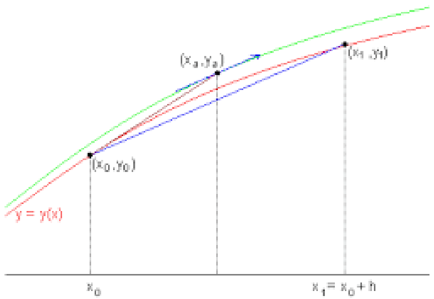
Lecture 14: Runge-Kutta Methods



Ordinary Differential Equations

- Sometimes referred to as a *rate equation*.
- When the function involves one independent variable, the equation is called an *ordinary differential equation* (or *ODE*).
- Differential equations are also classified as to their order: First order, second order, …
- Higher-order equations can be reduced to a system of first-order equations.

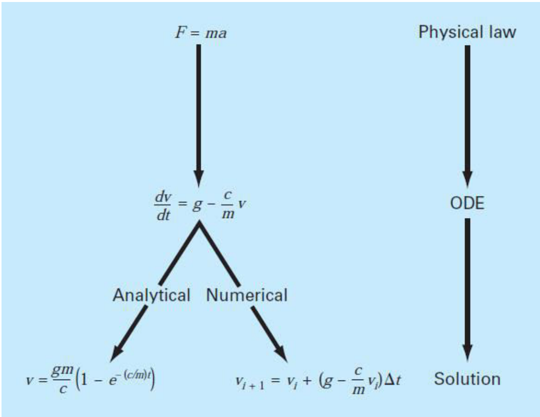
Methods for Solving ODEs

- ODEs are usually solved with analytical integration techniques.
- However, exact solutions for many ODEs of practical importance are not available.
- Numerical methods are required.

ODEs and Engineering Practice

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)

ODEs and Engineering Practice



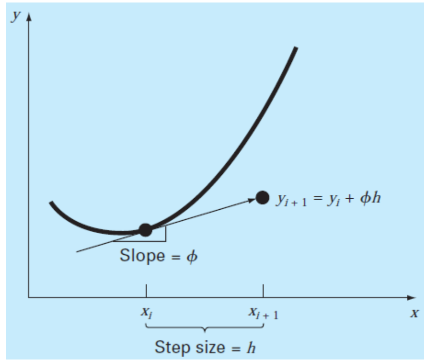
One-step Method for solving ODEs

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

New value = old value + slope × step size

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

One-step method



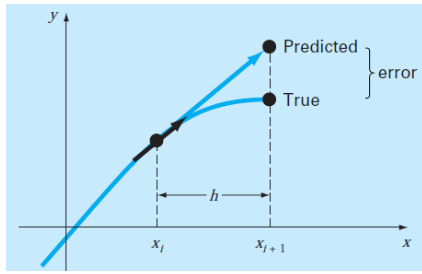
Runge-Kutta methods

- One-step methods → Slope estimation differs
- *Euler's method* → a one step method
  - The slope at the beginning of the interval is taken as an approximation of the average slope over the whole interval.

Euler's Method

- The first derivative provides a direct estimate of the slope at x_i.

$$\frac{dy}{dx} = f(x, y) \Longrightarrow \phi = f(x_i, y_i)$$



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

Euler's (or the *Euler-Cauchy* or the *point-slope*) *method.*

Example 1

Problem Statement. Use Euler's method to numerically integrate Eq. (PT7.13):

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

from $x = 0$ to $x = 4$ with a step size of 0.5. The initial condition at $x = 0$ is $y = 1$. Recall that the exact solution is given by Eq. (PT7.16):

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

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

$$y(0.5) = y(0) + f(0, 1)0.5 \qquad f(0, 1) = -2(0)^3 + 12(0)^2 - 20(0) + 8.5 = 8.5$$

$$y(0.5) = 1.0 + 8.5(0.5) = 5.25$$

Example 1

The true solution at $x = 0.5$ is

$$y = -0.5(0.5)^4 + 4(0.5)^3 - 10(0.5)^2 + 8.5(0.5) + 1 = 3.21875$$

Thus, the error is

$$E_t = \text{true} - \text{approximate} = 3.21875 - 5.25 = -2.03125$$

$$\varepsilon_t = -63.1\%$$

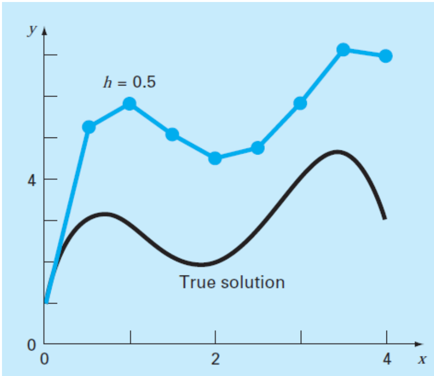
$$y(1) = y(0.5) + f(0.5, 5.25)0.5$$
$$= 5.25 + [-2(0.5)^3 + 12(0.5)^2 - 20(0.5) + 8.5]0.5$$
$$= 5.875$$

