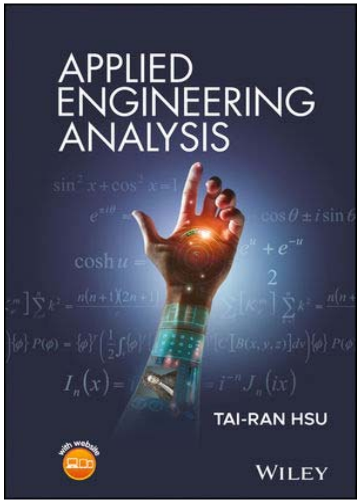
Applied Engineering Analysis
- slides for class teaching*

Chapter 10

Numerical Solution Methods for Engineering Analysis

* Based on the textbook on "Applied Engineering Analysis", by Tai-Ran Hsu, published by John Wiley & Sons, 2018 (ISBN 97811119071204)



Chapter Learning Objectives (p.339)

- Learn the alternative ways of using numerical methods to solve nonlinear equations, perform integrations, and solve differential equations.

- Learn the principles of various numerical techniques for solving nonlinear equations, performing integrations, and solving differential equations by the Runge-Kutta methods.

- Learn the fact that numerical methods offer approximate but credible accurate solutions to the problems that are not readily or possibly solved by closed-form solution methods.

- Learn the fact that numerical solutions are available to the users only at the preset solution points, and the accuracy of the solution is largely depending on the size of the increments of the variable selected for the solutions.

- Become familiar with the value of commercially available numerical solution software packages such as Mathematica and MatLAB.

10.1 Introduction

Numerical methods are techniques by which the mathematical problems involved with the engineering analysis cannot readily or possibly be solved by analytical methods such as those presented in previous chapters of this book.

We will learn from this chapter on the use of some of these numerical methods that will not only enable engineers to solve many mathematical problems, but they will also allow engineers to minimize the needs for the many hypotheses and idealization of the conditions, as stipulated in Section 1.4 (p.8) for engineering analysis.

This chapter will cover the principles of commonly used numerical techniques for:

(1) the solution of nonlinear polynomial and transcendental equations,

(2) Integration with integrals that involve complex forms of functions, and

(3) the solution of differential equations by selected finite difference methods,

(4) overviews of two popular commercial software packages called Mathematica and MatLAB.

10.2 **Engineering Analysis with Numerical Solutions** (p.340)

There are a number of unique characteristics of numerical solution methods in engineering analysis. Following are just a few obvious ones:

1) Numerical solutions are available only at selected (discrete) solution points, but not at all points covered by the functions as in the case with analytical solution methods.

2) Numerical methods are essentially “trail-and-error” processes. Typically, users need to estimate an initial solution with selected increment of the variable to which the intended solution will cover. Unstable numerical solutions may result from improper selection of step sizes (the incremental steps) with solutions either in the form of “wild oscillation” or becoming unbounded in the trend of values.

3) Most numerical solution methods results in errors in the solutions. There are two types of errors that are inherent with numerical solutions:

   (a) Truncation errors – Because of the approximate nature of numerical solutions, they often consists of lower order terms and higher order terms. The latter terms are often dropped in the computations for the sake of computational efficiency, resulting in error in the solution, and

   (b) Round-off errors –Most digital computers handle either numbers with 7 decimal points, or 14 decimal points in numerical solutions. In the case of 32-bit computer with double precision (i.e. 14 decimal points length numbers), any number after the 14th decimal point will be dropped. This may not sound like a big deal, but if a huge number of operations are involved in the computation, such error can accumulate and result in significant error in the end results.

   Both these errors are of accumulative natures. Consequently, errors in numerical solution may grow to be significant with solutions obtained after many step with the set increments.

10.3 Solution of Nonlinear Equations (p.341)

We have learned the distinction between linear and nonlinear algebraic equations in Section 4.1. There are numerous occasions that engineers are requested to solve nonlinear equations such as the equation for the solution t_f of the following nonlinear equation in Example 8.9 on page 270:

$$\left(0.005 - \frac{t_f}{20}\right) cos\left(10t_f\right) + \frac{1}{200}\sin\left(10t_f\right) = 0.03 \qquad (10.2)$$

We reported a solution of t_f=0.7 in Equation (10.2) by a “short cut” solution method, and also t_f= 0.862 by a more accurate solution method such as the Newton- Raphson method described in Section 10.3.2.

There are a number of numerical methods available to solve nonlinear equations such as in Equation (10.2); what we will introduce here in the book are the following two methods that are readily available by using digital computers:

10.3.1 Solution using Microsoft Excel software (Example 10.1) (p.342):

In this method, we will first express the equation in the form of f(x)=0 as shown in Figure 10.1. For example, we will express Equation (10.3) in Example 10.1 from the form of $x^4-2x^3+x^2-3x=-3$ into the form: $x^4-2x^3+x^2-3x+3=0$, in which we will get the function $f(x) = x^4-2x^3+x^2-3x+3$. The roots “x” would lie in the range between $x=x_i$ and x_{i+1}, with which the values of $f(x_i)$ and $f(x_{i+1})$ bearing different positive or negative sign. The difference of (x_{i-1}) and (x_i), or between x_{i+1} and x_i is referred to be the increment of x-value, or is expressed as Δx.

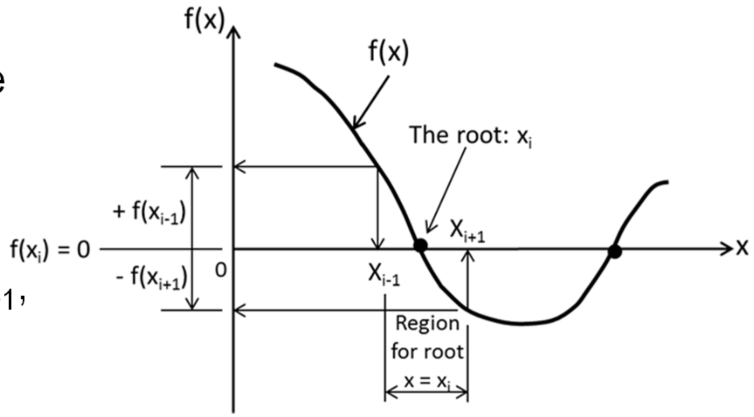


Figure 10.1 Roots in Nonlinear Equation f(x) = 0

10.3.1 Solution using Microsoft Excel software (Example 10.1) – Cont'd:

Example 10.1

Solve the nonlinear polynomial Equation in (10.3): $x^4-2x^3+x^2-3x+3=0$

Solution:

We have Equation 10.3 expressed as f(x)=0 with $f(x) = x^4-2x^3+x^2-3x+3$. We will use Microsoft Excel software to evaluate the function f(x) with an increment of the variable x, $\Delta x=0.5$ beginning at $x =0$. The values of the function f(x) with x_i (i = 1,2,3,….,9) are shown in the Table in the right, and the plot of function f(x) vs. variable x is depicted in Figure 10.2

i	x	f(x)
1	0	3.00
2	0.5	1.56
3	1.0	0
4	1.5	-0.94
5	2.0	+1.00
6	2.5	9.56
7	3.0	30.00
8	3.5	69/06
9	4.0	135.00

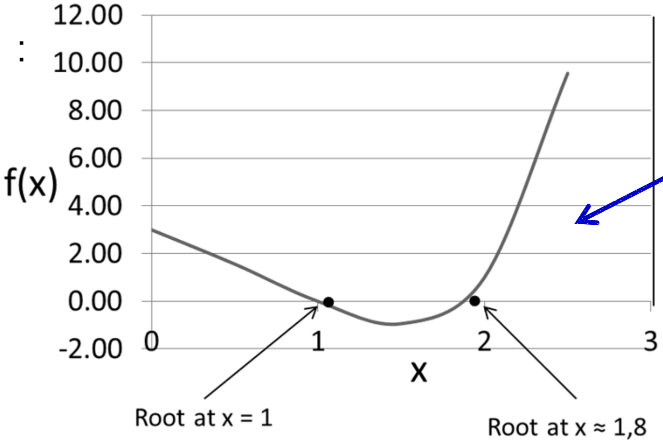


Figure 10.2

We notice from the computed values of f(x) with variable x in Figure 10.2 that there are two roots of the equation in the ranges of (x=1.0 and 1.5) and the other root in the range of (x = 1.5 and 2.0) because the sign changes of the function f(x) cross these two ranges of x variable. The first root of x =1 is obvious because it resulted in f(x) = 0. The search of the second root with computations of the function f(x) with smaller increment of x between x = 1.5 and x=2.0 indicated an approximate root at x = 1.8 as illustrated in the plot of the results in Figure 10.2.

10.3.2 The Newton-Raphson Method – a popular method for solving nonlinear equations (p.342)

This method offers rapid convergence to the roots of many nonlinear equations from the initial estimated roots.

Figure 10.3 illustrates the principle of Newton/Raphson’s method in solving nonlinear equations.

The user needs to estimate a root at x = x_i for the equation f(x) = 0, from which he (she) may compute the function $f(x_i)$ and at the same time the slope of the curve generated by the function f(x). This slope may be expressed $f'(x_i)$, as expressed in the following equation:.

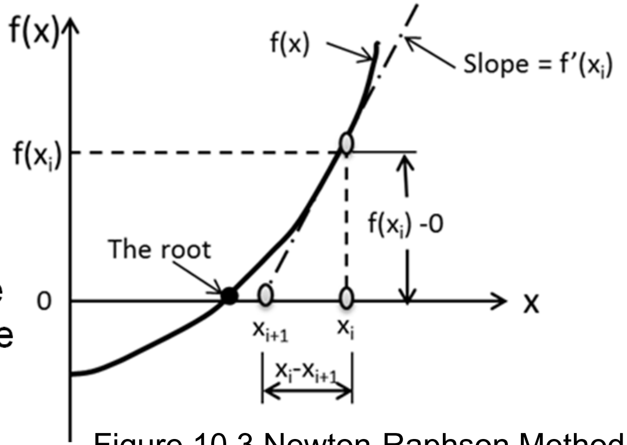


Figure 10.3 Newton-Raphson Method

$$f'(x_i) = \frac{f(x_i) - 0}{x_i - x_{i+1}} \qquad (10.4)$$

which leads to the following expression for the next estimated root at x = x_{i+1} to be:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \qquad (10.5)$$

One would readily notice from Figure 10.3 that the computed approximated next root x_{i+1} is much closer to the real root (shown in filled circle) than the previously estimated value at x_i.

10.3.2 The Newton-Raphson Method-Cont'd

Example 10.2 (p.343)

Use the Newton-Raphson's method to find the roots of the following nonlinear polynomial equation:

$$f(x) = x^4-2x^3+x^2-3x+3=0 \qquad \text{(a)}$$

Solution:

We will express the first order derivative of f(x) in Equation (a) that represent the slope of the curve as required by Equation (10.4):

$$f'(x) = 4x^3 - 6x^2 + 2x - 3 \qquad \text{(b)}$$

Substituting f(x_i) and f'(x_i) into Equation (10.5) for the Newton-Raphson method, we will have the following expression for finding the estimate roots begin with x = x_i:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} = x_i - \frac{4x_i^4 - 2x_i^3 + x_i^2 - 3x_i + 3}{16x_i^3 - 6x_i^2 + 2x_i - 3} \qquad \text{(c)}$$

Thus by estimating the first root at x = x_1 = 0.5 with i = 1, we will have the next estimated x value at x_2 with i = i+1 by using Equation (c) as:

$$x_2 = 0.5 - \frac{4(0.5)^4 - 2(0.5)^3 + (0.5)^2 - 3(0.5) + 3}{16(0.5)^3 - 6(0.5)^2 + 2(0.5) - 3} = 2.03226$$

By following the same procedure, we will find the convergence of the x-values to the root of Equation.(a). The table on the right shows the attempts made to find the first two roots

We will notice from this table that it only took 3 attempts to find the convergence to the first root at x = 1.0. It, however, took 6 attempts to reach a convergence to the second root at x = 1.8637 with an initial estimate of the root at x = 4.

Attempt number (i)	x_i	Computed x_{i+1}	Notes
1	0.5	1.0208	Estimate of first root
2	1.0208	0.9998	
3	0.9998	1.0010	Converges to first root
4	4.0	3.1818	Estimate of second root
5	3.1818	2.4655	
6	2.4655	2.1247	
7	2.1247	1.9382	
8	1.9382	1.8723	Begins to converge to root
9	1.8723	1.8638	
10	1.8638	1.8637	Converges to second root

10.3.2 The Newton-Raphson Method-Cont'd

Example 10.3 (p.344)

Determine the locate the mark line for the content volume of 500 milliliter in a measuring cup with its dimensions shown in the right of Figure 10.4 (p.345).:

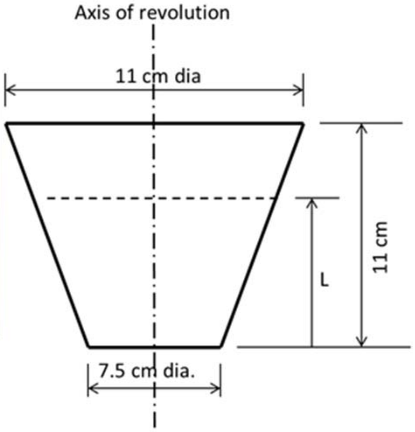
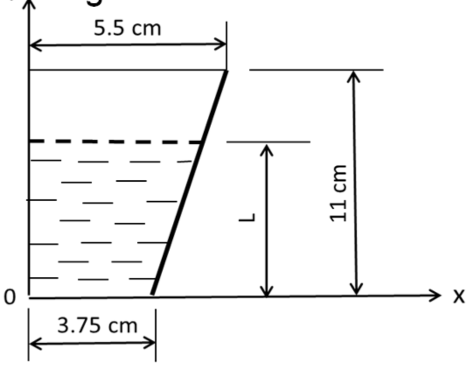


Figure 10.4

Solution:

We assume the mark on the measuring cup for 500 milliliter (mL) is situated at L as indicated in Figure 10.4. We thus need to determine the value of L, so that the volume of the measuring cup with the content at the height L to be 500 mL.

