Contents

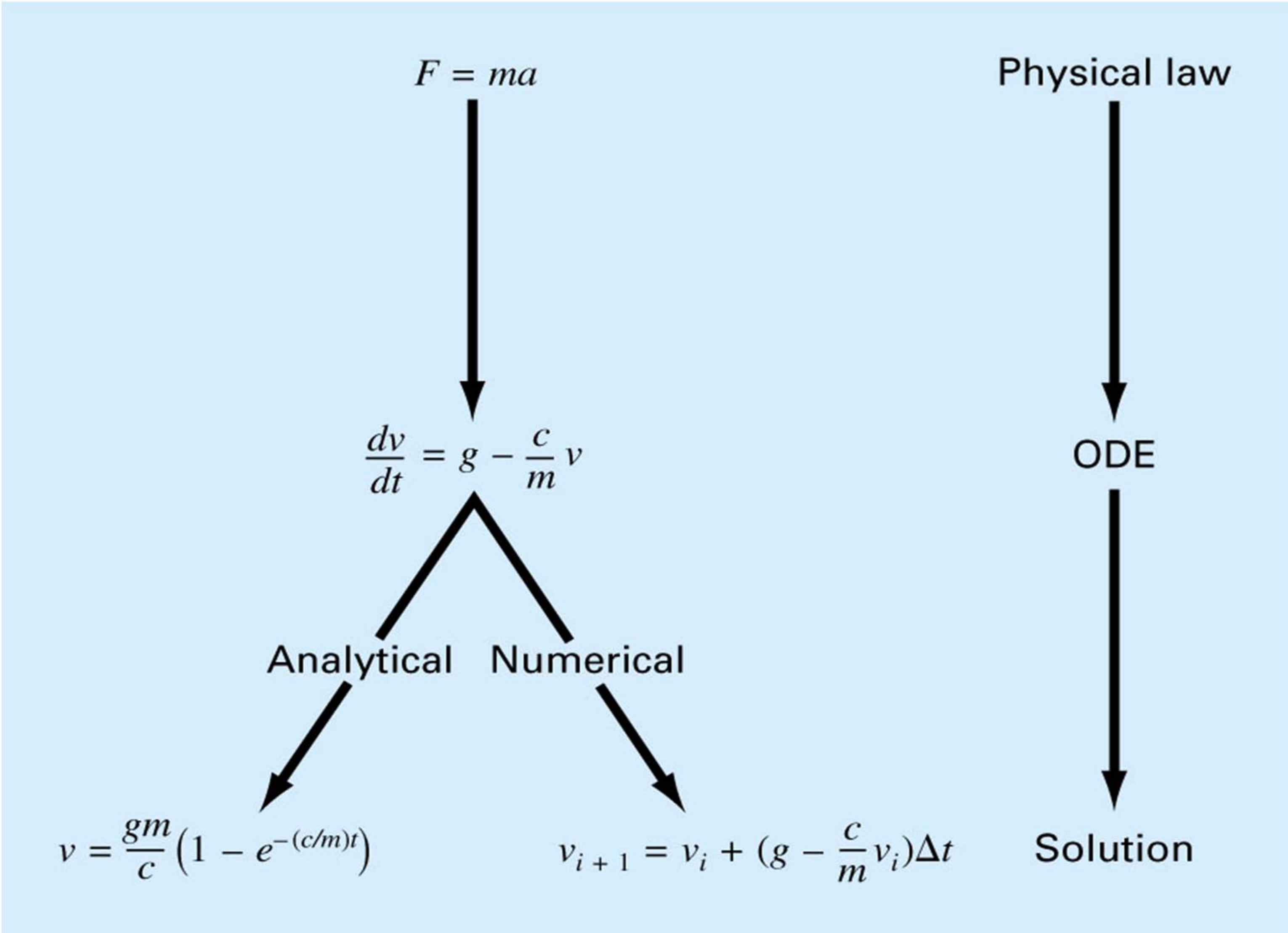
- One-step method: how to estimate the slope?
- Euler's (Euler-Cauchy, point-slope) method: first-order
- Improvement of Euler's method
  - Heun's method
  - Midpoint (improved polygon) method
- Runge-Kutta methods
  - Second-order
  - Third-order
  - Fourth-order
  - Higher-order

Fundamental Laws

In terms of the rate of change of variables (t = time, x = position)

Law	Mathematical Expression	Variables and Parameters
Newton's second law of motion	$\frac{dv}{dt} = \frac{F}{m}$	Velocity (v), force (F), and mass (m)
Fourier's heat law	$q = -k' \frac{dT}{dx}$	Heat flux (q), thermal conductivity (k') and temperature (T)
Fick's law of diffusion	$J = -D \frac{dc}{dx}$	Mass flux (J), diffusion coefficient (D), and concentration (c)
Faraday's law (voltage drop across an inductor)	$\Delta V_L = L \frac{di}{dt}$	Voltage drop (ΔV_L), inductance (L), and current (i)

Engineering Problem Solving



Ordinary Differential Equation

- Differential Equation
  - ODE: # of independent variable = 1
  - PDE: # of independent variable ≥ 2
- ODE
  - Linear / nonlinear

① $y' = x + 1$ ② $y' = y + 1$ ③ $y' = x^3 + 1$

④ $y' = \left(x^3 + 1\right) y$ ⑤ $y' = y^2 + 1$ ⑥ $y' = \sin y$

- Definition of linear ODE

$$\underbrace{a_n(x)\, y^{(n)} + \cdots + a_1(x)\, y' + a_0(x)\, y}_{\text{linear combination of } y, y', y'', \ldots, y^{(n)}} = F(x)$$

Higher Order Linear ODE

- Conversion to coupled 1st order ODEs

$$a_n(x)\,y^{(n)} + \cdots + a_1(x)\,y' + a_0(x)\,y = F(x) \leftarrow n\text{-th order ODE}$$

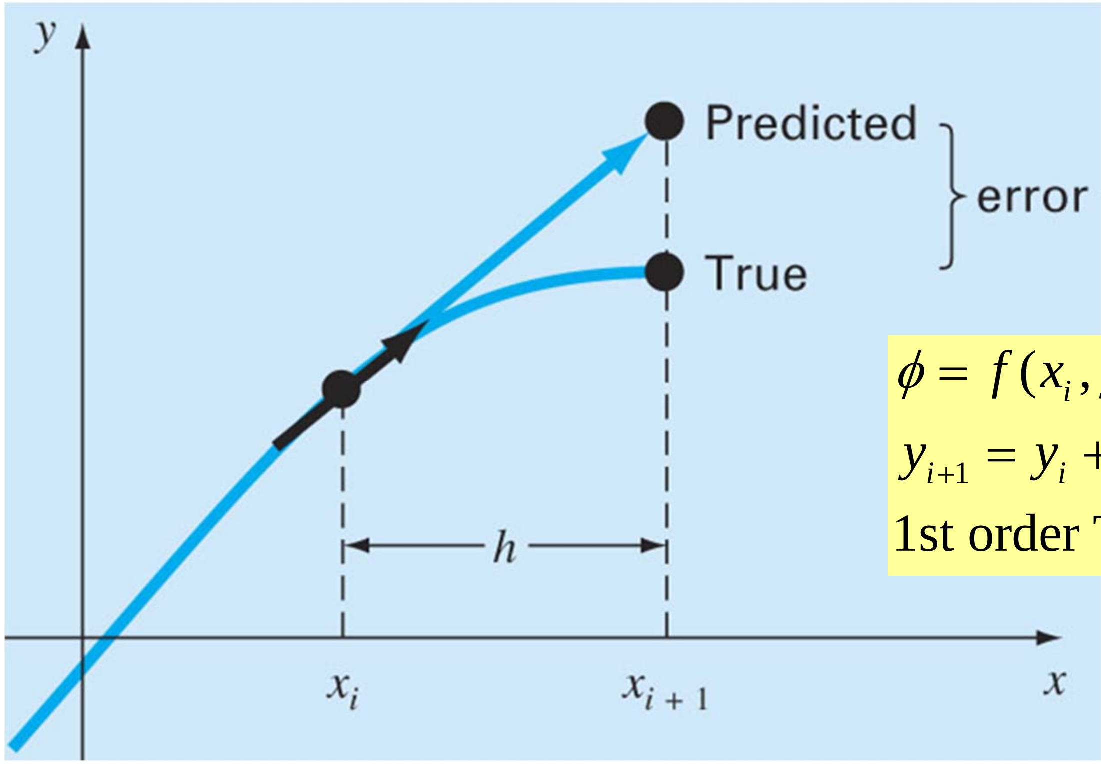
$$\left.\begin{aligned} y_0 &= y \\ y_1 &= y' \\ y_2 &= y'' \\ &\vdots \\ y_{n-1} &= y^{(n-1)} \end{aligned}\right\} \rightarrow \left\{\begin{aligned} y_0' &= y_1 \\ y_1' &= y_2 \\ y_2' &= y_3 \\ &\vdots \\ y_{n-1}' &= \frac{F(x) - a_0 y_0 - a_1 y_1 - \cdots - a_{n-1} y_{n-1}}{a_n} \end{aligned}\right.$$

$$\Rightarrow \text{vector form}: \mathbf{y}' = \mathbf{f}(x, \mathbf{y}) \quad \begin{cases} x: \text{independent variable} \\ \mathbf{y}: \text{dependent variable} \end{cases}$$

Euler's Method

- solve ordinary differential equations of the form

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



$\phi = f(x_i, y_i) \leftarrow$ slope at x_i

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

1st order Taylor approximation

Error Analysis

- Global truncation error: O(h) = # of steps x local error
  - Local truncation error: Taylor series, O(h^2)
  - Propagated truncation error: ?
- Round-off error

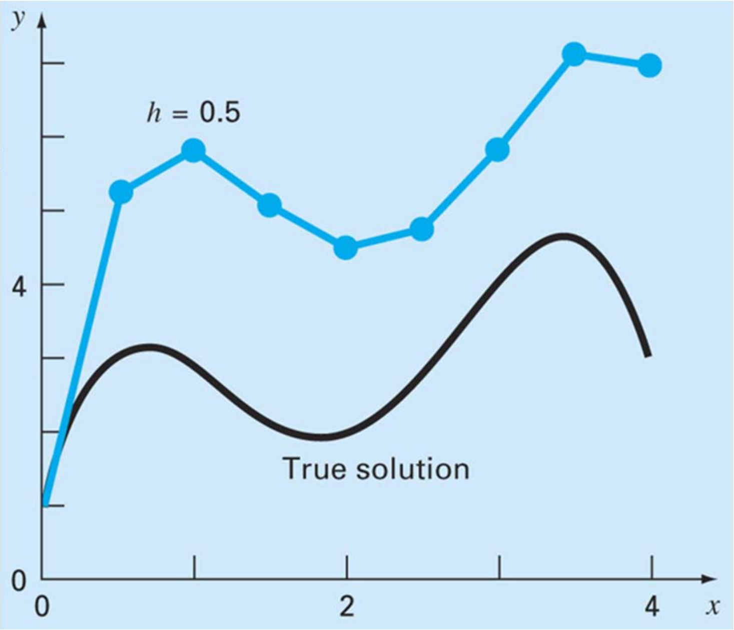
$$y_{i+1} = y_i + y_i' h + \frac{y_i''}{2!} h^2 + \cdots + \frac{y_i^{(n)}}{n!} h^n + R_n$$

$$y_{i+1} = y_i + f\left(x_i, y_i\right) h + \frac{f'\left(x_i, y_i\right)}{2!} h^2 + \cdots + \frac{f^{(n-1)}\left(x_i, y_i\right)}{n!} h^n + O\left(h^{n+1}\right)$$

$$E_t = \frac{f'\left(x_i, y_i\right)}{2!} h^2 + \cdots + \frac{f^{(n-1)}\left(x_i, y_i\right)}{n!} h^n + O\left(h^{n+1}\right)\text{: true local truncation error}$$

$$\xrightarrow{\text{small } h} E_a = \frac{f'\left(x_i, y_i\right)}{2!} h^2 = O\left(h^2\right)\text{: approximate local truncation error}$$

Example (1)

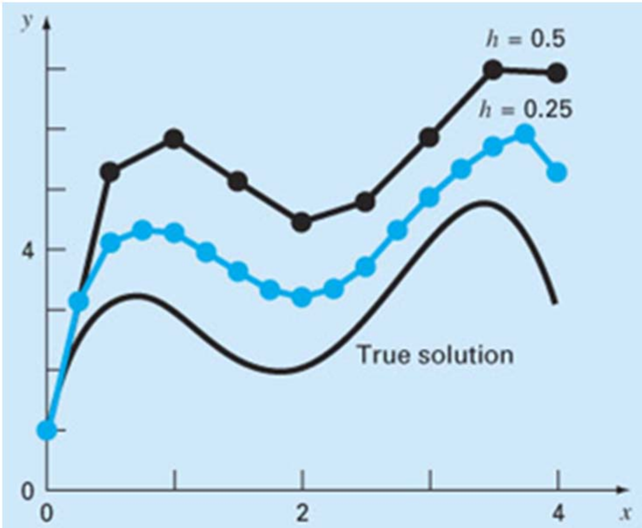


$$\frac{dy}{dx} = -2x^3 + 12x^2 - 20x + 8.5$$

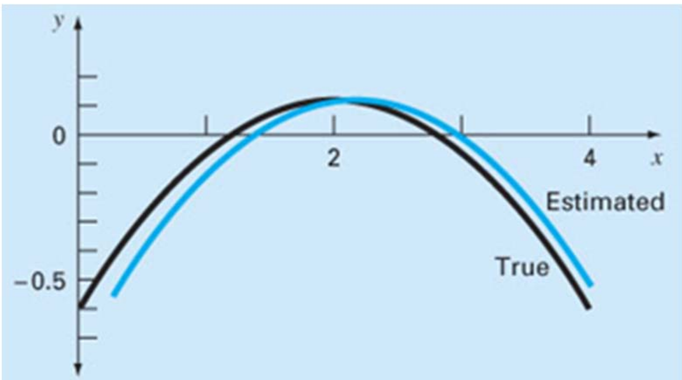
$$y = -0.5x^4 + 4x^3 - 10x^2 + 8.5x + 1$$

			Percent Relative Error	
x	y_{true}	y_{Euler}	**Global**	**Local**
0.0	1.00000	1.00000		
0.5	3.21875	5.25000	−63.1	−63.1
1.0	3.00000	5.87500	−95.8	−28.0
1.5	2.21875	5.12500	131.0	−1.41
2.0	2.00000	4.50000	−125.0	20.5
2.5	2.71875	4.75000	−74.7	17.3
3.0	4.00000	5.87500	46.9	4.0
3.5	4.71875	7.12500	−51.0	−11.3
4.0	3.00000	7.00000	−133.3	−53.0

