MATH2089
Numerical Methods
Lecture 10

Runge Kutta Method

Predictor-Corrector Methods

Runge-Kutta Method

- The Runge-Kutta methods do not require the calculation of higher derivation, but produce results equivalent in accuracy to the higher order Taylor formulas.

$$y_{i+1} = y_i + h\underbrace{\alpha(x_i, y_i, h)}_{\text{increment function}};$$

- The increment function is chosen to represent the average slope over the interval, $x_i \leq x \leq x_{i+1}$

$$\alpha(x_i, y_i, h) = c_1 k_1 + c_2 k_2 + \cdots + c_n k_n;$$

- n denotes the order of the Runge-Kutta method

Runge-Kutta Method (continue)

- $c_1, c_2, \ldots, c_n$ are constants and $k_1, k_2, \ldots, k_n$ are recurrence relations given by

$$
\begin{aligned}
k_1 &= f(x_i, y_i),\\
k_2 &= f(x_i + p_2 h, y_i + a_{21} h k_1),\\
k_3 &= f(x_i + p_3 h, y_i + a_{31} h k_1 + a_{32} h k_2)\\
&\vdots\\
k_n &= f(x_i + p_n h, y_i + a_{n1} h k_1 + a_{n2} h k_2 + \cdots + a_{n,n-1} h k_{n-1})
\end{aligned}
$$

- The compact form of the Runge-Kutta method:

$$y_{i+1} = y_i + h \sum_{j=1}^{n} c_j k_j$$

where $$k_j = f\left(x_i + p_j h, y_i + \sum_{l=1}^{j-1} a_{jl} h k_l\right)$$

Runge-Kutta Method (continue)

- ***First-order Runge-Kutta Method*** *(n = 1)*

$$y_{i+1} = y_i + hc_1k_1 = y_i + hc_1 f(x_i, y_i)$$

- With $c_1 = 1$ the first order Runge-Kutta method is the same as Euler's method.

- ***Second-order Runge-Kutta Method*** *(n = 2)*

$$y_{i+1} = y_i + hc_1k_1 + hc_2k_2 = y_i + hc_1 f(x_i, y_i) + hc_2 f(x_i + p_2h, y_i + a_{21}hk_1) \quad \text{(A1)}$$

Runge-Kutta Method (continue)

- Unknowns : c_1, c_2, p_2, a_{21}
- Step (1) By Taylor's series expansion:

$$y_{i+1} = y_i + h f(y_i, x_i) + \frac{h^2}{2!} f'(y_i, x_i) + O(h^3)$$

- With the chain rule: $f' = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f = f_x + f_y f$

$$y_{i+1} = y_i + h f_i + \frac{h^2}{2!}\left(f_{x_i} + f_{y_i} f_i \right) + O(h^3) \qquad \text{(B)}$$

- Step (2) Use Taylor's series expansion for function of two variables:

$$\overbrace{f(x_i + p_2 h, y_i + a_{21} h \underbrace{k_1}_{=f_i})}^{k_2} = f_i + p_2 h f_{x_i} + a_{21} h f_{y_i} f_i + O(h^2) \qquad \text{(A2)}$$

Runge-Kutta Method (continue)

- Substituting Eq. (A2) into (A1) and rearranging the equation yields

$$y_{i+1} = y_i + h f_i (c_1 + c_2) + h^2 f_{x_i} (c_2 p_2) + h^2 f_{y_i} f_i (c_2 a_{21}) + O(h^3) \qquad \text{(A3)}$$

- The coefficients of the corresponding terms from Eq. (B) and (A3) must be the same:

$$c_1 + c_2 = 1, \quad c_2 p_2 = \frac{1}{2} \quad \text{and} \quad c_2 a_{21} = \frac{1}{2}$$

- 4 unknowns and 3 equations → one degree of freedom
- Choose one parameter and determine the others. Let choose c_2 to be specified:

Runge-Kutta Method (continue)

$$c_2 = 1 \rightarrow c_1 = 0,\ p_2 = \frac{1}{2},\ a_{21} = \frac{1}{2} \Rightarrow \qquad \begin{aligned} y_{i+1} &= y_i + hk_2 \\ k_1 &= f(x_i, y_i) \\ k_2 &= f(x_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}hk_1) \end{aligned}$$

The same as modified Euler's method !

$$c_2 = \frac{1}{2} \rightarrow c_1 = \frac{1}{2},\ p_2 = 1,\ a_{21} = 1 \Rightarrow \qquad \begin{aligned} y_{i+1} &= y_i + \frac{h}{2}(k_1 + k_2) \\ k_1 &= f(x_i, y_i) \\ k_2 &= f(x_i + h, y_i + hk_1) \end{aligned}$$

The same as Heun's method!