The volume of a solid of revolution of a given length may be determined by Equations (2.16) or (2.17) (pp. 43-44). The profile of the measuring cup in Figure 10.4 is represented by a function x(y) = 0.16y + 3.75 in an x-y coordinate system shown in the right diagram.



We will use Equation (2.17) to determine the volume of the content of the measuring cup with a content level L by the following expreession:

$$V = \pi \int_0^L [x(y)]^2 \, dy = \pi \int_0^L (0.16y + 3.75)^2 \, dy = 0.0268 L^3 + 1.884 L^2 + 44.15 L$$

Since the volume of the measuring cup with the content level L is 500 mL or 500 cm^3, we will have the following equation for the unknown quantity L: $500 = 0.0268L^3 + 1.884L^2 + 44.15L$ or in an alternative form of:

$$L^3 + 70.3L^2 + 1647.39L - 18656.72 = 0 \qquad \text{(a)}$$

We recognize that Equation (a) is a nonlinear cubic equation, and one of the roots of this equation will be the length L, which is the solution of this example.

We will use the Newton/Raphson method to solve the cubic equation in Equation (a).

10.3.2 The Newton-Raphson Method-Cont'd

Let us substitute the unknown L in Equation (a) with usual unknown of nonlinear equations by x. The solution we will seek will be of Equation (b) instead. We will thus have the equation:

$$f(x) = x^3 + 70.3x^2 + 1647.39x - 18656.72 \qquad \text{(b)}$$

and its derivative:

$$f'(x) = 3x^2 + 140.6x + 1647.39 \qquad \text{(c)}$$

We will estimate the root of Equation (a) to be L = x_1 = 4.0 (i = 1), and using Equation (10.5) for the next estimate root to be:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 4 - \frac{f(4)}{f'(4)} = 4 - \frac{(-10878.36)}{2257.79} = 8.818$$

We found the subsequent estimation of the roots of Equation (c) rapidly converging to the following values

$$x_3 = 8.818 - \frac{2022}{3120.47} = 8.17 \quad \text{and} \quad x_4 = 8.17 - \frac{56.5825}{2996.3287} = 8.15$$

The last estimated root x_4 = 8.15 which is close to the solution x = 8.1566 obtained from an online software: Wolfram/Alpha Widgets (www.wolframalpha.com/widgets/view.jsp?id).

We may thus conclude that the mark line of 500 mL for the measuring cup in Figure 10.4 is located at the length L = 8.15 cm from the bottom of the cup.

10.3.2 The Newton-Raphson Method-Cont'd

Example 10.4 (p.346)

In Example 8.9 in Chapter 8 (P.268), we derived the an equation to describe a mass that is attached to a spring that would break when its elongation reached 0.03 m during resonant vibration of the spring-mass system. We need to determine the time t_f at which the spring breaks from Equation (a):

$$\left(0.05 - \frac{t_f}{20}\right)\cos 10t_f + \frac{1}{200}\sin 10t_f - 0.03 = 0 \qquad \text{(a)}$$

Solution:

We will use the Newton-Raphson's method to solve the unknown quantity t_f in Equation (a) by first assuming a solution on t_f = 0.75. We made this assumed solution based on a crude approximated value of t_f = 0.7 in Example 8.9.

Again, let us replace the unknown quantity t_f in Equation (a) by conventional unknown symbol x in the following alternative form:

$$\left(0.05 - \frac{x}{20}\right)\cos 10x + \frac{1}{200}\sin 10x - 0.03 = 0 \qquad \text{(b)}$$

with

$$f(x) = \left(0.05 - \frac{x}{20}\right)\cos 10x + \frac{1}{200}\sin 10x - 0.03 \qquad \text{(c)}$$

and the derivative:

$$f'(x) = -(0.05-0.5x)\sin 10x \qquad \text{(d)}$$

Thus, the estimated root x_{i+1} after the previously estimated root x_i may be computed by using the expression in Equation (10.5), as will be shown in the next slide.

10.3.2 The Newton-Raphson Method-Cont'd

Solution x from the equation: $\left(0.05-\frac{x}{20}\right)\cos 10x+\frac{1}{200}\sin 10x-0.03\ =\ 0$ (a)

We will begin the solution process with an initial estimate of the root of Equation (b) to be x_1 = 0.75, which leads to the following subsequent approximate root x_2 by Equation (10.5):

$$x_2\ =\ x_1-\frac{f\left(x_1\right)}{f'\left(x_1\right)}\ =\ 0.75-\frac{f\left(0.75\right)}{f'\left(0.75\right)}$$

$$=0.75-\left[\left(0.05-0.75/20\right)\cos(10\times 0.75)+\sin(10\times 0.75/200-0.03\right]/$$

$$\left[-\left(0.05-0.5\times 0.75\right)\sin\left(10\times 0.75\right)\right]=0.869979$$

Result of the above computation with x_2 = 0.869979 is presented in Trial No. 1 in the following table.

Trial No.	Assigned x_i	f(x)	f'(x)	x_{i+1}	% Difference
1	0.75	-0.036576	0.30486	0.869979	16
2	0.8	-0.019961	0.346275	0.857644	7.21
3	0.85	-0.003432	0.299433	0.86146	1.35
4	0.9	0.0085058	0.164847	0.848402	-5.73312
5	0.855	-0.001959	0.289694	0.861762	0.790885
6	0.857	-0.001384	0.28555	0.861846	0.565415
7	0.86	0.005367	0.279071	0.861923	0.223604
8	0.863	0.0002905	0.27228	0.861933	-0.12361

We observe that better results lied in the range of: Trial Nos.. 3-4 and 7-8 where the % difference crossed the +ve to –ve numbers in the last column in the above diagram. Rapid convergences to accurate solutions emerged with this method.

10.4 **Numerical Integration Methods** (p.347)

Definite integration of functions over specific intervals of variables that define the functions is a frequent requirement in engineering analysis.

Some of the practical applications of integration are presented in Section 2.3 in Chapter 2.(p.38)

Exact evaluation of many definite integrals can be found in handbooks [Zwillinger 2003] but many others with functions to be integrated are so complicated that analytical solutions for these integrals are not possible. Numerical methods provide viable ways for such evaluation.

In this section, we will present three numerical integration methods:
(1) the trapezoidal rule,
(2) the Simpson's one-third rule, and
(3) the Gaussian quadrature.

The first two methods require given functions in the integrand of the definite integrals.
The last method, the "Gaussian quadrature" will automatically define optimal locations on which the integrations are performed. It is a popular numerical integration method in advanced numerical analyses, such as the finite element method presented in the subsequent chapter.

We will focus our effort on refreshing the principles that are relevant to the development of algorithms of these particular numerical integration methods but not on their proofs.

10.4.1 The trapezoidal rule for numerical integration (p.348)

We have learned from Section 2.2.6 (p.36) that the value of a definite integral of a function y(x) is equal to the area under the curve produced by this function between the upper and lower limits variables of the function as illustrated in Figure 10.6.

Mathematically, the integral of function y(x) can be expressed as:

$$I = \int_{x_a}^{x_b} y(x)dx = Area\ A \qquad (10.6)$$

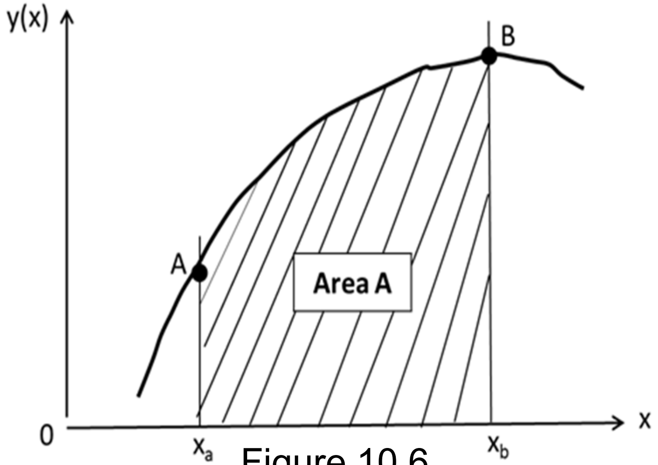


Figure 10.6

The value of the integral of a function may thus be determined by computing the **area** covered by the function between the two specified limits.

For example, the value of the function y(x) in Figure 10.7 may be approximated by the sum of the 3 plane **trapezoids** with areas A_1, A_2 and A_3, with the values of:

$$A_1 = \left(\frac{y_0 + y_1}{2}\right)h \qquad A_2 = \left(\frac{y_1 + y_2}{2}\right)h \qquad A_3 = \left(\frac{y_2 + y_3}{2}\right)h$$

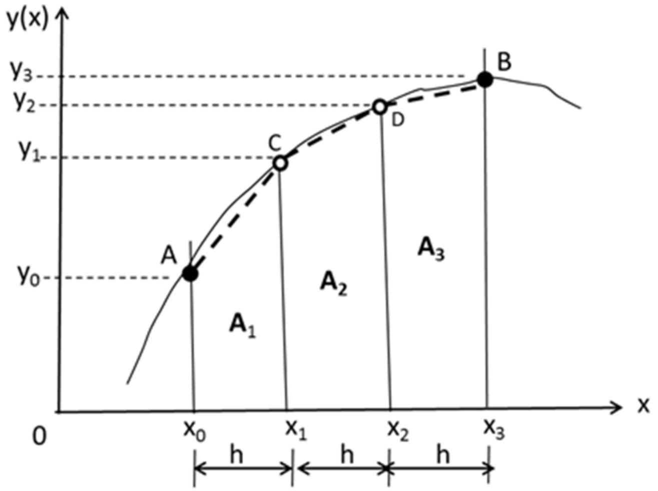


Figure 10.7

The **sum** of A_1, A_2 and A_3 is equal to:

$$A_1 + A_2 + A_3 \approx \int_{x_0}^{x_3} y(x)\,dx \approx \left(\frac{y_o + y_1}{2}\right)h + \left(\frac{y_1 + y_2}{2}\right)h + \left(\frac{y_2 + y_3}{2}\right)h$$

$$= \frac{h}{2}\left(y_0 + 2y_1 + 2y_2 + y_3\right) \qquad (10.7)$$

in which h is the assigned increment of variable x, and y_0, y_1, y_2 and y_3 are the values of the function evaluated at x_0, x_1, x_2 and x_3 respectively.

10.4.1 The trapezoidal rule for numerical integration – Cont'd

Example 10.5 (p.349)

Use the trapezoidal rule to evaluate the integral $\int_{x_a.}^{x_b} y(x)dx$ in which the function $y(x) = x\sqrt{\left(16 - x^2\right)^3}$ with x_a = 0.5 and x_b = 3.5 and assigned increment h = 1.0

Solution:

We will demonstrate the use of trapezoidal rule for numerical integration by plotting the function y(x) verses x as shown in the Figure 10.8:

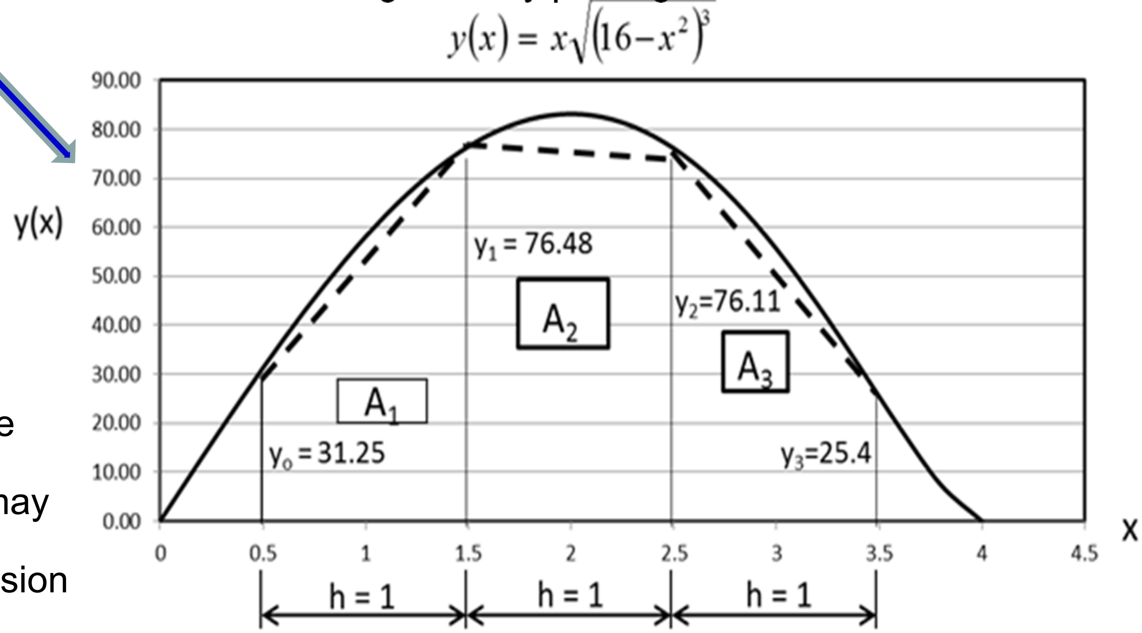


Figure 10.8

The value of the integral that we need to determine is: $I = \int_{0.5}^{3.5} x\sqrt{\left(16 - x^2\right)^3}\,dx$

By using the trapezoidal rule with the three trapezoidal illustrated in Figure 10.8, we may evaluate the integral I by using the expression in Equation (10.7) as:

$$I = \int_{0.5}^{3.5} x\sqrt{\left(16 - x^2\right)^3}\,dx \approx \frac{h}{2}\left(y_o + 2y_1 + 2y_2 + y_3\right) = \frac{1}{2}\left(31.25 + 2x76.48 + 2x76.11 + 25.4\right) = 180.92$$

The above numerical integration resulted with I=180.92 with increment h=1.0 is actually less than the value I=191.45 that one would obtain from "exact integration of the same function from an integration table In a handbook. This discrepancy in results should not be a surprise, as a usual rule of thumb in using numerical approximation method with the "accuracy of the result" closely related to the size of increment h used in the approximations. We will demonstrate such rule in the following description.