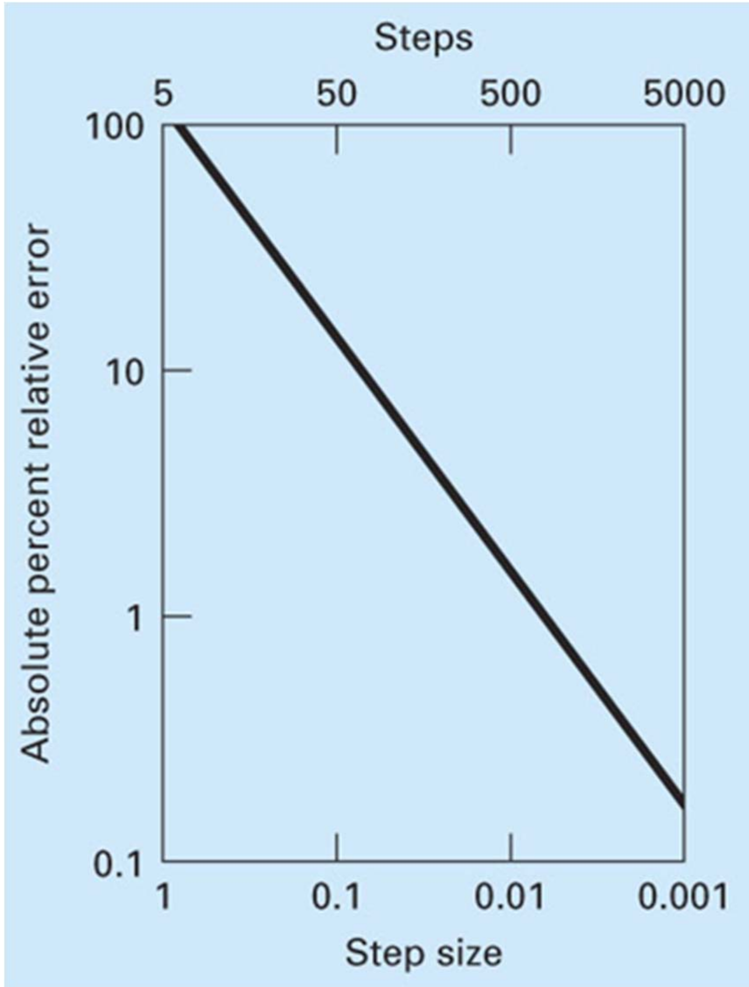
Example (2)



y
h = 0.5
h = 0.25
4
True solution
0
0
2
4
x



y
0
2
4
x
Estimated
True
−0.5



Steps
5
50
500
5000
100
10
1
0.1
Absolute percent relative error
1
0.1
0.01
0.001
Step size

Improvements of Euler's method

- A fundamental source of error in Euler's method
  - derivative at the beginning of the interval is assumed to apply across the entire interval
- Two simple modifications to circumvent this shortcoming
  - Heun's Method
  - Midpoint (or Improved Polygon) Method

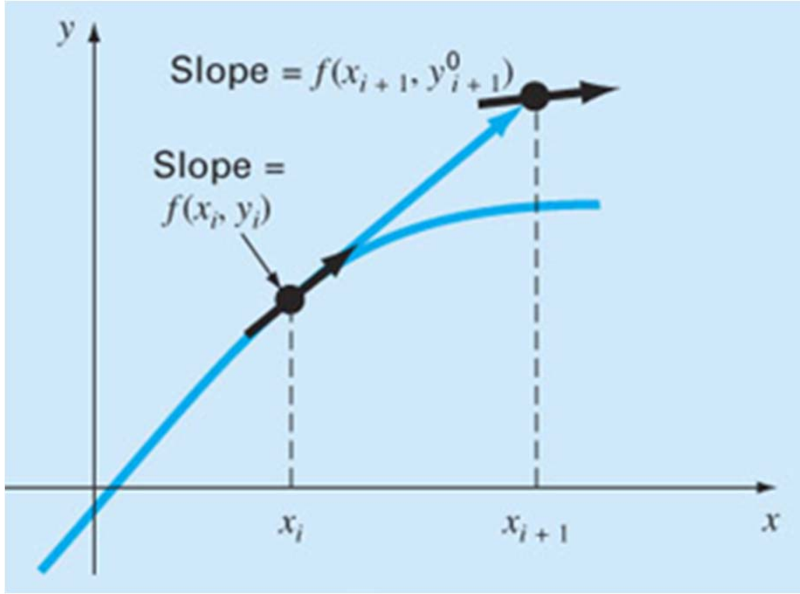
Heun's Method

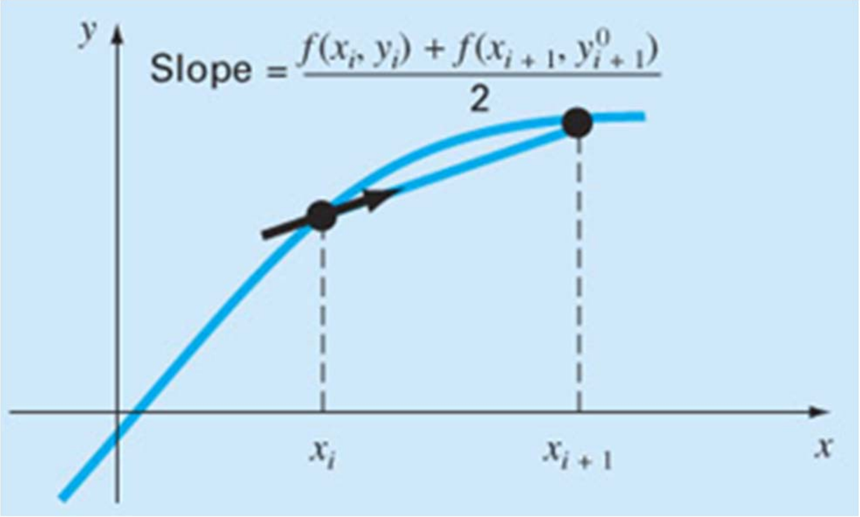
- Predictor-corrector approach

$$\text{slope: } f(x_i, y_i) \xrightarrow{y_{i+1}^0} f(x_{i+1}, y_{i+1}^0) \rightarrow \frac{1}{2}\left[f(x_i, y_i) + f(x_{i+1}, y_{i+1}^0) \right]$$

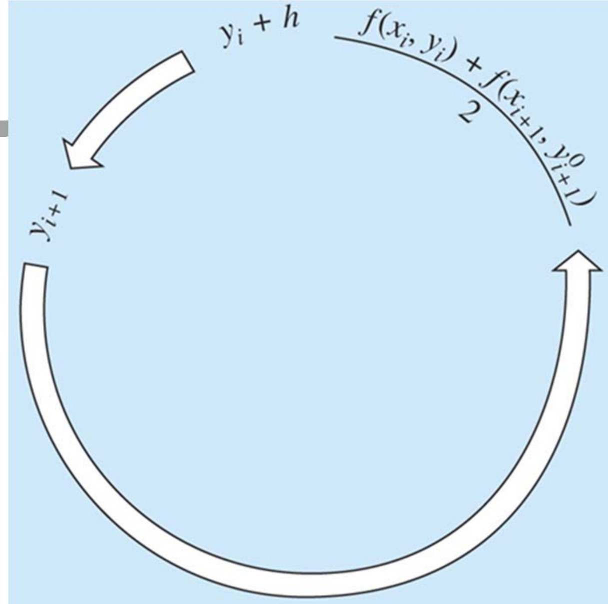
$$\text{Predictor: } y_{i+1}^0 = y_i + f(x_i, y_i)h$$

$$\text{Corrector: } y_{i+1} = y_i + \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^0)}{2}h$$





Example (1)



$$\frac{dy}{dx} = 4e^{0.8x} - 0.5y$$

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

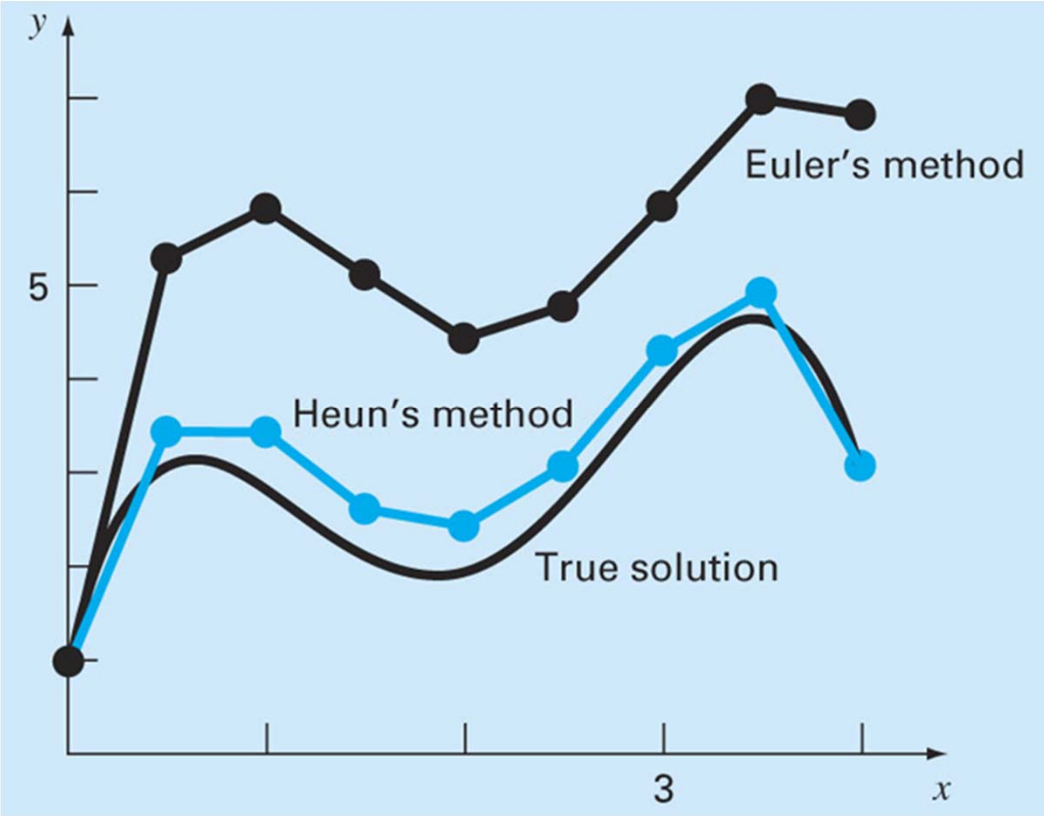
		Iterations of Heun's Method			
		1		15	
x	y_{true}	y_{Heun}	$\lvert\varepsilon_t\rvert$ (%)	y_{Heun}	$\lvert\varepsilon_t\rvert$ (%)
0	2.0000000	2.0000000	0.00	2.0000000	0.00
1	6.1946314	6.7010819	8.18	6.3608655	2.68
2	14.8439219	16.3197819	9.94	15.3022367	3.09
3	33.6771718	37.1992489	10.46	34.7432761	3.17
4	75.3389626	83.3377674	10.62	77.7350962	3.18

Example (2)

slope: $f(x_i, y_i) \rightarrow f(x_i)$

Predictor: no need

Corrector: $y_{i+1} = y_i + \dfrac{f(x_i) + f(x_{i+1})}{2} h \leftarrow$ Trapezoidal rule: $E_t = -\dfrac{f''(\xi)}{12} h^3$

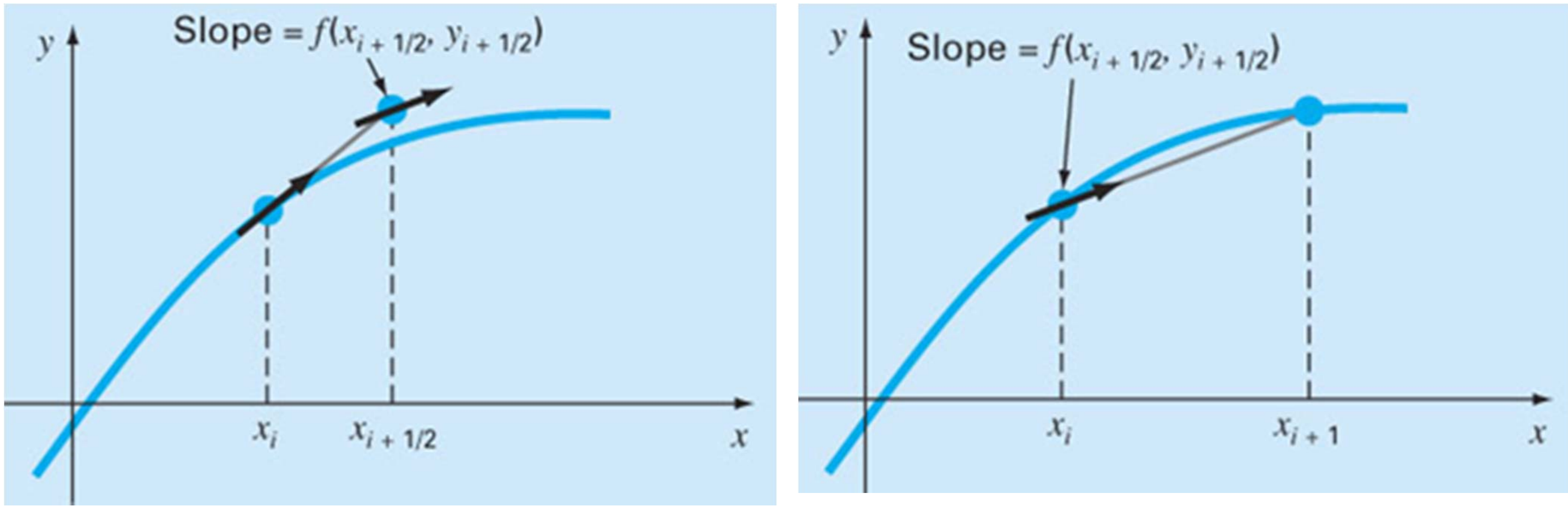


Midpoint (or Improved Polygon) Method

- Use Euler's method to predict a value of y at the midpoint of the interval (without iteration for corrector)

$$f(x_i, y_i) \rightarrow y_{i+1/2} = y_i + f(x_i, y_i)\frac{h}{2} \rightarrow y'_{i+1/2} = f(x_{i+1/2}, y_{i+1/2})$$

$$y_{i+1} = y_i + f(x_{i+1/2}, y_{i+1/2})h$$



Runge-Kutta Methods

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

$$\begin{cases} \phi = a_1 k_1 + a_2 k_2 + \cdots + a_n k_n : \text{ increment function} \\ a's : \text{constants} \\ k's : \text{recurrence relationships} \\ k_1 = f(x_i, y_i) \\ k_2 = f(x_i + p_1 h, y_i + q_{11} k_1 h) : \ p\text{'s and } q\text{'s are constants} \\ k_3 = f(x_i + p_2 h, y_i + q_{21} k_1 h + q_{22} k_2 h) \\ \vdots \\ k_n = f(x_i + p_{n-1} h, y_i + q_{n-1} k_1 h + q_{n-1,2} k_2 h + \cdots + q_{n-1,n-1} k_{n-1} h) \end{cases}$$

