Module 10 – Ordinary Differential Equations

Lesson goals

1. Understanding the meaning of local and global truncation errors and their relationship to step size for one-step methods for solving ODEs.
2. Knowing how to implement the following Runge-Kutta (RK) methods for a single ODE:
   - Euler
   - Midpoint
   - Fourth-order RK

$$\begin{cases} y' = f(t,y) \\ y(a) = \alpha \end{cases} \qquad t \in [a,b]$$

Introduction

This module is devoted to solving initial-value ordinary differential equations of the form

$$\begin{cases} \dfrac{dy}{dt} = f\,(t, y), & t \in [a, b] \\ y(a) = \alpha & \text{initial condition} \end{cases}$$

It has been shown that if f satisfies certain conditions (such as being continuous on $D = \{(t, y) | t \in [a, b], y \in \mathbb{R}\}$, and $\frac{\partial f}{\partial y}$ satisfies a Lipschitz condition in the variable y on D) this initial-value ODE has a unique solution $y(t)$ for $t \in [a, b]$.

$F(y) \qquad |F(y) - F(z)| \leq K |y - z|$

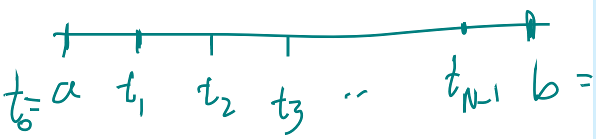
Euler's method

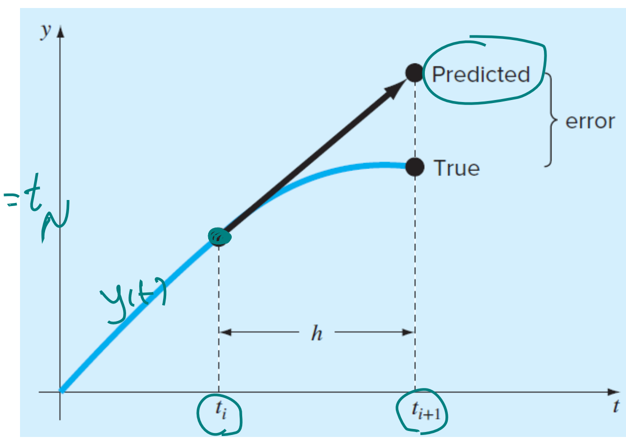
Euler's method is the most elementary approximation technique for solving initial-value problems. Although it is seldom used in practice, the simplicity of its derivation can be used to illustrate the techniques involved in the construction of some of the more advanced techniques.

Consider the mesh points that are equally distributed throughout the interval $[a, b]$.

$h = \frac{b-a}{N}$

$$t_i = a + ih, \quad i = 0,1, \dots, N, \qquad \text{and} \qquad h = \frac{b-a}{N} = t_{i+1} - t_i$$





$y' = f(t, y)$

$y(t) = y(t_i) + y'(t_i)(t - t_i) + \frac{y''(t_i)}{2}(t - t_i)^2 + \cdots$

$= y(t_i) + f(t_i, y_i)(t - t_i) + \frac{1}{2} f'(t_i, y_i)(t - t_i)^2 + \cdots$

Using Taylor series about t_i, we get

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + O(h^2)$$

$$\Rightarrow y(t_{i+1}) = y(t_i) + hf(t_i, y_i) + O(h^2)$$

Euler's method constructs $y_i \approx y(t_i)$, for each $i = 1, 2, \dots, N$, by deleting the remainder term. Thus, Euler's method is

$$\begin{cases} y_0 = \alpha = y(t_0) = y(a) \\ y_{i+1} = y_i + hf(t_i, y_i), \quad i = 0,1,2, \dots, N-1 \end{cases}$$

$y(t_{i+1}) = y(t_i) + f(t_i, y_i) h + \frac{1}{2} f'(t_i, y_i) h^2 + \cdots$

With the truncation error as $O(h^2)$. A truncation error bound for the method can be $\frac{Mh^2}{2}$, where M is an upper bound for the second derivative of y.

