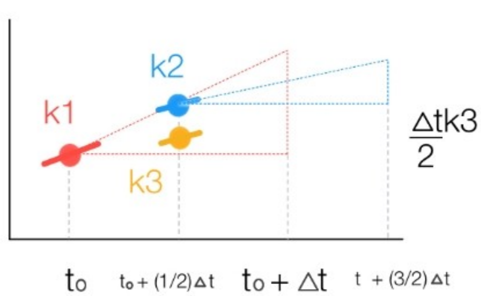
Graphical Explanation

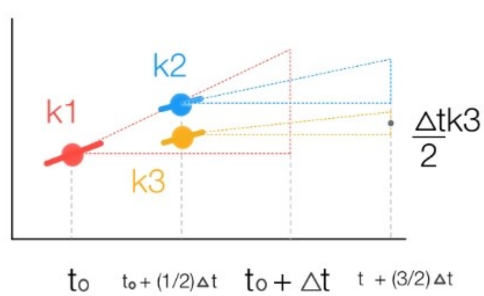


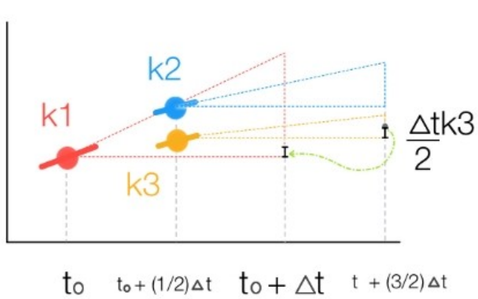


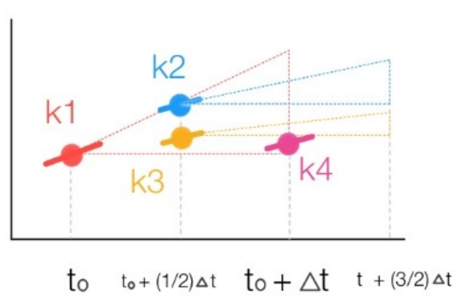












Problem

We want to find the function that be true in the condition:

Given Problem
$y' = f(t, y(t));\, y(t_0) = y_0$

How do we extend a quadrature formula for evaluating $\int f(\tau, y(\tau))\, d\tau$?

If we can extend a quadrature formula, we can calculate $y(t_{n+1})$ through:

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

Formula i

To see how we're going to do this, consider:

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

Now by replacing the Integral term in the above formula by quadrature and considering the above formula, we have:

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

But now we have a **<u>problem</u>**: What is the value of $y(t);\; t = t_n + c_j h;\; j = 1, 2, \ldots, \nu$?

We should find approximations of $y(t_n + c_j h);\; j = 1, 2, \ldots, \nu$. Let K_j , be the approximations of $y(t_n + c_j h)$.

Since the approximation should use the former steps, let $c_1 = 0$. The idea is to express each $K_j;\; j = 1, \ldots, \nu$, by the linear combination of $f(t_n + c_i h, K_i);\; i = 1, 2, \ldots, j-1$ so we have:

$$K_1 = y(t_n + c_1 h) = y_n$$
$$K_2 = y(t_n + c_2 h) = y_n + h \cdot a_{21} \cdot f(t_n, K_1)$$
$$K_3 = y(t_n + c_3 h) = y_n + h \cdot a_{31} \cdot f(t_n, K_1) + h \cdot a_{32} \cdot f(t_n + c_2 h, K_2)$$
.
.
.
$$K_\nu = y(t_n + c_\nu h) = y_n + h \cdot a_{\nu 1} \cdot f(t_n, K_1) + \ldots + h \cdot a_{\nu\, \nu-1} \cdot f(t_n + c_{\nu-1} h, K_{\nu-1}) = y_n + h \cdot \sum_{j=1}^{\nu-1} a_{\nu j} \cdot f(t_n + c_j h, K_j)$$

by these approximations we're reaching to:

$$y_{n+1} = y_n + h \cdot \sum_{j=1}^{\nu} \omega_j \cdot f\left(t_n + c_j h,\ K_j\right)$$

The matrix $A = \left(a_{j,i}\right)_{j,i=1,2,\dots,\nu}$, where the missing elements are defined to be zero, is called the RK matrix.

The vector $\omega = \left(\omega_j\right)_{j=1,2,\dots,\nu}$ is called RK weights and the vector $c = \left(c_j\right)_{j=1,2,\dots,\nu}$ is called RK nodes.