2nd Step

Example 1: Comparison of true and approximate values

x	y_{true}	y_{Euler}	Percent Relative Error Global	Percent Relative Error 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 1: Comparison of true and approximate values



Error Analysis for Euler's Method

- The numerical solution of ODEs involves two types of error:
  - Truncation, or discretization errors
  - Round-off errors
- The truncation errors are composed of two parts:
  - Local truncation error that results from an application of the method in question over a single step.
  - Propagated truncation error that results from the approximations produced during the previous steps.
  - The sum of the two is the total, or global truncation, error.

Truncation Errors

$$y' = f(x, y)$$

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

$$R_n = \frac{y^{(n+1)}(\xi)}{(n+1)!}h^{n+1}$$

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

Euler's method corresponds to the Taylor series

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

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

$$E_t = \frac{f'(x_i, y_i)}{2!}h^2 + \cdots + O(h^{n+1})$$

E_t = the true local truncation error.

$$E_a = \frac{f'(x_i, y_i)}{2!}h^2$$

E_a = the approximate local truncation error.

$$E_a = O(h^2)$$

Example 2

$$E_t = \frac{f'(x_i, y_i)}{2!}h^2 + \cdots + O(h^{n+1}) \qquad (25.7)$$

Problem Statement. Use Euler's method to numerically integrate Eq. (PT7.13):

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

from $x = 0$ to $x = 4$ with a step size of 0.5. The initial condition at $x = 0$ is $y = 1$. Recall that the exact solution is given by Eq. (PT7.16):

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

Problem Statement. Use Eq. (25.7) to estimate the error of the first step of Example 25.1. Also use it to determine the error due to each higher-order term of the Taylor series expansion.

Example 2

$$E_t = \frac{f'(x_i, y_i)}{2!}h^2 + \cdots + O(h^{n+1}) \qquad (25.7)$$

$$E_t = \frac{f'(x_i, y_i)}{2!}h^2 + \frac{f''(x_i, y_i)}{3!}h^3 + \frac{f^{(3)}(x_i, y_i)}{4!}h^4$$

$$f'(x_i, y_i) = -6x^2 + 24x - 20 \qquad f''(x_i, y_i) = -12x + 24$$

$$f^{(3)}(x_i, y_i) = -12$$

$$E_{t,2} = \frac{-6(0.0)^2 + 24(0.0) - 20}{2}(0.5)^2 = -2.5 \qquad E_{t,3} = \frac{-12(0.0) + 24}{6}(0.5)^3 = 0.5$$

$$E_{t,4} = \frac{-12}{24}(0.5)^4 = -0.03125 \qquad E_t = E_{t,2} + E_{t,3} + E_{t,4} = -2.5 + 0.5 - 0.03125 = -2.03125$$

$$E_{t,2} > E_{t,3} > E_{t,4}$$

Local and Global Truncation Errors

$$E_a = \frac{f'(x_i, y_i)}{2!}h^2$$

- Local: O(h^2)
- Global: O(h) (Carnahan et al.1969)
- Observations:
  - The error can be reduced by decreasing the step size.
  - The method will provide error-free predictions if the underlying function is linear.
- Euler's method uses straight-line segments to approximate the solution → a first-order method.

Example 3

Problem Statement. Use Euler's method to numerically integrate Eq. (PT7.13):

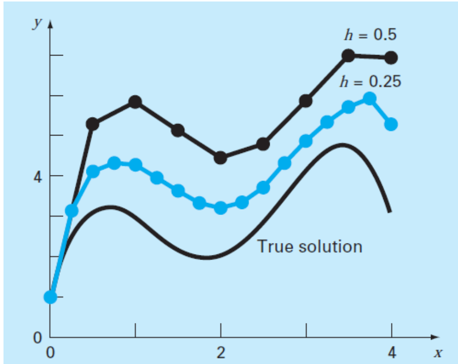
$$\frac{dy}{dx} = -2x^3 + 12x^2 - 20x + 8.5$$

from $x = 0$ to $x = 4$ with a step size of 0.5. The initial condition at $x = 0$ is $y = 1$. Recall that the exact solution is given by Eq. (PT7.16):

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

Problem Statement. Repeat the computation of Example 25.1 but use a step size of 0.25.