How to determine coefficients?

$$n \rightarrow \left(a's, p's, q's \leftrightarrow \text{Taylor series expansion} \right)$$

$$n = 1: \text{ Euler's method}$$

2^{nd}-order RK (1)

$$\begin{cases} y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h = y_i + \left(a_1 + a_2\right) f(x_i, y_i)h + \left(a_2 p_1 \dfrac{\partial f}{\partial x} + a_2 q_{11} f(x_i, y_i) \dfrac{\partial f}{\partial y} \right) h^2 + O\left(h^3\right) \\ k_1 = f(x_i, y_i) \\ k_2 = f(x_i + p_1 h, y_i + q_{11} k_1 h) \left[= f(x_i, y_i) + p_1 h \dfrac{\partial f}{\partial x} + q_{11} k_1 h \dfrac{\partial f}{\partial y} + O\left(h^2\right) \right] \end{cases}$$

$$\leftrightarrow y_{i+1} = y_i + f(x_i, y_i)h + \frac{f'(x_i, y_i)}{2!} h^2 = y_i + f(x_i, y_i)h + \frac{1}{2!} \left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \frac{dy}{dx} \right) h^2$$

$$\rightarrow \begin{cases} a_1 + a_2 = 1 \\ a_2 p_1 = \dfrac{1}{2} \\ a_2 q_{11} = \dfrac{1}{2} \end{cases} \rightarrow \begin{cases} a_1 = 1 - a_2 \\ p_1 = q_{11} = \dfrac{1}{2a_2} \end{cases} \rightarrow \begin{cases} a_2 = \dfrac{1}{2}: \text{ Heun's method} \\ a_2 = 1: \text{ Midpoint method} \\ a_2 = \dfrac{2}{3}: \text{ Ralson's method} \\ \leftarrow \text{minimum bound on truncation error} \end{cases}$$

2^{nd}-order RK (2)

Heun's method $\left(a_2 = \dfrac{1}{2}\right): a_1 = \dfrac{1}{2},\ p_1 = q_{11} = 1 \rightarrow \begin{cases} k_1 = f(x_i, y_i) \\ k_2 = f(x_i + h, y_i + k_1 h) \end{cases}$

$$y_{i+1} = y_i + \left(\frac{1}{2}k_1 + \frac{1}{2}k_2\right)h = y_i + \frac{f(x_i, y_i) + f(x_i + h, y_i + f(x_i, y_i)h)}{2}h = y_i + \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^0)}{2}h$$

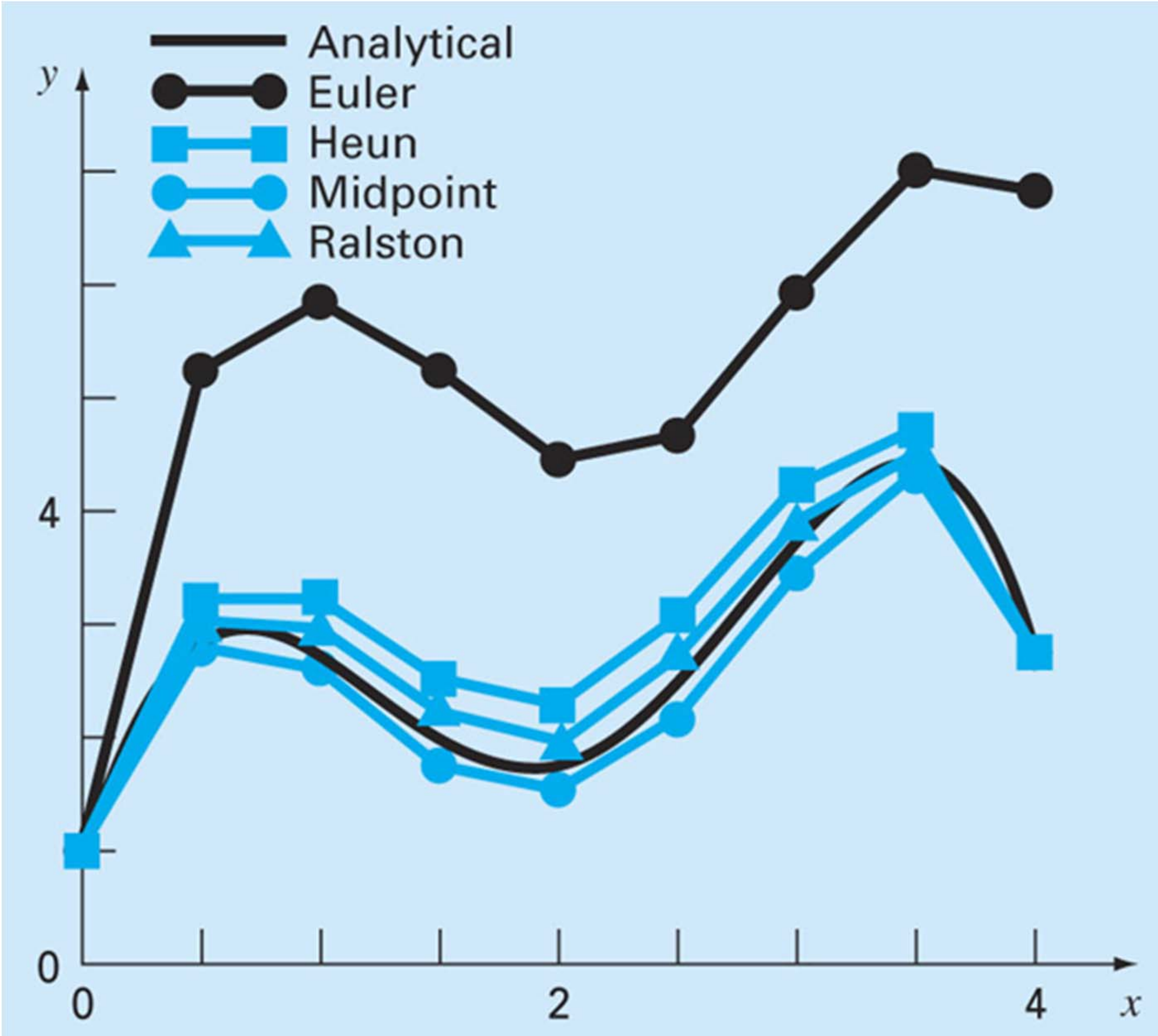
Midpoint method $(a_2 = 1): a_1 = 0,\ p_1 = q_{11} = \dfrac{1}{2} \rightarrow \begin{cases} k_1 = f(x_i, y_i) \\ k_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1 h) \end{cases}$

$$y_{i+1} = y_i + k_2 h = y_i + f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}f(x_i, y_i)h)h = y_i + f(x_{i+1/2}, y_{i+1/2})h$$

Ralson's method $\left(a_2 = \dfrac{2}{3}\right): a_1 = \dfrac{1}{3},\ p_1 = q_{11} = \dfrac{3}{4} \rightarrow \begin{cases} k_1 = f(x_i, y_i) \\ k_2 = f(x_i + \frac{3}{4}h, y_i + \frac{3}{4}k_1 h) \end{cases}$

$$y_{i+1} = y_i + \left(\frac{1}{3}k_1 + \frac{2}{3}k_2\right)h = y_i + \left[\frac{f(x_i, y_i)}{3} + \frac{2f(x_i + \frac{3}{4}h, y_i + \frac{3}{4}f(x_i, y_i)h)}{3}\right]h$$

Example



3rd-order and 4th-order RK

[Third-order RK]

$$y_{i+1} = y_i + \frac{1}{6}\left(k_1 + 4k_2 + k_3\right)h$$

$$\begin{cases} k_1 = f(x_i, y_i) \\ k_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1 h) \\ k_3 = f(x_i + h, y_i - k_1 h + 2k_2 h) \end{cases}$$

eight unknowns

$$\left(a_1, a_2, a_3, p_1, p_2, q_{11}, q_{21}, q_{22}\right)$$

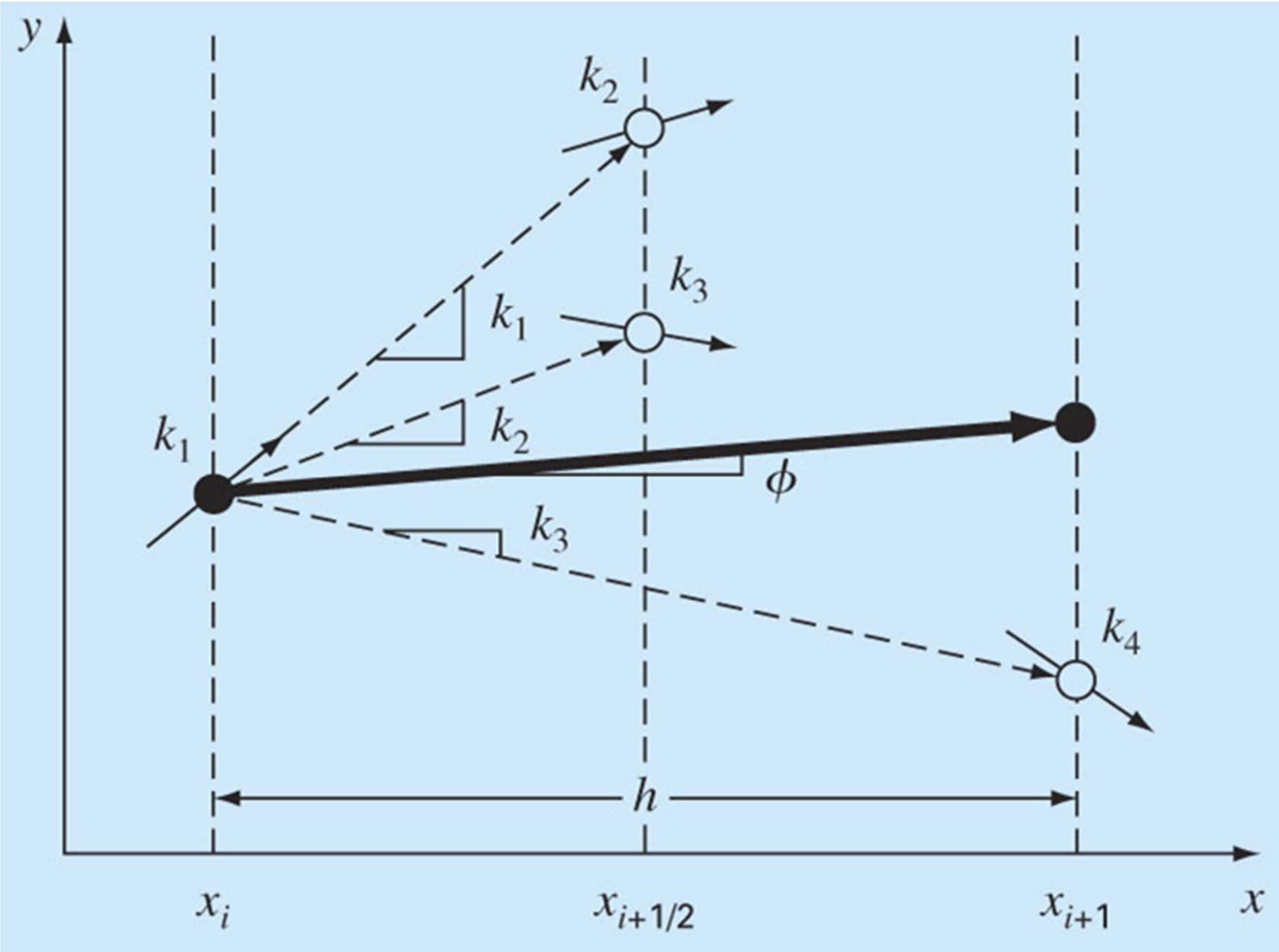
six equations

[Fourth-order RK]

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

$$\begin{cases} k_1 = f(x_i, y_i) \\ k_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1 h) \\ k_3 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2 h) \\ k_4 = f(x_i + h, y_i + k_3 h) \end{cases}$$

Slope Estimates



Higher-Order RK

[Fifth-order RK]

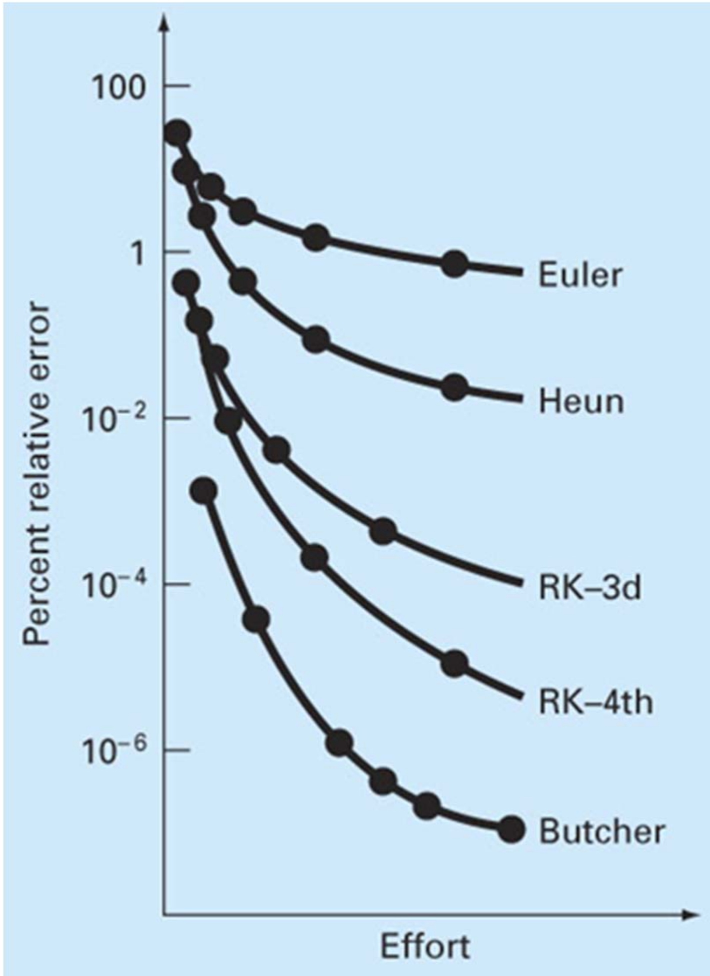
$$y_{i+1} = y_i + \frac{1}{90}\left(7k_1 + 32k_3 + 12k_4 + 32k_5 + 7k_6\right)h$$

$$\begin{cases} k_1 = f(x_i, y_i) \\ k_2 = f(x_i + \frac{1}{4}h, y_i + \frac{1}{4}k_1 h) \\ k_3 = f(x_i + \frac{1}{4}h, y_i + \frac{1}{8}k_1 h + \frac{1}{8}k_2 h) \\ k_4 = f(x_i + \frac{1}{2}h, y_i - \frac{1}{2}k_2 h + k_3 h) \\ k_5 = f(x_i + \frac{3}{4}h, y_i + \frac{3}{16}k_1 h + \frac{9}{16}k_4 h) \\ k_6 = f(x_i + h, y_i - \frac{3}{7}k_1 h + \frac{2}{7}k_2 h + \frac{12}{7}k_3 h - \frac{12}{7}k_4 h + \frac{8}{7}k_5 h) \end{cases}$$