What is RK tableau or Butcher's tableau?

The tableau of the form $\begin{array}{c|c} c & A \\ \hline & \omega^T \end{array}$

in which the vector c indicates the positions, within the step, of the stage values, the matrix A indicates the dependence of the stages on the derivatives found at other stages, and ω is a vector of quadrature weights, showing how the final result depends on the derivatives, computed at the various stages.

How should we choose the RK matrix?

Notice to the Taylor series of y_{n+1} :

Formula ii: Taylor Series

$$y_{n+1}=y(t_n+h)= y_n + h\cdot f(t_n,y_n)+\frac{h^2}{2}[f_t(t_n,y_n)+f_y(t_n,y_n)\cdot f(t_n,y_n)] +O(h^3)$$

Let us consider the simplest nontrivial case: $\nu=2$

$$y_{n+1}= y_n + h\cdot\omega_1\cdot f(t_n,K_1)+h\cdot\omega_2\cdot f(t_n+c_2h,K2) \qquad (1)$$

$$K_1= y_n$$
$$K_2= y_n + h\cdot a_{21}\cdot f(t_n,K_1)= y_n + h\cdot a_{21}\cdot f(t_n,y_n)$$

$$f(t_n+c_2h,K2)=f(t_n+c_2h,\; y_n + h\cdot a_{21}\cdot f(t_n,y_n))=f(t_n,y_n)+h[c_2\frac{\partial f(t_n,y_n)}{\partial t}+a_{21}\frac{\partial f(t_n,y_n)}{\partial y_n}\cdot f(t_n,y_n)] +O(h^2)$$

Now by adding the values above to the (1) we reach at:

Formula iii: Quadrature

$$y_{n+1}=y_n+h(\omega_1+\omega_2)f(t_n,y_n)+h^2\cdot\omega_2[c_2\frac{\partial f(t_n,y_n)}{\partial t}+a_{21}\frac{\partial f(t_n,y_n)}{\partial y}f(t_n,y_n)]+ O(h^3)$$

By comparing Formula iii: Quadrature to Formula ii: Taylor Series, we get:

$$\omega_1+\omega_2=1 \qquad \omega_2c_2=\frac{1}{2} \qquad a_{21}=c_2$$

In the sense of RK tableaux:

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ \hline & 0 & 1 \end{array}, \qquad \begin{array}{c|cc} 0 & & \\ \frac{2}{3} & \frac{2}{3} & \\ \hline & \frac{1}{4} & \frac{3}{4} \end{array} \qquad \text{and} \qquad \begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}.$$

Notice that as we watch for $\nu=2$, $\sum_{j=1}^{\nu}\omega_j=1$.

Since we want our method be precise for $y'=f(x,y(x))=1$

Always we should have:

$K_1 = y_n = t_n$
$t_n + c_2 h = y(t_n + c_2 h) = K_2 = y_n + h \cdot a_{21} \cdot f(t_n, K_1) = t_n + a_{21} \cdot h \quad \Rightarrow \quad c_2 = a_{21}$
$t_n + c_3 h = y(t_n + c_3 h) = K_3 = y_n + h \cdot a_{31} \cdot f(t_n, K_1) + h \cdot a_{32} \cdot f(t_n + c_2 h, K_2) = t_n + h(a_{31} + a_{32}) \quad \Rightarrow \quad c_3 = a_{31} + a_{32}$
.
.
.
$t_n + c_\nu h = y(t_n + c_\nu h) = K_\nu = y_n + h \cdot a_{\nu 1} \cdot f(t_n, K_1) + \ldots + h \cdot a_{\nu \nu-1} \cdot f(t_n + c_{\nu-1} h, K_{\nu-1})$

$$= y_n + h \cdot \sum_{j=1}^{\nu-1} a_{\nu j} \cdot f(t_n + c_j h, K_j) = t_n + h \sum_{i=1}^{\nu-1} a_{\nu i} \quad \Rightarrow \quad \sum_{i=1}^{\nu-1} a_{\nu i} = c_\nu$$

Conclusion: $\forall\, 1 \le j \le \nu : \quad \sum_{i=1}^{j-1} a_{ji} = c_j$

Now consider the case $y' = f(y)$ and $\nu = 3$.