$y(t_{i+1}) = y(t_i) + f(t_i, y_i) h + O(h^2)$

Example. Use Euler's method to integrate

$$y' = 4e^{0.8t} - 0.5y, \quad y(0) = 2$$

from $t = 0$ to 4 with a step size of 1. Note that the exact solution can be determined analytically as

$$y = \frac{4}{1.3}\left(e^{0.8t} - e^{-0.5t}\right) + 2e^{-0.5t} \quad \checkmark$$

$h=1$,

t_i	0	1	2	3	4
$\tilde{y}_i$	2	5	11.4022	25.5132	56.8493
y_i	2	6.1946	14.8439	33.6772	75.339

$\tilde{y}(0) = y(0) = 2$

$\tilde{y}(1) = \tilde{y}(0) + h f(0, \tilde{y}(0)) = 2 + (1) f(0, 2) = 2 + (1)(3) = 5$

$\tilde{y}(2) = \tilde{y}(1) + h f(1, \tilde{y}(1)) = 5 + (1) f(1, 5) = 5 + (1)(6.4022) = 11.4022$

$\tilde{y}(3) = \tilde{y}(2) + h f(2, \tilde{y}(2)) = 11.4022 + (1) f(2, 11.4022) = 11.4022 + (1)(14.111)$
$= 25.5132$

$\tilde{y}(4) = \tilde{y}(3) + h f(3, \tilde{y}(3)) = 25.5132 + (1) f(3, 25.5132) = 25.5132 + (1)(31.3361) = 56.8493$

Stability of Euler's method. The truncation error of Euler's method depends on the step size in a predictable way based on the Taylor series. This is an *accuracy issue*.

The stability of a solution method is another important consideration that must be considered when solving ODEs. A numerical solution is said to be *unstable* if errors grow exponentially for a problem for which there is a bounded solution. Stability issue depends on three factors: the differential equation, the numerical method, and the step size.

Insight into the step size required for stability can be examined by studying a very simple ODE:

$y_1 = y_0(1-ah)$

$y_2 = y_1(1-ah) = y_0(1-ah)^2$

$y_3 = y_2(1-ah) = y_0(1-ah)^3$

$\vdots$

$y_{n+1} = y_0(1-ah)^{n+1}$

$(a>0)$

$= f(t,y)$

$$\frac{dy}{dt} = -ay, \quad y(x_0) = y_0$$

$y' = -ay \rightarrow \frac{dy}{y} = -a\,dx$

$\ln y = -ax + c$

$y = Ce^{-ax}$

$y = y_0 e^{-ax}$

The exact solution for this equation is $y = y_0 e^{-at}$. Thus, the solution starts at y_0 and asymptotically approaches zero.

Now suppose that we use Euler's method to solve the same problem numerically:

$$y_{i+1} = y_i - ay_ih = y_i(1 - ah)$$

The parenthetical quantity $1 - ah$ is called an amplification factor. If its absolute value is greater than unity, the solution will grow in an unbounded fashion. So clearly, the stability depends on the step size h. That is, if $h > \frac{2}{a}$, $|y_i| \to \infty$ as $i \to \infty$. Based on this analysis, Euler's method is said to be *conditionally stable*.

Note that there are certain ODEs where errors always grow regardless of the method. Such ODEs are called *ill-conditioned*.

Modifications on Euler's method.

A fundamental source of error in Euler's method is that the derivative at the beginning of the interval is assumed to apply across the entire interval.

Heun's method. This method uses two derivatives for the interval–one at the beginning and another at the end. These derivatives are then averaged to obtain an improved estimate of the slope for the entire interval.

The slope at the beginning of an interval

$$y_i' = f(t_i, y_i)$$

is used to make an intermediate prediction of y_{i+1} according to:

at (t_i, y_i) $\longrightarrow$

$$y^0_{i+1} = y_i + f(t_i, y_i)h \qquad \text{(predictor equation)}$$

Then, the slope at the end of the interval is given by

Slope

$$y'_{i+1} = f(t_{i+1}, y^0_{i+1})$$

Now these to slopes are combined to get an average slope for the entire interval according to