10.4.1 The trapezoidal rule for numerical integration with multiple variable increments– Cont'd (p.350)

As a general rule, more accurate results may be obtained in numerical analyses by using smaller variable increment h, which means more trapezoids are involved in the present case of evaluating the integral:

$$I = \int_{x_0}^{x_n} y(x)dx$$

illustrated in Figure 10.9

The approximate value of the integral of function y(x) in Figure 10.9 is equal to the sum of all the trapezoids in the figure by the following equation derived from the same principle as in the foregoing case with three trapezoidal:

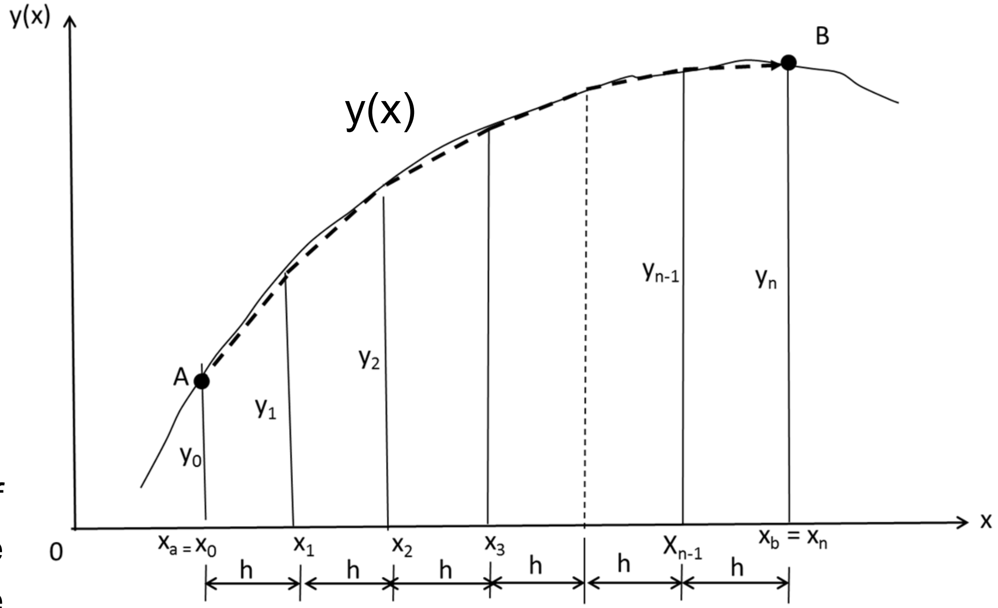


Figure 10.9 Integral of function y(x) with multiple trapezoids

$$I = \int_{x_a}^{x_b} y(x)dx \approx \sum\left(A_1 + A_2 + A_3 + ... + A_{n-1}\right)$$

$$= \frac{h}{2}\left(y_0 + 2y_1 + 2y_2 + 2y_3 + ... + 2y_{n-1} + y_n\right) \qquad (10.8)$$

where h is the size of the variable increment along the x-coordinate in the numerical integration, as shown in Figure 10.9.

10.4.1 The trapezoidal rule for numerical integration – Cont'd

Example 10.6 (p.350)

Evaluate the same integral in Example 10.5 but with a reduced increment h = 0.5.

The value of the integral that we need to determine is: $I = \int_{0.5}^{3.5} y(x)dx = \int_{0.5}^{3.5} x\sqrt{\left(16-x^2\right)^3}\, dx$

with the function $y(x) = x\sqrt{\left(16-x^2\right)^3}$

The area covered by the function between x=0 and x=3.5 is approximated by what is shown in Figure 10.10.

We will first determine the function values y(x) at x = 0.5, 1.0, 1.5, 2.0, 2.5, 3.0 and 3.5, as shown in the following table:

n	x_n	y_{n-1}
1	0.5	31.25
2	1.0	58.09
3	1.5	76.48
4	2.0	83.18
5	2.5	76.11
6	3.0	55.56
7	3.5	25.42

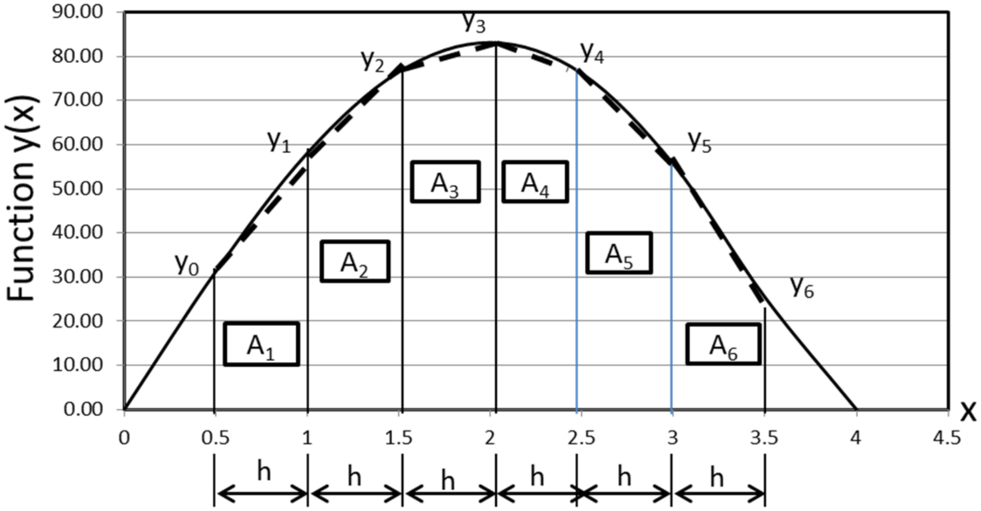


Figure 10.10 Integration of a function y(x) with 6 trapezoids

This table also shows the function values, y_1, y_2, y_3, y_4, y_5 and y_6 required in the computation of the areas of the five trapezoids in Figure 10.10. We may use Eq.(10.8) to compute the sum of the areas of the five trapezoids to be:

$$I \approx \frac{0.5}{2}\left(31.25 + 2x58.09 + 2x76.48 + 2x83.18 + 2x76.11 + 2x55.56 + 25.42\right) = 188.88$$

We obtained the value of 188.88 for the same integral I with h = 0.5 is much closer to the analytical value of 191.45 than what we obtained with I=180 with h = 1.0 in Example 10.5, which demonstrates the fact that the smaller increment h, leads to more accurate results using numerical methods.

10.4.2 **Numerical Integration by Sampson's One-Third Rule** (p.352)

We have learned that we may evaluate an integral by summing the plane areas under the curve representing the function (the integrand) in the integral between two limits of the variable in the integration, as illustrated in Figure 10.11(a):

We notice from this figure that the area under the function y(x) is made up by 3 straight edges and the arc AB (not a straight line).

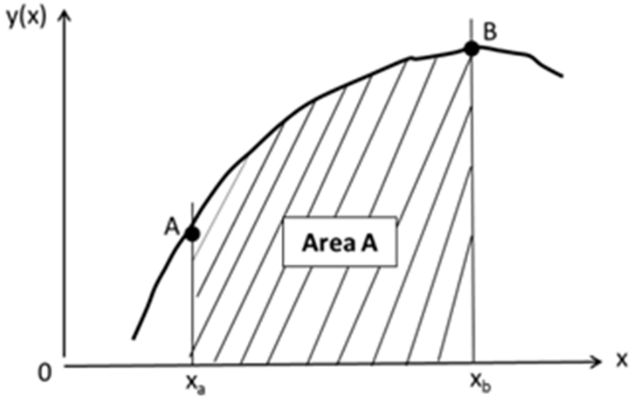


Figure 10.11(a)

We use the trapezoidal method to evaluate the integral

$$I = \int_{x_a}^{x_b} y(x)dx$$

by approximating the area under the function y(x) between x_a and x_b by a trapezoid as illustrated in Figure 10.11(b) with a straight edge in dotted line, instead of an arc AB.

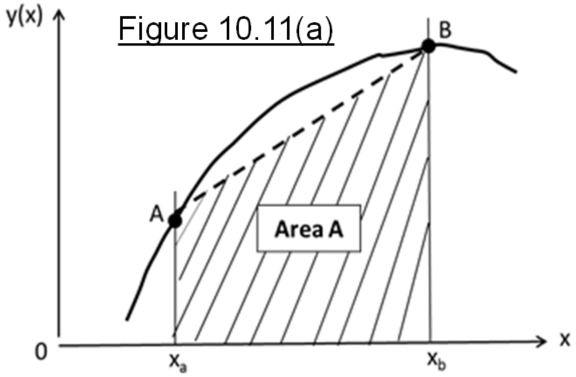


Figure 10.11 (b)

One would not hesitate to recognize the discrepancy in results from using the trapezoidal approximation method using trapezoids, instead of using the arc AB for the area.

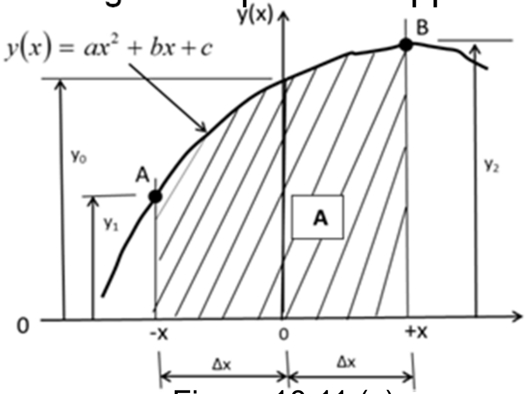


Figure 10.11 (c)

The **Simpson rule**, in particular, the "Simpson one-third rule." differs from the trapezoidal method by assuming a parabolic function y(x) = ax^2 + bx + c connecting adjacent points A and B on the curve representing the function y(x) as shown in Figure 10.11(c). This parabolic function with constant coefficients a,b and c to be determined by the Sampson one-third rule will not only result in faster convergence to results but also offers much accurate results in numerical integrations.

10.4.2 Numerical Integration using Sampson's One-Third Rule – Cont'd

Math formulation of Sampson One-third Rule for numerical integration (p.353):

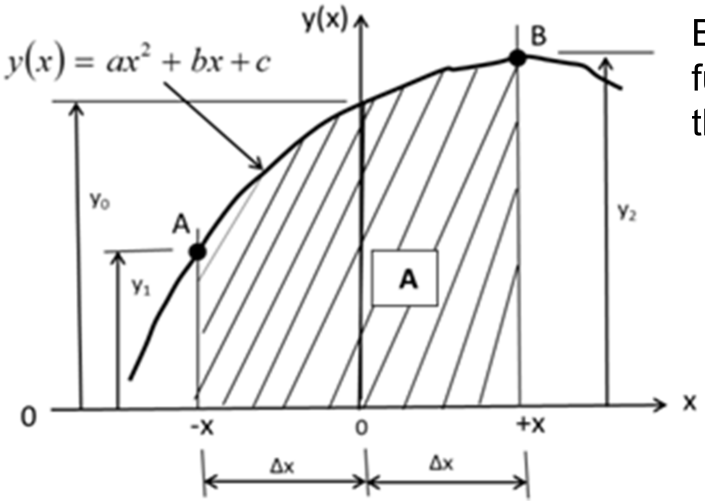


Figure 10.11(c)

By referring to the diagram in Figure 10.11(c), in which the function in the integrand is: y(x) = ax^2 + bx + c , we will have the function evaluated at three points at x = -x, x=0 and x=+x:

$$y_0 = a(-x)^2 + b(-x) + c$$

$y_1 = c$ and $y_2 = a(x)^2 + b(x) + c$ from which we may solve for:

$$a = \frac{1}{2x^2}\left(y_0 - 2y_1 + y_2\right) \qquad (10.9a)$$

$$c = y_1 \qquad (10.9b)$$

The value of the integral of the function y(x) is equal to the plane area A in Figure 10.11(c), or

$$I = AREA\,(-xABx) = \int_{-\Delta x}^{+\Delta x} y(x)dx = \int_{-\Delta x}^{+\Delta x}\left(ax^2 + bx + c\right)dx = \left(\frac{ax^3}{3} + \frac{bx^2}{2} + cx\right)\Bigg|_{-\Delta x}^{+\Delta x}$$

$$= \frac{\Delta x}{3}\left[2a(\Delta x)^2 + 6c\right]$$

By substituting the constant coefficients a and c in Equation (10.9a, b), together with Δx = x in Figure 10.11(c) into the above expression, we will get the following relation for the Simpson one-third rule for the integral I:

$$I = \int_{-\Delta x}^{+\Delta x} y(x)dx = \int_{-x}^{+x}\left(ax^2 + bx + c\right)dx = \frac{\Delta x}{3}\left(y_0 + 4y_1 + y_2\right) \qquad (10.10)$$