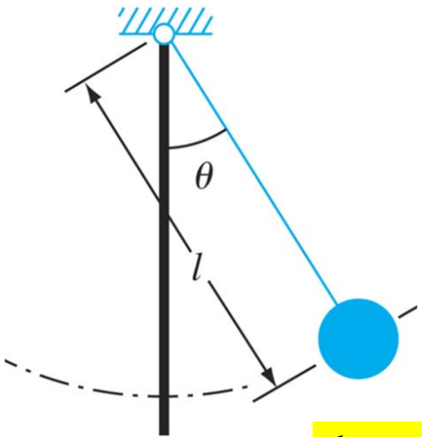
Example

$$\frac{dy}{dx} = 4e^{0.8x} - 0.5y$$

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

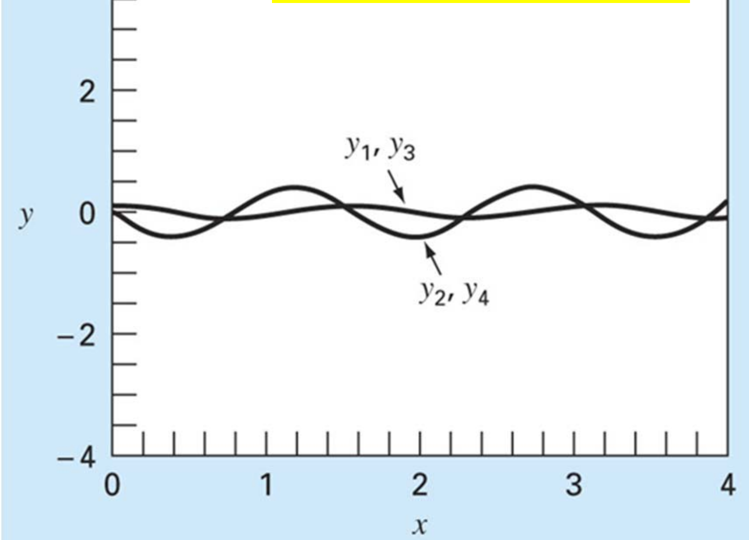


Example: Swinging Pendulum

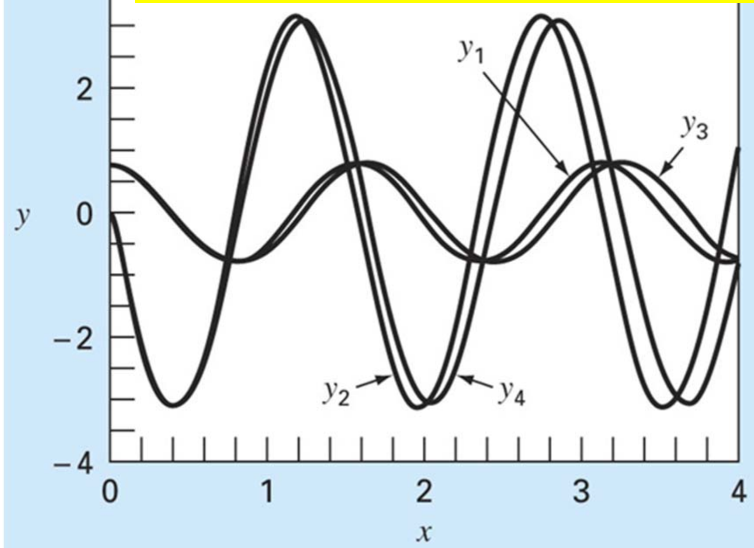


$$\left[\begin{array}{l} \dfrac{d^2\theta}{dt^2}+\dfrac{g}{l}\sin\theta=0 \\ \xrightarrow{\theta=y_3}\begin{cases}\dfrac{dy_3}{dt}=y_4\\ \dfrac{dy_4}{dt}=-16.1\sin(y_3)\end{cases}\end{array}\right] \xrightarrow[\sin\theta\cong\theta]{\text{linearization}} \left[\begin{array}{l} \dfrac{d^2\theta}{dt^2}+\dfrac{g}{l}\theta=0 \\ \xrightarrow{\theta=y_1}\begin{cases}\dfrac{dy_1}{dt}=y_2\\ \dfrac{dy_2}{dt}=-16.1y_1\end{cases}\end{array}\right]$$

$$\begin{cases}y_1=y_3=0.1\text{ radian}\\ y_2=y_4=0\end{cases}$$



$$\begin{cases}y_1=y_3=\pi/4=0.7854\text{ radian}\\ y_2=y_4=0\end{cases}$$

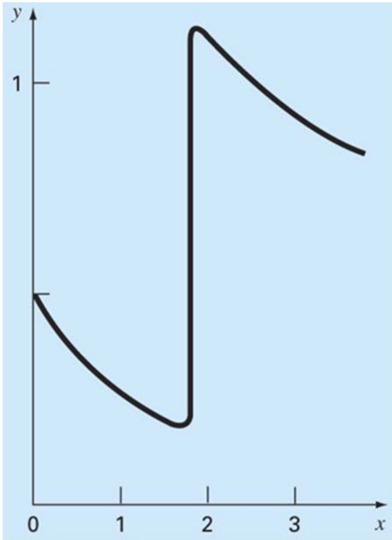


Adaptive RK Methods

  - Very small step size to accurately capture the impulse behavior

- Adaptive step-size control
  - Adjust the step size automatically?
  - Based on the error estimate
    - Same order / different step sizes: adaptive RK (step having)
    - Different order / same step size: RK Fehlberg (embedded RK)

$$h_{new} = h_{present} \left| \frac{\Delta_{new}}{\Delta_{present}} \right|^{\alpha} \quad \text{where} \begin{cases} h: \text{ step size} \\ \Delta: \text{ accuracy} \\ \Delta_{new} = \varepsilon y_{scale} \\ y_{scale} = |y| + \left| h \dfrac{dy}{dx} \right| \end{cases}$$

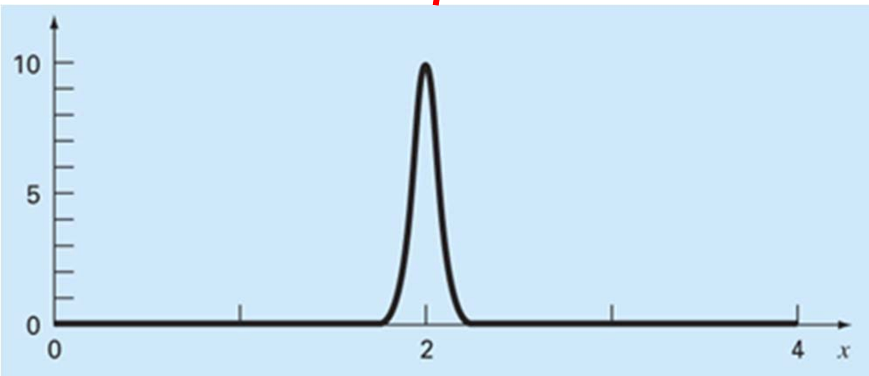


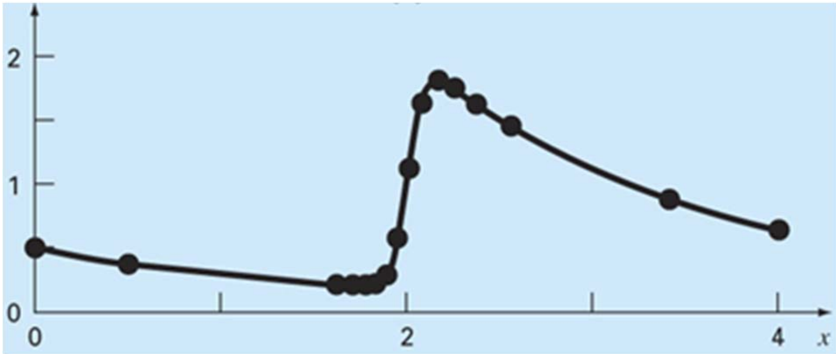
Adaptive Fourth-Order RK Scheme

$$\frac{dy}{dx} + 0.6y = \underbrace{10e^{-(x-2)^2/\left[2(0.075)^2\right]}} \quad \text{and} \quad y(0) = 0.5$$

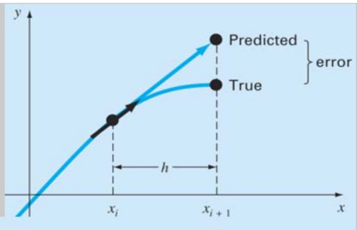
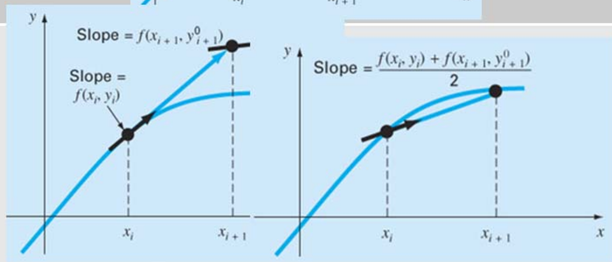
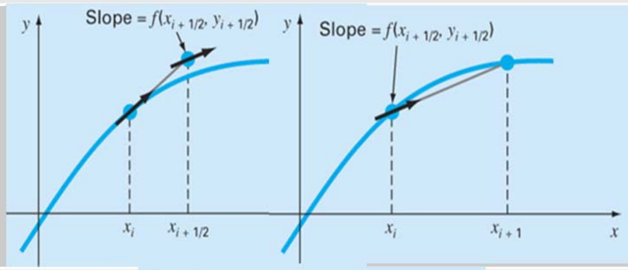
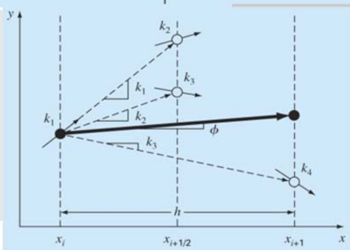
$$y_h = 0.5e^{-0.6x}$$

$$y_p : \text{ abrupt transition in the vicinity of } x = 2$$





Summary: ODE Integration Scheme

method		Global error	Exact for	Runge-Kutta
Euler's		O(h)	Linear function	1st order
Heun's		$O(h^2)$	Quadratic function	2nd order
Midpoint				
Classical Runge-Kutta		$O(h^4)$	Quartic function	4th order

Contents

- Stiff ODEs
- Multistep methods
- Boundary Value Problems
- Eigenvalue Problems

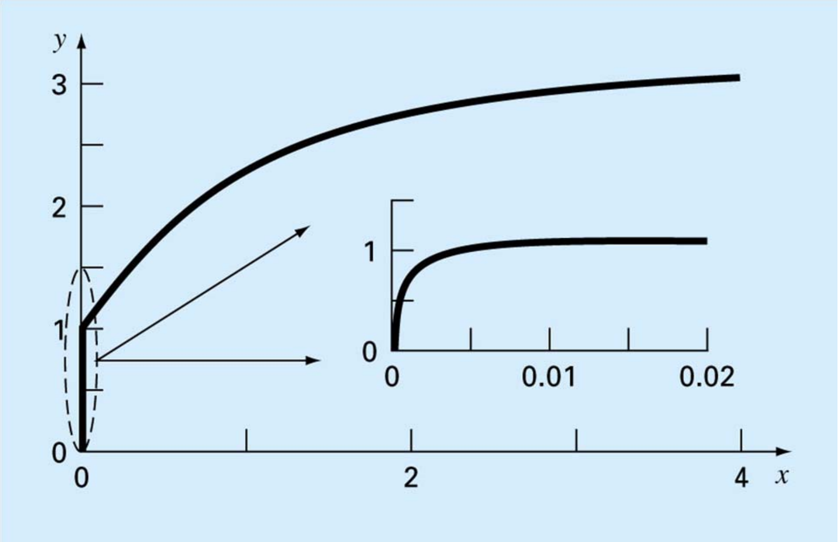
- Stiff ODEs
  - ODEs that have both fast and slow components to their solution
  - Remedy: implicit solution technique
- Multistep methods
  - Algorithms that retain information of previous steps to more effectively capture the trajectory of the solution

Stiffness

- Stiff system
  - one involving rapidly changing components together with slowly changing ones

$$\frac{dy}{dt} = -1000y + 3000 - 2000e^{-t} \text{ and } y(0) = 0$$

$$y = 3 - 0.998\underbrace{e^{-1000t}}_{\substack{\text{dominate}\\ \text{during}\\ t<0.005}} - 2.002e^{-t}$$



  - Step size for stability

$$\frac{dy}{dt} = -ay \text{ and } y(0) = y_0 \to \begin{cases} \text{analytic: } y = y_0 e^{-at} \\ \text{numerical (Euler's method): } y_{i+1} = y_i + \dfrac{dy_i}{dt} h \end{cases}$$

$$y_{i+1} = y_i - ay_i h \to y_{i+1} = y_i(1 - ah) \xrightarrow{\text{stability}} |1 - ah| < 1 \to 0 < h < \frac{2}{a}$$