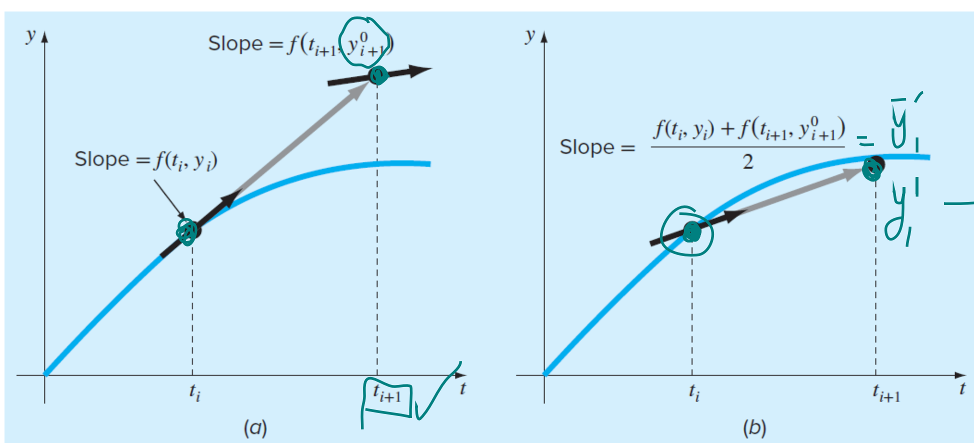
New slope $\longrightarrow$

$$\bar{y}_i' = \frac{f(t_{i+1}, y^0_{i+1}) + f(t_i, y_i)}{2}$$

Thus, the new point is obtained by

$$y_{i+1} = y_i + \frac{f(t_{i+1}, y^0_{i+1}) + f(t_i, y_i)}{2} h \qquad \text{(corrector equation)}$$

$y' = f(t, y)$



$\to f(t_1, y'_1)$

1st predict $y^0_{i+1} = y^m_i + h f(t_i, y^m_i)$

$\bar{y}'_1 = \frac{f(t_i, y^m_i) + f(t_{i+1}, y^0_{i+1})}{2}$

Current point (t_i, y^m_i)

2nd predict $y^1_{i+1} = y^m_i + \bar{y}'_1 h$

$\bar{y}'_2 = f(t_i, y^m_i) + f(t_{i+1}, y^1_{i+1})$

3rd predict: $y^{2}_{i+1} = y^{m}_{i} + \bar{y}'_{2}\,h$

⋮

Correct: $y_{i+1} = y^{m}_{i} + \bar{y}'_{m}\,h$

A termination criterion for convergence of the corrector is provided by

$y' = f(t,y) = h(t)$

$$|\varepsilon_a| = \left|\frac{y^{j}_{i+1} - y^{j-1}_{i+1}}{y^{j}_{i+1}}\right| \times 100\% \quad < \varepsilon_s$$

Note. For the case $f(t,y)$ is solely a function of independent variable t, then it is shown that the Heun's method can be obtained by the Trapezoidal rule in the integration.

Example. Use Heun's method with iteration to integrate

$$y' = 4e^{0.8t} - 0.5y, \quad y(0) = 2$$

from $t = 0$ to 4 with a step size of $h = 1$. Employ a stopping criterion of 0.001% to terminate the corrector iterations.

$\varepsilon_s = 0.00001$ $\quad |\varepsilon_a| < \varepsilon_s$ Stop predictions and introduce the corrector eqn.

t	0	1
$y(t)$	2	

y''_0

To estimate $y(1)$, we do several predictions:

1st predict: $y^{0}_{1} = y_0 + f(t_0,y_0)h = 2 + 3(1) = 5$

$\bar{y}'_1 = \dfrac{f(t_0,y_0) + f(t_1,y^{0}_1)}{2} = 4.7011$

2nd predict: $y^{1}_{1} = y_0 + \bar{y}'_1 h = 2 + (4.7011)(1) = 6.7011$ $\qquad |\varepsilon_a| = \left|\dfrac{6.7011 - 5}{5}\right| \times 100\% = 34\%$

$\bar{y}'_2 = \dfrac{f(t_0,y_0) + f(t_1,y^{1}_1)}{2} = 4.275$

3rd predict: $y^{2}_{1} = y_0 + \bar{y}'_2 h = 2 + (4.275)(1) = 6.2758$ $\qquad |\varepsilon_a| = \left|\dfrac{6.2758 - 6.7011}{6.7011}\right| \times 100\% = 6.35\%$

$\bar{y}'_3 = \dfrac{f(t_0,y_0) + f(t_1,y^{2}_1)}{2} = 4.3821$

We continue this procedure until $|\varepsilon_a| < 0.00001$. In this case an estimate for $y(1)$ will be provided using last slope and the original point.