Example 3



Some useful conclusions

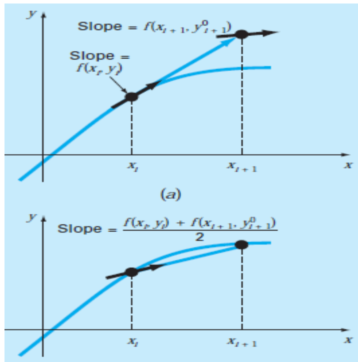
- The error can be reduced by decreasing the step size.
- The method will provide error-free predictions if the underlying function is linear.
- For an *n*th-order method, the local truncation error will be $O(h^{n+1})$ *and the global error* $O(h^n)$.

Improvements of Euler's Method

- <u>A fundamental source of error:</u> the derivative at the beginning of the interval is assumed to apply across the entire interval.
- Two simple modifications are available.

Heun's Method

- Determine **two derivatives**—one at the initial point and another at the end point.
- **Average them** to obtain an improved estimate of the slope for the entire interval.



Heun's Method

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

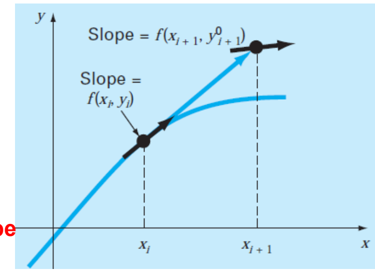
$$y_{i+1}^0 = y_i + f(x_i, y_i)h$$

An intermediate prediction

$$y_{i+1}' = f(x_{i+1}, y_{i+1}^0)$$

An estimated slope

Predictor equation



$$\bar{y}' = \frac{y_i' + y_{i+1}'}{2} = \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^0)}{2}$$

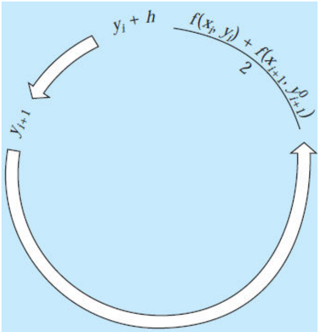
<u>Heun method is a *predictor-corrector approach.*</u>

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

Corrector equation

Heun's Method in Iterative Fashion

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



Termination criterion:

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

Example 4

Problem Statement. Use Heun's method to integrate $y' = 4e^{0.8x} - 0.5y$ from $x = 0$ to $x = 4$ with a step size of 1. The initial condition at $x = 0$ is $y = 2$.

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

Analytical solution

$$y'_0 = 4e^0 - 0.5(2) = 3$$

$$y^0_1 = 2 + 3(1) = 5$$

$$y'_1 = f(x_1, y^0_1) = 4e^{0.8(1)} - 0.5(5) = 6.402164$$

$$y' = \frac{3 + 6.402164}{2} = 4.701082$$

$$y_1 = 2 + 4.701082(1) = 6.701082$$

Example 4: Iteration

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

$$y_1 = 2 + \frac{[3 + 4e^{0.8(1)} - 0.5(6.701082)]}{2} 1 = 6.275811$$

$$y_1 = 2 + \frac{[3 + 4e^{0.8(1)} - 0.5(6.275811)]}{2} 1 = 6.382129$$

Example 4: Iteration

		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

Heun's Method

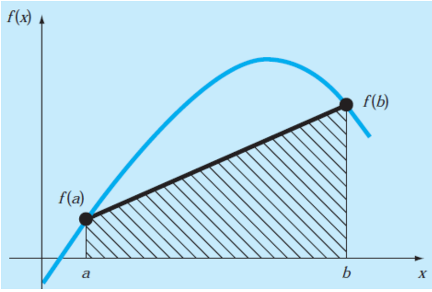
- For cases such as polynomials, where the ODE is solely a function of the independent variable, the predictor step is not required and the corrector is applied only once for each iteration.