Runge-Kutta Method (continue)

Third-order Runge-Kutta Method ***(n = 3)***

- The derivation is similar to the one described for the second-order method. 8 unknowns and 6 equations → two degree of freedom. Choose two parameters and determine the others.

$$
\begin{aligned}
y_{i+1} &= y_i + \frac{h}{6}\left(k_1 + 4k_2 + k_3\right) \\
k_1 &= f(x_i, y_i) \\
k_2 &= f(x_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}hk_1) \\
k_3 &= f(x_i + h, y_i - hk_1 + 2hk_2)
\end{aligned}
$$

- If $y' = f(x)$, the third order Runge-Kutta method is reduced to Simpson's 1/3 rule which yields exact results when $f(x)$ is a cubic equation.

Runge-Kutta Method (continue)

Fourth-order Runge-Kutta Method ***(n = 4)***

- The fourth order Runge-Kutta methods are most popular with the global truncation is $O(h^4)$.

Method Attributed to Runge

$$y_{i+1} = y_i + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right)$$
$$k_1 = f(x_i, y_i)$$
$$k_2 = f(x_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}hk_1)$$
$$k_3 = f(x_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}hk_2)$$
$$k_4 = f(x_i + h, y_i + hk_3)$$

Method Attributed to Kutta

$$y_{i+1} = y_i + \frac{h}{8}\left(k_1 + 3k_2 + 3k_3 + k_4\right)$$
$$k_1 = f(x_i, y_i)$$
$$k_2 = f(x_i + \tfrac{1}{3}h, y_i + \tfrac{1}{3}hk_1)$$
$$k_3 = f(x_i + \tfrac{2}{3}h, y_i - \tfrac{1}{3}hk_1 + hk_2)$$
$$k_4 = f(x_i + h, y_i + hk_1 - hk_2 + hk_3)$$

Example

- Find the solution of the problem $y' = y + 2x - 1$ with $y(x = 0) = 1$ over the interval $0 \le x \le 1$ using the fourth order Runge-Kutta method (attributed to Runge)

Solution:

$$h = 0.1;\ x_0 = 0,\ x_1 = 0.1,\ x_2 = 0.2,\ x_3 = 0.3,\ \ldots,\ x_{10} = 1$$

Start with $y_0 = 1$

$$k_1 = f(x_0, y_0) = y_0 + 2x_0 - 1 \quad \Rightarrow \quad k_1 = f(0,1) = 1 + 2(0) - 1 = 0$$

$$k_2 = f(x_0 + \tfrac{1}{2}h, y_0 + \tfrac{1}{2}hk_1) = f(0 + \tfrac{1}{2}(0.1), 1 + \tfrac{1}{2}(0.1)(0)) = f(0.05, 1) = 1 + 2(0.05) - 1 = 0.1$$

$$k_3 = f(x_0 + \tfrac{1}{2}h, y_0 + \tfrac{1}{2}hk_2) = f(0 + \tfrac{1}{2}(0.1), 1 + \tfrac{1}{2}(0.1)(0.1)) = f(0.05, 1.005) = 0.105$$

$$k_4 = f(x_0 + h, y_0 + hk_3) = f(0 + 0.1, 1 + 0.1(0.105)) = f(0.1, 1.0105) = 0.2105$$

$$y_1 = y_0 + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \quad \Rightarrow \quad \underline{y_1 = y(0.1) = 1 + \frac{h}{6}(0 + 2(0.1 + 0.105) + 0.2105) = 1.0103}$$

Example (continue)

$$k_1 = f(x_1, y_1) = y_1 + 2x_1 - 1 \quad \Rightarrow \quad k_1 = f(0.1, 1.013) = 1.013 + 2(0.1) - 1 = 0.2103$$

$$k_2 = f(x_1 + \tfrac{1}{2}h, y_1 + \tfrac{1}{2}hk_1) = f(0.1 + \tfrac{1}{2}(0.1), 1.013 + \tfrac{1}{2}(0.1)(2.103)) = f(0.15, 1.0208) = 0.32082$$

$$k_3 = f(x_1 + \tfrac{1}{2}h, y_1 + \tfrac{1}{2}hk_2) = f(0.1 + \tfrac{1}{2}(0.1), 1.013 + \tfrac{1}{2}(0.1)(0.32082)) = f(0.15, 1.0263) = 0.32634$$

$$k_4 = f(x_1 + h, y_1 + hk_3) = f(0.1 + 0.1, 1.013 + 0.1(0.32634)) = f(0.2, 1.0546) = 0.44293$$

$$\underline{y_2 = y(0.2) = 1 + \frac{h}{6}(0.2103 + 2(0.32082 + 0.32634) + 0.44293) = 1.0428}$$