The Midpoint Method. Another simple modification of Euler's method is called the midpoint method. A value of y is predicted at the midpoint of the interval by the Euler's method.

$y' = f(t,y) = h(t)$

$$y_{i+\frac{1}{2}} = y\left(t_i + \frac{h}{2}\right) = y_i + \frac{h}{2} f(t_i\,, y_i)$$

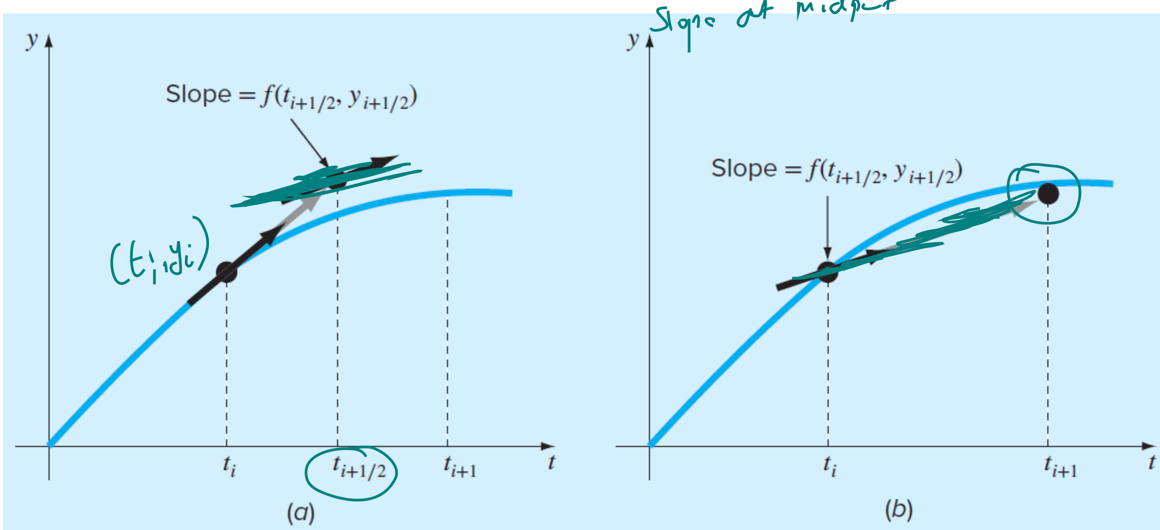
Then, this predicted value is used to calculate a slope at the midpoint:

$$y'_{i+\frac{1}{2}} = f\left(t_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}\right), \quad \text{where } t_{i+\frac{1}{2}} = t_i + \frac{h}{2} \text{ and } y_{i+\frac{1}{2}} = y_i + \frac{h}{2} f(t_i, y_i)$$

which is assumed to represent a valid approximation of the average slope for the entire interval. This slope is then used to extrapolate linearly from t_i to t_{i+1}:

$$y_{i+1} = y_i + f\left(t_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}\right) h$$

Slope at midpoint



Note. For the case $f(t, y)$ is solely a function of independent variable t, then it is shown that the midpoint method can be obtained by the Midpoint rule in the integration.

Example. Use the Midpoint method with $N = 10$, $h = 0.2$, $t_i = 0.2i$, and $y(0) = 0.5$ to approximate the solution of the following equation.

$$y' = y - t^2 + 1, \qquad 0 \le t \le 2$$

t_i	0	0.2	0.4	0.6	0.8	....
$y(t_i)$	0.5	0.828	2.7448	.	.	–

$y(0.2) \simeq$

① estimate for $y(0.1) = y(t_0) + \frac{h}{2} f(t_0, y_0) = 0.5 + \frac{0.2}{2}(1.5) = 0.65$

② slope at midpoint $= f(0.1, 0.65) = 1.64$

③ $y(0.2) = y(0) + h\, f(0.1, 0.65) = 0.5 + (0.2)(1.64) = 0.828$

$y(0.4) \simeq$

① $y(0.3) = y(0.2) + \frac{0.2}{2} f(0.2, 0.828) = 0.828 + (0.1)(1.788) = 1.0068$

② slope at midpoint $= f(0.3, 1.0068) = 1.9168$

③ $y(0.4) = y(0.2) + (h)(1.9168) = 0.828 + 1.9168 = 2.7448$

Follow the same approach to get estimations for the rest of values of t_i.