By finding $K_j ; \; j = 1,2,3$ and their Taylor Series, and the Taylor series of y_{n+1} , we could've found that:

$$\omega_1 + \omega_2 + \omega_3 = 1 \qquad \omega_2 c_2 + \omega_3 c_3 = \frac{1}{2} \qquad \omega_2 c_2^2 + \omega_3 c_3^2 = \frac{1}{3} \qquad \omega_3 a_{32} c_2 = \frac{1}{6}$$

What we've done here should be the necessary condition but for Orders less than or equal to 3, it's sufficient[1].

Third-order, three-stage Runge-Kutta methods are called the classical RK method.

In the sense of RK tableaux:

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ 1 & -1 & 2 & \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array} \qquad \begin{array}{c|ccc} 0 & & & \\ \frac{2}{3} & \frac{2}{3} & & \\ \frac{2}{3} & 0 & \frac{2}{3} & \\ \hline & \frac{1}{4} & \frac{3}{8} & \frac{3}{8} \end{array} .$$

1 For more information and calculations, see the page 40 of the book

Fourth order is not beyond the capabilities of a Taylor expansion, although a great deal of persistence and care (or, alternatively, a good symbolic manipulator) are required. The best-known fourth-order, four-stage RK method is

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

The derivation of higher-order ERK methods requires a substantially more advanced technique based upon graph theory.

Local Error Estimator

Let use κ for denoting the local error.

In order to estimate κ , we need to use to method, our main method of order p and a method of higher-order.

Let consider $x_n,\ y_n$ for the method of higher-order and the method of order p .

$$
\begin{aligned}
y_{n+1} &= \tilde{y}(t_{n+1}) + lh^{p+1} + O(h^{p+2}) \\
x_{n+1} &= \tilde{y}(t_{n+1}) + O(h^{p+2})
\end{aligned}
$$

$\tilde{y}$ is exact solution. By subtracting the two method above, we reach that $lh^{p+1} \approx y_{n+1} - x_{n+1}$.

So we define: $\kappa = \|y_{n+1} - x_{n+1}\|$.

Local Truncation Error

Now the question is how good are these methods, which we for now quantify by the Local Truncation Error. The LTE is the one-step error in the method assuming you have an exact solution up until that point. So for a one step method of the form $x_{n+1} = x_n + \phi(t_n, x_n)$, we define LTE as:

$$
y_{n+1} = y_n + h\phi(t_n, y_n) + h\tau_{n+1} \Rightarrow \tau_{n+1} = \frac{y_{n+1} - y_n}{h} - \phi(t_n, y_n)
$$

Examples

(Explicit) Euler's Method: $w_{n+1} = w_n + hf(t_n, w_n)$

$$\begin{aligned}
\tau_{n+1} &= \frac{y_{n+1} - y_n}{h} - f(t_n, y_n) \\
&= \frac{\cancel{y_n} + hy_n' + \frac{h^2}{2!}y_n'' + \frac{h^3}{3!}y'''(\xi) - \cancel{y_n}}{h} - f(t_n, y_n) \quad \xi \in (t_n, t_{n+1}) \\
&= \cancel{y_n'} + \frac{h}{2}y_n'' + \frac{h^2}{3!}y'''(\xi) - \cancel{f(t_n, y_n)} \\
&= O(h)
\end{aligned}$$

(Implicit) Euler's Method: $w_{n+1} = w_n + hf(t_{n+1}, w_{n+1})$

$$\begin{aligned}
\tau_{n+1} &= \frac{y_{n+1} - y_n}{h} - f(t_{n+1}, y_{n+1}) \\
&= \frac{\cancel{y_n} + hy_n' + \frac{h^2}{2!}y_n'' + \frac{h^3}{3!}y'''(\xi) - \cancel{y_n}}{h} - \underbrace{f(t_{n+1}, y_{n+1})}_{y_{n+1}'} \quad \xi \in (t_n, t_{n+1}) \\
&= y_n' + \frac{h}{2}y_n'' + \frac{h^2}{3!}y'''(\xi_1) - (y_n' + hy_n'' + \frac{h^2}{2!}y^{(3)}(\xi_2)) \\
&= O(h)
\end{aligned}$$

This quantifies our intuition earlier that Implicit and Explicit Euler should be equally good (or bad).