10.4.2 Numerical Integration by Sampson’s One-Third Rule – Cont’d

Example 10.7 (p.353)

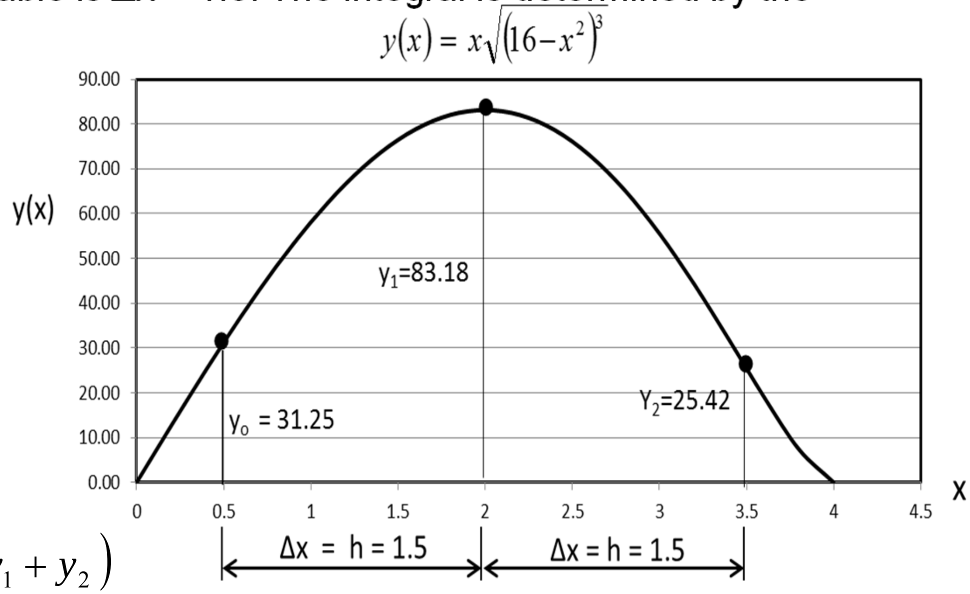
Use Simpson one-third rule to find the numerical value of the integral in Example 10.5: I= $\int_{x_a}^{x_b} y(x)dx$ in which

$$y(x) = x\sqrt{\left(16-x^2\right)^3}$$

Solution:

We will use the three function values at x = 0.5, 2.0 and 3.5 to compute the value of the integral.

In such case, the increment of the integration variable is Δx = 1.5. The integral is determined by the expression in Equation (10.10).



We may obtain the function values y_0, y_1 and y_2 at x = 0.5, 2 and 3.5 from the Table in Example 10.6 as:

y_0 = 31.25, y_1 = 83.18 and y_2 = 25.42.

Integration of the function y(x) in this example5 can thus be determined by substituting the values of y_0, y_1 and y_2 and the increment of x, Δx = 1.5 into Equation (10.10) to give:

$$I = \int_{-\Delta x}^{+\Delta x} y(x)dx = \int_{-x}^{+x}\left(ax^2 + bx + c\right)dx = \frac{\Delta x}{3}\left(y_0 + 4y_1 + y_2\right)$$

$$= \int_{0.5}^{3.5} y(x)dx = \int_{0.5}^{3.5} x\sqrt{\left(16-x^2\right)^3}\,dx = \frac{1.5}{3}\left(31.25+4x83.18+25.42\right) = 194.70$$

The “exact” solution of the above integral is 191.45 from a math handbook, from which we may have the following comparison of results between the I = 188.88 by three trapezoidal in Example 10.5 and the current solution I = 194.70 with three function values using the Simpson one-third rule.

10.4.2 Numerical Integration by Sampson's One-Third Rule – Cont'd

(use multi-values of the function in integrand) (p.354)

One may use Equation (10.10) to evaluate integrals involving more than three function values between two specified limits for the integration. We will use the same illustration in Figure 10.13 to derive the general expression for the Simpson one-third rule for numerical integration.

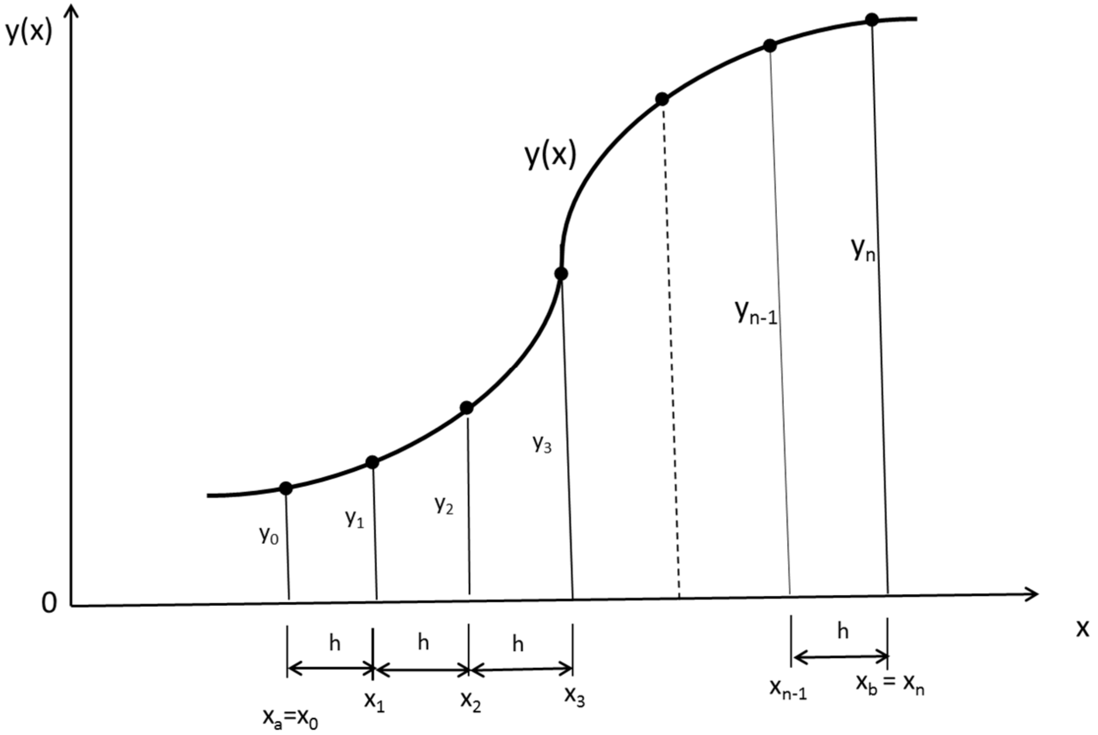


Figure 10.13

From which, we may thus express the general expression of Simpson one-third rule in the following equation:

$$I = \int_{x_a}^{x_b} y(x)dx = \frac{h}{3}\left(y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 + \ldots\ldots\ldots\ldots\ldots\ldots + 2y_{n-2} + 4y_{n-1} + y_n\right) \qquad (10.11)$$

10.4.2 Numerical Integration by Sampson's One-Third Rule – Cont'd

Example 10.8 (p.355)

Use the Simpson one-third rule in Equation (10.10) to evaluate the integral in Example 10.6:

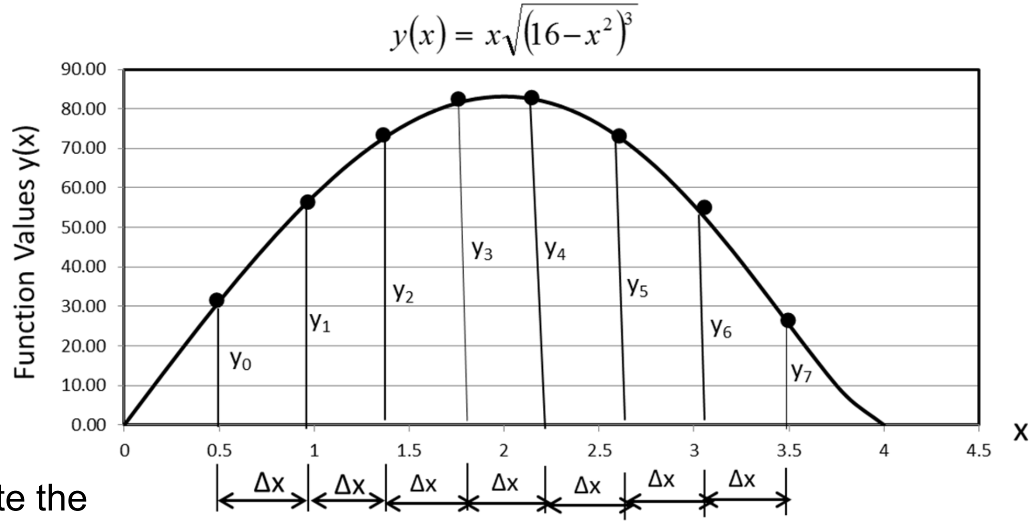
$$I = \int_{0.5}^{3.5} y(x)dx = \int_{0.5}^{3.5} x\sqrt{\left(16-x^2\right)^3}\,dx$$

Solution:

Because the function y(x) in the integral that we need to evaluate is same as we had in Example 10.6, so we will use the same table for the values of the present function with Δx = 0.43 for the Sampson method.
The eight function values are presented in the following Table with these data points shown in Figure 10.14.

n	1	2	3	4	5	6	7	8
x_n	0.5	0.93	1.36	1.79	2.21	2.64	3.07	3.5
$y = y(x_n)$	31.25	54.69	72.3	81.89	81.85	71.54	51.68	25.41

Figure 10.14



We can thus use Equation (10.11) and evaluate the integral I to be:

$$I = \int_{0.5}^{3.5} y(x)dx = \int_{0.5}^{3.5} x\sqrt{\left(16-x^2\right)^3}\,dx \approx \frac{\Delta x}{3}\left(y_0+4y_1+2y_2+4y_3+2y_4+4y_5+2y_6+y_7\right)$$

$$= \frac{0.43}{3}\left(31.25+4x54.69+2x72.30+4x81.89+2x81.85+4x71.54+2x51.68+25.41\right) = 186.45$$

10.4.3 Numerical Integration by Gaussian Quadrature (p.356)

Most numerical methods for evaluating definite integrals require the users to select the sampling points and evaluate the integral in terms of the discrete values of the function at these points.

These methods usually work well with well-behaved functions in the integral, but they do not offer any guideline on the selection of the size of the increment (h or Δx), in numerical computations, as in the previous formulations.

There are times when engineers are expected to find numerical values of integrals involving functions that have drastic change of shapes over the range of the required integrations. These methods do not yield good approximation of the numerical values of the integral because of improper selection of sampling points.

Gaussian quadrature method has the advantage of offering the users with criteria on optimal sampling points in numerical integration. It was established on the basis of strategically selected sampling points. The normal form of a Gaussian integral can be expressed as:

$$I = \int_{-1}^{1} F(\xi)d\xi = \sum_{i=1}^{n} H_i F(a_i) \qquad (10.12)$$

where n is the total number of sampling points, H_i are the weighting coefficients corresponding to sampling points located at : $\xi = \pm a_i$ as given in Table 10.3 on P. 357.

The form of Gaussian integral shown in Equation (10.12) is rarely seen in practice. A transformation of coordinate is required to convert the general form of integration such as shown in Equation (10.6) on p.348 to the form shown in Equation (10.12), as illustrated in Figure 10.15.