$y(t) = y(t_i) + y'(t_i)(t - t_i) + \frac{1}{2} y''(t_i)(t - t_i)^2 + \frac{1}{3!} y'''(t_i)(t - t_i)^3 + \cdots$

Taylor's method of order n

$y' = t^2 - y + 1$

$y'' = 2t - y'$

$= 2t - t^2 + y - 1$

$y''' = 2 - 2t + y'$

$= 2 - 2t + t^2 - y + 1$

Suppose the solution $y(t)$ to the initial value problem

$$y' = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha,$$

has $(n+1)$ continuous derivatives. If we expand the solution, $y(t)$, in terms of its nth Taylor polynomial about t_i and evaluate at t_{i+1}, we obtain

error term $= O(h^{n+1})$

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \frac{h^2}{2} y''(t_i) + \cdots + \frac{h^n}{n!} y^{(n)}(t_i) + \frac{h^{n+1}}{(n+1)!} y^{(n+1)}(\xi_i),$$

for some $\xi_i \in (t_i, t_{i+1})$. Successive differentiation of the solution, $y(t)$, gives

$$y'(t) = f(t, y(t)), \quad y''(t) = f'(t, y(t)), \qquad \text{and, generally,} \quad y^{(k)}(t) = f^{(k-1)}(t, y(t)).$$

Substituting these results into the previous equation gives

$$y(t_{i+1}) = \underbrace{y(t_i) + h f(t_i, y(t_i)) + \frac{h^2}{2} f'(t_i, y(t_i)) + \cdots + \frac{h^n}{n!} f^{(n-1)}(t_i, y(t_i))}_{T^{(n)}(t_i, y_i)} + \underbrace{\frac{h^{n+1}}{(n+1)!} f^{(n)}(\xi_i, y(\xi_i))}_{O(h^{n+1})}.$$

Thus,

$$y_0 = \alpha$$

$$y_{i+1} = y_i + hT^{(n)}(t_i, y_i), \quad i = 0, 1, 2, \dots, N-1$$

provides an estimate for the function values based on Taylor's method and has the truncation error of $O(h^{n+1})$. Note that

$$T^{(n)}(t_i, w_i) = f(t_i, w_i) + \frac{h}{2} f'(t_i, w_i) + \cdots + \frac{h^{n-1}}{n!} f^{(n-1)}(t_i, w_i).$$

and Euler's method is Taylor's method of order one.

Example. Apply Taylor's method of order two with $N = 10$ to the initial value problem

$$y' = \underbrace{y - t^2 + 1}_{f(t,y)}, \quad 0 \le t \le 2, \quad y(0) = 0.5.$$

t_i	0	0.2	0.4	0.6	... 0.8	...
$y(t_i)$	0.5	0.666	0.8691	1.1149	1.4103 ...	

$$T^{(2)}(t_i, y_i) = y(t_i) + h\, y'(t_i) + \frac{h^2}{2} y''(t_i)$$

$$= y(t_i) + h\left[y(t_i) - t_i^2 + 1\right] + \frac{h^2}{2}\left[y(t_i) - t_i^2 + 1 - 2t_i\right]$$

$$y'' = (y - t^2 + 1)' = y' - 2t$$

$$= (y - t^2 + 1) - 2t$$

$$y(t_{i+1}) = y(t_i) + h\, T^{(2)}(t_i, y_i) \checkmark$$

$$y' = f(t,y)$$

Runge-Kutta methods

These methods have the high-order local truncation error of the Taylor methods but eliminate the need to compute and evaluate the derivatives of $f(t, y)$.

Many variations exist but all can be cast in the following generalized form:

$$y_{i+1} = y_i + \phi h$$

where ϕ is called an *increment function*, which can be interpreted as a representative slope over the interval, and can be written in general form as

$$\phi = a_1 k_1 + a_2 k_2 + \cdots + a_n k_n$$

where the a_i's are constants and the k_i's are defined by

$$k_1 = f(t_i, y_i)$$

$$k_2 = f(t_i + p_1 h, y_i + q_{11} k_1 h)$$

$$k_3 = f(t_i + p_2 h, y_i + q_{21} k_1 h + q_{22} k_2 h)$$

$$\vdots$$

$$k_n = f\left(t_i + p_{n-1} h, y_i + q_{n-1,1} k_1 h + q_{n-1,2} k_2 h + \cdots + q_{n-1,n-1} k_{n-1} h\right)$$

where the p's and q's are constants.