Explicit vs. Implicit

- Implicit
  - Unknown on both sides of the equation

$$\text{Euler's method}\begin{cases}\text{explicit (forward): } y_{i+1} = y_i + \dfrac{dy_i}{dt}h \\ \text{implicit (backward): } y_{i+1} = y_i + \dfrac{dy_{i+1}}{dt}h\end{cases}$$

$$y_{i+1} = y_i - a y_{i+1} h \rightarrow y_{i+1} = \frac{y_i}{1+ah} \leftarrow \begin{cases}\text{unconditionally stable regardless of step size} \\ \text{only first-order accurate} \\ \rightarrow \text{second-order? implicit trapezoidal rule integration scheme}\end{cases}$$

Example

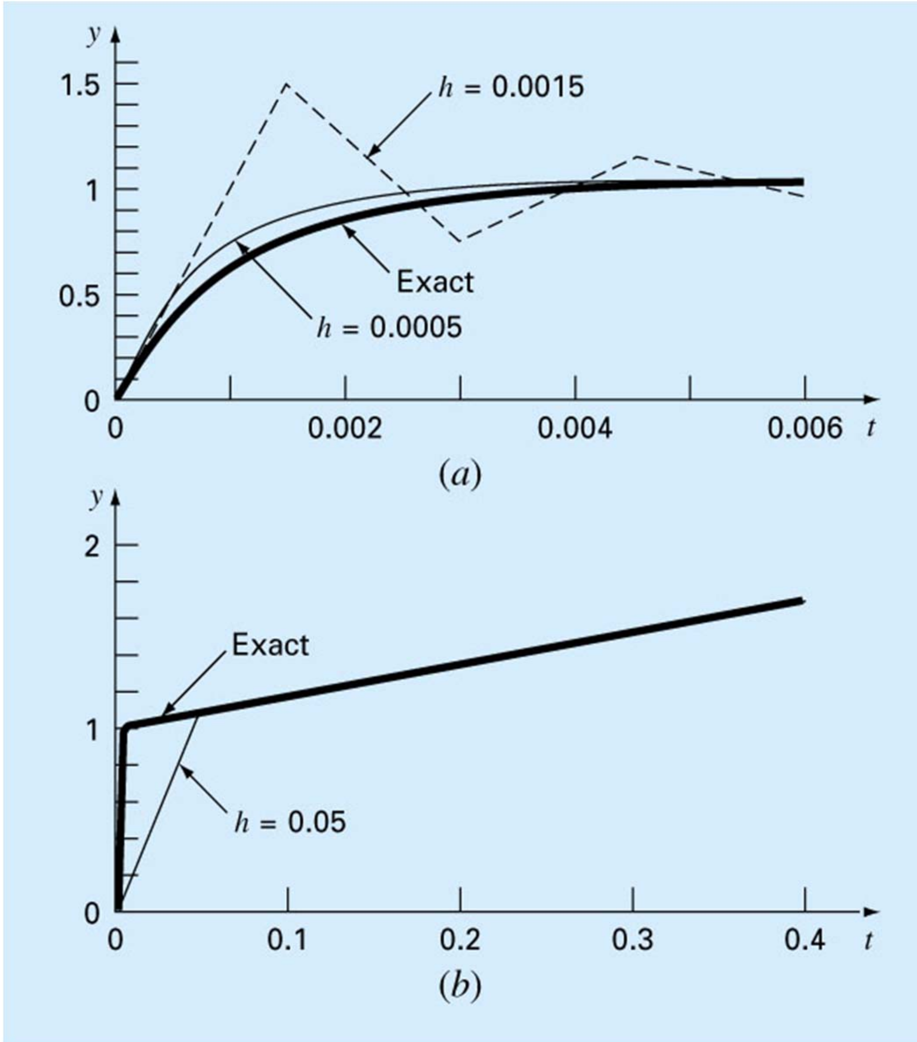
$$\frac{dy}{dt} = -1000y + 3000 - 2000e^{-t} \text{ and } y(0) = 0$$

$$y = 3 - 0.998e^{-1000t} - 2.002e^{-t}$$

(a) explicit $(t = 0 \sim 0.006)$:

$$h = 0.0005,\ h = 0.0015\ \left(h_{th} = 0.002\right)$$

(b) implicit $(t = 0 \sim 0.4)$: $h = 0.05$



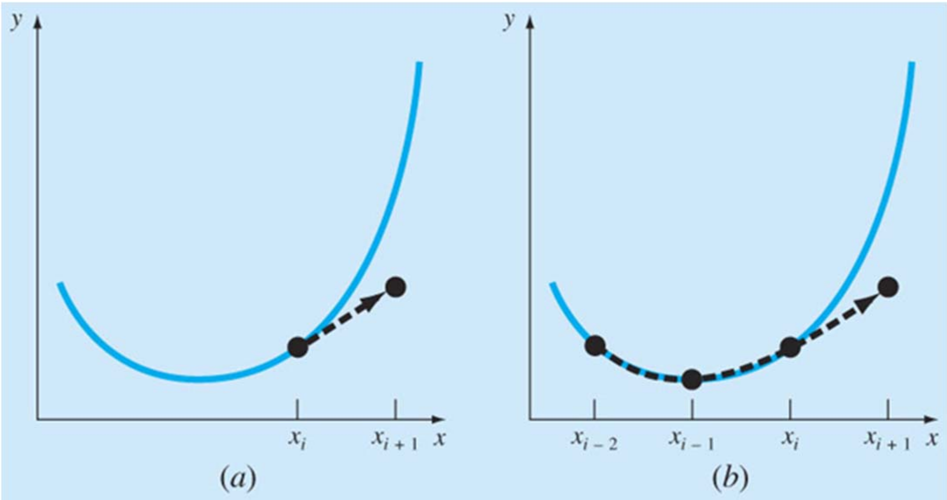
Systems of ODEs

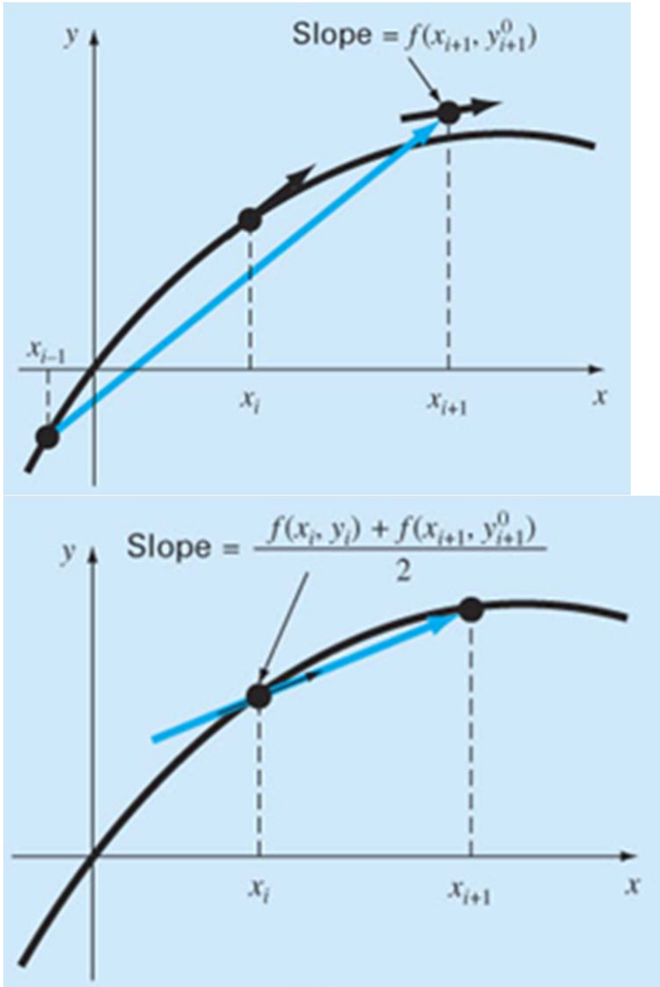
$$\begin{cases} \dfrac{dy_1}{dt} = -5y_1 + 3y_2 \text{ and } y_1(0) = 52.29 \\ \dfrac{dy_2}{dt} = 100y_1 - 301y_2 \text{ and } y_2(0) = 83.82 \end{cases} \rightarrow \begin{cases} y_1 = 52.96e^{-3.9899t} - 0.67e^{-302.0101t} \\ y_2 = 17.83e^{-3.9899t} + 65.99e^{-302.0101t} \end{cases}$$

$$\begin{cases} y_{1,i+1} = y_{1,i} + \left(-5y_{1,i+1} + 3y_{2,i+1}\right)h \\ y_{2,i+1} = y_{2,i} + \left(100y_{1,i+1} - 301y_{2,i+1}\right)h \end{cases} \rightarrow \begin{cases} (1+5h)\, y_{1,i+1} - 3hy_{2,i+1} = y_{1,i} \\ -100y_{1,i+1} + (1+301)\, hy_{2,i+1} = y_{2,i} \end{cases}$$

- Solve a set of simultaneous equations
- Nonlinear ODEs: difficult to solve

One-step vs. Multistep





- Non-self starting Heun Method

Predictor: Euler's method

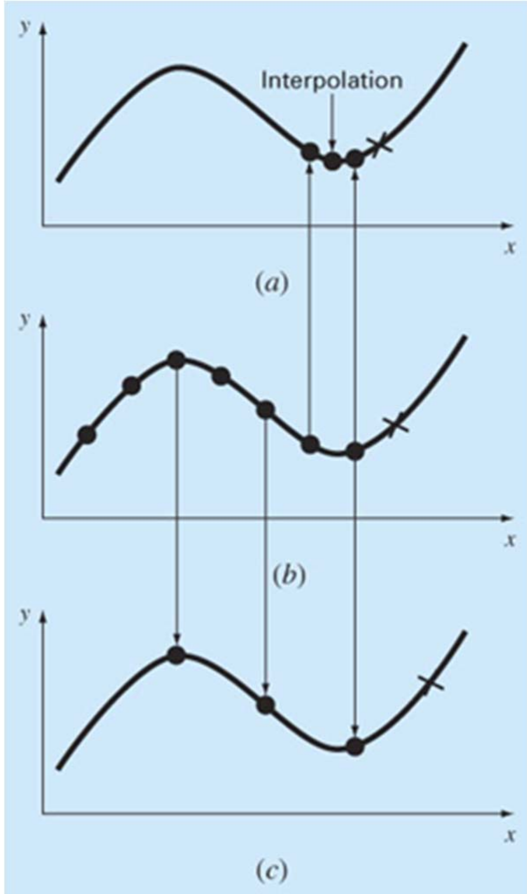
$$\begin{cases} y^0_{i+1} = y_i + f(x_i, y_i)h \to O\left(h^2\right) \\ y^0_{i+1} = y_{i-1} + f(x_i, y_i)2h \to O\left(h^3\right) \end{cases}$$

Corrector: trapezoidal rule

$$y_{i+1} = y_i + \frac{f(x_i, y_i) + f(x_{i+1}, y^0_{i+1})}{2} h \to O\left(h^3\right)$$

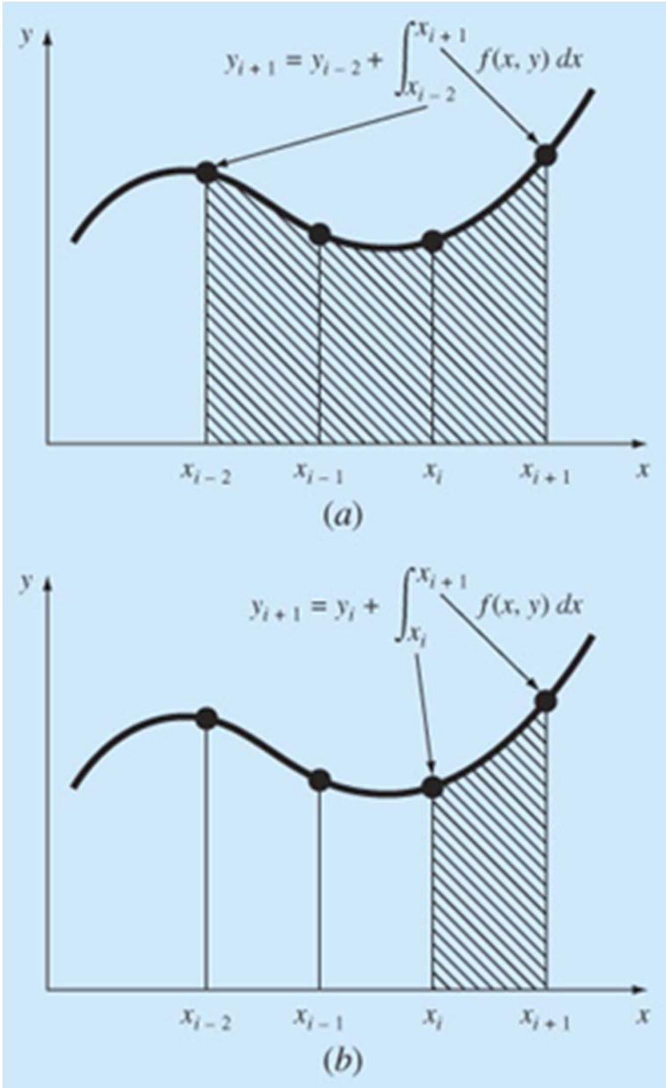
Step-Size Control

- Constant Step Size
  - A value for *h* must be chosen prior to computation.
  - It must be small enough to yield a sufficiently small truncation error.
  - It should also be as large as possible to minimize run time cost and round-off error.
- Variable Step Size
  - If the corrector error is greater than some specified error, the step size is decreased.
  - A step size is chosen so that the convergence criterion of the corrector is satisfied in two iterations.
  - A more efficient strategy is to increase and decrease by doubling and halving the step size.