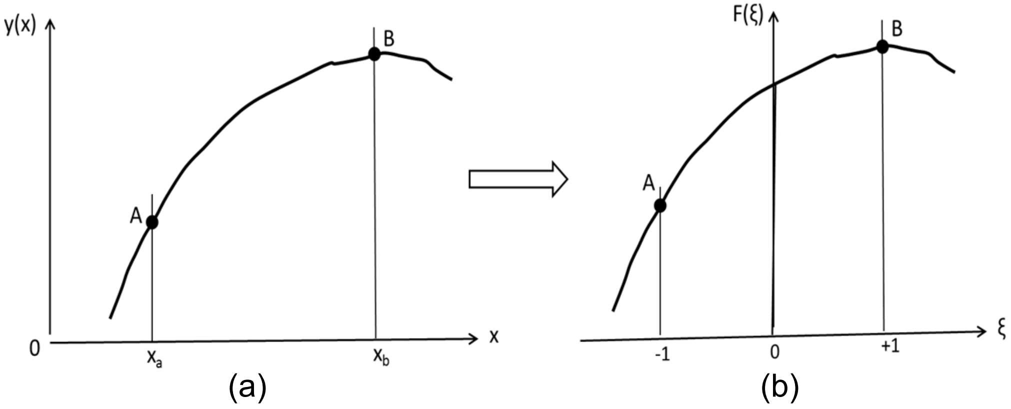


Figure 10.15 Transform of coordinates

10.4.3 Numerical Integration by Gaussian Quadrature – Cont’d on formulation (p.356)

The transformation of coordinates from y(x) in the x-coordinate to the function F(ξ) in the coordinate ξ may be accomplished by using the following relationship:

$$x = \frac{1}{2}(x_b - x_a)\xi + \frac{1}{2}(x_b + x_a) \qquad (10.13)$$

which leads to the following expression:

$$\int_{x_a}^{x_b} y(x)dx = \frac{x_b - x_a}{2}\int_{-1}^{1} F(\xi)d\xi \qquad (10.14)$$

where $F(\xi) = y(x)\Big|_{x=\frac{1}{2}(x_b - x_x)\xi + \frac{1}{2}(x_b - x_a)}$ and $d\xi = \left[\frac{1}{2}(x_b - x_a)\right]^{-1} dx$

We will obtain the expression for the required evaluation of the integral in Equation (10.6) using Gaussian quadrature by substituting the relationship in Equation (10.12) into the above expression in Equation (10.14) in the following form:

$$I = \int_{x_a}^{x_b} y(x)dx = \frac{x_b - x_a}{2}\sum_{i=1}^{n} H_i\, F(a_i) \qquad (10.15)$$

10.4.3 Numerical Integration by Gaussian Quadrature – Cont'd

Example 10.9 (p.358)

Evaluate the following integral by using the Gaussian quadrature in Equation (10.14).

$$I = \int_0^{\pi} \cos x\, dx \qquad \text{(a)}$$

Solution:

We have the function y(x) = cosx over the integration limits x_a = 0 and x_b = π. The transformation of coordinates makes use of the relationship $x = \frac{\pi}{2}\xi + \frac{\pi}{2}$ from Equation (10.13), from which we get:

$$y(x) = \cos x = F(\xi) = \cos\left(\frac{\pi}{2}\xi + \frac{\pi}{2}\right) = \sin\left(\frac{\pi}{2}\xi\right)$$

Also, from Equation (10.14) with the use of the trigonometric relationships such as:

$$\sin\left(\frac{\pi}{2} + \theta\right) = \cos\theta \quad \textit{and} \quad \cos\left(\frac{\pi}{2} + \theta\right) = \sin\theta$$

We may arrive at the following expression for integrating I in Equation (a) using Gaussian quadrature:

$$I = \int_0^{\pi} \cos x\, dx = \int_{-1}^{1}\left[\sin\left(\frac{\pi}{2}\xi\right)\left(\frac{\pi}{2}d\xi\right)\right] = \frac{\pi}{2}\int_{-1}^{1}\sin\frac{\pi}{2}\xi\, d\xi = \frac{\pi}{2}\sum_{i=1}^{n} H_i \sin\left(\frac{\pi}{2}a_i\right) \qquad \text{(b)}$$

Let us take, for example, 3 sampling points, i.e., n = 3 from Table 10.3 on P.357 with:

$a_1 = 0$	$a_2 = +0.77459$	$a_3 = -0.77459$
$H_1 = 0.88888$	$H_2 = 0.55555$	$H_3 = 0.55555$

Substituting the above numbers into Equation (b) will lead to the solution:

$$I = \frac{\pi}{2}\left[0.88888 \;\sin(0) + 0.55555 \;\sin\left(\frac{\pi}{2}\times 0.77459\right) + 0.55555 \;\sin\left(-\frac{\pi}{2}\times 0.77459\right)\right]$$

$$= \frac{\pi}{2}\left[0.5555\;5 \sin(1.2167) - 0.55555 \;\sin(1.2167)\right] = 0$$

Example 10.11 (p.360)

Evaluate the following integral in Example 10.8 using Gaussian quadrature method

$$I = \int_{0.5}^{3.5} y(x)dx = \int_{0.5}^{3.5} x\sqrt{\left(16-x^2\right)^3}\, dx \qquad \text{(a)}$$

Solution:

We notice the function y(x) in the integral in Equation (a) is identical to what we had in previous Examples 10.5 (p.349), 6 (p.350) and 8 (p.355).

We will first derive the expression of the function F(ξ) for the function y(x) in Equation (a) from the integration limits of 0.5 and 3.5 to the limit -1 and +1, as required in Gaussian quadrature.

The transformation of coordinate systems as illustrated in Figure 10.15 begins with the transformation of variable from x to ξ using the relationship in Equation (10.13), leading to the following relationship between the variables x and ξ as shown below:

$x = \frac{1}{2}(x_b - x_a)\xi + \frac{1}{2}(x_b + x_a) = 1.5\xi + 2$, from which we obtain: dx = 1.5 dξ (c)

The integral in Equation (a) in the y(x) vs.x coordinates can thus be transformed to the F(ξ) vs. ξ coordinates by using Equation (10.14), yielding the following expression:

$$I = \int_{0.5}^{3.5} y(x)dx = \int_{0.5}^{3.5} x\sqrt{\left(16-x^2\right)^3}\, dx = \frac{x_b - x_a}{2}\int_{-1}^{1} F(\xi)d\xi$$

$$= \frac{3.5-0.5}{2}\int_{-1}^{1} y(x)\Big|_{x=1.5\xi+2} d\xi \qquad \text{(d)}$$

We may thus evaluate the integral with the expression in Equation (d) as:

$$I = 1.5\int_{-1}^{1}(1.5\xi+2)\sqrt{\left[16-(1.5\xi+2)^2\right]^3}\, d\xi = 1.5\int_{-1}^{1}(1.5\xi+2)\sqrt{\left(-2.25\xi^2-6\xi+12\right)^3}\, d\xi$$

10.4.3 Numerical Integration by Gaussian Quadrature – Cont'd on solution of Example 10.11

We have arrived at the following integral:

$$I = 1.5\int_{-1}^{1}(1.5\xi+2)\sqrt{\left[16-(1.5\xi+2)^2\right]^3}\,d\xi = 1.5\int_{-1}^{1}(1.5\xi+2)\sqrt{\left(-2.25\xi^2-6\xi+12\right)^3}\,d\xi$$

from which we have the function in the form of Gaussian quadrature:

$$F(\xi) = (1.5\xi+2)\sqrt{\left(-2.25\xi^2-6\xi+12\right)^3} \qquad \text{(e)}$$

We may thus use Equation (10.12) for the value of the integral I in Equation (a) to be:

$$I = 1.5\sum_{i=1}^{n} H_i\, F(a_i) \qquad \text{(f)}$$

Let us choose 3 sampling points for the integral in Equation (f), i.e., n= 3 in Equation (e) with a_i and H_i (i = 1, 2, 3) from Table 10.3, as tabulated below:

i	a_i	H_i
1	0	0.88888
2	0.77459	0.55555
3	-0.77459	0.55555

By substituting the above numbers into Equation (f), we will have the value of the integral I as:

$$I = 1.5\,[H_1F(0) + H_2F(0.7746) + H_3F(-0.7746)] \qquad \text{(g)}$$

We may evaluate the function values at a_i using Equation (e) as:

F(0) = 83.1384, F(0.77459) = 46.4981, and F(-0.77459) = 50.1454 (10.16)

Substituting these function values into Equation (e) will result the integral I in Equation (a) as:

$$I = \int_{0.5}^{3.5} y(x)dx = \int_{0.5}^{3.5} x\sqrt{\left(16-x^2\right)^3}\,dx$$

$$=1.5\left(0.8888x83.1384+0.5555x46.4981+0.5555x50.1454\right) = 191.3684$$

This value of the integral in Equation (a) obtained by Gaussian quadrature with three sampling points is remarkably close to the exact value of 191.45 obtained from a math handbook.

10.5 Numerical Methods for Solving Differential Equations (p.361)

We have learned the solutions of "linear differential equations," in Chapters 7, 8 and 9, such as shown in the following equation: $\frac{d^2\,u(x)}{dx^2}$- $4\frac{du(x)}{dx} = -12x - 5$ with solution method available in Section 8.6.

There are, however, occasions in which engineers need to solve either highly complicated linear differential equations or nonlinear differential equations; such as in the form of: $\left[\frac{du(x)}{dx}\right]^2 - 4\frac{du(x)}{dx} = 9x^2$- 4, in such cases numerical solution methods become viable alternative methods for the solutions.

Numerical solution methods for differential equations relating to two types of engineering analysis problems: (1) The "initial value" problems, and (2) the "boundary value" problems.

Numerical solution methods will offer the users with solution of the differential equations at "solution points", but not everywhere within the variable domains.

Solution of initial value problems involves a starting point with the variable of the function, say at x_0, which is a specific value of variable x for solution y (x). With the solution given at this starting point, one may find the solutions at x= x_o+h, x_o+2h, x_o+3h,…………., x_o+nh, in which h is the selected "step size" in the numerical computations and n is an integer number of steps used in the analysis.

Numerical solution to boundary value problems is more complicated, in which function values are often specified at certain variables, and there are restrictions on how the selected steps for solution values may be restricted by the specified values at these variable points.

The number of steps n in the computation can be as large as what takes to cover the entire range of the variable in the analysis, or as small as selected by the users. Like all numerical solution methods, smaller solution steps would offer more accurate results as in general cases.

There are many numerical solution methods available for engineers to solve differential equations. We will present: (1) The finite difference method to illustrate the principles of converting "differential equations" to "difference equations", and (2) the Runge-Kutta method - a popular method by engineers..

10.5.1 The principle of finite difference method (p.362)

The essence of the “finite difference method” (FDM) is to convert the “derivatives” in the differential equations into “difference”, so that differential equation may be expressed in algebraic equations in terms of the converted format of “differences.”

With regard to Figure 10.16, we have a continuous function f(x) that has values f_{i-1}, f_i and f_{i+1} at the variables x_{i-1}, x_i and x_{i+1} respectively. We may also designate the three function values at the three x-values to be:

$$f_i = f(x) \qquad (10.16a)$$

$$f_{i+1} = f(x + \Delta x), \text{ and} \qquad (10.16b)$$

$$f_{i-1} = f(x - \Delta x) \qquad (10.16c)$$

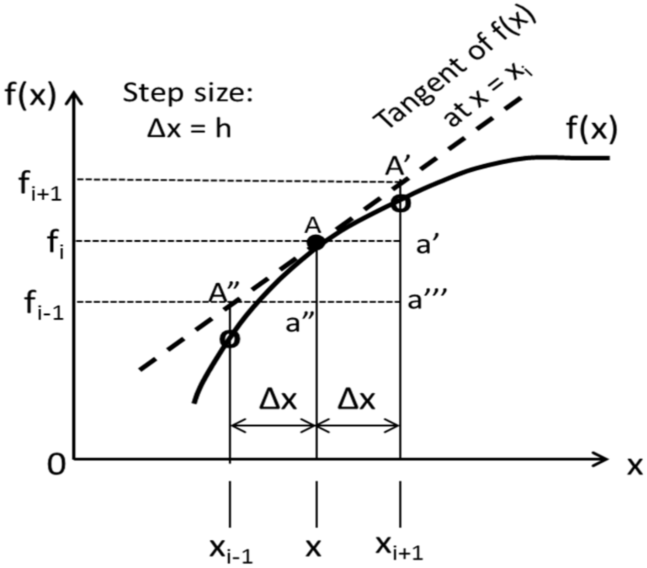


Figure 10.16 Function Evaluated at 3 positions

The derivative of the function f(x) at Point A with $x = x_i$ in Figure 10.16 is graphically represented by the tangent line A”-A’ to the curve representing function f(x) at point A. Mathematically, we may express the derivative as given in Equation (2.9) on p.34, or in the form:

$$\frac{df(x)}{dx} = \ell im_{\Delta x \to o} \frac{\Delta f}{\Delta x} \approx \frac{\Delta f}{\Delta x} \qquad (10.17)$$

where Δx is the increment of variable x used in the above derivative.

One may observe an important relation from Equation (10.17) that the derivative may be approximated by the finite increments of Δf corresponding to Δx as indicated in Equation (10.18).

$$\frac{df(x)}{dx} \approx \frac{\Delta f}{\Delta x} \qquad (10.18)$$

We thus realize that derivatives of continuous functions may be approximated by adopting **finite**, but not the infinitesimally small increments of variable x. Formulation with such approximation is called “***finite difference***”.

10.5.2 The 3 Basic Finite-Difference Schemes (p.363)

1) *The forward difference scheme:*

In this "forward difference scheme," the rate of change of the function f(x) with respect to the variable x is accounted for between the function value at the current value at x = x_i and the value of the same function at the next step, i.e. $x_{i+1} = x + \Delta x$ in the triangle ΔA'Aa' in Figure 10.16.

The mathematical expression of this scheme is shown in Equation (10.19).

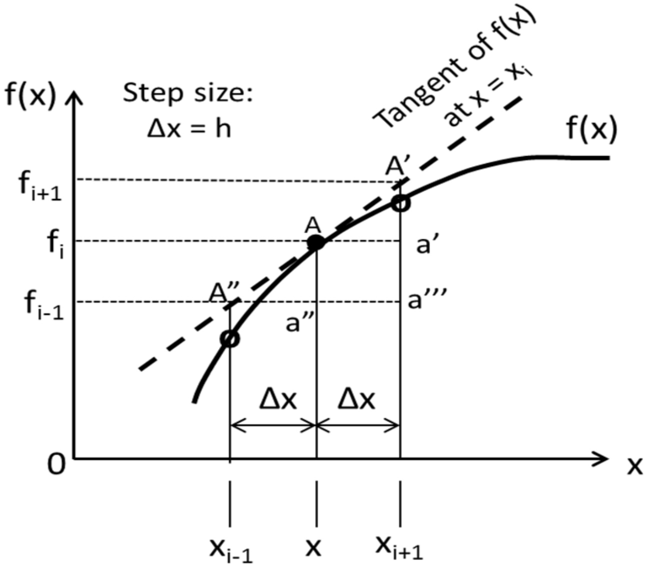


Figure 10.16 Function Evaluated at 3 positions

$$\nabla f_i = \left.\frac{df(x)}{dx}\right|_{x=x_i} \approx \left.\frac{\Delta f}{\Delta x}\right|_{x=x_i} = \frac{f_{i+1}-f_i}{x_{i+1}-x_i} = \frac{f_{i+1}-f_i}{\Delta x} = \frac{f_{i+1}-f_i}{h} \qquad (10.19)$$

in which h = Δx is the "step size."

