數值分析

Chapter 7

Ordinary Differential Equations: Initial Value Problems

授課教師：劉耀先

國立陽明交通大學 機械工程學系 EE464

yhliu@nctu.edu.tw

110學年度第一學期

Outline

- Background (7.1)
- Euler’s method (7.2, 7.3)
- Midpoint method (7.4)
- Runge-Kutta methods (7.5)
- Multistep methods (7.6)
- Predictor-corrector methods (7.7)
- Systems of first-order ODEs (7.8)
- Higher order initial value ODE (7.9)
- MATLAB built-in functions (7.10)

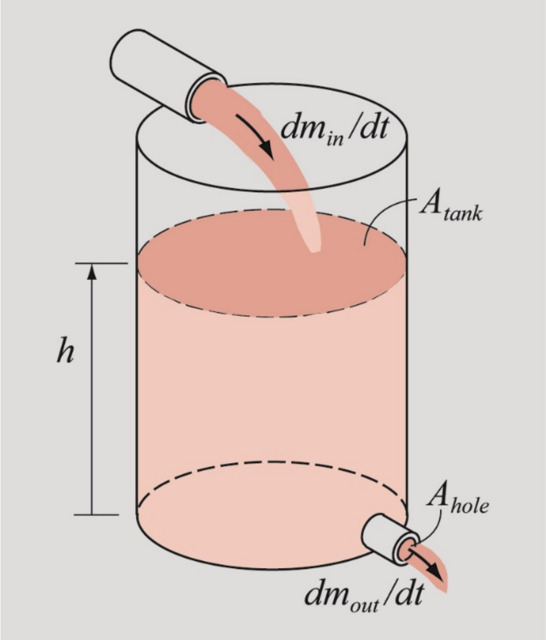
7.1 Background

- First order ODEs involve the first derivative of the dependent variable (y) with respect to the independent variable(x):

$$\frac{dy}{dx} + ax^2 + by = 0 \qquad \text{(Linear)} \qquad \frac{dy}{dx} + ayx + b\sqrt{y} = 0 \qquad \text{(Nonlinear)}$$

- Consider the water flow out of the tank:

$$\frac{dm_{in}}{dt} = K_1 + K_2 \cos(Ct)$$



- The equation can be rewritten as:

$$\rho A_{\tan k} \frac{dh}{dt} = K_1 + K_2 \cos(Ct) - \rho A_{pipe} \sqrt{2gh}$$

- Initial condition: h = h_o at t = t_o
- Initial-value problem(IVP)

7.1 Background

- First-order ODE problem

$$\frac{dy}{dx} = f(x, y) \quad \text{with the initial condition} \quad y(x_1) = y_1$$

- Analytical solution of a first-order ODE:
  - Mathematical expression
- Numerical solution of a first-order ODE:
  - A set of discrete points that approximate the function y(x)
  - **Single-step** and **multi-step**

Explicit vs. Implicit method

- Explicit:
  - use an explicit formula for calculating the value of the dependent variable
  - The right-hand side of the equation only has **known** quantities

$$y_{i+1} = F(x_i, x_{i+1}, y_i)$$

- Implicit:
  - calculate the unknown from unknown quantities
  - The unknown quantities appears on both sides of the equation

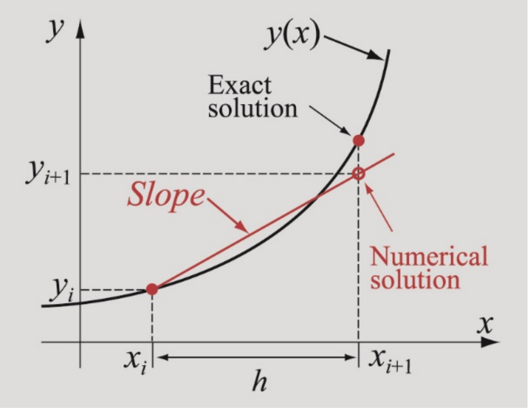
$$y_{i+1} = F(x_i, x_{i+1}, \underline{y_{i+1}})$$

Single-step explicit method

- Calculate the approximate numerical solution (x_{i+1}, y_{i+1}) from the known solution at point (x_i, y_i):

$$x_{i+1} = x_i + h$$

$$y_{i+1} = y_i + Slope \cdot h$$



- Slope: the estimate value of dy/dx

- The procedure continues with i=3 and so on until the points cover the while domain of the solution

7.2 Euler's methods

- Euler's method is the simplest technique for solving a first-order ODE of the form:

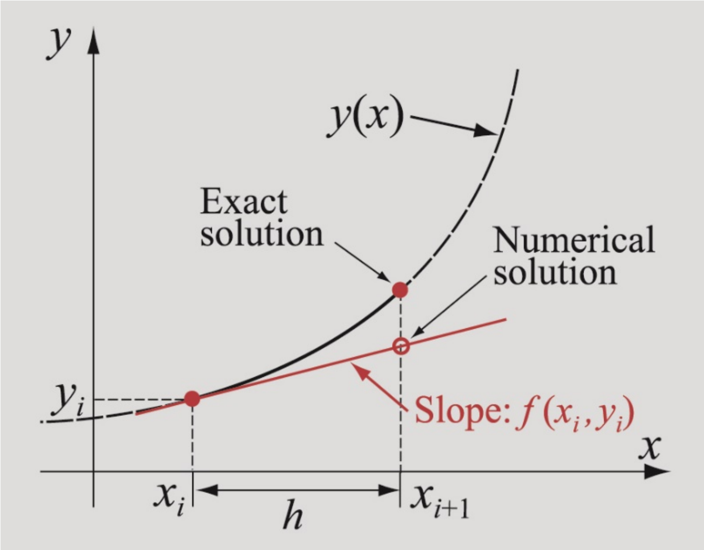
$$\frac{dy}{dx} = f(x, y) \quad \text{with the initial condition} \quad y(x_1) = y_1$$

- Euler's explicit method (forward Euler method):

$$Slope = \left.\frac{dy}{dx}\right|_{x=x_i} = f(x_i, y_i)$$

$$x_{i+1} = x_i + h$$

$$y_{i+1} = y_i + f(x_i, y_i) \cdot h$$



Euler's method

- Using numerical integration:

$$\int_{y_i}^{y_{i+1}} dy = \int_{x_i}^{x_{i+1}} f(x,y)dx \quad \Rightarrow \quad y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x,y)dx$$

- Evaluate the integral using the numerical integration (ex Rectangle method):

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x,y)dx = y_i + f(x_i, y_i)(x_{i+1} - x_i)$$

- The derivate can also be estimated by finite difference approximation:

$$\left.\frac{dy}{dx}\right|_{x=x_i} \approx \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = f(x_i, y_i)$$