- Subsequently calculate y_3, y_4, y_5, …, y_9, y_{10}

Example (continue)

- Compare with exact solution

x	y	k_1	k_2	k_3	k_4	exact	% error
0	**1**	0	0.1	0.105	0.2105	1	0
0.1	**1.0103**	0.21034	0.32086	0.32638	0.44298	1.0103	1.6775E-05
0.2	**1.0428**	0.44281	0.56495	0.57105	0.69991	1.0428	3.5924E-05
0.3	**1.0997**	0.69972	0.83470	0.84145	0.98386	1.0997	5.6471E-05
0.4	**1.1836**	0.98365	1.1328	1.1403	1.2977	1.1836	7.7314E-05
0.5	**1.2974**	1.2974	1.4623	1.4706	1.6445	1.2974	9.7438E-05
0.6	**1.4442**	1.6442	1.8264	1.8356	2.0278	1.4442	1.1609E-04
0.7	**1.6275**	2.0275	2.2289	2.2389	2.4514	1.6275	1.3283E-04
0.8	**1.8511**	2.4511	2.6736	2.6848	2.9196	1.8511	1.4750E-04
0.9	**2.1192**	2.9192	3.1652	3.1775	3.4369	2.1192	1.6019E-04
1	**2.4366**					2.4366	1.7109E-04

Multistep Methods

- All the previous methods (Euler, modified Euler, Huen and Runge-Kutta methods) are called single-step methods because they use only the information from the last step computed

- Single-step methods – require information at a single point to find .
- Multi-step methods – require information at more than one point to find .
- Utilize the past values of *y* and/or to construct a polynomial that approximates the derivative function, and extrapolate this into the next interval.

Adams-Bashforth Open (or Explicit) Formula

- Using Taylor's series expansion of *y* around x_i to express y_{i+1}

$$y_{i+1} = y_i + h f_i + \frac{h^2}{2!} f_i' + \frac{h^3}{3!} f_i'' + \frac{h^4}{4!} f_i''' + \cdots$$

- By using the backward different formula,

$$f_i' = \frac{f_i - f_{i-1}}{h} + \frac{h}{2} f_i'' + O(h^2)$$

for the 2nd -order method

$$y_{i+1} = y_i + h\left(\frac{3}{2} f_i - \frac{1}{2} f_{i-1}\right) + \frac{5}{12} h^3 f_i'' + O(h^4)$$

Adams-Bashforth Open (or Explicit) Formula (continue)

- By using ,

$$f_i' = \frac{3f_i - 4f_{i-1} + f_{i-2}}{2h} + O(h^2)$$

$$f_i'' = \frac{f_i - 2f_{i-1} + f_{i-2}}{h^2} + O(h)$$

for the 3d -order method

$$y_{i+1} = y_i + h\left(\frac{23}{12}f_i - \frac{16}{12}f_{i-1} + \frac{5}{12}f_{i-2}\right) + O(h^4)$$

- The general expression for the *n*th-order Adams-Bashforth open formula

$$y_{i+1} = y_i + h\sum_{k=1}^{n}\alpha_{nk}\, f_{i-k+1} + O(h^{n+1})$$

Adams-Bashforth Open (or Explicit) Formula (continue)

Coefficients and local truncation error for Adams-Bashforth Open Formulas

Order of the formula (n)	$k=1$ α_{n1}	$k=2$ α_{n2}	$k=3$ α_{n3}	$k=4$ α_{n4}	$k=5$ α_{n5}	$k=6$ α_{n6}	Local truncation error, $O(h^{n+1})$
1	1						$\frac{1}{2}h^2 f'(\xi)$
2	$\frac{3}{2}$	$-\frac{1}{2}$					$\frac{5}{12}h^3 f''(\xi)$
3	$\frac{23}{12}$	$-\frac{16}{12}$	$\frac{5}{12}$				$\frac{9}{24}h^4 f'''(\xi)$
4	$\frac{55}{24}$	$-\frac{59}{24}$	$\frac{37}{24}$	$-\frac{9}{24}$			$\frac{251}{720}h^5 f^{(4)}(\xi)$
5	$\frac{1901}{720}$	$-\frac{2774}{720}$	$\frac{2616}{720}$	$-\frac{1274}{720}$	$\frac{251}{720}$		$\frac{475}{1440}h^6 f^{(5)}(\xi)$
6	$\frac{4277}{1440}$	$-\frac{7923}{1440}$	$\frac{9982}{1440}$	$-\frac{7298}{1440}$	$\frac{2877}{1440}$	$-\frac{475}{1440}$	$\frac{19087}{60480}h^7 f^{(6)}(\xi)$