The derivative of the function f(x) at other values of the variable x in the positive direction can be expressed following Equation (10.19) to be:

$$\nabla f_{i+1} = \frac{f_{i+2}-f_{i+1}}{h}$$
$$\nabla f_{i+2} = \frac{f_{i+3}-f_{i+2}}{h}, \ etc. \qquad (10.20)$$

The second order derivative of the function f(x) at x can be derived by the following procedure:

$$\nabla^2 f_i = \left.\frac{d}{dx}\left(\frac{df(x)}{dx}\right)\right|_{x=x_i} = \lim_{\Delta x\to 0}\frac{\nabla f|_{x_{i+1}} - \nabla f_i}{\Delta x} \approx \frac{\nabla f_{i+1}-\nabla f_i}{\Delta x} = \frac{\nabla f_{i+1}-\nabla f_i}{h}$$

$$= \frac{\frac{f_{i+2}-f_{i+1}}{h} - \frac{f_{i+1}-f_i}{h}}{h} = \frac{f_{i+2}-2f_{i+1}+f_i}{h^2} \qquad (10.21)$$

10.5.2 The 3 Basic Finite-Difference Schemes – Cont'd

2) *The backward difference scheme:*

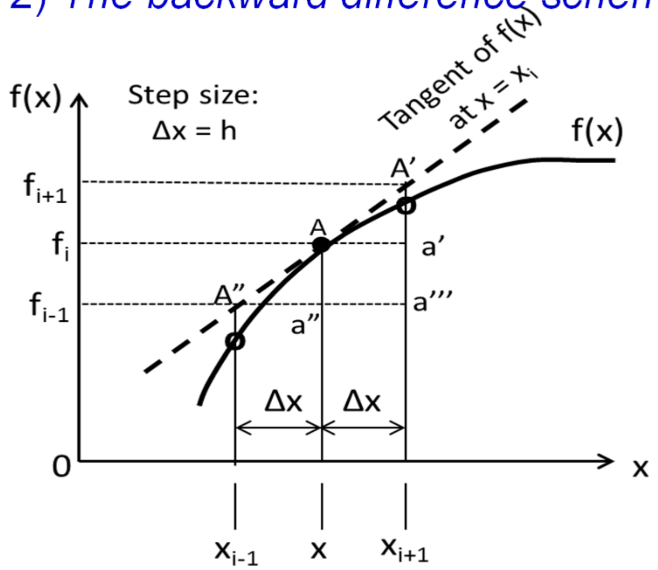


Figure 10.16 Function Evaluated at 3 positions

In this difference scheme, the rate of the change of the function with respect to the variable x is accounted for between the current value at x = x_i and the step backward, i.e. $x_{i-1} = x - \Delta x$ in the triangle ΔAA''a'' in Figure 10.16. The mathematical expression of this scheme is given in Equation (10.22):

$$\nabla f_i = \underset{\Delta x \to 0}{\ell im} \frac{f_i - f_{i-1}}{\Delta x} = \underset{\Delta x \to 0}{\ell im} \frac{f(x) - f(x-\Delta x)}{\Delta x} \approx \frac{f_i - f_{i-1}}{\Delta x} = \frac{f_i - f_{i-1}}{h} \qquad (10.22)$$

Following a similar procedure in the forward difference scheme, we may express the second order derivative in the following form:

$$\nabla^2 f_i \approx \frac{f_i - 2f_{i-1} + f_{i-2}}{h^2} \qquad (10.23)$$

3) *The Central difference scheme:*

The rate of change of function f(x) in this finite difference scheme accounts the function values between the previous step at (x-Δx) and the step ahead, i.e. (x+Δx). The triangle involved in this difference scheme is ΔA'A''a''' in Figure 10.16. We have the first order derivative expressed in Equation (10.24).

$$\nabla f_i \approx \frac{f_{i+1} - f_{i-1}}{x_{i+1} - x_{i-1}} = \frac{f_{i+1} - f_{i-1}}{2h} \qquad (10.24)$$

Equation (10.24) involves a larger step of the size "2h" in the first order derivative. These "coarse" steps will compromise the accuracy of the values of the derivatives. A better central difference scheme is to account for the "half" steps in both directions. We may improve the step size in Eq.(10.25):

$$f_{i+\frac{1}{2}} = f\left(x + \frac{\Delta x}{2}\right) \quad and \quad f_{i-\frac{1}{2}} = f\left(x - \frac{\Delta x}{2}\right) \qquad (10.25)$$

leading to:

$$\nabla f_i \approx \frac{f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}}{h} \qquad (10.26)$$

10.5.2 The 3 Basic Finite-Difference Schemes – Cont'd

Example 10.12 (p.364)

Solve the following differential equation using the forward finite difference scheme.

$$\frac{d^2 x(t)}{d t^2} + x(t) = 0 \qquad \text{(a)}$$

with specified initial conditions:

$$x(0) = 1 \qquad \text{(b)}$$

$$\dot{x}(0) = 0 \qquad \text{(c)}$$

Solution:

Let us use the *forward difference scheme* following Equations (10.19) and (10.21) with the finite difference scheme:

$$\frac{dx(t)}{dt} \approx \frac{x(t+\Delta t) - x(t)}{\Delta t} \qquad \text{(d)}$$

and

$$\frac{d^2 x(t)}{d t^2} \approx \frac{x(t+2\Delta t) - 2x(t+\Delta t) + x(t)}{(\Delta t)^2} \qquad \text{(e)}$$

Substituting Equation (e) into Equation (a) results in the following finite difference form of the differential equation:

$$\frac{x(t+2\Delta t) - 2x(t+\Delta t) + x(t)}{(\Delta t)^2} + x(t) = 0$$

Upon re-arranging the terms in the above equation, we get the following "recurrence relation" for the approximate solution of Equation (a):

$$x(t+2\Delta t) - 2x(t+\Delta t) + [1+(\Delta t)^2]\,x(t) = 0 \qquad \text{(f)}$$

with $\mathrm{x}(0) = 1$ (g)

and $\mathrm{x}(\Delta \mathrm{t}) = \mathrm{x}(0) = 1$ (h)

Example 10.12 – Cont'd

The finite difference equation: $x(t+2\Delta t)-2x(t+\Delta t)+[1+(\Delta t)^2]x(t) = 0$ (f)

with $x(0) = 1$ (g)

and $x(\Delta t) = x(0) = 1$ (h)

We are now ready to solve for x(t) in Equation (a) using the finite difference operator by repeated use of the recurrence relation in Equation (f).
The solution of x(t) will be on the incremental steps of Δt chosen by the user. By referring to the initial conditions in Equations (g) and (h) we may get the x(t) at all subsequent steps

Choice of solution steps, Δt:

Let us assume that a step size *Δt = 0.05* is chosen for the solution.
The corresponding solution becomes, from Equation (f) with *Δt = 0.05:*

$$x(t+0.1) - 2x(t+0.05) + 1.0025x(t) = 0 \quad (j)$$

and from Equation (h): $x(0.05) = 1$

$$x(0.1) - 2x(0.05) + 1.0025x(0) = 0$$

Because *x(0)* is the initial condition in Equation (g), so the above relation yields

$$x(0+0.1) - 2x(0+0.05) = -1.0025$$

But since x(0.05) = 1 from Equation (h), we have:

$$\mathbf{x(0.1)} = -1.0025 + 2x(0.05) = \mathbf{0.9975} \quad (k)$$

Example 10.12 – Cont'd

The finite difference equation: $x(t+2\Delta t)-2x(t+\Delta t)+[1+(\Delta t)^2]x(t) = 0$ (f)

with $x(0) = 1$ (g)

and $x(\Delta t) = x(0) = 1$ (h)

We may now move to the next time point by letting **t = t +** Δt = 0 + 0.05 = **0.05** and having t = 0.05 substituted into Equation (f) and get:

$$x(0.05+0.1) - 2x(0.05 + 0.05) + 1.0025x(0.05) = 0,$$

or $$x(0.15)\text{-}\ 2x(0.1) + 1.0025x(0.05) = 0 \qquad \text{(m)}$$

Since we have already obtained x(0.1) = 0.9975 from Equation (k) and x(0.05) = 1 from Equation (h), we will thus have another solution point from Equation (m):

$$\mathbf{x(0.15)} = 2\text{x}0.9975 - 1.0025\text{x}1 = \mathbf{0.9925} \qquad \text{(p)}$$

We will move to the next time point that is **t = t +** Δ**t** = 0.05 + 0.05 = **0.**1. Substituting t=0.1 onto Equation (f), we get:

$$x(0.1 + 0.1) - 2x(0.1 + 0.05) + 1.0025x(0.1) = 0$$

But with x(0.15) = 0.9925 from the last step in Equation (p) and x(0.1) = 0.9975 from Equation (k), we will have: $x(0.2) - 2x(0.15) + 1.0025x(0.1) = 0$

or $$\mathbf{x(0.2)} = 2\text{x}0.9925 - 1.0025\text{x}0.9975 = \mathbf{0.9850} \qquad \text{(q)}$$

10.5.2 The 3 Basic Finite-Difference Schemes – Cont'd

Example 10.12 – Cont'd

Thus, by following the same procedure as illustrated above, we may obtain solution of Equation (a) at all time points with an increment $\Delta t = 0.05$.

$$\frac{d^2 x(t)}{d t^2} + x(t) = 0 \tag{a}$$

with given conditions:

$$x(0) = 1 \tag{b}$$

$$\dot{x}(0) = 0 \tag{c}$$

The results obtained from the forward difference scheme are summarized in the Table below, with the comparison to the exact solution of x(t) = cost:

Variable, t	Solution by the finite difference method	Exact solution	% Error
0	1	1	0
0.05	1	0.999996	≈0
0.10	0.9975	0.9950041	0.25
0.15	0.9925	0.98877	0.38
0.20	0.9850	0.980066	0.503

One will observe from the above-tabulated values that the percentage of error of the results obtained from the finite difference method increases with the increase of variable t, and *prove that the accuracy of the finite difference method improves with smaller increments of the variables, e.g. the Δt in the present example*

10.5.4 A Popular Numerical Solution Method of Solving Differential Equations

-The Runge-Kutta Methods (p.367)

There are a number of numerical solution methods available for solving differential equations relating to both types of initial value and boundary value problems. Readers can find detailed description of these methods from special books on this subject from the references cited in this book. What we will present in this Section are the two popular numerical solution methods on both the 2nd-order and the 4th-order Runge-Kutta methods.

Both these Runge-Kutta methods are integrative methods for approximation of solutions of differential equations. This method with several versions were developed around 1900s by German mathematicians C. Runge and M.W. Kutta.

The essence of the Runge-Katta methods involves the use of numerical integrating the function in differential equations by using a trial step at mid-point of an interval, e.g., within a step Δx or h by using numerical integration techniques such as trapezoidal or Simpson rules as presented in Section10.4. The numerical integrations would allow the cancellation of low-order error terms for more accurate solutions.

As in the preceding sections on numerical methods for engineering analysis, we will present the formulations of the Runge-Kutta methods in this section without proof these formulations mathematically. We will also use commercially available software by the name of MatLAB to sole a selected differential equation with plot of its solution in Case 3 in Appendix 4 on page 478.

10.5.4.1 The Second-order Runge-Kutta Method (p.367)

This is the simplest form of the Runge-Katta method with the formulations for the solution of first order differential equation in the following form:

$$y'(x) = f(x,y) \qquad (10.30)$$

with a specified solution point corresponding to one specific condition for the Equation (10.30). The solution points of this differential equation can be expressed as:

$$y_{i+1} = y_i + k_2 + O\left(h^3\right) \qquad (10.31)$$

where O(h^3) is the order of error of the step h^3, and

$$k_2 = h\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1\right) \qquad (10.32a)$$

where $$k_1 = h\, f\left(x_i, y_i\right) \qquad (10.32b)$$

Example 10.13 (p.368)

Use the second order Runge-Kutta method shown in Equations (10.31) and (10. 32a,b) to solve the following first order ordinary differential equation similar to that in Example 7.4 on p.203:

$$\frac{dy(x)}{dx} + 2\,y(x) = 2 \qquad (a)$$

with a given condition y(0) =2.

We will solve Equation (a) with condition y(0) = 2 in Example 7.4 with an exact solution of $y(x) = 1 + e^{-2x}$.

10.5.4.1 The Second-order Runge-Kutta Method – cont'd

Example 10.13 – Cont'd

Use the second order Runge-Kutta method shown in Equations (10.31) and (10. 32a,b) to solve the following first order ordinary differential equation similar to that in Example 7.4:

$$\frac{dy(x)}{dx} + 2\,y(x) = 2 \qquad \text{(a)}$$

with a given condition y(0) =2.