Example 7-1: First-order ODE with Euler's explicit method

- Use Euler's explicit method to solve the ODE from x=0 to x=2.5 with the initial condition y=3 at x=0

$$\frac{dy}{dx} = -1.2y + 7e^{-0.3x}$$

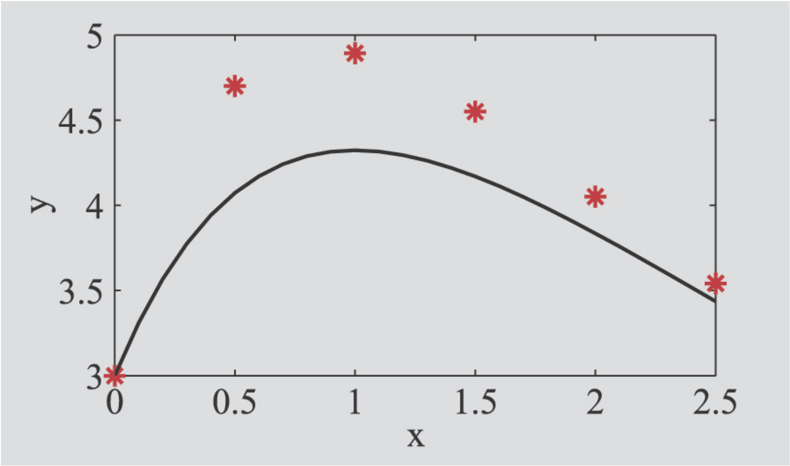
- (a) solve by hand using h=0.5
- (b) write a MATLAB program that solves the equation using h=0.1
- (c) use the program from (b) to solve the equation using h=0.01

- (a) $x_{i+1} = x_i + h = x_i + 0.5$

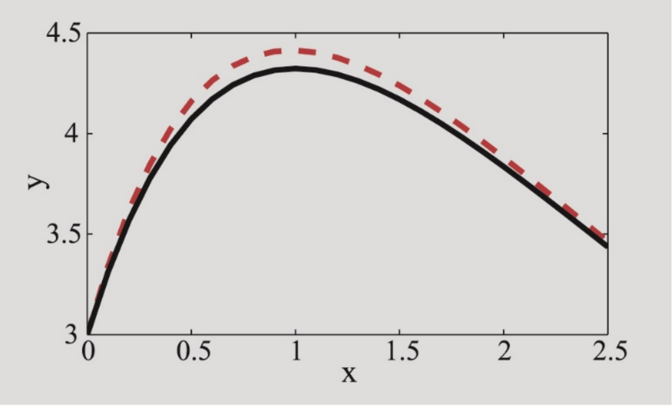
$$y_{i+1} = y_i + f(x_i, y_i) \cdot h = y_i + (-1.2y_i + 7e^{-0.3x_i}) \cdot 0.5$$

Example 7-1

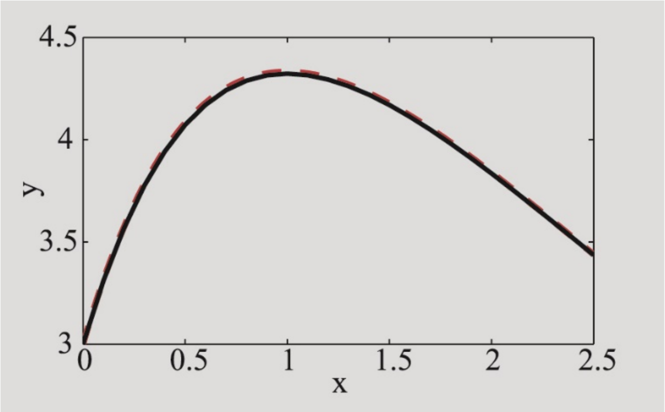
(a)



(b)



(c)



7.2.2 Truncation error in Euler's explicit method

- Local truncation error:
  - the error inherent in the formula used to obtain the numerical solution in a single step (subinterval)
- Using two-term Taylor series expansion:

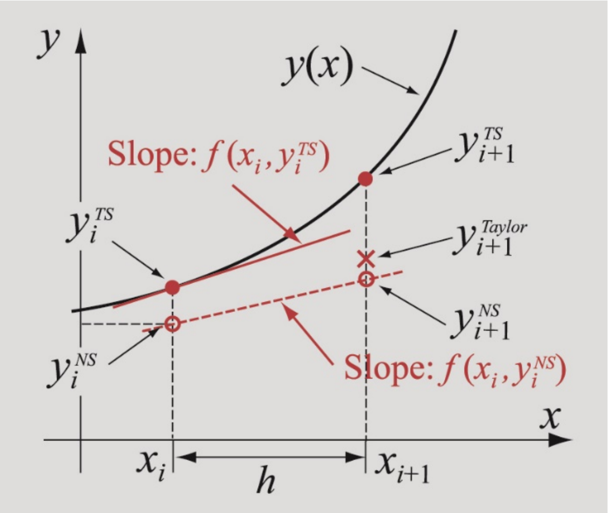
$$y_{i+1}^{Taylor} = y_i^{NS} + f(x_i, y_i^{NS})h + \left.\frac{d^2y}{dx^2}\right|_{x=\xi_i} \frac{h^2}{2}$$

- With Euler's explicit method:

$$y_{i+1}^{NS} = y_i^{NS} + f(x_i, y_i^{NS})h$$

- The local truncation error:

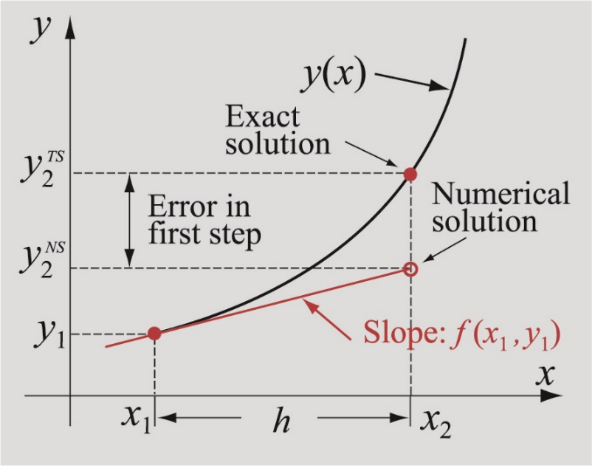
$$e_i^{TR} = y_{i+1}^{Taylor} - y_{i+1}^{NS} = \left.\frac{d^2y}{dx^2}\right|_{x=\xi_i} \frac{h^2}{2} = O(h^2)$$