The 4th -order method is the most popular $y_{i+1} = y_i + \frac{h}{24}\left(55 f_i - 59 f_{i-1} + 37 f_{i-2} - 9 f_{i-3}\right)$

Adams-Moulton Closed (or Implicit) Formula

- Using Taylor's series expansion of *y* around to express y_i (with *-h*)

$$y_i = y_{i+1} - h f_{i+1} + \frac{h^2}{2!} f'_{i+1} - \frac{h^3}{3!} f''_{i+1} + \frac{h^4}{4!} f'''_{i+1} - \cdots$$

- and rearrange the equation:

$$y_{i+1} = y_i + h f_{i+1} - \frac{h^2}{2!} f'_{i+1} + \frac{h^3}{3!} f''_{i+1} - \frac{h^4}{4!} f'''_{i+1} + \cdots$$

The method is implicit because $f_{i+1} = f(x_{i+1}, y_{i+1})$

appears on the RHS.

Adams-Moulton Closed (or Implicit) Formula (continue)

- By using the backward different formula

$$f'_{i+1} = \frac{f_{i+1} - f_i}{h} + \frac{h}{2} f''_{i+1} + O(h^2)$$

- for the 2nd -order method

$$y_{i+1} = y_i + h\left(\frac{1}{2} f_i + \frac{1}{2} f_{i+1}\right) - \frac{1}{12} h^3 f''_{i+1} + O(h^4)$$

- By using

$$f'_{i+1} = \frac{3 f_{i+1} - 4 f_i + f_{i-1}}{2h} + O(h^2) \quad f_{i+1}'' = \frac{f_{i+1} - 2 f_i + f_{i-1}}{h^2} + O(h)$$

- for the 3d -order method

$$y_{i+1} = y_i + h\left(\frac{5}{12} f_{i+1} + \frac{8}{12} f_i - \frac{1}{12} f_{i-1}\right) + O(h^4)$$

Adams-Moulton Closed (or Implicit) Formula (continue)

- The general expression for the *n*th-order Adams-Moulton closed formula

$$y_{i+1} = y_i + h\sum_{k=1}^{n}\alpha_{nk}\, f_{i-k+2} + O(h^{n+1})$$

Adams-Moulton Closed (or Implicit) Formula (continue)

Coefficients and local truncation error for Adams-Moulton Closed Formulas.

Order of the formula (n)	$k=1$ α_{n1}	$k=2$ α_{n2}	$k=3$ α_{n3}	$k=4$ α_{n4}	$k=5$ α_{n5}	$k=6$ α_{n6}	Local truncation error, $O(h^{n+1})$
1	1						$-\frac{1}{2}h^2 f'(\xi)$
2	$\frac{1}{2}$	$\frac{1}{2}$					$-\frac{1}{12}h^3 f''(\xi)$
3	$\frac{5}{12}$	$\frac{8}{12}$	$-\frac{1}{12}$				$-\frac{1}{24}h^4 f'''(\xi)$
4	$\frac{9}{24}$	$\frac{19}{24}$	$-\frac{5}{24}$	$\frac{1}{24}$			$-\frac{19}{720}h^5 f^{(4)}(\xi)$
5	$\frac{251}{720}$	$\frac{646}{720}$	$-\frac{264}{720}$	$\frac{106}{720}$	$-\frac{19}{720}$		$-\frac{27}{1440}h^6 f^{(5)}(\xi)$
6	$\frac{475}{1440}$	$\frac{1427}{1440}$	$-\frac{798}{1440}$	$\frac{482}{1440}$	$-\frac{173}{1440}$	$\frac{27}{1440}$	$-\frac{863}{60480}h^7 f^{(6)}(\xi)$

The 4th -order method is the most popular $y_{i+1} = y_i + \frac{h}{24}\left(9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}\right)$

Fourth-Order Adams Predictor-Corrector Method

- The open and closed formulas are used in combination. The open formula is used in a predictor step and the closed formula is applied as a corrector step.

1. Select the value of h and ε and compute $x_i = x_0 + ih,\ i = 1, 2, \ldots$

2. From the known initial condition $y(x_0) = y_0$

   use the fourth Runge-Kutta method (because it has the same level of accuracy), find the solution

$$y_1, y_2, y_3$$

3. Compute $f_1 = f(x_1, y_1),\ f_2 = f(x_2, y_2),\ f_3 = f(x_3, y_3)$

Fourth-Order Adams Predictor-Corrector Method

Step 1 Start with $i = 3$, find $y_{i+1}^{(0)} = y_4^{(0)}$ using the fourth-order Adams-Bashforth open formula $y_4^{(0)} = y_3 + \frac{h}{24}\left(55 f_3 - 59 f_2 + 37 f_1 - 9 f_0\right)$: predictor