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

Comparison with Trapezoidal Rule

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

$$I = (b-a)\frac{f(a) + f(b)}{2}$$



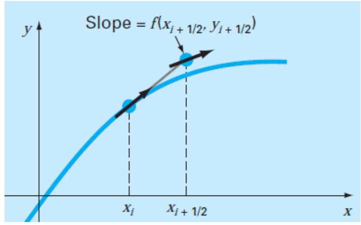
Local truncation error: $E_t = -\frac{f''(\xi)}{12} h^3$

The Midpoint Method

- Alternatively known as Improved Polygon or modified Euler.
- It uses Euler's method to predict a value of y at the midpoint of the interval.

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

$$y'_{i+1/2} = f(x_{i+1/2}, y_{i+1/2})$$



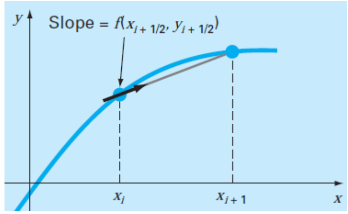
The Midpoint Method

- This slope is assumed to represent a valid approximation of the average slope for the entire interval.

$$y'_{i+1/2} = f(x_{i+1/2}, y_{i+1/2})$$

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

It cannot be applied iteratively to improve the solution.



Linked to Newton-Cotes integration formulas

$$\int_a^b f(x)\,dx \cong (b-a)\,f(x_1)$$

- The midpoint method is superior to Euler's method because it utilizes a slope estimate at the midpoint of the prediction interval.
- The local and global errors of the midpoint method are $O(h^3)$ and $O(h^2)$, respectively.

Runge-Kutta Methods

- Runge-Kutta (*RK*) methods achieve the accuracy of a Taylor series approach without requiring the calculation of higher derivatives.

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

where $\phi(x_i, y_i, h)$ is called an *increment function*,

- The increment function can be written in general form as

$$\phi = a_1k_1 + a_2k_2 + \cdots + a_nk_n$$

Runge-Kutta Methods

$$\phi = a_1k_1 + a_2k_2 + \cdots + a_nk_n$$

$$\begin{aligned}
k_1 &= f(x_i, y_i)\\
k_2 &= f(x_i + p_1h,\ y_i + q_{11}k_1h)\\
k_3 &= f(x_i + p_2h,\ y_i + q_{21}k_1h + q_{22}k_2h)\\
&\ \ \vdots\\
k_n &= f(x_i + p_{n-1}h,\ y_i + q_{n-1,1}k_1h + q_{n-1,2}k_2h + \cdots + q_{n-1,n-1}k_{n-1}h)
\end{aligned}$$

Notice that the k's are recurrence relationships.

Runge-Kutta Methods

- Various types of Runge-Kutta methods can be devised by employing different *n's.*
- n = 1, first-order *RK* method (Euler's method).

Second-Order Runge-Kutta Methods

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

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h$$

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

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

How to obtain these equations?

Second-Order Runge-Kutta Methods

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h$$

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

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

2nd order Taylor series

$$f'(x_i, y_i) = \frac{\partial f(x,y)}{\partial x} + \frac{\partial f(x,y)}{\partial y}\frac{dy}{dx}$$

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

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h \quad = \quad y_{i+1} = y_i + f(x_i, y_i)h + \left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{dy}{dx}\right)\frac{h^2}{2!}$$

Second-Order Runge-Kutta Methods

- We first use a Taylor series to expand

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

$$g(x + r, y + s) = g(x, y) + r\frac{\partial g}{\partial x} + s\frac{\partial g}{\partial y} + \cdots$$

$$f(x_i + p_1 h, y_i + q_{11} k_1 h) = f(x_i, y_i) + p_1 h\frac{\partial f}{\partial x} + q_{11} k_1 h\frac{\partial f}{\partial y} + O(h^2)$$

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h \qquad k_1 = f(x_i, y_i)$$

$$y_{i+1} = y_i + a_1 h f(x_i, y_i) + a_2 h f(x_i, y_i) + a_2 p_1 h^2\frac{\partial f}{\partial x} + a_2 q_{11} h^2 f(x_i, y_i)\frac{\partial f}{\partial y} + O(h^3)$$

Second-Order Runge-Kutta Methods

$$y_{i+1} = y_i + a_1 h f(x_i, y_i) + a_2 h f(x_i, y_i) + a_2 p_1 h^2\frac{\partial f}{\partial x} + a_2 q_{11} h^2 f(x_i, y_i)\frac{\partial f}{\partial y} + O(h^3)$$

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

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

Compare like terms

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

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

Heun Method with a Single Corrector ($a_2 = 1/2$)

$$a_1 = 1 - a_2$$

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

$$a_1 = 1/2 \text{ and } p_1 = q_{11} = 1$$

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h$$

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

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

- k_1 is the slope at the beginning of the interval and k_2 is the slope at the end of the interval.