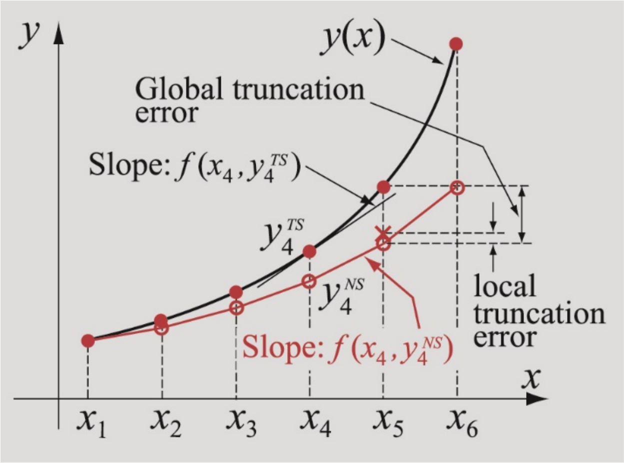


Global truncation error

- The total error due to truncation alone
- The truncation error is propagated from the previous subinterval to the next subinterval
- The global truncation error at the last point is:

$$E_{N+1}^{TR} \le \frac{hM}{C}\left[e^{hCN} - 1\right] = \frac{hM}{C}\left[e^{C(x_{N+1}-x_i)} - 1\right]$$





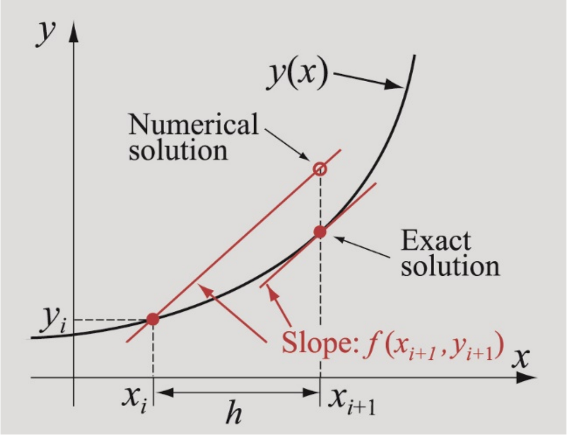
7.2.3 Euler's Implicit Method

- The form of Euler's implicit method (**backward Euler's method**) uses the slope of the function y(x) is taken to be a constant at the endpoint **(x_{i+1}, y_{i+1})**

$$x_{i+1} = x_i + h \qquad y_{i+1} = y_i + f(x_{i+1}, y_{i+1}) \cdot h$$

- This is a nonlinear equation for the unknown y_{i+1}
- Evaluate the integral by the Rectangle method with the end point of the interval
- The approximation of the derivative:

$$\left.\frac{dy}{dx}\right|_{x=x_{i+1}} \approx \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = f(x_{i+1}, y_{i+1})$$



Example 7-2: First-order ODE using Euler's Implicit Method

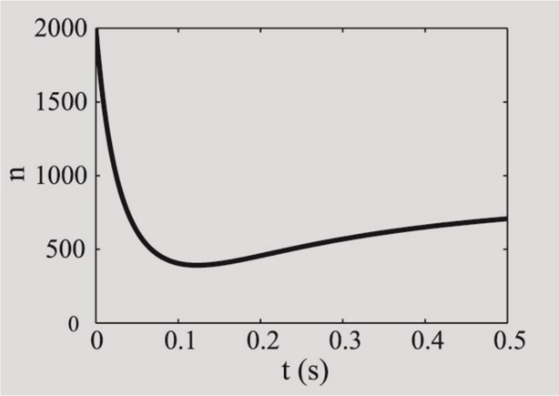
- The differential equation :

$$\frac{dn(t)}{dt} = -0.8n^{3/2} + 10n_1(1-e^{-3t})$$

- Where n_1=2000 at t=0 is the initial condition. Solve the differential equation from t=0 until t=0.5s using h=0.002.

$$t_{i+1} = t_i + h$$

$$n_{i+1} = n_i + [-0.8n_{i+1}^{3/2} + 10n_1(1-e^{-3t_{i+1}})] \cdot h$$

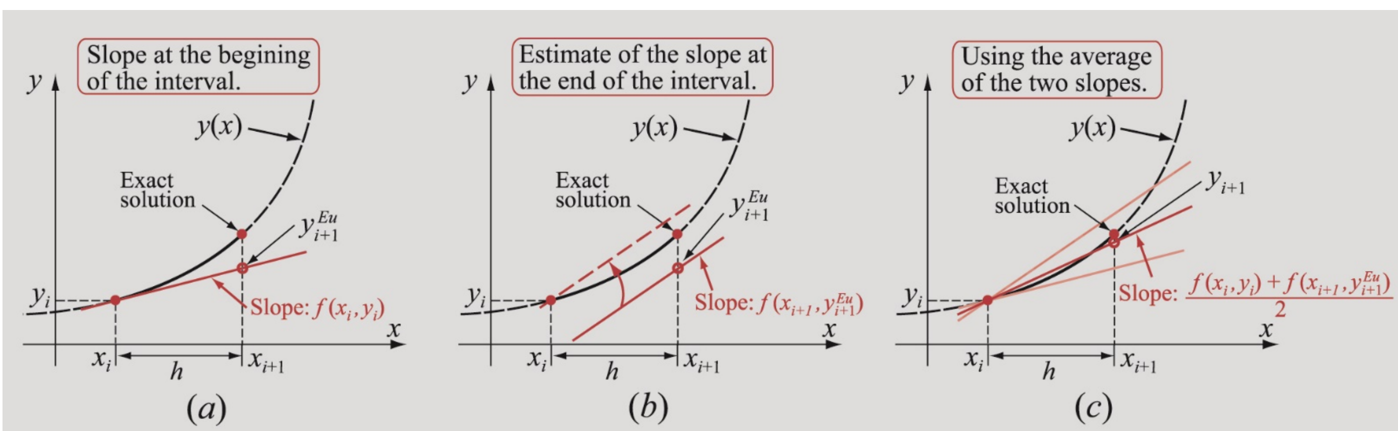


7.3 Modified Euler's Method

- Single step, explicit method for solving first-order ODE
- The slope is calculated based on the average of the slope at the **beginning** and the estimate at the **end** point

$$\left.\frac{dy}{dx}\right|_{x=x_i} = f(x_i, y_i) \qquad \left.\frac{dy}{dx}\right|_{\substack{y=y_{i+1}^{\mathrm{Eu}} \\ x=x_{i+1}}} = f(x_{i+1}, y_{i+1}^{Eu}) \qquad y_{i+1}^{Eu} = y_i + f(x_i, y_i)\cdot h$$