Step 2 Compute $f_4^{(0)} = f(x_4, y_4^{(0)})$ and start with $j = 0$ to find $y_{i+1}^{(j+1)} = y_4^{(1)}$ using the fourth-order Adams-Moulton closed formula

$$y_4^{(1)} = y_3 + \frac{h}{24}\left(9 f_4^{(0)} + 19 f_3 - 5 f_2 + f_1\right) \text{ : corrector}$$

Step 3 Check convergence

if $\left|\dfrac{y_4^{(1)} - y_4^{(0)}}{y_4^{(1)}}\right| \le \varepsilon$, the answer is $y_4^{(1)}$, compute $f_4 = f(x_4, y_4^{(1)})$ and go to step 1 to compute a new predictor $y_{i+1}^{(0)} = y_5^{(0)}$ for next i,

$$y_{i+1}^{(0)} = y_i + \frac{h}{24}\left(55 f_i - 59 f_{i-1} + 37 f_{i-2} - 9 f_{i-3}\right) \text{ : predictor}$$

Fourth-Order Adams Predictor-Corrector Method

if $\left|\dfrac{y_4^{(1)} - y_4^{(0)}}{y_4^{(1)}}\right| > \varepsilon$, compute $f_4^{(1)} = f(x_4, y_4^{(1)})$ and repeat step 2 – 3

re-compute corrector $y_{i+1}^{(j+1)} = y_4^{(2)}$

($y_{i+1}^{(j+1)} = y_i + \dfrac{h}{24}\left(9 f_{i+1}^{(j)} + 19 f_i - 5 f_{i-1} + f_{i-2}\right)$: corrector) for next j until

the convergence is reached, $\left|\dfrac{y_{i+1}^{(j+1)} - y_{i+1}^{(j)}}{y_{i+1}^{(j+1)}}\right| \le \varepsilon$, then compute

$f_{i+1} = f(x_{i+1}, y_{i+1}^{(j+1)})$ and go to step 1 to compute a new predictor $y_{i+1}^{(0)}$ for next i.

If the convergence criterion is not satisfied in a reasonable number of iterations, repeat the whole procedure with a reduced step size, h.

Example

- Find the solution of the problem $y' = y + 2x - 1$ with $y(x = 0) = 1$ over the interval $0 \le x \le 1$ using the Adams predictor-corrector method.

Solution:

- $h = 0.1$; $x_0 = 0$, $x_1 = 0.1$, $x_2 = 0.2$, $x_3 = 0.3$, …, $x_{10} = 1$
- Use the fourth Runge-Kutta method to find the solution of y_1, y_2, y_3 with $y_0 = 1$

i	x	y	k_1	k_2	k_3	k_4
0	0	**1.00000000000000**	0.000000000	0.100000000	0.105000000	0.210500000
1	0.1	**1.01034166666667**	0.210341667	0.320858750	0.326384604	0.442980127
2	0.2	**1.04280514170139**	0.442805142	0.564945399	0.571052412	0.699910383
3	0.3	**1.09971699412508**				

Example (continue)

Compute $f_1 = f(x_1, y_1), f_2 = f(x_2, y_2), f_3 = f(x_3, y_3)$ where $f(x,y) = y + 2x - 1$

i	x	y	f(x, y)
0	0	1.00000000000000	**0.00000000000000**
1	0.1	1.01034166666667	**0.21034166666667**
2	0.2	1.04280514170139	**0.442805141701389**
3	0.3	1.09971699412508	**0.699716994125075**

Step 1 Start with $i = 3$, find $y_{i+1}^{(0)} = y_4^{(0)}$ using the fourth-order Adams-Bashforth open formula

predictor: $y_4^{(0)} = y_3 + \frac{h}{24}\left(55 f_3 - 59 f_2 + 37 f_1 - 9 f_0\right) =$ 1.183640214888258,

Step 2 Compute $f_4^{(0)} = f(x_4, y_4^{(0)})$ = 0.983640 and start with $j = 0$ to find $y_{i+1}^{(j+1)} = y_4^{(1)}$ using the fourth-order Adams-Moulton closed formula

corrector $^{(1)}$: $y_4^{(1)} = y_3 + \frac{h}{24}\left(9 f_4^{(0)} + 19 f_3 - 5 f_2 + f_1\right) =$ 1.1836491

Example (continue)

<u>Step 3</u> Check convergence

$$\left|\frac{y_4^{(1)} - y_4^{(0)}}{y_4^{(1)}}\right| = 7.4903017\text{E-}06 > \varepsilon = 10^{-6}$$

compute $f_4^{(1)} = f(x_4, y_4^{(1)}) = 0.9836402148882581$ and repeat step 2 – 3

corrector $^{(2)}$: $y_4^{(2)} = y_3 + \dfrac{h}{24}\left(9 f_4^{(1)} + 19 f_3 - 5 f_2 + f_1\right) =$ 1.18364941317895

$$\left|\frac{y_4^{(2)} - y_4^{(1)}}{y_4^{(2)}}\right| = 2.808842\text{E-}07 < \varepsilon = 10^{-6}$$ → converge, stop the iteration,

then compute $f_4 = f(x_4, y_4^{(2)}) = 0.9836491$ and go to step 1 to compute a new predictor $y_{i+1}^{(0)}$ for next *i*.