(Explicit) Midpoint Method: $w_{n+1} = w_n + hf\left(t_n + \frac{h}{2}, w_n + \frac{h}{2}f(t_n, w_n)\right)$

$$\begin{aligned}
\tau_{n+1} &= \frac{y_{n+1} - y_n}{h} - f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}f(t_n, y_n)\right) \\
&= y_n' + \frac{h}{2!}y_n'' + \frac{h^2}{3!}y_n^{(3)} + O(h^3) - \left[f(t_n, y_n) + \frac{\partial f}{\partial t}(t_n, y_n)\left(\frac{h}{2}\right)\right. \\
&\quad + \frac{\partial f}{\partial y}(t_n, y_n)\left(\frac{h}{2}\right)f(t_n, y_n) + \frac{\partial^2 f}{\partial t^2}(t_n, y_n)\left(\frac{h}{2}\right)^2 + \frac{\partial^2 f}{\partial t \partial y}(t_n, y_n)\left(\frac{h}{2}\right)^2 f(t_n, y_n) \\
&\quad \left. + \frac{\partial^2 f}{\partial y^2}(t_n, y_n)\left(\frac{h}{2}\right)^2 (f(t_n, y_n))^2 + O(h^3)\right]
\end{aligned}$$

$$\begin{aligned}
\tau_{n+1} = \cancel{y_n'} + \frac{h}{2!}y_n'' + \frac{h^2}{3!}y_n^{(3)} + O(h^3) - &\left[\cancel{f(t_n, y_n)} + \frac{\partial f}{\partial t}(t_n, y_n)\left(\frac{h}{2}\right)\right. \\
&+ \frac{\partial f}{\partial y}(t_n, y_n)\left(\frac{h}{2}\right)f(t_n, y_n) + \frac{\partial^2 f}{\partial t^2}(t_n, y_n)\left(\frac{h}{2}\right)^2 + \frac{\partial^2 f}{\partial t \partial y}(t_n, y_n)\left(\frac{h}{2}\right)^2 f(t_n, y_n) \\
&\left. + \frac{\partial^2 f}{\partial y^2}(t_n, y_n)\left(\frac{h}{2}\right)^2 (f(t_n, y_n))^2 + O(h^3)\right]
\end{aligned}$$

$$
\begin{aligned}
\tau_{n+1} = \cancel{\frac{h}{2!}y_n''} + \frac{h^2}{3!}y_n^{(3)} + O(h^3) - \Bigg[& \cancel{\frac{\partial f}{\partial t}(t_n, y_n)\left(\frac{h}{2}\right) + \frac{\partial f}{\partial y}(t_n, y_n)\left(\frac{h}{2}\right) f(t_n, y_n)} \\
& + \frac{\partial^2 f}{\partial t^2}(t_n, y_n)\left(\frac{h}{2}\right)^2 + \frac{\partial^2 f}{\partial t \partial y}(t_n, y_n)\left(\frac{h}{2}\right)^2 f(t_n, y_n) \\
& + \frac{\partial^2 f}{\partial y^2}(t_n, y_n)\left(\frac{h}{2}\right)^2 (f(t_n, y_n))^2 + O(h^3)\Bigg]
\end{aligned}
$$

(Implicit) Trapezoid Rule: $w_{n+1} = w_n + \frac{h}{2}[f(t_n, w_n) + f(t_{n+1}, w_{n+1})]$

$$
\begin{aligned}
\tau_{n+1} &= \frac{y_{n+1} - y_n}{h} - \frac{1}{2}\left(\underbrace{f(t_n, y_n)}_{y_n'} + \underbrace{f(t_{n+1}, y_{n+1})}_{y_{n+1}'}\right) \\
&= y_n' + \frac{h}{2}y_n'' + \frac{h^2}{3!}y_n^{(3)} + \frac{h^3}{4!}y^{(4)}(\xi_1) - \frac{1}{2}\left(y_n' + y_n' + hy_n'' + \frac{h^2}{2!}y_n^{(3)} + \frac{h^3}{3!}y^{(3)}(\xi_2)\right) \\
&= O(h^2)
\end{aligned}
$$

Here is a **4th-order RK** method

$$
\begin{aligned}
k_1 &= f(t_n, w_n) \\
k_2 &= f(t_n + \frac{h}{2}, w_n + \frac{h}{2}k_1) \\
k_3 &= f(t_n + \frac{h}{2}, w_n + \frac{h}{2}k_2) \\
k_4 &= f(t_{n+1}, w_n + hk_3) \\
w_{n+1} &= w_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}
$$

References

A First Course in the Numerical Analysis of Differential Equations, ARIEH ISERLES

Numerical Methods for Ordinary Differential Equations, J. C. Butcher

Berkeley Notes, GSI: Andrew Shi

www.haroldserrano.com