- **Second-Order Runge-Kutta Methods.** This version of RK methods is given by

$$y_{i+1} = y_i + \overbrace{(a_1 k_1 + a_2 k_2)}^{\phi} h \qquad (*)$$

where

$$k_1 = f(t_i, y_i)$$

$$k_2 = f(t_i + p_1 h, y_i + q_{11} k_1 h)$$

The values for a_1, a_2, p_1, and q_{11} are evaluated by setting equation $(*)$ equal to a second-order Taylor series. By doing this, three equations can be derived to evaluate the four unknown constants:

$$a_1 = 1 - a_2, \quad p_1 = q_{11} = \frac{1}{2a_2}.$$

Because we can choose an infinite number of values for a_2, there are an infinite number of second-order RK methods. Every version would yield the same results if the solution to the ODE were quadratic, linear, or a constant.

Note that the local truncation error (the error caused by one iteration) for second-order RK method is $O(h^3)$ and the global truncation error (the cumulative error caused by many iterations) is $O(h^2)$.

Heun's and midpoint methods are specific cases of second-order RK method.

- *Heun's method:* Choose $a_2 = \frac{1}{2} = a_1$ and $p_1 = q_{11} = 1$

$$y_{i+1} = y_i + \left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right)h$$

$$k_1 = f(t_i, y_i)$$

$$k_2 = f(t_i + h, y_i + k_1 h)$$

- *Midpoint method:* Choose $a_2 = 1, a_1 = 0,$ and $p_1 = q_{11} = \frac{1}{2}$

$$y_{i+1} = y_i + k_2 h$$

$$k_1 = f(t_i, y_i)$$

$$k_2 = f\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1 h\right)$$

- **Classical Fourth-Order Runge-Kutta Method.** The most popular RK methods are fourth order. The following form is the most commonly used one, and is called the classical fourth-order RK method:

$$y_{i+1} = y_i + \left(\underset{\frac{1}{6}}{\underset{\downarrow}{a_1}} k_1 + \underset{\frac{1}{3}}{\underset{\downarrow}{a_2}} k_2 + \underset{\frac{1}{3}}{\underset{\downarrow}{a_3}} k_3 + \underset{\frac{1}{6}}{\underset{\downarrow}{a_4}} k_4\right) \rightarrow$$

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$$

$$k_1 = f(t_i, y_i)$$

$$k_2 = f\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1 h\right)$$

$$k_3 = f\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2 h\right)$$

$$k_4 = f(t_i + h, y_i + k_3 h)$$

Note that this method has the global truncation error of $O(h^4)$.

$a_2 = \frac{1}{2} \to a_1 = \frac{1}{2} \to p_1 = q_{11} = 1$

Example. Employ the second-order and the classical fourth-order RK methods to estimate the solution of the following equation at $t = 1$ using a step-size of $h = 1$.

$$y' = 4e^{0.8t} - 0.5y, \qquad y(0) = 2.$$

t_i	0	1	1
y_i	2	6.7011	6.20105
		2nd order RK	4th order RK

2nd order RK with $a_2 = a_1 = \frac{1}{2}$, $p_1 = q_{11} = 1$

$$\begin{cases} y(t_1) = y(t_0) + h\left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right) \\ k_1 = f(t_0, y_0) \longrightarrow k_1 = f(0, 2) = 3 \\ k_2 = f(t_0 + h, y_0 + k_1 h) = f(1, 2+3) = 6.4022 \end{cases}$$

$$y(1) = y(0) + (1)\left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right) = 2 + (1.5 + 3.2011) = 6.7011$$

4th order RK:

$$y(t_1) = y(t_0) + h\left(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4\right)$$

$$\begin{cases} k_1 = f(t_0, y_0) = f(0,2) = 3 \\ k_2 = f(t_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_1 h) = f(0.5, 3.5) = 4.2173 \\ k_3 = f(t_0 + \frac{1}{2}h, y_0 + \frac{1}{2}k_2 h) = f(0.5, 2 + 2.10865) = 3.913 \\ k_4 = f(t_0 + h, y_0 + k_3 h) = f(1, 5.913) = 5.9457 \end{cases}$$

$$y(1) = 2 + (1)\left(\frac{1}{2} + \frac{4.2173}{3} + \frac{3.913}{3} + \frac{5.9457}{6}\right) = 6.20105$$