- The value is calculated from: $y_{i+1} = y_i + \dfrac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{Eu})}{2}\cdot h$



Algorithm for modified Euler's method

- Given a solution at point (x_i, y_i), calculate the next value of the independent variable:

$$x_{i+1} = x_i + h$$

- Calculate $f(x_i, y_i)$
- Estimate y_{i+1} using Euler's method:

$$y_{i+1}^{Eu} = y_i + f(x_i, y_i) \cdot h$$

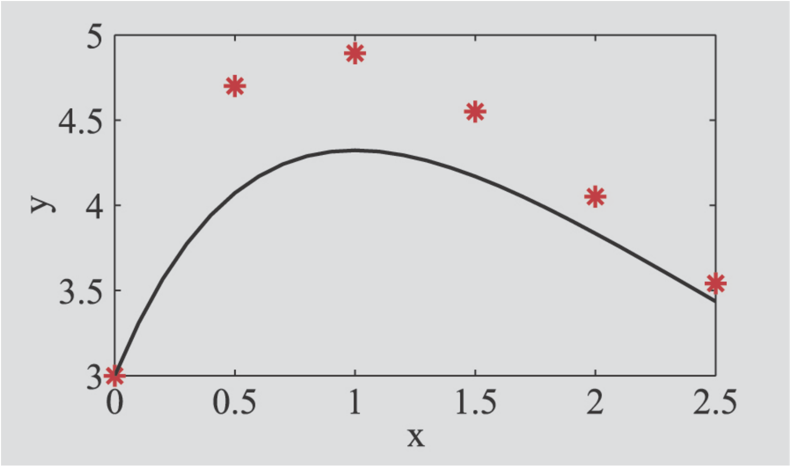
- Calculate $f(x_{i+1}, y_{i+1}^{Eu})$
- Calculate the numerical solution at x_{i+1}

$$y_{i+1} = y_i + \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{Eu})}{2} \cdot h$$

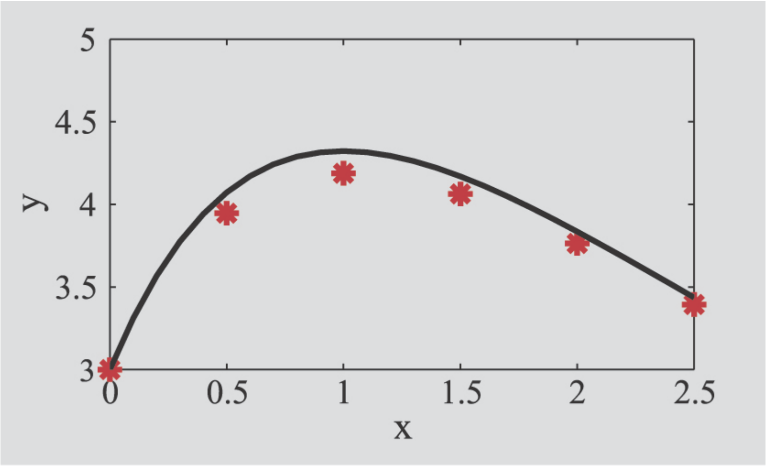
Ex 7-3: Modified Euler's Method

- Use modified Euler's method to solve the following ODE from x=0 to x=2.5 with the initial condition y=3 at x=0, using h=0.5:

$$\frac{dy}{dx} = -1.2y + 7e^{-0.3x}$$



Euler's method



Modified Euler's method

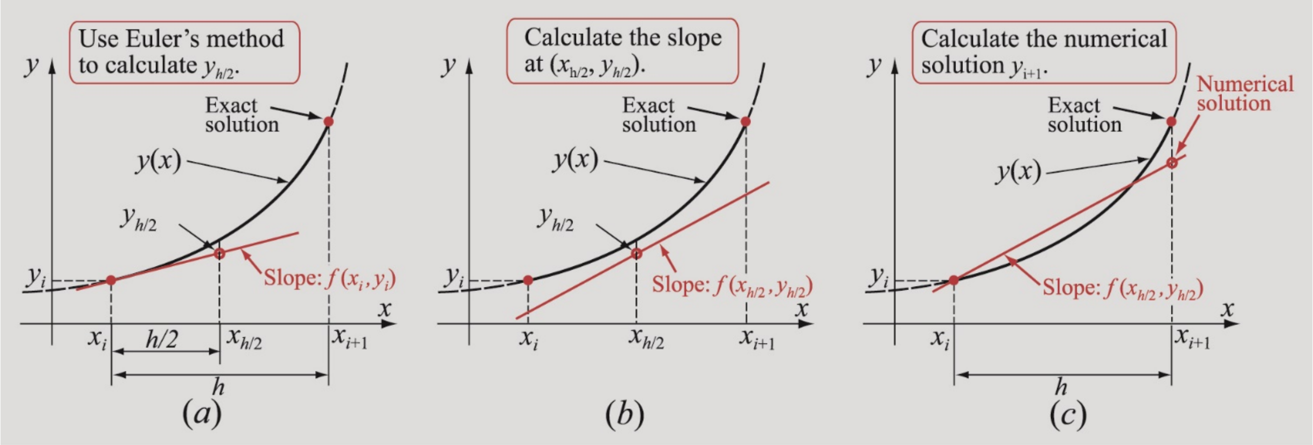
7.4 Midpoint Method

- Estimate the slope at the **middle** point of the interval
- First calculate the approximate value at the middle point of the interval:

$$x_m = x_i + h/2 \qquad\qquad y_m = y_i + f(x_i, y_i)\cdot h/2$$

- Use the estimated midpoint to calculate the slope:

$$y_m = y_i + f(x_m, y_m)\cdot h \qquad\qquad \left.\frac{dy}{dx}\right|_{x=x_m} = f(x_m, y_m)$$



7.5 Runge-Kutta Methods

- A single-step, explicit method for solving first-order ODE
- The slope is calculated from number of points (**order**) within the subinterval
- The second-order Runge-Kutta methods use the slope at **two** points, third-order Runge-Kutta methods uses the slope at **three** points, and so on.
- The classical Runge-Kutta method is of **fourth** order
- The accuracy increases with increasing order

7.5.1 Second-Order Runge-Kutta Methods

- The general form of second-order Runge-Kutta method:

$$y_{i+1} = y_i + (c_1 K_1 + c_2 K_2) \cdot h$$

- with

$$K_1 = f(x_i, y_i) \qquad K_2 = f(x_i + a_2 h, y_i + b_{21} K_1 h)$$

- The constants vary with the specific second-order method
  - Modified Euler’s method
  - Midpoint method
  - Heun’s method