Solution:

Let us first re-arrange Equation (a) in the following form:

$\frac{dy(x)}{dx} = y'(x) = 2 - 2\,y(x) = f(x,y)$ from which we have: $f(x,y) = 2 - 2y$ (b)

and the specified solution point: $y(0) = y_0 = 2$

We are now ready to determine the first solution point using Equation (10.30) to (10.32a,b):

Step 1: With i = 0 and selected increment h = 0.1:

$$y_1 \approx y_0 + k_2$$

$$k_1 = 0.1(2-2y_0) = 0.1(2-2x2) = -0.2$$

$$k_2 = 0.1\,f\left(x_0+\frac{h}{2}, y_0+\frac{1}{2}k_1\right) = 0.1\left[2-2\left(2+\frac{-0.2}{2}\right)\right] = -0.18$$

We thus have our first solution point:

$y_1 = y_0 + k_2 = 2 - 0.18 = 1.82$ (the exact solution is: $y_1 = 1.8187$)

10.5.4.1 The Second-order Runge-Kutta Method – cont'd

Example 10.13 – Cont'd

Use the second order Runge-Kutta method shown in Equations (10.31) and (10. 32a,b) to solve the following first order ordinary differential equation similar to that in Example 7.4:

$$\frac{dy(x)}{dx} + 2\,y(x) = 2 \qquad \text{(a)}$$

with a given condition y(0) =2.

We may move the solution point forward to: i = 1, h = 0.1 with y_1 = 1.82:

We should have the solution point: $y_2 = y_1 + k_2$ as in Equation (10.31), with:

$$k_1 = h\,f(x_1, y_1) = 0.1(2-2y_1) = 0.1(2-2x1.82) = 0.164$$

$$k_2 = h\,f\left(x_1 + \frac{1}{2}h, y_1 + \frac{1}{2}k_1\right) = 0.1\left[2-2\left(1.82+\frac{1}{2}0.164\right)\right] = -0.1804$$

Hence the second solution point,y_2 is:

$y_2 = y_1 + k_2 = 1.82 – 0.1804 = 1.6396$ (exact solution is: y_2 = 1.67)

We observe that the error of numerical solution accumulates from 0.07% for y_1 to 1.82% for y_2.

10.5.4.2 The 4th – Order Runge-Kutta Method (p.369)

This is the most popular version of the Runge-Kutta method for solving differential equation of initial value problems. Formulation of this order solution method is similar to that of the second order method.

The differential equation is similar to that shown in Equation (10.30) with $y'(x) = f(x,y)$ but with the solution point given by the following formula:

$$y_{i+1} = y_i + \frac{h}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \qquad (10.33)$$

where $k_1 = f\left(x_i, y_i\right)$ (10.34a)

$$k_2 = f\left(x_i + \frac{h}{2},\ y_i + \frac{k_1 h}{2}\right) \qquad (10.34b)$$

$$k_3 = f\left(x_i + \frac{h}{2},\ y_i + \frac{k_2 h}{2}\right) \qquad (10.34c)$$

$$k_4 = f\left(x_i + h,\ y_i + k_3 h\right) \qquad (10.34d)$$

Example 10.14 (p.369)

Use the Runge-Kutta 4th order method to solve the same differential equation in Example 10.13 but only for the second solution point y_2.

Solution:

The differential equation we will solve is: $\frac{dy(x)}{dx} + 2y(x) = 2$ with given condition of $y(0) = 2$.

Example 10.13 already solve the first solution point with $y_1 = 1.82$ with a chosen step size of h = 0.1. We are required to find the next solution point at y_2 with i = 1 using the same step size of h = 0.1.

10.5.4.2 The 4th – Order Runge-Kutta Method-Cont'd

Example 10.14 – cont'd

We need to translate the differential eqation $\frac{dy(x)}{dx} + 2y(x) = 2$ into the form of the differential equation using the Runge-Kutta solution method in the form of y'(x) = f(x,y) as shown in Equation (10.30) with: f(x,y) = 2-2y (a)

and y_0 = 2

Example 10.13 already solve the first solution point with y_1 = 1.82 with a chosen step size of h = 0.1. We are required to find the next solution point at y_2 with i = 1 using the same step size of h = 0.1.

We may use the expression in Equation (10.33) to obtain solution point 2 for the differential equation in Example 10.13 to be:

$$y_2 = y_1 + \frac{0.1}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right) \qquad \text{(b)}$$

where k_1, k_2, k_3 and k_4 can be obtained by using the expressions in Equations (10.34a, b, c and d) respectively:

$$k_1 = (x_1, y_1) = 2-2y_1 = 2-2x1.82 = -1.64 \qquad \text{(c)}$$

$$k_2 = f\left(x_1 + \frac{h}{2}, y_1 + \frac{k_1 h}{2}\right) = 2 - 2\left[1.82 + \frac{(-1.64)\times 0.1}{2}\right] = -1.476 \qquad \text{(d)}$$

$$k_3 = f\left(x_1 + \frac{h}{2}, y_1 + \frac{k_2 h}{2}\right) = 2\text{-}2\left[1.82 + \frac{(-1.476)\times 0.1}{2}\right] = -1.4924 \qquad \text{(e)}$$

$$k_4 = f\left(x_1 + \frac{h}{2}, y_1 + k_3 h\right) = 2 - 2(1.82 - 1.4924 \times 0.1) = -1.34152 \qquad \text{(f)}$$

Substituting the constants k_1, k_2, k_3 and k_4 in Equations (c,d,e,f) into Equation (b), resulting in:

$$y_2 = 1.82 + \frac{0.1}{6}[-1.64 + 2(-1.476) + 2(-1.4924) + (-1.34152)] = 1.6714$$

We find the above solution y_2 using the 4th order Runge-Katta method is remarkably close to the exact solution of y_2 = 1.67. This is a much more accurate result than what can be obtained by using the 2nd order Runge-Kutta method as illustrated in Example 10.13.

10.5.4.3 Runge-Kutta Method for Higher Order Differential Equations (p.370)

- We have learned that Runge-Kutta method can solve differential equations often with remarkable accuracy as demonstrated in Example 10.14.
- Unfortunately most textbooks only offer the application of this valuable method for solving first order differential equations.
- Its application to solve higher order differential equations requires the conversion of higher order differential equations to the first order-equivalent forms such as in Equation (10.30) on p. 367.
- The solution of the converted higher order differential equations can be obtained by using the expressions such as given in Equation (10.33) for the 4th order Runge-Kutta formulation.
- We will present the following formulation to illustrate how the fourth order Runge-Kutta method can be used to solve second order ordinary differential equations

We will first express the given second order ordinary differential equation in the form:

$$\frac{d^2y(x)}{dx^2} = f\left(x, y, \frac{dy(x)}{dx}\right) = f(x,y,y') \qquad (10.35)$$

Since the left-hand-side of Equation (10.35) may be expressed as: $\frac{dy'(x)}{dx}$, we may convert the second order differential equation in (10.35) into a first order differential equation in the form:

$$\frac{dy'(x)}{dx} = f[x,y,y'(x)] \qquad (10.36a)$$

with

$$\frac{dy(x)}{dx} = y'(x) = F[x,y,y'(x)] \qquad (10.36b)$$

10.5.4.3 Runge-Kutta Method for Higher Order Differential Equations – Cont'd

We will present the converted 2nd order differential in Equation (10.35) in Equations (10.36a) and (10.36b) below:

$$\frac{dy'(x)}{dx} = f[x, y, y'(x)] \qquad (10.36a)$$

with

$$\frac{dy(x)}{dx} = y'(x) = F[x, y, y'(x)] \qquad (10.36b)$$

Solution points yi+1 of these second order differential equations may be obtained by the following expressions derived from the 4th order Runge-Kutta method:

$$y_{i+1} = y_i + \frac{1}{6}(F_1 + 2F_2 + 2F_3 + F_4)h \qquad (10.37)$$

and its derivative:

$$y'_{i+1} = y_i + \frac{1}{6}(f_1 + 2f_2 + 2f_3 + f_4)h \qquad (10.38)$$

where h = the increment of the solution points in the x-coordinate.

The coefficients F_1, F_2, F_3 and F_4, and f_1, f_2, f_3 and f_4 in Equations (10.37) and (10.38) can be obtained by the expressions given in the following table (Table 10.4) on p.371:

$f_1 = f(x_i, y_i, y'_i)$	$F_1 = y'_i$
$f_2 = f\left(x_i + \frac{1}{2}h,\ y_i + \frac{1}{2}F_1h,\ y'_i + \frac{1}{2}f_1h\right)$	$F_2 = y'_i + \frac{1}{2}f_1h$
$f_3 = f\left(x_i + \frac{1}{2}h,\ y_i + \frac{1}{2}F_2h + \frac{1}{4}f_1h^2,\ y'_i + \frac{1}{2}f_2h\right)$	$F_3 = y'_i + \frac{1}{2}f_2h$
$f_4 = f\left(x_i + h,\ y_i + F_3h + \frac{1}{2}f_2h,\ y'_i + f_3h\right)$	$F_4 = y'_i + f_3h$

10.5.4.3 Runge-Kutta Method for Higher Order Differential Equations – Cont’d

Example 10.15 (p.371)

Use the 4th order Runge-Kutta method to solve the following second order differential equation

$$\frac{d^2y(x)}{dx^2} - 2\frac{dy(x)}{dx} + y(x) = x^2 - 4x + 2 \qquad \text{(a)}$$

with given conditions:

$$y(x)\big|_{x=0} = y(0) = y_0 = 0 \qquad \text{(b1)}$$

$$\left.\frac{dy(x)}{dx}\right|_{x=0} = y'(0) = y_0' = 0 \qquad \text{(b2)}$$

Solution:

By comparing Equation (a) with Equation (10.35), we will obtain the following expression for the function f(x,y,y’) in Equation (10.35):

$$f(x, y, y') = (x^2 - 4x + 2) – y + 2y' \qquad \text{(c)}$$

with the specified conditions of $y_0 = 0$ and y’(0) = 0 for our subsequent numerical solution of the differential equation.

We will select 3 different step sizes: h = 0.2, 0.6 and 1.2 for the three case illustrations using by following a procedure, starting with variable at x = 0.

10.5.4.3 Runge-Kutta Method for Higher Order Differential Equations – Cont’d

Example 10.15 - Cont’d

We begin the numerical solution for Equation (a) by letting i = 0 in Equation (10.37) with $x_0 = y_0 = y'_0 = 0$, and $f(x,y,y')=(x^2-4x+2)-y+2y'$ for the first solution point at x = h = 0.2:

We will obtain the following coefficients by using the expressions for the coefficients given in Table 10.4:

$$F_1 = y_0' = 0 - (a\ given\ condition)$$

$$f_1 = f\left(x_o, y_{0,}\, y_0'\right) = x_0^2 - 4x_0 + 2 - y_0 + 2y_0' = 0-0+2-0+2x0 = 2$$

$$F_2 = y_0' + \frac{1}{2} f_1 h = 0+\frac{1}{2}2x0.2 = 0.2$$

$$f_2 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{1}{2}F_1 h, y_0' + \frac{1}{2} f_1 h\right) = \left(x_0 + \frac{0.2}{2}\right)^2 - 4\left(x_0 + \frac{0.2}{2}\right) + 2$$

$$-\left(y_0 + \frac{1}{2} \times 0 \times 0 \times 0.2\right) + 2\left(y_0' + \frac{1}{2} \times 2 \times 0.2\right) = 2.01$$

$$F_3 = y_0' + \frac{1}{2} f_2 h = 0+\frac{1}{2}(2.01)x0.2 = 0.201$$

$$f_3 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{1}{2}F_2 h, y_0' + \frac{1}{4} f_1 h^2 + \frac{1}{2} f_2 h\right) = \left(x_0 + \frac{0.2}{2}\right)^2 - 4\left(x_0 + \frac{0.2}{2}\right) + 2$$

$$-\left(y_0 + \frac{1}{2} \times 0.2 \times 0.2\right) + 2\left(y_0' + \frac{1}{4}(2)(0.2)^2 + \frac{1}{2} \times 2.01 \times 0.2\right) = 2.012$$

10.5.4.3 Runge-Kutta Method for Higher Order Differential Equations – Cont'd

Example 10.15 - Cont'd

We begin the numerical solution for Equation (a) by letting i = 0 in Equation (10.37) with $x_0 = y_0 = y'_0 = 0$, and $f(x,y,y') = (x^2-4x+2)-y+2y'$ for the first solution point at $x = h = 0.2$:

$$F_4 = y_0' + f_3\text{h} = 0+2.012\times 0.2 = 0.4024$$

$$f_4 = f\left(x_0 + h, y_0 + F_3 h + \frac{1}{2} f_2 h, y_0' + f_3 h\right) = [(0+0.2)^2 - 4(0+0.2) + 2] - \left(0 + 0.201 \times 0.2 + \frac{1}{2} \times 2.01 \times 0.2\right) + 2(0 + 2.012 \times 0.2) = 2.2056$$

We are ready to find the numerical solution of the differential equation in Equation (a) by substituting the values of F_1, F_2, F_3 and F_4 into Equation (10.37), and obtain a solution point y_1 with i = 0 and h = 0.2:

$$y_1 = y(0.2) = y_0 + \frac{1}{6}(F_1 + 2F_2 + 2F_3 + F_4)\text{h} = 0 + \frac{1}{6}(0 + 2\times 0.2 + 2\times 0.201 + 0.4024)0.2 = 0.0401467 \qquad \text{(d)}$$

The exact solution of Equation is $y(x) = x^2$, which yields an exact solution of $y(0.2) = 0.04$. The numerical solution in Equation (d) has a .42% error from the exact solution.

We will also use Equation (10.38) to approximate the value of the first order derivation y'(0.2) as:

$$y_1' = y'(0.2) = y_0 + \frac{1}{6}(f_1 + 2f_2 + 2f_3 + f_4)\text{h} = 0 + \frac{1}{6}(2 + 2\times 2.01 + 2\times 2.012 + 2.2056)0.2 = 0.40798$$

10.5.4.3 Runge-Kutta Method for Higher Order Differential Equations – Cont'd

Example 10.15 - Cont'd

We have thus obtained the numerical solution for Equation (a) as computed in the previous slide, in which we let i = 0 in Equation (10.37) with $x_0 = y_0 = y'_0 = 0$, and $f(x,y,y') = (x^2-4x+2)-y+2y'$ for the first solution point at x = h = 0.2.