Heun's method: A Second order method

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

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

- Prove that Heun's method is a second order method.
- Hints:
  - Alternatively, prove the truncation error for Heun's method is $O(h^3)$.
  - Write the expression for E_t
  - E_t = *Heun's formula – Exact solution*
  - E_t = *Heun's formula - $y(x_i+h)$*
  - *Expand K_2 up to $O(h^2)$ and $y(x_i+h)$* up to $O(h^3)$ using Taylor's expansion.

The Midpoint Method ($a_2 = 1$).

$$a_1 = 0, p_1 = q_{11} = 1/2$$

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2)h$$

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

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

Ralston's Method ($a_2 = 2/3$)

- Ralston (1962) and Ralston and Rabinowitz (1978) determined that choosing a_2 = *2/3 provides a minimum bound on the truncation error for* the second-order RK algorithms.

$$a_1 = 1/3 \text{ and } p_1 = q_{11} = 3/4$$

$$y_{i+1} = y_i + \left(\frac{1}{3}k_1 + \frac{2}{3}k_2\right)h$$

$$k_1 = f(x_i, y_i)$$
$$k_2 = f\left(x_i + \frac{3}{4}h, y_i + \frac{3}{4}k_1 h\right)$$

Example 5

Problem Statement. Use the midpoint method [Eq. (25.37)] and Ralston's method [Eq. (25.38)] to numerically integrate Eq. (PT7.13)

$$f(x, y) = -2x^3 + 12x^2 - 20x + 8.5$$

from $x = 0$ to $x = 4$ using a step size of 0.5. The initial condition at $x = 0$ is $y = 1$. Compare the results with the values obtained using another second-order RK algorithm, that is, the Heun method without corrector iteration (Table 25.3).

$$y_{i+1} = y_i + k_2 h \qquad (25.37)$$

$$y_{i+1} = y_i + \left(\frac{1}{3}k_1 + \frac{2}{3}k_2\right)h \qquad (25.38)$$

Midpoint method:

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

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

Example 5

Midpoint method:

$$y_{i+1} = y_i + k_2 h \qquad (25.37)$$

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

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

$$k_1 = -2(0)^3 + 12(0)^2 - 20(0) + 8.5 = 8.5$$

$$k_2 = -2(0.25)^3 + 12(0.25)^2 - 20(0.25) + 8.5 = 4.21875$$

$$y(0.5) = 1 + 4.21875(0.5) = 3.109375 \qquad \varepsilon_t = 3.4\%$$

Example 5

- **Ralston's method**

$$y_{i+1} = y_i + \left(\frac{1}{3}k_1 + \frac{2}{3}k_2\right)h \qquad (25.38)$$

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

$$k_2 = f\left(x_i + \frac{3}{4}h, y_i + \frac{3}{4}k_1 h\right)$$

$$k_2 = -2(0.375)^3 + 12(0.375)^2 - 20(0.375) + 8.5 = 2.58203125$$

$$\phi = \frac{1}{3}(8.5) + \frac{2}{3}(2.58203125) = 4.5546875$$

$$y(0.5) = 1 + 4.5546875(0.5) = 3.27734375 \qquad \varepsilon_t = -1.82\%$$

Example 5

		Heun		Midpoint		Second-Order Ralston RK	
x	y_{true}	y	$\|\varepsilon_t\|$ (%)	y	$\|\varepsilon_t\|$ (%)	y	$\|\varepsilon_t\|$ (%)
0.0	1.00000	1.00000	0	1.00000	0	1.00000	0
0.5	3.21875	3.43750	6.8	3.109375	3.4	3.277344	1.8
1.0	3.00000	3.37500	12.5	2.81250	6.3	3.101563	3.4
1.5	2.21875	2.68750	21.1	1.984375	10.6	2.347656	5.8
2.0	2.00000	2.50000	25.0	1.75	12.5	2.140625	7.0
2.5	2.71875	3.18750	17.2	2.484375	8.6	2.855469	5.0
3.0	4.00000	4.37500	9.4	3.81250	4.7	4.117188	2.9
3.5	4.71875	4.93750	4.6	4.609375	2.3	4.800781	1.7
4.0	3.00000	3.00000	0	3	0	3.031250	1.0

Example 5