Initial Value Problem.
Math 353 Section 12, Fall 2008, University of Delaware

Let consider a first order initial value ordinary differential equation

$$y' = f(t, y), \quad y(a) = y_a, \ \ a \le t \le b. \tag{1}$$

One step constant step size methods.

We want to find approximations to the solution of (1) at $t_k = a + (k-1)h,\ (k = 1, 2, ..., m+1)$, where $h = (b-a)/m$. Numerical methods will produce a discrete solution of (1) $y = (y_0, y_1, ..., y_m)^\top$ such that $y_0 = y_a$ and $y_k \approx y(t_k),\ k = 1, 2, ..., m$.

Euler Method (explicit).

$$y_0 = y_a, \ \ y_{k+1} = y_k + hf(t_k, y_k), \ \ k = 0, 1, ..., m-1.$$

with a global error bound

$$g_k = |y(t_k) - y_k| \le \frac{Mh}{2L}(e^{L(t_k - a)} - 1) = O(h)$$

where L is a Lipschitz constant of $f(t, y)$ in the variable y and $M = max_{a \le c \le b}|y''(c)|$.

Euler Method (implicit).

$$y_0 = y_a, \ \ y_{k+1} = y_k + hf(t_{k+1}, y_{k+1}), \ \ k = 0, 1, ..., m-1.$$

Taylor method of order n.

$$y_0 = y_a, \ \ y_{k+1} = y_k + hf(t_k, y_k) + \frac{h^2}{2} f'(t_k, y_k) + \cdots + \frac{h^n}{n!} f^{(n-1)}(t_k, y_k), \ \ k = 0, 1, ..., m-1.$$

Here, $f^{(i)}(t, y),\ i = 1, 2, ..., n-1$, denotes the i-th total derivative of $f(t, y(t))$ with respect to t.

Theorem. If Taylor method of order n is used to approximate the solution of (1) with step size h and if $y \in C^{n+1}[a, b]$, then

$$e_k = O(h^{n+1}), \quad g_k = |y(t_k) - y_k| = O(h^n), \quad k = 1, 2, ..., m.$$

Runge-Kutta method of order 2 (midpoint method).

$$y_0 = y_a, \ \ y_{k+1} = y_k + hf\left(t_k + \frac{h}{2}, y_k + \frac{h}{2} f(t_k, y_k)\right), \ \ k = 0, 1, ..., m-1.$$

Runge-Kutta method of order 2 (explicit trapezoid).

$$y_0 = y_a, \ \ y_{k+1} = y_k + \frac{h}{2}\Big(f(t_k, y_k) + f(t_k + h, y_k + hf(t_k, y_k))\Big), \ \ k = 0, 1, ..., m-1.$$

Runge-Kutta method of order 4 (RK4).

$$y_0 = y_a, \ \ y_{k+1} = y_k + \frac{h}{6}(f_1 + 2f_2 + 2f_3 + f_4), \ \ k = 0, 1, ..., m-1,$$

where

$$f_1 = f(t_k, y_k), \quad f_2 = f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2} f_1\right),$$

$$f_3 = f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2} f_2\right), \ \ f_4 = f(t_k + h, y_k + hf_3).$$

Variable step size methods

Bogacki-Shampine order2/order3 embedded pair (ode23-MatLab command).

$$y_0 = y_a, \;\; y_{k+1} = y_k + \frac{h}{24}(7f_1 + 6f_2 + 8f_3 + 3f_4)$$

and

$$z_0 = y_a, \;\; z_{k+1} = z_k + \frac{h}{9}(2f_1 + 3f_2 + 4f_3)$$

where

$$f_1 = f(t_k, y_k), \quad f_2 = f\left(t_k + \frac{h}{2}, y_k + \frac{h}{2}f_1\right),$$

$$f_3 = f\left(t_k + \frac{3h}{4}, y_k + \frac{3h}{4}f_2\right), \;\; f_4 = f(t_k + h, z_{k+1}).$$

The error is for step-size control is

$$e_k = |z_{k+1} - y_{k+1}| = \frac{h}{72}| - 5f_1 + 6f_2 + 8f_3 - 9f_4|.$$

Dormand-Prince order4/order5 embedded pair (ode45-MatLab Command).

$$y_0 = y_a, \;\; y_{k+1} = y_k + h\left(\frac{5179}{57600}f_1 + \frac{7571}{16695}f_3 + \frac{393}{640}f_4 - \frac{92097}{339200}f_5 + \frac{187}{2100}f_6 + \frac{1}{40}f_7\right)$$

and

$$z_0 = y_a, \;\; z_{k+1} = z_k + h\left(\frac{35}{384}f_1 + \frac{500}{1113}f_3 + \frac{125}{192}f_4 - \frac{2187}{6784}f_5 + \frac{11}{84}f_6\right)$$

where

$$f_1 = f(t_k, y_k), \quad f_2 = f\left(t_k + \frac{h}{5}, y_k + \frac{h}{5}f_1\right),$$

$$f_3 = f\left(t_k + \frac{3h}{10}, y_k + \frac{3h}{40}f_1 + \frac{9h}{40}f_2\right), \;\; f_4 = f\left(t_k + \frac{4h}{5}, y_k + \frac{44h}{45}f_1 - \frac{56h}{15}f_2 + \frac{32h}{9}f_3\right),$$

$$f_5 = f\left(t_k + \frac{8h}{9}, y_k + h\left(\frac{19372}{6561}f_1 - \frac{25360}{2187}f_2 + \frac{64448}{6561}f_3 - \frac{212}{729}f_4\right)\right),$$

$$f_6 = f\left(t_k + h, y_k + h\left(\frac{9017}{3168}f_1 - \frac{355}{33}f_2 + \frac{46732}{5247}f_3 - \frac{49}{176}f_4 - \frac{5103}{18656}f_5\right)\right), \; f_7 = f(t_k + h, z_{k+1}).$$