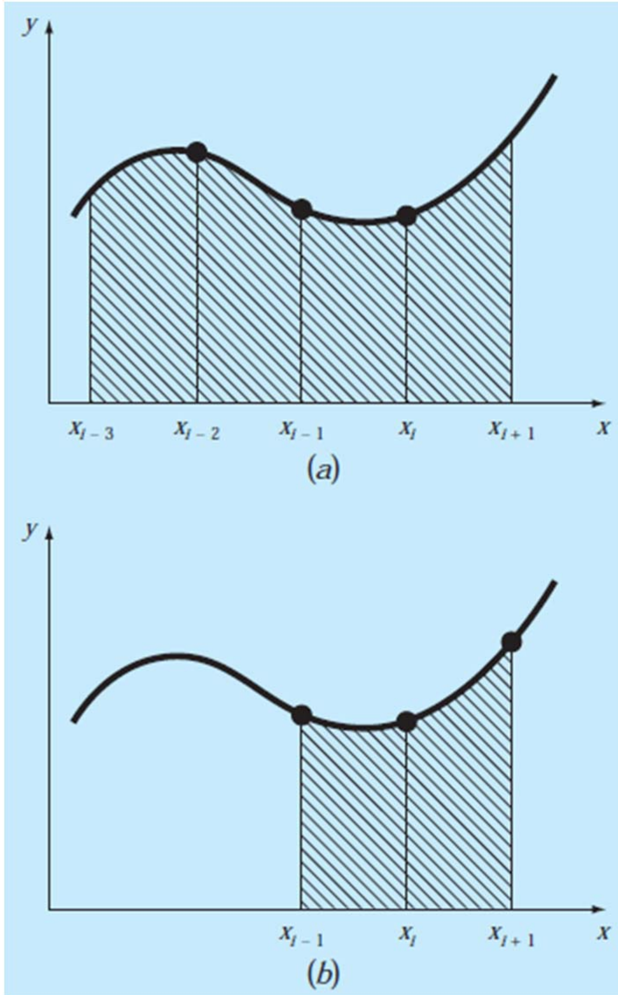
Integration Formulas

- Predictor step
  - Open formula (midpoint method): initial estimate
  - Closed formula (trapezoidal rule): improve the solution
- Higher-order integration formula
  - Newton-Cotes formula
    - Use a series of points to obtain an integral estimate over a number of segment
    - Project across the entire range
  - Adams formula
    - Use a series of points to obtain an integral estimate for a single segment
    - Project across the segment



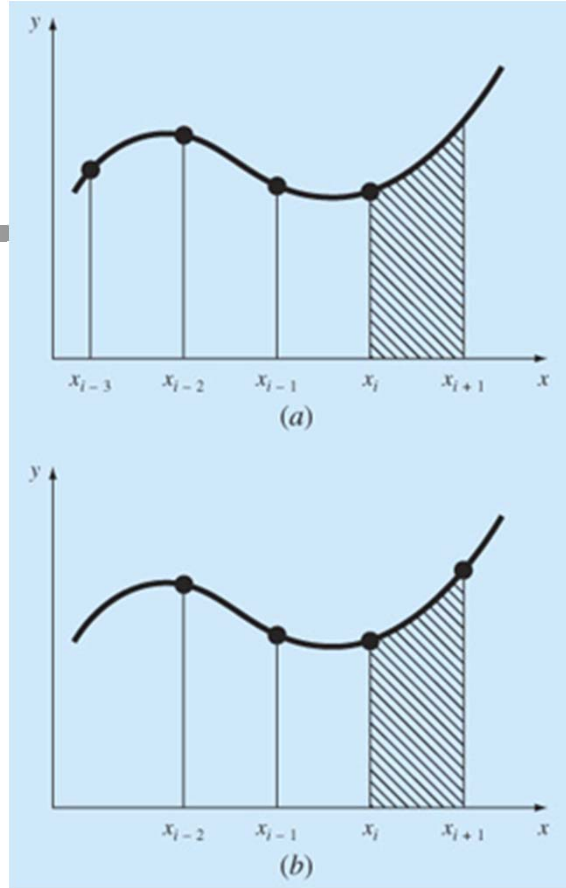
Newton-Cotes Formulas

$f_n(x)$ is an n-th order interpolating polynomial



$$
\begin{cases}
\text{open formula: } y_{i+1} = y_{i-n} + \displaystyle\int_{x_{i-n}}^{x_{i+1}} f_n(x)dx \\
n=1: y_{i+1} = y_{i-1} + 2hf_i \rightarrow \text{midpoint method} \\
n=2: y_{i+1} = y_{i-2} + \dfrac{3h}{2}\left(f_i + f_{i-1}\right) \\
n=3: y_{i+1} = y_{i-3} + \dfrac{4h}{3}\left(2f_i - f_{i-1} + 2f_{i-2}\right) \\
\text{closed formula: } y_{i+1} = y_{i-n+1} + \displaystyle\int_{x_{i-n+1}}^{x_{i+1}} f_n(x)dx \\
n=1: y_{i+1} = y_i + \dfrac{h}{2}\left(f_i + f_{i+1}\right) \rightarrow \text{trapezoidal rule} \\
n=2: y_{i+1} = y_{i-1} + \dfrac{h}{3}\left(f_{i-1} + 4f_i + f_{i+1}\right) \rightarrow \text{Simpson's 1/3 rule}
\end{cases}
$$

Adams Formulas



open formula (Adams – Bashforth): forward Taylor series around x_i

$$y_{i+1} = y_i + f_i h + \frac{f_i'}{2!}h^2 + \frac{f_i''}{3!}h^3 + \cdots = y_i + h\left(f_i + \frac{h}{2} f_i' + \frac{h^2}{3!} f_i'' + \cdots \right)$$

$$\xrightarrow[\text{backward difference}]{f_i' = \frac{f_i - f_{i-1}}{h} + \frac{f_i''}{2}h + O\left(h^2\right)} y_{i+1} = y_i + h\left[f_i + \frac{h}{2}\left\{ \frac{f_i - f_{i-1}}{h} + \frac{f_i''}{2}h + O\left(h^2\right) \right\} + \frac{h^2}{3!} f_i'' + \cdots \right]$$

$$y_{i+1} = y_i + h\left(\frac{3}{2} f_i - \frac{1}{2} f_{i-1} \right) + \frac{5}{12}h^3 f_i'' + O(h^4) \rightarrow y_{i+1} = y_i + h\sum_{k=0}^{n-1} \beta_k f_{i-k} + O(h^{n+1})$$

closed formula (Adams – Moulton): backward Taylor series around x_{i+1}

$$y_i = y_{i+1} - f_{i+1}h + \frac{f_{i+1}'}{2!}h^2 - \frac{f_{i+1}''}{3!}h^3 + \cdots \rightarrow y_{i+1} = y_i + h\left(f_{i+1} - \frac{h}{2} f_{i+1}' + \frac{h^2}{3!} f_{i+1}'' + \cdots \right)$$

$$\xrightarrow[\text{backward difference}]{f_{i+1}' = \frac{f_{i+1} - f_i}{h} + \frac{f_{i+1}''}{2}h + O\left(h^2\right)} y_{i+1} = y_i + h\left[f_{i+1} - \frac{h}{2}\left\{ \frac{f_{i+1} - f_i}{h} + \frac{f_{i+1}''}{2}h + O\left(h^2\right) \right\} + \frac{h^2}{3!} f_{i+1}'' + \cdots \right]$$

$$y_{i+1} = y_i + h\left(\frac{1}{2} f_{i+1} + \frac{1}{2} f_i \right) - \frac{1}{12}h^3 f_{i+1}'' - O\left(h^4\right) \rightarrow y_{i+1} = y_i + h\sum_{k=0}^{n-1} \beta_k f_{i+1-k} + O(h^{n+1})$$

Summary

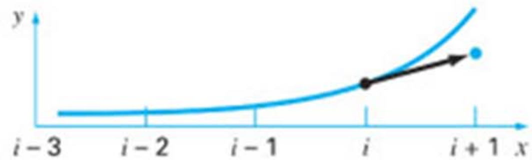
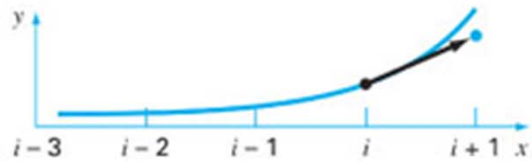
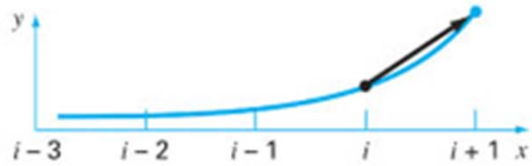
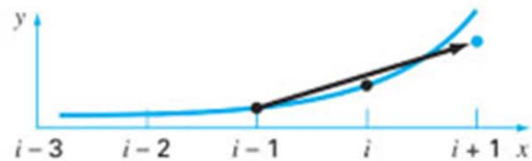
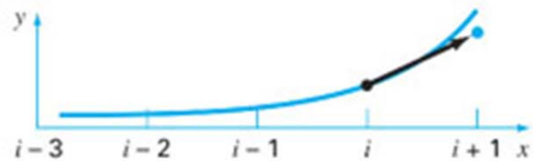
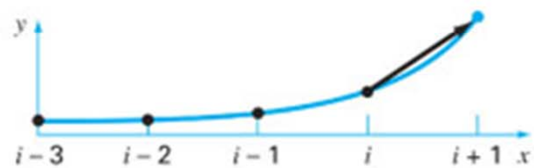
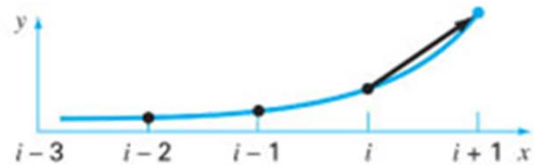
$$\begin{cases} \text{Heun's corrector: } y_{i+1} = y_i + \dfrac{f(x_i) + f(x_{i+1})}{2} h \\ \leftrightarrow \text{ trapezoidal rule: } \displaystyle\int_{x_i}^{x_{i+1}} f(x) dx \cong \dfrac{f(x_i) + f(x_{i+1})}{2} h \to E_t = -\dfrac{f''(\xi)}{12} h^3 \end{cases}$$

$$\begin{cases} \text{midpoint method: } y_{i+1} = y_i + f(x_{i+1/2}, y_{i+1/2}) h \\ \leftrightarrow \text{ Newton} - \text{Cotes open integration formula } (\text{midpoint method}): \\ \displaystyle\int_a^b f(x) dx \cong (b-a) f(x_1) \to \int_{x_i}^{x_{i+1}} f(x) dx \cong hf(x_{i+1/2}) \end{cases}$$

$$\begin{cases} \text{third-order RK: } y_{i+1} = y_i + \dfrac{1}{6}(k_1 + 4k_2 + k_3) h \\ \leftrightarrow \text{Simpson's 1/3 rule: } \displaystyle\int_a^b f(x) dx \cong (b-a) f_2(x) \\ \cong \dfrac{h}{3}\left[f(x_0) + 4f(x_1) + f(x_2) \right] = (b-a) \dfrac{f(x_0) + 4f(x_1) + f(x_2)}{6} \to E_t = -\dfrac{f^{(4)}(\xi)}{90} h^5 \end{cases}$$

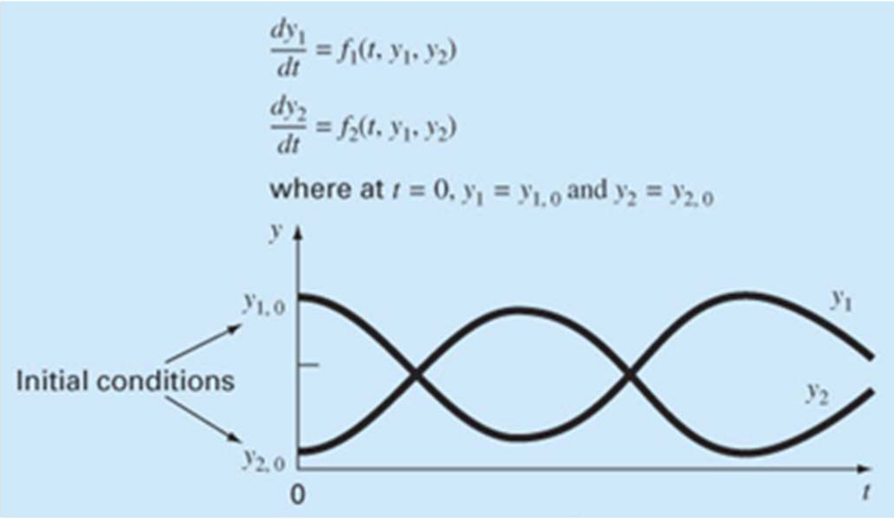
Numerical Solution of ODEs

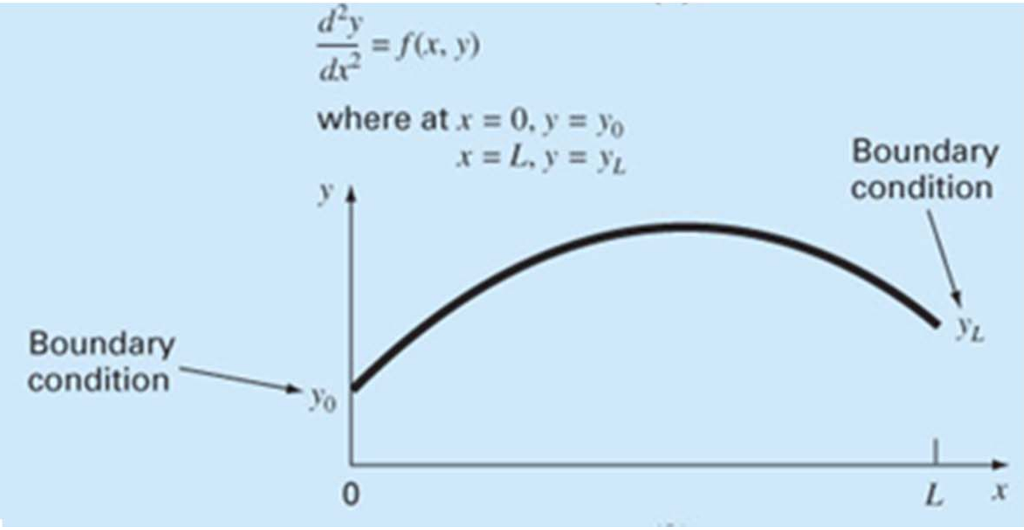
Method	Starting Values	Iterations Required	Global Error	Ease of Changing Step Size	Programming Effort	Comments
One step						
Euler's	1	No	$O(h)$	Easy	Easy	Good for quick estimates
Heun's	1	Yes	$O(h^2)$	Easy	Moderate	—
Midpoint	1	No	$O(h^2)$	Easy	Moderate	—
Second-order Ralston	1	No	$O(h^2)$	Easy	Moderate	The second-order RK method that minimizes truncation error
Fourth-order RK	1	No	$O(h^4)$	Easy	Moderate	Widely used
Adaptive fourth-order RK or RK-Fehlberg	1	No	$O(h^5)^*$	Easy	Moderate to difficult	Error estimate allows step-size adjustment
Multistep						
Non-self-starting Heun	2	Yes	$O(h^3)^*$	Difficult	Moderate to difficult†	Simple multistep method
Milne's	4	Yes	$O(h^5)^*$	Difficult	Moderate to difficult†	Sometimes unstable
Fourth-order Adams	4	Yes	$O(h^5)^*$	Difficult	Moderate to difficult†	