Modified Euler Method / Midpoint Method

- Modified Euler Method for 2nd-order RK method:

$$c_1 = \frac{1}{2} \qquad c_2 = \frac{1}{2} \qquad a_2 = 1 \qquad b_{21} = 1$$

$$y_{i+1} = y_i + \frac{1}{2}(K_1 + K_2) \cdot h$$

$$K_1 = f(x_i, y_i) \qquad K_2 = f(x_i + h, y_i + K_1 h)$$

- Midpoint Method for 2nd-order RK method:

$$c_1 = 0 \qquad c_2 = 1 \qquad a_2 = \frac{1}{2} \qquad b_{21} = \frac{1}{2}$$

$$y_{i+1} = y_i + K_2 \cdot h$$

$$K_1 = f(x_i, y_i) \quad K_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_1 h)$$

Heun's method

- Heun's Method for 2nd-order RK method:

$$c_1 = \frac{1}{4} \qquad c_2 = \frac{3}{4} \qquad a_2 = \frac{2}{3} \qquad b_{21} = \frac{2}{3}$$

$$y_{i+1} = y_i + (\frac{1}{4}K_1 + \frac{3}{4}K_2)\cdot h$$

$$K_1 = f(x_i, y_i) \quad K_2 = f(x_i + \frac{2}{3}h, y_i + \frac{2}{3}K_1 h)$$

- From Taylor's series expansion, the second-order Runge-Kutta method has the following equations:

$$c_1 + c_2 = 1 \qquad c_2 a_2 = 1 \qquad c_2 b_{21} = 1$$

Third-Order Runge-Kutta Methods

- General form:

$$y_{i+1} = y_i + (c_1 K_1 + c_2 K_2 + c_3 K_3) \cdot h$$

$$K_1 = f(x_i, y_i) \qquad K_2 = f(x_i + a_2 h, y_i + b_{21} K_1 h)$$

$$K_3 = f(x_i + a_3 h, y_i + b_{31} K_1 h + b_{31} K_2 h)$$

- For the classical 3rd-order RK method:

$$y_{i+1} = y_i + \frac{1}{6}(K_1 + 4K_2 + K_3) \cdot h$$

$$K_1 = f(x_i, y_i) \qquad K_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_1 h) \quad K_3 = f(x_i + h, y_i - K_1 h + 2K_2 h)$$

Method	c_1	c_2	c_3	a_2	b_{21}	a_3	b_{31}	b_{31}
Classical	1/6	4/6	1/6	1/2	1/2	1	–1	2
Nystrom's	2/8	3/8	3/8	2/3	2/3	2/3	0	2/3
Nearly Optimal	2/9	3/9	4/9	1/2	1/2	3/4	0	3/4
Heun's Third	1/4	0	3/4	1/3	1/3	2/3	0	2/3

Fourth-Order Runge-Kutta Method

- General form:

$$y_{i+1} = y_i + (c_1 K_1 + c_2 K_2 + c_3 K_3 + c_4 K_4) \cdot h$$

$$K_1 = f(x_i, y_i) \qquad K_2 = f(x_i + a_2 h, y_i + b_{21} K_1 h)$$

$$K_3 = f(x_i + a_3 h, y_i + b_{31} K_1 h + b_{31} K_2 h)$$

$$K_4 = f(x_i + a_4 h, y_i + b_{41} K_1 h + b_{42} K_2 h + b_{43} K_3 h)$$

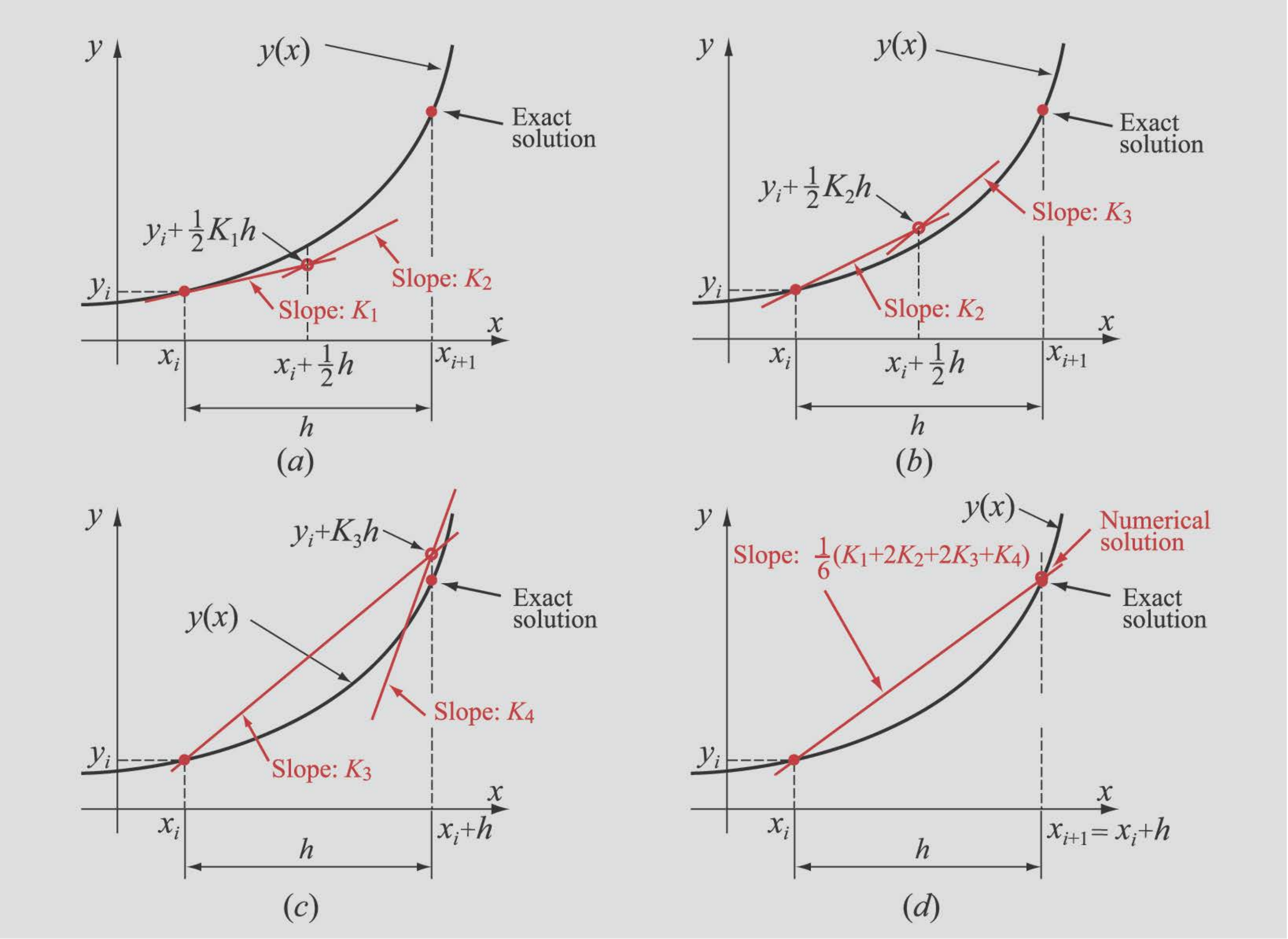
- The classical fourth-order RK method:

$$y_{i+1} = y_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \cdot h$$

$$K_1 = f(x_i, y_i) \qquad K_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_1 h) \qquad K_3 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_2 h)$$

$$K_4 = f(x_i + h, y_i + K_3 h)$$

Classical Fourth-order Runge-Kutta Method

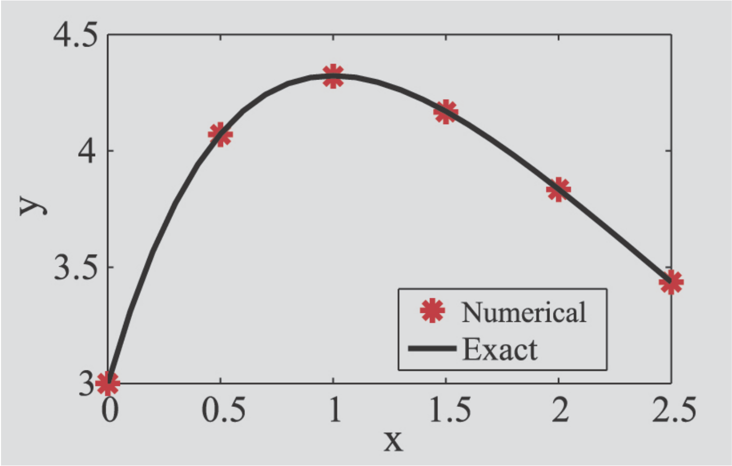


Ex 7-6:First-order ODE with 4th-order RK method

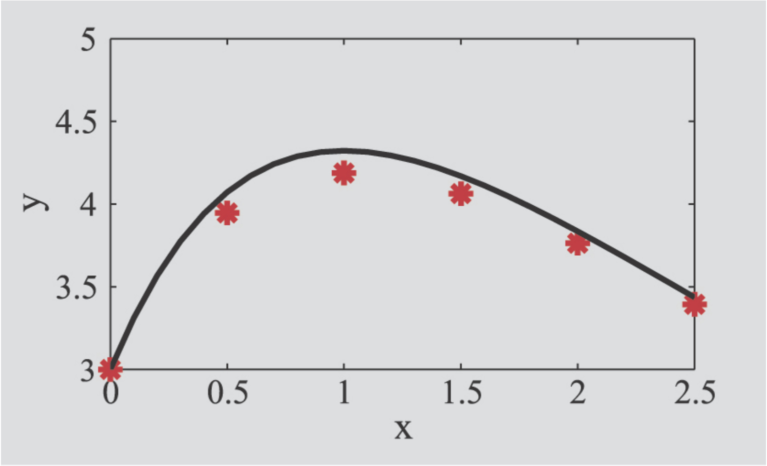
- Write a user-defined MATLAB function that solve a first-order ODE using classical 4th-order RK method:
- From x=0 to 2.5 with y(0)=3 and h=0.5
- [x, y] = odeRK4(ODE, a, b, h)

$$\frac{dy}{dx} = -1.2y + 7e^{-0.3x}$$

- The results show small error even with large step size



4th order RK



Modified Euler's method

7.6 Multistep Methods

- Calculate the solution of y_{i+1} at x_{i+1} using two or more previous points
- Explicit or implicit
- If three points are used for the explicit form:

$$y_{i+1} = F(x_{i-2}, y_{i-2}, x_{i-1}, y_{i-1}, x_i, y_i, x_{i+1})$$

- The first few points can be determined by single-step methods or by multistep methods that use fewer prior points
- In implicit multistep methods, the unknown y_{i+1} appears on both sides of the equation

7.6.1 Adams-Bashforth Method

- An explicit multistep method for solving a first-order ODE
- The order is determined by the number of points used in the formula
- Consider integrating the differential equation over an arbitrary interval:

$$\frac{dy}{dx} = f(x, y) \qquad\qquad y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x, y)dx$$

- The integration is carried out by approximating f(x,y) with a polynomial that interpolates the value of f(x,y) at (x_i, y_i) and at previous points

7.6.1 Adams-Bashforth Method

- 2^{nd} order Adams-Bashforth method

$$y_{i+1} = y_i + \frac{h}{2}[3f(x_i, y_i) - f(x_{i-1}, y_{i-1})]$$

- 3^{rd} order Adams-Bashforth method

$$y_{i+1} = y_i + \frac{h}{12}[23f(x_i, y_i) - 16f(x_{i-1}, y_{i-1}) + 5f(x_{i-2}, y_{i-2})]$$

- 4^{th} order Adams-Bashforth method

$$y_{i+1} = y_i + \frac{h}{24}[55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}]$$

7.6.2 Adams-Moulton Method

- An implicit multistep method for solving a first-order ODE
- Use the points (x_i, y_i) and (x_{i+1}, y_{i+1}) to determine the interpolation points
- 2nd order Adams-Moulton method:

$$y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, y_{i+1})]$$

- 3rd order Adams-Moulton method:

$$y_{i+1} = y_i + \frac{h}{12}[5f(x_{i+1}, y_{i+1}) + 8f(x_i, y_i) - f(x_{i-1}, y_{i-1})]$$

- 4th order Adams-Moulton method:

$$y_{i+1} = y_i + \frac{h}{24}[9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}]$$

7.7 Predictor-Corrector Method

- Solving ODEs using two formulas: **predictor** and **correcto**r formulas
- The **predicto**r is an *explicit* formula and is used first to determine an estimate of the solution y_{i+1}
- The corrector is applied once an estimate of y_{i+1} is found, which uses the estimated value of y_{i+1} on the right hand side of an otherwise implicit formula for calculating a new, more accurate value of y_{i+1}on the left hand side
- No solution of a nonlinear equation is required
- The application of corrector formula can be repeated several times to obtain a more refined value

Algorithm for the Predictor-Corrector

- Given a solution at points (x_1, y_1), (x_2, y_2)…(x_i, y_i)
- (1) Calculate y_{i+1} using an explicit method
- (2) Substitute y_{i+1} from step (1), as well as any required values from already known solution at previous points, in the right hand side of an implicit formula to obtain a refined value of y_{i+1}
- (3) Repeat Step 2 as many times as necessary untilthe desired level of accuracy