Repeat step 1 – 3 for next *i*, we get

Example (continue)

x	y
0	1.00000000000000
0.1	1.01034166666667
0.2	1.04280514170139
0.3	1.09971699412508
0.4	1.18364941317895
0.5	1.29744332717520
0.6	1.44423931921767
0.7	1.62750825205359
0.8	1.85108602902678
0.9	2.11921197874592
1	2.43657128484701

Initial value

Use the fourth-order Runge-Kutta method

Use the fourth-order Adams predictor-corrector method

Note:

- The fourth-order Runge-Kutta and Adams predictor-corrector methods are found to be very efficient and reliable compared with all method.
- The Runge-Kutta methods are self-starting and are easy to program on a digital computer. Although the predictor-corrector methods are not self-starting, a fourth-order Runge-Kutta method can be used to start the solution.
- The Runge-Kutta methods might require a smaller step size in order to achieve a specified accuracy.
- The predictor-corrector methods yield the results faster in most cases.
- For large systems of differential equations, a predictor-corrector method is preferred not only because of smaller computational time, but also due to the simplicity to estimate the error at any given stage.

Simultaneous Differential Equations

- In analysis of many engineering systems involves the solution of systems of simultaneous differential equations including higher order ordinary differential equations.

- The numerical solution of one or more higher ordinary differential equations is most conveniently obtained if the system of equations is represented in the form of a set of *n* simultaneous first-order differential equations

Simultaneous Differential Equations (continue)

- consider the *n*th-order differential equation:

$$a_n(x)\frac{d^n y}{dx^n} + a_{n-1}(x)\frac{d^{n-1} y}{dx^{n-1}} + \cdots + a_1(x)\frac{dy}{dx} + a_0 y = g(x)$$

- This can be rewritten as: $\dfrac{d^n y}{dx^n} = F\left(y, \dfrac{dy}{dx}, \dfrac{d^2 y}{dx^2}, \ldots, \dfrac{d^{n-1} y}{dx^{n-1}}, x\right)$

- By using the transformation:

$$\begin{aligned} y &= y_1, \\ \frac{dy}{dx} &= \frac{dy_1}{dx} = y_2, \\ \frac{d^2 y}{dx^2} &= \frac{dy_2}{dx} = y_3, \\ &\vdots, \\ \frac{d^{n-1} y}{dx^{n-1}} &= \frac{dy_{n-1}}{dx} = y_n, \\ \frac{d^n y}{dx^n} &= \frac{dy_n}{dx} \end{aligned}$$

Simultaneous Differential Equations (continue)

$$\left.\begin{aligned}
\frac{dy_1}{dx} &= f_1(y_1, y_2, \ldots, y_n, x) \equiv y_2\,, \\
\frac{dy_2}{dx} &= f_2(y_1, y_2, \ldots, y_n, x) \equiv y_3 \\
&\;\;\vdots, \\
\frac{dy_{n-1}}{dx} &= f_{n-1}(y_1, y_2, \ldots, y_n, x) \equiv y_n\,, \\
\frac{d^n y}{dx^n} &= f_n(y_1, y_2, \ldots, y_n, x) \equiv \frac{dy_n}{dx}
\end{aligned}\right\} \frac{d\vec{y}}{dx} = \vec{f}(\vec{y}, x) \;\Rightarrow\; \vec{y} = \begin{Bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{Bmatrix}, \; \vec{f} = \begin{Bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{Bmatrix}$$

The initial conditions are usually stated as:

$$\vec{y}(x=0) = \begin{Bmatrix} y_1(x=0) \\ y_2(x=0) \\ \vdots \\ y_n(x=0) \end{Bmatrix}$$

Example

- Solve the following differential equation $\frac{d^2y}{dx^2}+2\frac{dy}{dx}+4y=0$

  using Euler's method and 4th-order Runge-Kutta method, assuming that at x = 0, y = 2 and $\frac{dy}{dx}=0$

  Integrate to x = 3 with a step size of 0.1 and 0.01.