Method	Formulation	Graphic Interpretation	Errors
Euler (First-order RK)	$y_{i+1} = y_i + hk_1$ $k_1 = f(x_i, y_i)$		Local error $\simeq O(h^2)$ Global error $\simeq O(h)$
Ralston's second-order RK	$y_{i+1} = y_i + h(\frac{1}{3}k_1 + \frac{2}{3}k_2)$ $k_1 = f(x_i, y_i)$ $k_2 = f(x_i + \frac{3}{4}h, y_i + \frac{3}{4}hk_1)$		Local error $\simeq O(h^3)$ Global error $\simeq O(h^2)$
Classic fourth-order RK	$y_{i+1} = y_i + h(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4)$ $k_1 = f(x_i, y_i)$ $k_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1)$ $k_3 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_2)$ $k_4 = f(x_i + h, y_i + hk_3)$		Local error $\simeq O(h^5)$ Global error $\simeq O(h^4)$
Non-self-starting Heun	Predictor: (midpoint method) $y^0_{i+1} = y^m_{i-1} + 2hf(x_i, y^m_i)$		Predictor modifier: $E_p \simeq \frac{4}{5}(y^m_{i,u} + y^0_{i,u})$
	Corrector: (trapezoidal rule) $y^j_{i+1} = y^m_i + h\dfrac{f(x_i, y^m_i) + f(x_{i+1}, y^{j-1}_{i+1})}{2}$		Corrector modifier: $E_c \simeq -\dfrac{y^m_{i+1,u} - y^0_{i+1,u}}{5}$
Fourth-order Adams	Predictor: (fourth Adams-Bashforth) $y^0_{i+1} = y^m_i + h(\frac{55}{24}f^m_i - \frac{59}{24}f^m_{i-1} + \frac{37}{24}f^m_{i-2} - \frac{9}{24}f^m_{i-3})$		Predictor modifier: $E_p \simeq \frac{251}{270}(y^m_{i,u} - y^0_{i,u})$
	Corrector: (fourth Adams-Moulton) $y^j_{i+1} = y^m_i + h(\frac{9}{24}f^{j-1}_{i+1} + \frac{19}{24}f^m_i - \frac{5}{24}f^m_{i-1} + \frac{1}{24}f^m_{i-2})$		Corrector modifier: $E_c \simeq -\frac{19}{270}(y^m_{i+1,u} - y^0_{i+1,u})$

IVP vs. BVP

- ODE (n-th order) + auxiliary conditions (n)
- Initial-value problem (IVP)
  - All the conditions are specified at the same value of the independent variable
- Boundary-value problem (BVP)
  - All the conditions are specified at different values of the independent variable (extreme points or boundaries of a system)





BVP

- Shooting method
  - Converts the boundary value problem to initial-value problem
  - A trial-and-error approach is then implemented to solve the initial value approach
- Finite-difference method
  - Most common alternatives to the shooting method
  - Finite differences are substituted for the derivatives in the original equation
  - Finite differences equation applies for each of the interior nodes
  - first and last interior nodes, T_{i-1} and T_{i+1}, respectively, are specified by the boundary conditions
  - linear equation transformed into a set of simultaneous algebraic equations can be solved efficiently

1-D Heat Equation (1)

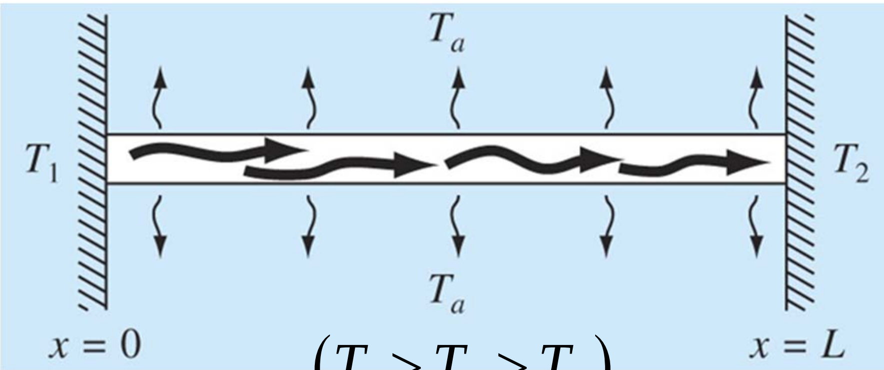
- Uniform rod of length L with non-uniform temperature
  - Heat (or thermal) energy of a body with uniform properties:

$$\text{heat energy} = cmT \left(= \rho C_p T\right)$$

$$\begin{cases} c: \text{specific heat (capacity) } [J/(kg \cdot K)] \\ \text{energy required to raise a unit mass of the substance 1 unit in temperature} \\ m: \text{body mass [kg]} \end{cases}$$

  - Fourier's law of heat transfer: $\dfrac{\text{Rate of heat transfer}}{\text{area}} = -k \dfrac{\partial T}{\partial x}$

k : thermal conductivity $[J/(kg \cdot K)]$



$\left(T_1 > T_2 > T_a\right)$

1-D Heat Equation (2)

– Conservation of energy:

$$\begin{pmatrix}\text{change of}\\ \text{heat energy of}\\ \text{segment in time } \Delta t\end{pmatrix} = \begin{pmatrix}\text{heat in from}\\ \text{left boundary}\end{pmatrix} - \begin{pmatrix}\text{heat out from}\\ \text{right boundary}\end{pmatrix} - \begin{pmatrix}\text{heat out along}\\ \text{its length}\end{pmatrix}$$

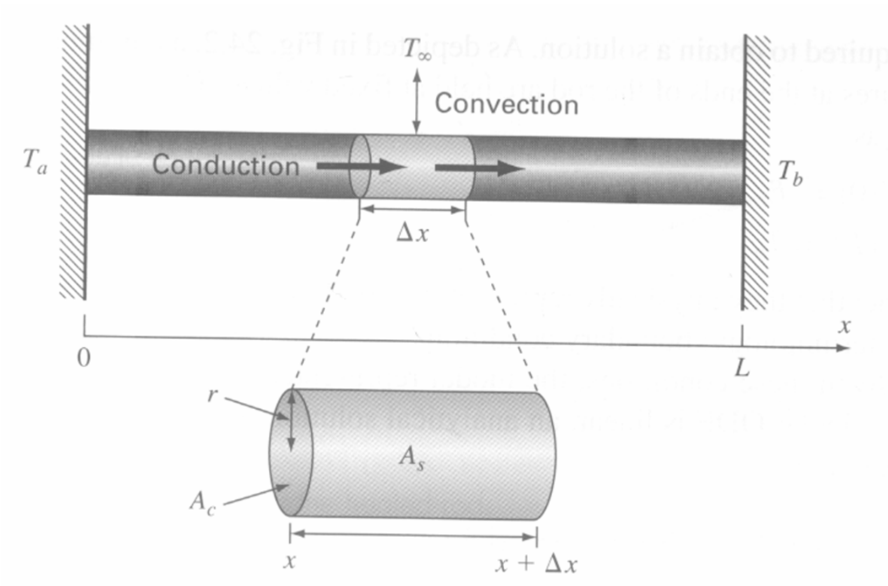
$$c(\rho A\Delta x)T(x,t+\Delta t) - c(\rho A\Delta x)T(x,t) = \Delta t A\left(-k\frac{\partial T}{\partial x}\right)_x - \Delta t A\left(-k\frac{\partial T}{\partial x}\right)_{x+\Delta x} + \Delta t h\Delta x(T_a - T)$$

$$\frac{T(x,t+\Delta t) - T(x,t)}{\Delta t} = \frac{k}{c\rho}\left[\frac{\left(\frac{\partial T}{\partial x}\right)_{x+\Delta x} - \left(\frac{\partial T}{\partial x}\right)_x}{\Delta x}\right] + \frac{h}{c\rho A}(T_a - T)$$

$$\xrightarrow{\Delta t,\Delta x\to 0} \frac{\partial T}{\partial t} = \kappa\frac{\partial^2 T}{\partial x^2} + \frac{h}{c\rho A}(T_a - T)$$

$$\xrightarrow{\text{steady state, } \frac{\partial}{\partial t}=0} 0 = \frac{\partial^2 T}{\partial x^2} + \frac{h}{k}(T_a - T) \to \frac{d^2 T}{dx^2} + h'(T_a - T) = 0$$

1-D Heat Equation (3)



heat balance around a differential element of thickness Δx

$$0 = q(x)A_c - q(x+\Delta x)A_c + hA_s(T_\infty - T)$$

$$\xrightarrow{\div \pi r^2 \Delta x} 0 = \frac{q(x) - q(x+\Delta x)}{\Delta x} + \frac{2h}{r}(T_\infty - T)$$

$$\xrightarrow{\Delta x \to 0} 0 = -\frac{\partial q}{\partial x} + \frac{2h}{r}(T_\infty - T)$$

$$\xrightarrow{q=-k\frac{\partial T}{\partial x}} 0 = \frac{d^2T}{dx^2} + h'(T_\infty - T)$$

Example

- Heat balance for a long, thin rod
  - Not insulated along its length
  - Steady state

$$\frac{d^2T}{dx^2} + h'(T_a - T) = 0$$

$$\left.\begin{aligned} T(0) &= T_1 = 40°C \\ T(L) &= T_2 = 200°C \end{aligned}\right\} \text{boundary conditions}$$

$$T_a = 20°C \left(\text{temperature of the surrounding air}\right)$$

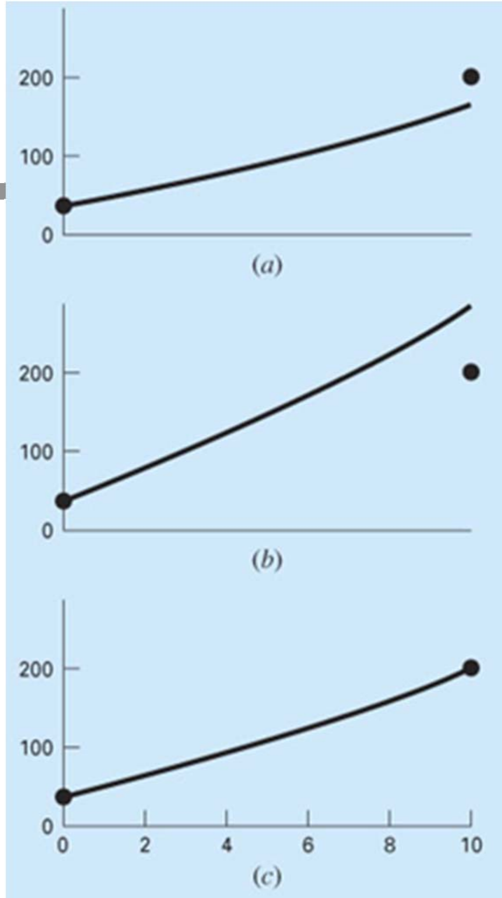
$$L = 10\,m$$

$$h' = 0.01 m^{-2} \left(\text{heat transfer coefficient}\right)$$

rate of heat dissipation to the surrounding air

analytic solution: $T = 73.4523e^{0.1x} - 53.4523e^{-0.1x} + 20$

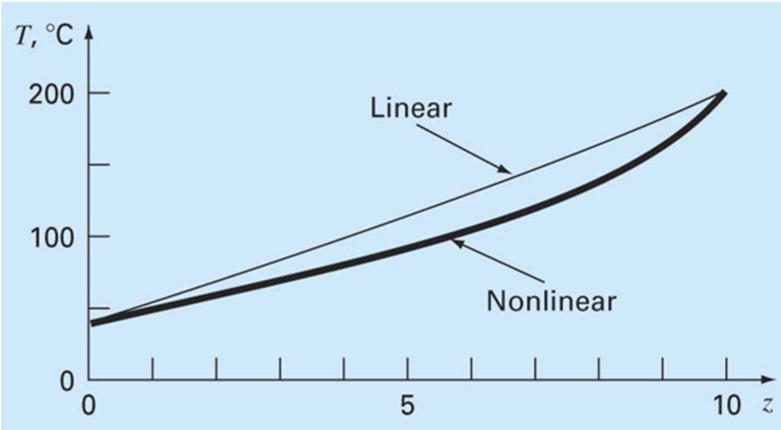
Shooting Method



$$\begin{cases} \dfrac{dT}{dx} = z \\ \dfrac{dz}{dx} = h'(T - T_a)\left[\text{linear}\right] \quad \dfrac{dz}{dx} = h''(T - T_a)^4 \left[\text{nonlinear}\right] \end{cases}$$

$4th$ RK, $h = 2,\ T(10) = 200$

$$\rightarrow \begin{cases} \text{guess / linear: } \begin{cases} z(0) = 10 \rightarrow T(10) = 168.3797 \\ z(0) = 20 \rightarrow T(10) = 285.8980 \end{cases} \\ \xrightarrow[\text{interpolation}]{\text{linear}} z(0) = 12.6907 \\ \text{non-linear: } T(10) = f\left(z(0)\right) \\ \rightarrow g\left(z(0)\right) = f\left(z(0)\right) - 200 \end{cases}$$



Finite Difference Method

$$\frac{d^2T}{dx^2} + h'(T_a - T) = 0 \xrightarrow{\frac{d^2T}{dx^2} = \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2}} \frac{T_{i+1} - 2T_i + T_{i-1}}{\Delta x^2} - h'(T_i - T_a) = 0$$

$$-T_{i-1} + (2 + h'\Delta x^2)T_i - T_{i+1} = h'\Delta x^2 T_a$$

[fixed / Dirichlet boundary condition]

$$\rightarrow \begin{cases} i = 1 : -T_0 + (2 + h'\Delta x^2)T_1 - T_2 = h'\Delta x^2 T_a \\ \vdots \\ i = n : -T_{n-1} + (2 + h'\Delta x^2)T_1 - T_{n+1} = h'\Delta x^2 T_a \end{cases}$$

[natural / Neumann boundary condition]

eg. insulation = zero heat flux

$$\frac{dT}{dx}(0) = T'_a \rightarrow \frac{dT}{dx} = \frac{T_1 - T_{-1}}{2\Delta x} \rightarrow T_{-1} = T_1 - 2\Delta x \frac{dT}{dx}$$

$$i = 0 : -T_{-1} + (2 + h'\Delta x^2)T_0 - T_1 = h'\Delta x^2 T_a$$

$$\rightarrow (2 + h'\Delta x^2)T_0 - T_1 = h'\Delta x^2 T_a + \left(T_1 - 2\Delta x \frac{dT}{dx}\right)$$