Algorithm: Euler predictor-corrector

- Modified Euler predictor-corrector method
- Calculate a first estimate using Euler's explicit method as a predictor

$$y_{i+1}^{(1)} = y_i + f(x_i, y_i) \cdot h$$

- Calculate better estimates using corrector for k = 2, 3,..

$$y_{i+1}^{(k)} = y_i + \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{(k-1)})}{2} \cdot h$$

- Stop the iteration when:

$$\left| \frac{y_{i+1}^{(k)} - y_{i+1}^{(k-1)}}{y_{i+1}^{(k-1)}} \right| \le \varepsilon$$

Algorithm: Adams-Bashforth and Adams-Moulton

- The explicit formula (Adams-Bashforth) and implicit formula(Adams-Moulton) can be used together as a predictor-corrector method
- For the 3rd order formula, the first estimate is:

$$y_{i+1}^{1} = y_i + \frac{h}{12}[23 f(x_i, y_i) - 16 f(x_{i-1}, y_{i-1}) + 5 f(x_{i-2}, y_{i-2})]$$

- The corrector equation is: (for k = 2, 3…)

$$y_{i+1}^{k} = y_i + \frac{h}{12}[5 f(x_{i+1}, y_{i+1}^{k-1}) + 8 f(x_i, y_i) - f(x_{i-1}, y_{i-1})]$$

7.8 System of 1st Order ODEs

- General form of a system of first-order ODEs:

$$\frac{dy_1}{dt} = f_1(t, y_1, y_2, y_3, ..., y_n) \qquad y_1(t_1) = Y_1$$

$$\frac{dy_2}{dt} = f_2(t, y_1, y_2, y_3, ..., y_n) \qquad y_2(t_1) = Y_2$$

$$\vdots \qquad\qquad\qquad \vdots$$

$$\frac{dy_n}{dt} = f_n(t, y_1, y_2, y_3, ..., y_n) \qquad y_n(t_1) = Y_n$$

- Solving the system of first-order ODEs:

$$t_{i+1} = t_i + h \qquad y_{1,i+1} = y_{1,i} + Slope_1 \cdot h$$

$$y_{2,i+1} = y_{2,i} + Slope_2 \cdot h$$

$$\vdots$$

$$y_{n,i+1} = y_{n,i} + Slope_n \cdot h$$

System of 1st Order ODEs

- Euler's explicit method
- Consider a system of two equations with y and z as the dependent variables:

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

- The initial condition:

$$y(a) = y_1 \qquad z(a) = z_1$$

- The Euler's explicit method is given by:

$$x_{i+1} = x_i + h$$

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

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

System of 1st Order ODEs: RK method

- Second-order RK Method (modified Euler):
- Consider a system of three equations with y, z, and w as the dependent variables:

$$\frac{dy}{dx} = f_1(x, y, z, w) \qquad \frac{dz}{dx} = f_2(x, y, z, w) \qquad \frac{dw}{dx} = f_3(x, y, z, w)$$

- The initial condition:

$$y(a) = y_1 \qquad z(a) = z_1 \qquad w(a) = w_1$$

- The solution process is given by:

$$x_{i+1} = x_i + h$$

$$K_{y,1} = f_1(x_i, y_i, z_i, w_i)$$
$$K_{z,1} = f_2(x_i, y_i, z_i, w_i)$$
$$K_{w,1} = f_3(x_i, y_i, z_i, w_i)$$

System of 1st Order ODEs: RK method

- Next calculate the slope K_2:

$$K_{y,2} = f_1(x_i + h, y_i + K_{y,1}h, z_i + K_{z,1}h, w_i + K_{w,1}h)$$

$$K_{z,2} = f_2(x_i + h, y_i + K_{y,1}h, z_i + K_{z,1}h, w_i + K_{w,1}h)$$

$$K_{w,2} = f_3(x_i + h, y_i + K_{y,1}h, z_i + K_{z,1}h, w_i + K_{w,1}h)$$

- Finally the values of the three dependent variables can be calculated:

$$y_{i+1} = y_i + \frac{1}{2}(K_{y,1} + K_{y,2}) \cdot h$$

$$z_{i+1} = z_i + \frac{1}{2}(K_{z,1} + K_{z,2}) \cdot h$$

$$w_{i+1} = w_i + \frac{1}{2}(K_{w,1} + K_{w,2}) \cdot h$$

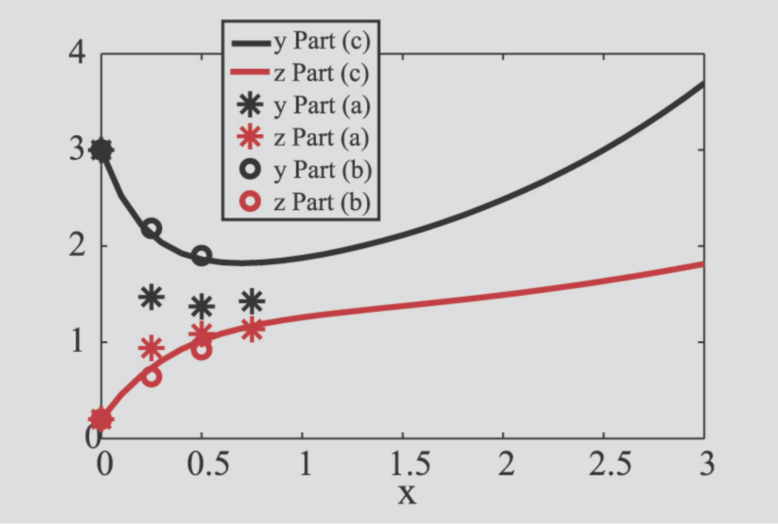
Example 7-7: Two first-order ODEs

- Consider the following initial value problem using 2nd order Runge-Kutta method:

$$\frac{dy}{dx} = (-y+z)e^{(1-x)} + 0.5y \qquad y(0) = 3$$

$$\frac{dz}{dx} = y - z^2 \qquad z(0) = 0.2$$

- Write a MATLAB program using h=0.1



Fourth-order RK method

- Solve the value of K_1:

$$K_{y,1} = f_1(x_i, y_i, z_i, w_i) \quad K_{z,1} = f_2(x_i, y_i, z_i, w_i) \quad K_{w,1} = f_3(x_i, y_i, z_i, w_i)$$