<u>Solution:</u>

- By defining

$$\frac{dy}{dx}=z=f_1(x,y,z), \qquad \frac{dz}{dx}=-2z-4y=f_2(x,y,z)$$

The initial conditions : $y(0)=2, \;\; z(0)=0$

Example (continue)

$$y_{i+1} = y_i + h\, f_1(x_i, y_i, z_i) = y_i + hz_i$$

$$z_{i+1} = z_i + h\, f_2(x_i, y_i, z_i) = z_i + h(-2z_i - 4y_i)$$

- Euler's method: *h* = 0.1, *x*0 = 0, *x*1 = 0.1, *x*2 = 0.2, *x*3 = 0.3 , …, *x*30 = 3, with an initial *y*0 = 2, *z*0 = 0

$$y_1 = y_0 + h\, f_1(x_0, y_0, z_0)$$
$$= y_0 + hz_0 = 2 + 0.1(0) = 2$$
$$z_1 = z_0 + h\, f_2(x_0, y_0, z_0) = z_0 + h(-2z_0 - 4y_0)$$
$$= 0 + 0.1(-2(0) - 4(2)) = -0.8$$

$$y_2 = y_1 + hz_1 = 2 + 0.1(-0.8) = 1.92$$
$$z_2 = z_1 + h(-2z_1 - 4y_1)$$
$$= -0.8 + 0.1(-2(-0.8) - 4(2)) = -1.44$$

Example (continue)

$$y_3 = y_2 + hz_2 = 1.92 + 0.1(-0.8) = 1.776$$

$$\begin{aligned} z_3 &= z_2 + h(-2z_2 - 4y_2) \\ &= -1.44 + 0.1(-2(-1.44) - 4(1.92)) = -1.92 \end{aligned}$$

$$y_4 = y_3 + hz_3 = 1.776 + 0.1(-1.92) = 1.584$$

$$\begin{aligned} z_4 &= z_3 + h(-2z_3 - 4y_3) \\ &= -1.92 + 0.1(-2(-1.92) - 4(1.776)) = -2.2464 \end{aligned}$$

$$\left.\begin{aligned} y_{i+1} &= y(x_{i+1}) = y_i + hz_i \\ z_{i+1} &= z(x_{i+1}) = z_i + h(-2z_i - 4y_i) \end{aligned}\right\} i = 5, 7, \ldots, 30$$

- Then calculate $y5$, $z5$, …, $y29$, $z29$, $y30$…

Example (continue)

4th-order Runge-Kutta method

h = 0.1, $x0$ = 0, $x1$ = 0.1, $x2$ = 0.2, $x3$ = 0.3, …, $x30$ = 3, with an initial $y0$ = 2, $z0$ = 0

$$k_{1,y} = f_1(z_0) = z_0 = 0$$

$$k_{1,z} = f_2(y_0, z_0) = -2z_0 - 4y_0 = -2(0) - 4(2) = -8$$

$$k_{2,y} = f_1(z_0 + \tfrac{1}{2}hk_{1,z}) = f_1(0 + \tfrac{1}{2}(0.1)(-8)) = f_1(-0.4) = -0.4$$

$$k_{2,z} = f_2(y_0 + \tfrac{1}{2}hk_{1,y}, z_0 + \tfrac{1}{2}hk_{1,z}) = f_2(2 + \tfrac{1}{2}(.01)(0), 0 + \tfrac{1}{2}(0.1)(-8))$$
$$= f_2(2, -0.4) = -2(-0.4) - 4(2) = -7.2$$

$$k_{3,y} = f_1(z_0 + \tfrac{1}{2}hk_{2,z}) = f_1(0 + \tfrac{1}{2}(0.1)(-7.2)) = f_1(-0.36) = -0.36$$

$$k_{3,z} = f_2(y_0 + \tfrac{1}{2}hk_{2,y}, z_0 + \tfrac{1}{2}hk_{2,z}) = f_2(2 + \tfrac{1}{2}(0.1)(-0.4), 0 + \tfrac{1}{2}(0.1)(-7.2))$$
$$= f_2(1.98, -0.36) = -2(-0.36) - 4(1.98) = -7.2$$

$$k_{4,y} = f(z_0 + hk_{3,z}) = f_1(0 + 0.1(-7.2)) = f_1(-0.72) = -0.72$$

$$k_{4,z} = f(y_0 + hk_{3,y}, z_0 + hk_{3,z}) = f_2(2 + 0.1(-0.36), 0 + 0.1(-7.2))$$
$$= f_2(1.964, -0.72) = -2(-0.72) - 4(1.964) = --6.416$$