Eigenvalue Problems

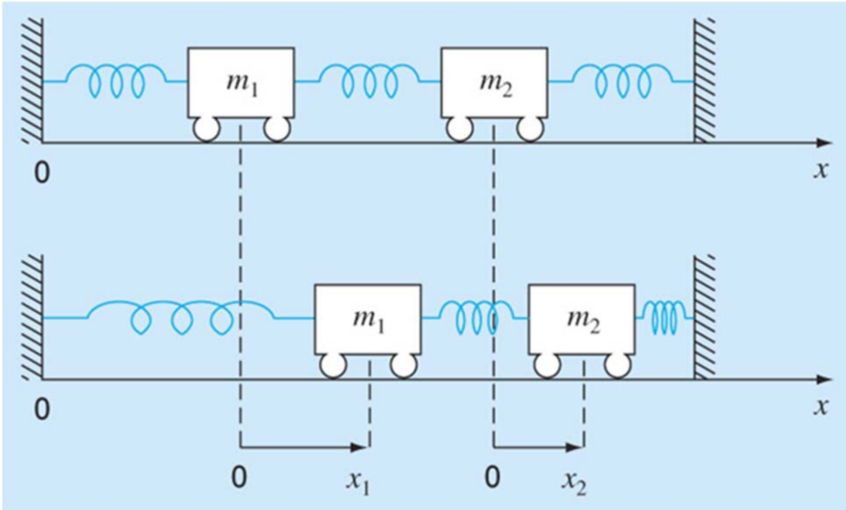
- Special class of boundary-value problems that are common in engineering involving vibrations, elasticity, and other oscillating systems

$$[A]\{X\} = \lambda\{X\} \rightarrow ([A] - \lambda[I])\{X\} = 0 \rightarrow \underbrace{\det([A] - \lambda[I])}_{\text{polynomial in } \lambda} = 0$$

λ : eigenvalue or characteristic value

$\{X\}$: eigenvector

Mass-Spring System: Oscillation



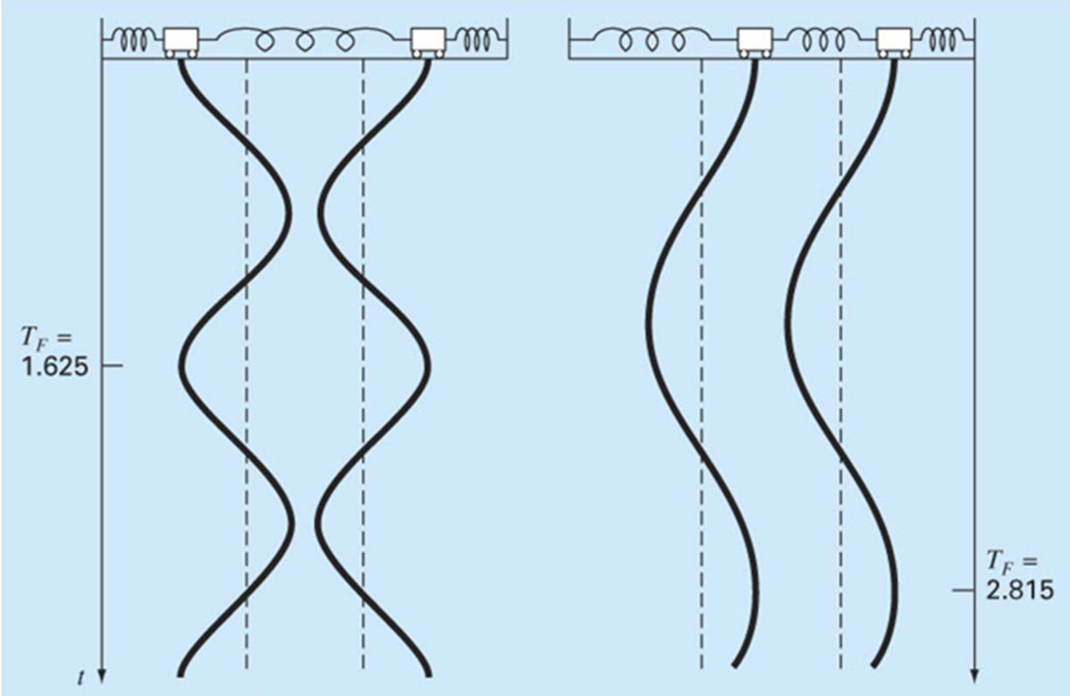
$$\begin{cases} -kx_1 + k\left(x_2 - x_1\right) = m_1 \dfrac{d^2 x_1}{dt^2} \rightarrow m_1 \dfrac{d^2 x_1}{dt^2} + 2kx_1 - kx_2 = 0 \\ -k\left(x_2 - x_1\right) - kx_2 = m_2 \dfrac{d^2 x_2}{dt^2} \rightarrow m_2 \dfrac{d^2 x_2}{dt^2} - kx_1 + 2kx_2 = 0 \end{cases}$$

$$x_i = A_i \sin \omega t \rightarrow \frac{d^2 x_i}{dt^2} = -\omega^2 A_i \sin \omega t \rightarrow \begin{cases} \left(\dfrac{2k}{m_1} - \omega^2\right) A_1 - \dfrac{k}{m_1} A_2 = 0 \\ -\dfrac{k}{m_2} A_1 + \left(\dfrac{2k}{m_2} - \omega^2\right) A_2 = 0 \end{cases}$$

Example

$$m_1 = m_2 = 40kg, k = 200N/m$$

$$\rightarrow \begin{cases} (10-\omega^2)A_1 - 5A_2 = 0 \\ -5A_1 + (10-\omega^2)A_2 = 0 \end{cases} \rightarrow \omega^2 = \begin{cases} 15 \\ 5 \end{cases} \rightarrow \omega = \begin{cases} 3.873 \\ 2.236 \end{cases} \xrightarrow{\omega=\frac{2\pi}{T}} T = \begin{cases} 1.625 \\ 2.815 \end{cases}$$



(a) First mode (b) Second mode

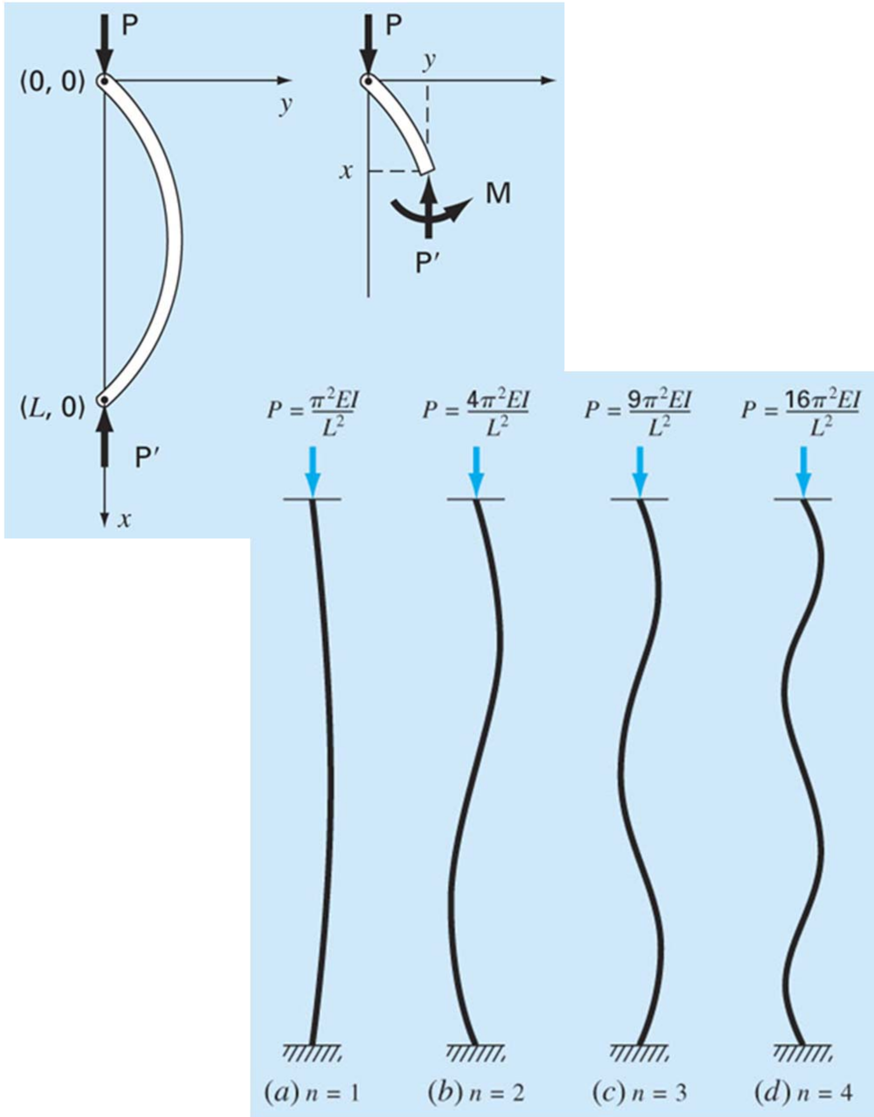
Column Buckling

$$\frac{d^2y}{dx^2} = \frac{M}{EI} \xrightarrow{M=-Py,\ p^2=\frac{P}{EI}}$$

$$\frac{d^2y}{dx^2} + p^2 y = 0 \quad \text{B.C. } y(0) = y(L) = 0$$

$$y = A\sin px + B\cos px$$

$$\begin{cases} y(0) = 0 = B \\ y(L) = 0 = A\sin pL \to pL = n\pi \to p = \dfrac{n\pi}{L} \\ \to P = \dfrac{n^2\pi^2 EI}{L^2}\text{: Euler's formula} \end{cases}$$



Polynomial Method

- Develop the set of equations $([A]-\lambda[I])\{X\}=0$
- Expand the determinant of $[A]-\lambda[I]$, which will be a polynomial whose roots are λ
- Solve for the roots with either Müller’s or Bairstow’s method, deflating in order to find all of the eigenvalues/eigenvectors

Power Method (1)

- Write the system as $AX = \lambda X$
- For an initial guess, assume that X = $\{1,...,1\}^T$, substitute and solve for a new set of X
- Normalize with respect to the largest value of X.
- Iterate until convergence
- Upon convergence, the normalization factor will be the largest eigenvalue, with eigenvector equal to X
- If matrix A is symmetric, it can then be deflated using Hotelling's method $A_2 = A_1 - \lambda_1 X_1 X_1^T$ where A_1 is the original matrix and λ_1, X_1 are the largest eigenvalue/eigenvector pair

Power Method (2)

- Proceed in this manner to find the largest several eigenvalues. This method cannot usually be used to find all of the eigenvalues due to the accumulation of significant round-off errors
- To find the smallest eigenvalue/eigenvector pairs, perform the power method on the inverse of A, deflating in order to eliminate those already found

Example: Polynomial Method

$$\frac{d^2y}{dx^2} + p^2 y = 0 \rightarrow \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p^2 y_i = 0$$

$$y_{i-1} - \left(2 - h^2 p^2\right) y_i + y_{i+1} = 0$$

$$y(0) = y(L = 3) = 0 \rightarrow y_0 = y_{n+1} = 0$$

one node $\left(h = 3/2\right): -\left(2 - 2.25 p^2\right) y_1 = 0 \rightarrow p = \pm 0.9428$

two nodes $\left(h = 3/3\right): \begin{bmatrix} 2 - p^2 & -1 \\ -1 & 2 - p^2 \end{bmatrix} \begin{Bmatrix} y_1 \\ y_2 \end{Bmatrix} = 0 \rightarrow p = \pm 1, \pm 1.73205$

three nodes $\left(h = 3/4\right): \begin{bmatrix} 2 - 0.5625 p^2 & -1 & 0 \\ -1 & 2 - 0.5625 p^2 & -1 \\ 0 & -1 & 2 - 0.5625 p^2 \end{bmatrix} \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \end{Bmatrix} = 0$

		Polynomial Method			
Eigenvalue	**True**	**h = 3/2**	**h = 3/3**	**h = 3/4**	**h = 3/5**
1	1.0472	0.9428 (10%)	1.0000 (4.5%)	1.0205 (2.5%)	1.0301 (1.6%)
2	2.0944		1.7321 (21%)	1.8856 (10%)	1.9593 (65%)
3	3.1416			2.4637 (22%)	2.6967 (14%)
4	4.1888				3.1702 (24%)

Example: Power Method

$$\frac{d^2y}{dx^2}+p^2y=0 \rightarrow -\frac{y_{i+1}-2y_i+y_{i-1}}{h^2}=p^2y_i$$

three nodes $(h=3/4)$:

highest eigenvalue: $\mathbf{Ax}=\lambda\mathbf{x}$

$$\begin{bmatrix} 3.556 & -1.778 & 0 \\ -1.778 & 3.556 & -1.778 \\ 0 & -1.778 & 3.556 \end{bmatrix}\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}=\lambda\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

$$\begin{bmatrix} 3.556 & -1.778 & 0 \\ -1.778 & 3.556 & -1.778 \\ 0 & -1.778 & 3.556 \end{bmatrix}\begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix}=1.778\begin{Bmatrix} 1 \\ 0 \\ 1 \end{Bmatrix}$$

$$\begin{bmatrix} 3.556 & -1.778 & 0 \\ -1.778 & 3.556 & -1.778 \\ 0 & -1.778 & 3.556 \end{bmatrix}\begin{Bmatrix} 1 \\ 0 \\ 1 \end{Bmatrix}=3.556\begin{Bmatrix} 1 \\ -1 \\ 1 \end{Bmatrix}$$

lowest eigenvalue: $\mathbf{A}^{-1}\mathbf{x}=\lambda^{-1}\mathbf{x}$

$$\begin{bmatrix} 0.422 & 0.281 & 0.141 \\ 0.281 & 0.562 & 0.281 \\ 0.141 & -1.778 & 0.422 \end{bmatrix}\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}=\lambda^{-1}\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix}$$

$$\begin{bmatrix} 0.422 & 0.281 & 0.141 \\ 0.281 & 0.562 & 0.281 \\ 0.141 & -1.778 & 0.422 \end{bmatrix}\begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix}=1.124\begin{Bmatrix} 0.751 \\ 1 \\ 0.751 \end{Bmatrix}$$

$$\begin{bmatrix} 0.422 & 0.281 & 0.141 \\ 0.281 & 0.562 & 0.281 \\ 0.141 & -1.778 & 0.422 \end{bmatrix}\begin{Bmatrix} 0.751 \\ 1 \\ 0.751 \end{Bmatrix}=0.984\begin{Bmatrix} 1 \\ -1 \\ 1 \end{Bmatrix}$$