Multi step methods

General formula for linear s step method is

$$y_{k+1} = \sum_{i=0}^{s} a_i y_{k-i} + h\sum_{j=-1}^{s} b_j f(t_{k-j}, y_{k-j}).$$

If $b_{-1} = 0$, then it is called explicit (forward) multi step method.
If $b_{-1} \neq 0$, then it is called implicit (backward) multi step method.
Adams-Bashforth two step method (second-order).

$$y_{k+1} = y_k + h\left(\frac{3}{2}f(t_k, y_k) - \frac{1}{2}f(t_{k-1}, y_{k-1})\right), \;\; k = 1, 2, ..., m-1.$$

Adams-Bashforth three step method (third-order).

$$y_{k+1} = y_k + \frac{h}{12}(23f(t_k, y_k) - 16f(t_{k-1}, y_{k-1}) + 5f(t_{k-2}, y_{k-2})), \quad k = 2, 3, ..., m-1.$$

Implicit trapezoid method (second-order).

$$y_{k+1} = y_k + \frac{h}{2}(f(t_k, y_k) + f(t_{k+1}, y_{k+1})), \quad k = 0, 1, ..., m-1.$$

Adams-Moulton two step method (third-order).

$$y_{k+1} = y_k + \frac{h}{12}(5f(t_{k+1}, y_{k+1}) + 8f(t_k, y_k) - f(t_{k-1}, y_{k-1})), \quad k = 1, 2, ..., m-1.$$

System of first order ordinary differential equations.

$$\begin{array}{c} y_1' = f_1(t, y_1, ..., y_n), \quad y_1(a) = y_{10} \\ \cdots \quad \cdots \\ y_n' = f_n(t, y_1, ..., y_n), \quad y_n(a) = y_{n0} \\ a \le t \le b. \end{array} \tag{2}$$

Let define $y(t) = (y_1(t), ..., y_n(t))^\top \ : \mathbf{R} \ \mapsto \ \mathbf{R}^n$ a vector valued function and $F(t, y_1, ..., y_n) = (f_1(t, y), ..., f_n(t, y)) \ : \ \mathbf{R}^{n+1} \ \mapsto \ \mathbf{R}^n$ a vector function, then (2) can be represented in a vector form

$$\frac{d}{dt}(y(t)) = F(t, y), \quad y(a) = (y_{10}, ..., y_{n0})^\top, \quad a \le t \le b.$$

Methods for a single differential equations can be extended for system of differential equations in a vector form.

Euler method (vector form).

$$y_0 = (y_{10}, ..., y_{n0})^\top, \quad \text{for } k = 0, 1, 2, ..., m-1 \text{ (number of the nodes)} \ :$$

$$y_{ik+1} = y_{ik} + hf_i(t_k, y_{1k}, ..., y_{nk}), \quad i = 1, 2, ...n \text{ (number of the equations)},$$

where $y_{ik} \approx y_i(t_k), \ \ i = 1, 2, ..., n, \ \ k = 0, 1, ..., m.$

Runge-Kutta method of order 4 for system of 2 differential equations (vector form).

$$y_0 = (y_{10}, y_{20})^\top, \quad \text{for } k = 0, 1, ..., m-1 \ :$$

$$y_{1k+1} = y_{1k} + \frac{h}{6}(f_{11} + 2f_{21} + 2f_{31} + f_{41}), \quad y_{2k+1} = y_{2k} + \frac{h}{6}(f_{12} + 2f_{22} + 2f_{32} + f_{42})$$

where

$$f_{11} = f_1(t_k, y_{1k}, y_{2k}), \quad f_{11} = f_2(t_k, y_{1k}, y_{2k}),$$

$$f_{21} = f_1\left(t_k + \frac{h}{2}, y_{1k} + \frac{h}{2}f_{11}, y_{2k} + \frac{h}{2}f_{12}\right), \quad f_{22} = f_2\left(t_k + \frac{h}{2}, y_{1k} + \frac{h}{2}f_{11}, y_{2k} + \frac{h}{2}f_{12}\right),$$

$$f_{31} = f_1\left(t_k + \frac{h}{2}, y_{1k} + \frac{h}{2}f_{21}, y_{2k} + \frac{h}{2}f_{22}\right), \quad f_{32} = f_2\left(t_k + \frac{h}{2}, y_{1k} + \frac{h}{2}f_{21}, y_{2k} + \frac{h}{2}f_{22}\right),$$

$$f_{41} = f_1(t_k + h, y_{1k} + hf_{31}, y_{2k} + hf_{32}), \quad f_{42} = f_2(t_k + h, y_{1k} + hf_{31}, y_{2k} + hf_{32}).$$

Higher order equations

Let consider n-th order a single ordinary differential equation

$$y^{(n)} = f(t, y, y', ..., y^{(n-1)}), \quad y(a) = y_{0a}, ..., y^{(n-1)}(a) = y_{n-1a}, \quad a \le t \le b. \tag{3}$$

If we define new variables

$$y_1 = y,\ y_2 = y',\ y_3 = y'', ..., y_n = y^{(n-1)},$$

then (3) will be converted to the following system of differential equations

$y_1' = y_2, \quad y_1(a) = y_{0a}$
$y_2' = y_3, \quad y_2(a) = y_{1a}$
$\cdots \quad \cdots$
$y_{n-1}' = y_n, \quad y_{n-1}(a) = y_{n-2a}$
$y_n' = f(t, y_1, ..., y_{n-1}), \quad y_n(a) = y_{n-1a}, \quad a \le t \le b.$