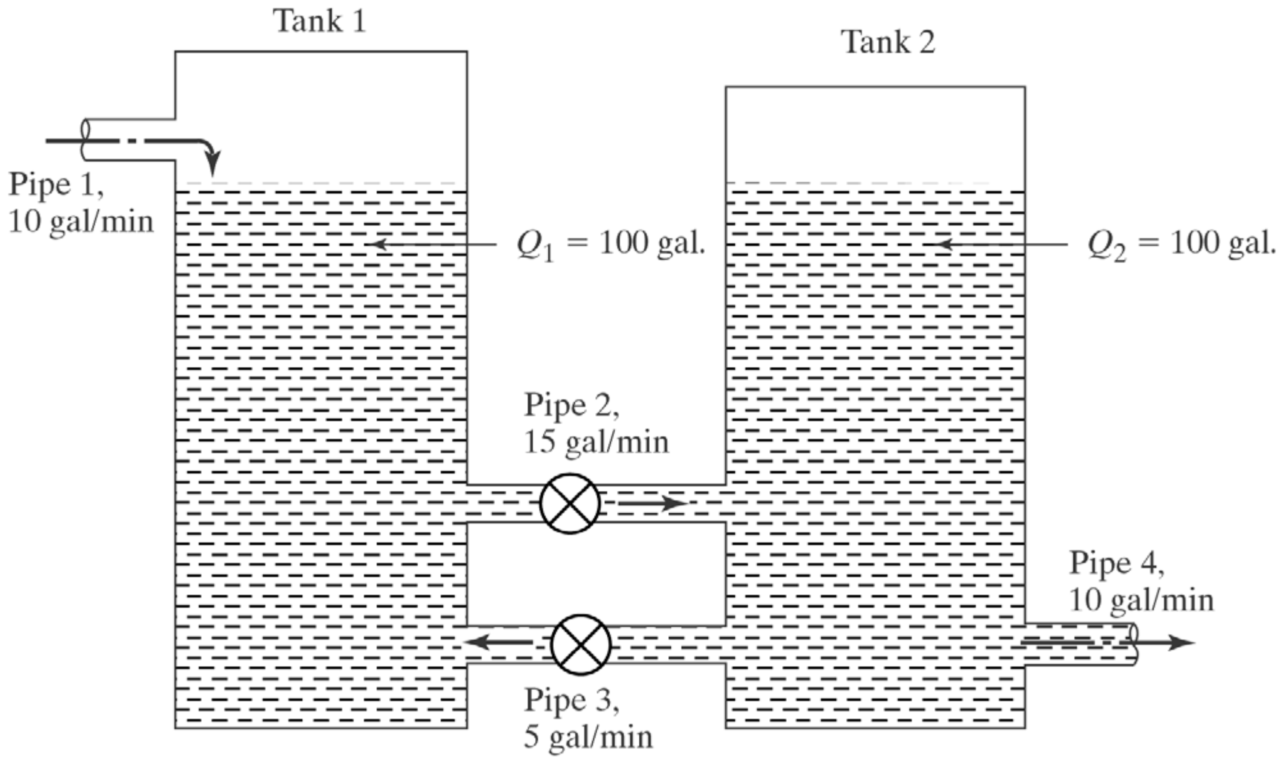
Example (continue)

$$y_1 = y_0 + \tfrac{h}{6}\left(k_{1,y} + 2k_{2,y} + 2k_{3,y} + k_{4,y}\right) = 2 + \tfrac{0.1}{6}(0 + 2(-0.4) + 2(-0.36) - 0.72) = 1.9627$$

$$z_1 = z_0 + \tfrac{h}{6}\left(k_{1,z} + 2k_{2,z} + 2k_{3,z} + k_{4,z}\right) = 0 + \tfrac{0.1}{6}(-8 + 2(-7.2) + 2(-7.2) - 6.416) = -0.72027$$

- Then calculate *y*5, *z*5, …, *y*29, *z*29, *y*30 …

Example – system of equations



Each of the tanks shown in Fig. contain 100 gallons of water. The water in tank 1 contains 1000 grams of dissolved salt, while the water in tank 2 contains 100 grams of dissolved salt. At time t=0, pipe 1 starts pumping pure water at the rate of 10 gallons/min while pipes 2,3,4 start pumping mixed water at rates of 15, 5 and 10 gallons/min, respectively. Determine the concentration of salt in the tanks at t=3 min.

Use of MATLAB

- ode45 is based on an explicit Runge-Kutta (4,5) formula, the Dormand-Prince pair. In general, ode45 is the best function to apply as a "first try" for most problems.
- ode23 is an implementation of an explicit Runge-Kutta (2,3) pair of Bogacki and Shampine. It may be more efficient than ode45 at crude tolerances and in the presence of moderate stiffness. Like ode45, ode23 is a one-step solver.
- ode113 is a variable order Adams-Bashforth-Moulton PECE solver.

  [t,Y] = solver(odefun,tspan,y0) where solver is one of ode45, ode23, ode113, ode15s, ode23s, ode23t, or ode23tb.