Systems of Equations

Many practical problems in engineering and science require the solution of a system of simultaneous ordinary differential equations rather than a single equation. Such systems may be represented generally as

$$\begin{cases} \dfrac{dy_1}{dt} = f_1(t, y_1, y_2, \ldots, y_n) \\ \dfrac{dy_2}{dt} = f_2(t, y_1, y_2, \ldots, y_n) \\ \quad\vdots \\ \dfrac{dy_n}{dt} = f_n(t, y_1, y_2, \ldots, y_n) \end{cases}$$

$$\begin{cases} y' = t x e^{y} & f_1(t,x,y) \\ x' = x^2 + y + t x^3 & f_2(t,x,y) \end{cases}$$

$x(t)$ $\quad$ $y(t)$

The solution of such a system requires that n initial conditions be known at the starting value of t.

All the methods discussed in this chapter for single equations can be extended to systems of ODEs. Engineering applications can involve thousands of simultaneous equations. In each case, the procedure for solving a system of equations simply involves applying the one-step technique for every equation at each step before proceeding to the next step.

$$\begin{cases} x(t_0) \\ y(t_0) \end{cases} \longrightarrow \begin{cases} x(t_1) \\ y(t_1) \end{cases} \longmapsto \begin{cases} x(t_2) \\ y(t_2) \end{cases} \longmapsto \cdots$$

Example. Transform the second-order initial-value problem

$$y'' - 2y' + 2y = e^{2t}\sin t, \quad 0 \le t \le 1, \quad \text{with } y(0) = -0.4 \text{ and } y'(0) = -0.6$$

into a system of first order initial-value problems, and use the Euler's method with $h = 0.5$ to approximate the solution at $t = 1$.

Note. This equation can be written as a system of ODEs using $u_1(t) = y(t)$ and $u_2(t) = y'(t)$. Then, the system will be of the form:

$$\begin{cases} \dfrac{du_1}{dt} = u_2 = f_1(t, u_1, u_2) \\ \dfrac{du_2}{dt} = e^{2t}\sin t + 2u_2 - 2u_1 = f_2(t, u_1, u_2) \end{cases}$$

With initial conditions $u_1(0) = -0.4$ and $u_2(0) = -0.6$.

$f_2(0, -0.4, -0.6)$

t_i	0	0.5	1
$y(t_i) = u_1(t_i)$	-0.4	-0.7	-1.1
$u_2(t_i)$	-0.6	-0.8	-0.2484

$u_1(0.5) \simeq ?$

$u_2(0.5) \simeq ?$

$$u_1(0.5) = u_1(0) + h f_1(0, u_1(0), u_2(0)) = -0.4 + 0.5 f_1(0, -0.4, -0.6)$$
$$= -0.4 + (0.5)(-0.6) = -0.7$$

$$u_2(0.5) = u_2(0) + h f_2(0, u_1(0), u_2(0))$$
$$= -0.6 + (0.5) f_2(0, -0.4, -0.6) = -0.6 + (0.5)[-0.4] = -0.8$$

$u_1(1) \simeq$

$u_2(1) \simeq$

$$u_1(1) = u_1(0.5) + h f_1(0.5, u_1(0.5), u_2(0.5)) = -0.7 + (0.5) f_1(0.5, -0.7, -0.8)$$
$$= -0.7 + (0.5)(-0.8) = -1.1$$

$$u_2(1) = u_2(0.5) + h f_2(0.5, u_1(0.5), u_2(0.5)) = -0.8 + (0.5) f_2(0.5, -0.7, -0.8)$$
$$= -0.8 + (0.5)(1.1032) = -0.2484$$

References

1. Chapra, Steven C. (2018). *Numerical Methods with* MATLAB *for Engineers and Scientists*, 4th Ed. McGraw Hill.
2. Burden, Richard L., Faires, J. Douglas (2011). *Numerical Analysis*, 9th Ed. Brooks/Cole Cengage Learning