- Solve the value of K_2 and K_3:

$$K_{y,2} = f_1(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_{y,1}h, z_i + \frac{1}{2}K_{z,1}h, w_i + \frac{1}{2}K_{w,1}h)$$
$$K_{z,2} = f_2(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_{y,1}h, z_i + \frac{1}{2}K_{z,1}h, w_i + \frac{1}{2}K_{w,1}h)$$
$$K_{w,2} = f_3(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_{y,1}h, z_i + \frac{1}{2}K_{z,1}h, w_i + \frac{1}{2}K_{w,1}h)$$

$$K_{y,3} = f_1(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_{y,2}h, z_i + \frac{1}{2}K_{z,2}h, w_i + \frac{1}{2}K_{w,2}h)$$
$$K_{z,3} = f_2(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_{y,2}h, z_i + \frac{1}{2}K_{z,2}h, w_i + \frac{1}{2}K_{w,2}h)$$
$$K_{w,3} = f_3(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_{y,2}h, z_i + \frac{1}{2}K_{z,2}h, w_i + \frac{1}{2}K_{w,2}h)$$

Fourth-order RK method

- Solve the value of K_4:

$$K_{y,4} = f_1(x_i + h, y_i + K_{y,3}h, z_i + K_{z,3}h, w_i + K_{w,3}h)$$

$$K_{z,4} = f_2(x_i + h, y_i + K_{y,3}h, z_i + K_{z,3}h, w_i + K_{w,3}h)$$

$$K_{w,4} = f_3(x_i + h, y_i + K_{y,3}h, z_i + K_{z,3}h, w_i + K_{w,3}h)$$

- The value of the dependent variables can be calculated:

$$y_{i+1} = y_i + \frac{1}{6}(K_{y,1} + 2K_{y,2} + 2K_{y,3} + K_{y,4}) \cdot h$$

$$z_{i+1} = z_i + \frac{1}{6}(K_{z,1} + 2K_{z,2} + 2K_{z,3} + K_{z,4}) \cdot h$$

$$w_{i+1} = w_i + \frac{1}{6}(K_{w,1} + 2K_{w,2} + 2K_{w,3} + K_{w,4}) \cdot h$$

7.9 Solving a Higher-Order Initial Value Problem

- Consider a second order ODE:

$$\frac{d^2 y}{dx^2} = f(x, y, \frac{dy}{dx})$$

- The equation can be solved if the initial condition:

$$y(a) = A \qquad \left.\frac{dy}{dx}\right|_{x=a} = B$$

- This equation can be transformed into a system of **two first-order** ODEs by introducing a new dependent variable w:

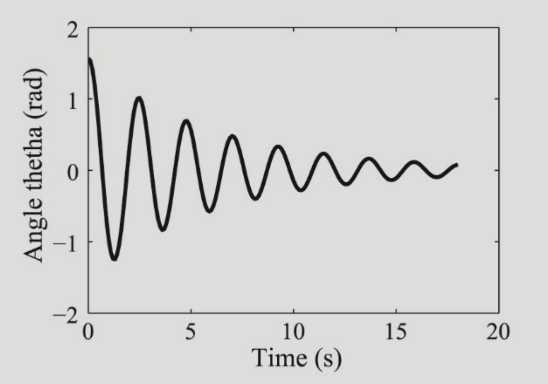
$$\frac{dy}{dx} = w \qquad y(a) = A$$

$$\frac{dw}{dx} = f(x, y, w) \qquad w(a) = B$$

Ex 7-8: Second order differential equation

- Solve the following second order differential equation with the fourth-order Runge-Kutta method

$$\frac{d^2\theta}{dt^2} = -\frac{c}{m}\frac{d\theta}{dt} - \frac{g}{L}\sin\theta$$



- This equation can be written as:

$$\frac{d\theta}{dt} = w \qquad \theta(0) = \frac{\pi}{2}$$

$$\frac{dw}{dt} = -\frac{c}{m}w - \frac{g}{L}\sin\theta \qquad w(0) = 0$$

- The domain is from t=0 to t=18 with h=0.1

Higher-order IVP

- For the nth-order IVP with the initial condition:

$$\frac{d^n y}{dx^n} = f(x, y, \frac{dy}{dx}, \frac{d^2 y}{dx^2}, ..., \frac{d^{n-1} y}{dx^{n-1}})$$

$$y(a) = A_1 \qquad \left.\frac{dy}{dx}\right|_{x=a} = A_2 \qquad \left.\frac{d^2 y}{dx^2}\right|_{x=a} = A_3, ..., \left.\frac{d^{n-1} y}{dx^{n-1}}\right|_{x=a} = A_n$$

- Transform the nth-order ODE into a system of n first-order ODEs using the similar method

$$w_1 = \frac{dy}{dx} \qquad y(a) = A_1$$

$$w_2 = \frac{dw_1}{dx} = \frac{d^2 y}{d^2 x} \qquad w_1(a) = A_2$$

$$w_3 = \frac{dw_2}{dx} = \frac{d^3 y}{d^3 x} \qquad w_2(a) = A_3$$

$$\vdots \qquad \vdots$$

$$w_{n-1} = \frac{dw_{n-2}}{dx} = \frac{d^{n-1} y}{d^{n-1} x} \qquad w_{n-2}(a) = A_{n-1}$$

System of higher-order IVP

- Any coupled system of higher-order ODEs can also be written as a system of first-order ODEs
- Consider a system of two second order ODEs:

$$\frac{d^2x}{dt^2} = f(x, y, t, \frac{dx}{dt}, \frac{dy}{dt}) \qquad \frac{d^2y}{dt^2} = f(x, y, t, \frac{dx}{dt}, \frac{dy}{dt})$$

- This can be written as a system of **four** first-order ODEs:

$$\frac{dx}{dt} = u \qquad \frac{dy}{dt} = w$$

$$\frac{du}{dt} = f(x, y, t, u, w) \qquad \frac{dw}{dt} = f(x, y, t, u, w)$$

MATLAB Built-in functions

Solver Name	Description
`ode45`	For nonstiff problems, best to apply as a first try for most problems. Single-step method based on fourth and fifth-order explicit Runge–Kutta methods.
`ode23`	For nonstiff problems. Single-step method based on second and third-order explicit Runge–Kutta methods. Often quicker but less accurate than `ode45`.
`ode113`	For nonstiff problems. Multistep method based on Adams–Bashforth–Moulton methods.
`ode15s`	For stiff problems. Multistep method that uses a variable-order method. Low to medium accuracy.
`ode23s`	For stiff problems. One-step solver. Can solve some problems that `ode15` cannot. Low accuracy.
`ode23t`	For moderately stiff problems. Low accuracy.
`ode23tb`	For stiff problems. Uses an implicit Runge–Kutta method. Often more efficient than `ode15s`.