One may follow the same procedure to obtain the solution of Equation (a) at point: x = x + h with h =0.1, 0.4,0.6 and 1.2 to see the difference in the solutions, and also the % of errors in these solutions with different h-values. Table 10.5 shows these solutions and the % errors in the solutions using different h-values.

Like all other numerical solution methods for solving differential equations, the accuracy, or error of the approximated solutions depends largely on the step size h chosen by the users.

We have demonstrated the effects of the chosen increment size h for the same differential equation in Equation (a) but with other two h-values with h = 0.6 and 1.2 in two separate cases. The results of these two cases, together with the cases of h = 0.2, 0.1 and 0.4 are summarized in the same Table 10.5.

Table 10.5 Enhanced Solutions of a Differential Equation by Runge-Kutta Method with 5 incremental sizes

Case Number:	1[2]	2[1,2]	3[2]	4[1]	5[1]
X_0	0	0	0	0	0
h	0.2	0.1	0.4	0.6	1.2
F_1	0	0	0	0	0
F_2	0.2	0.1	0.4	0.6	1.2
F_3	0.201	0.100125	0.408	0.627	1.416
F_4	0.3984	1.999537	0.7904	1.1784	2.4864
f_1	2	2	2	2	2
f_2	2.01	2.0025	2.04	2.09	2.36
f_3	1.992	1.99775	1.976	1.964	2.072
f_4	1.8078	1.999537	1.9446	1.9406	1.936
y'(x)	0.3937	0.200052	0.79844	1.20486	2.56
Approx. $y(x_o+h)$	0.040013	0.010000	0.16042	0.36324	1.5436
Exact $y(x_o+h)$	0.04	0.01	0.16	0.36	1.44
%Error: y(h)	0.033	0.0042	0.2667	0.9	7.2
%Error: y'(h)	0.005	0.000313	0.195		

NOTE: [1] Case runs with solutions by MatLAB in Case 1 in Appendix 4 of the book (p. 478).
[2] Cases presented in Table 10.5 on p.373

Runge-Kutta Method for Higher Order Differential Equations – Cont’d

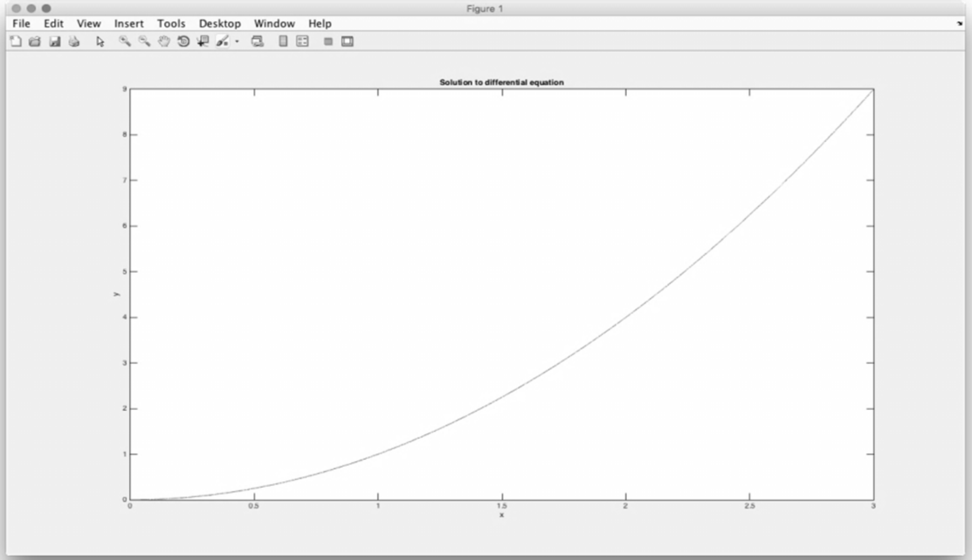
Solving a Higher Order Differential Equation using Runge-Kutta method and online MatLAB software

The solution with incremental size such as h=0.2 obtained by manual operations using Runge-Kutta method in Example 10.15 appeared tedious and time-consuming.

The same differential equation with pecified conditions was solved by a commercial software package MatLAB with the input/output information included in Case 3 of Appendix 4 (P.478).

The results obtained by using this software were remarkably accurate with solution at the same three points, i.e.: at x=0.2, x=0.6 and x = 0=1.2 with h=0.2 are plotted in the graphic output as shown in Figure 10.17 by the MatLAB software with I/O indicated in Case 3 of Appendix 4 (p. 483).

Figure 10.17 Graphic Output of the Solution of a 2nd Order Differential Equation Using MatLAB Sotware Package



Example 10.16 (p.373)

Use the 4th order Runge-Kutta method to solve the last solution of the function x(t) at t = 0.2 in the second order differential equation in Example 10.12 (p.364), in which the same 2nd order differential equation was solved by using a forward difference scheme. This example demonstrates the fact that Runge-Kutta method can be related to the finite difference method of solving differential equations.

Solution:

The differential equation to be solved is:

$$\frac{d^2x(t)}{dt^2}+x(t)=0 \qquad \text{(a)}$$

with specified conditions:

x(0) = 1

and $\left.\frac{dx(t)}{dt}\right|_{t=0} = 0$

Since we do not have the term $x'(t)$ in Equation (a), we will not need to evaluate F_1, F_2, F_3 and F_4 in Table 10.5. We will use the same step size h = 0.05 as was in Example 10.12.

By following the procedures adopted in the Example 10.15, we express Equation (a) in the folloiwng form:

$$\frac{d^2x(t)}{dt^2}= -\ x(t) = 0\text{-}x(t)+0=f(t,x,x')$$

where $$\frac{dx(t)}{dt}=x'(t)$$

and x_0 = 0.9925 (the solution obtained in Example (10.12) and $x_0^{'}=-0.14944$

10.5.4.3 Runge-Kutta Method for Higher Order Differential Equations – Cont'd

Example 10.16 (p.373)-Cont'd

We will use the 4th order Runge-Katta method as shown in Equations (10.33) and (10.38) and the coefficients given in Table 10.4 (p.371) to obtain the solution at x(0.2) as follows. Let i=0, h=0.05, x_0=0.9925 and x_0' = -0.14944 for the solution at t =0.2, or

$$x_1 = x(0.2) = x_0 + x_0' h + \frac{1}{6}(f_1 + f_2 + f_3)h^2 \qquad \text{(b)}$$

We will evaluate the coefficients f_1, f_2, f_3 and f_4 from Table 10.4 as follows:

$$f_1 = f\left(t_0, x_{0,}, x_0'\right) = x_0 = -0.14944$$

$$f_2 = x_0' + \frac{1}{2} f_1 h = -0.14944 + \frac{(-0.14944)x0.05}{2} = -0.153176$$

$$f_3 = x_0' + \frac{1}{2} f_2 h = -0.14944 + \frac{(-0.153176)x0.05}{2} = -0.153694$$

$$f_4 = x_0' + f_3 h = -0.14944 - 0.153694\,x0.05 = -0.15710347$$

According to Equation (10.38), we have the solution

$$x_1 = x(0.2) = x_0 + x_0' h + \frac{1}{6}(f_1 + f_2 + f_3)h^2$$

$$= 0.9925 + (-0.14944)x0.05 + \frac{1}{6}(-0.14944 - 0.153176 - 0.153694)(0.05)^2 = 0.9848$$

This numerical solution has an error of 0.48% from the exact solution, and it is more accurate than that obtained from the simple forward difference scheme in Example 10.12.

10.6 Introduction to Commercial Software Packages for Numerical Analysis (p.375)

We have demonstrated in this chapter that accurate solutions of almost all numerical solution methods require small increment step sizes the variable of the function for solutions. Smaller increments means more computational efforts and advanced digital computers often are the necessary tools to achieve accurate solutions..

Consequently, sophisticated computer packages such as the two popular commercially available packages with trade names of “Mathematica” and “MatLAB” have proven to be valuable tools for engineers in their engineering analyses.

In this section, we will briefly highlight these two numerical analysis packages, in particular, their capabilities in solving various engineering problems. Readers are referred to several excellent references cited in the book for more detailed descriptions of these packages, as well as effective use of a well documented commercial software package named MatLAB in Appendix 4 of the book.

10.6.1 Introduction to Mathematica (p.375)

Mathematica is a computational software program based on symbolic mathematics. It is used in many scientific, engineering, mathematical and computing fields. The programming languages used in Mathematica is the Wolfram Language by Stephen Wolfram and C, C++ and Java. This software package has been in the marketplace since June 1988. Following are noticeable capabilities for handling engineering analysis:

1) Determine roots of polynomial equations of cubic or higher orders.
2) Integrate and differentiate complicated expressions.
3) Solve linear and nonlinear differential equations
4) Elementary and special mathematical function libraries
5) Matrix and data manipulation tools
6) Numeric and symbolic tools for discrete and continuous calculus

It can also solve the following common analytical engineering problems involving:

1) The determination of Laplace and Fourier transforms of functions.
2) Generating graphics in two- and three-dimensions.
3) With simplify trigonometric and algebraic expressions.

10.6.1 Introduction to Mathematica (p.375)-Cont'd

Mathematica also has the following features that are of great values in advanced engineering analyses:

- Support for complex numbers, arbitrary precision, interval arithmetic and symbolic computation
- Solvers for systems of equations, Diophantine equations, ODEs, PDEs, etc.
- Multivariate statistics libraries including fitting, hypothesis testing, and probability and expectation calculations on over 140 distributions.
- Calculations and simulations on random processes and queues
- Computational geometry in 2D, 3D and higher dimensions
- Finite element analysis including 2D and 3D adaptive mesh generation
- Constrained and unconstrained local and global optimization
- Toolkit for adding user interfaces to calculations and applications
- Tools for 2D and 3D image processing and morphological processing, including image recognition
- Tools for visualizing and analyzing directed and undirected graphs
- Tools for combinatorics problems
- Data mining tools such as cluster analysis, sequence alignment and pattern matching

10.6.1 Introduction to Mathematica (p.375)-Cont'd

- Group theory and symbolic tensor functions
- Libraries for signal processing including wavelet analysis on sounds, images and data
- Linear and non-linear control systems libraries
- Continuous and discrete integral transforms
- Import and export filters for data, images, video, sound,CAD, GIS,[1] document and biomedical formats
- Database collection for mathematical, scientific, and socio-economic information and access to Wolfram alpha data and computations
- Technical word processing including *formula* editing and automated report generating
- Tools for connecting to DLL, SOL, Java, NET, C++, Fortran, CUDA, OpenCL and http based systems
- Tools for parallel programming
- Mathematica language in notebook when connected to the Internet

The last of the above itemized features is of particular value to engineers. For example, we were required to find the root of the following cubical equation in Example 10.3: $L^3 + 70.3L^2 + 1647.39L - 18656.72 = 0$

A meaningful root of this equation found by using the Newton/Raphson method was L = 8.15 as shown in Example 10.3. A similar solution of L = 8.1566 were obtained by the solution method offered at an Internet at the website Wolfram/Alpha Widgets (www.wolframalpha.com/widgets/view.jsp?id) with user's input of the coefficients of this equation. It offered an instant solution and with an excellent user interface feature.

10.6.2 Introduction to MatLAB (p.376)

MATLAB is an acronym of "**mat**rix **lab**oratory." This numerical analysis package was designed by Cleve Molar in the late 1970s with an initial release to the public in 1984. The latest version, Version 8.6 was released in September 2015.

MATLAB provides a multi-paradigm numerical omputing environment and 4th generation programming language, a proprietary programming language developed by MathWorks. It allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces that include interfacing with programs written in other languages, including ,C++, Java, Fortran and Python. It is a popular numerical analysis package mainly because of it has graphic and graphical user interfacing programming capability.

Like Mathematica, MATLAB is capable of handling the following common problems in engineering analysis [Malek-Madani 1998]:

1) Find roots of polynomials, sum series, and determine limits of sequences,
2) Symbolically integrate and differentiate complicated expressions,
3) Plot graphics in two and three dimensions,
4) Simplify trigonometric and algebraic expressions,
5) Solve linear and nonlinear differential equations, and
6) Determine the Laplace transform of functions,
7) Plus a variety of other mathematical operations.

Operation of MatLAB requires user to input simple program for the solution of the problems. These program usually consists of three "commands:" **1)** the "command window" for the user to enter commands and data, **2**) the "graphics window" to display the results in plots and graphs, and **3)** the "edit window," to create and edit the M-files, which provide alternative ways of performing operations that could expand MATLAB's problem-solving capabilities

Detailed instructions on using MATLAB for solving a variety of mathematical problems are available in MATLAB Primer published by the Mathworks, Inc.(www.mathworks.com) and two excellent references [Malek-Madani 1998, Chapra 2012].

10.6.2 Introduction to MatLAB (p.376)-Cont'd

Appendix 4 of this book will present the input/output (I/O) procedures for three cases of engineering analysis using the MatLAB software package. These cases include:

Case 1: Graphic solution of the amplitudes with "beats" offered by the solution in Equation (8.40) for the near-resonant vibration of a metal stamping machine,

Case 2: The numerical solution with graphic expressions of the amplitudes and mode shapes of a thin flexible rectangular pad subjected to transverse vibration as presented in Equation (9.76).

Case 3: The solution with graphic expression of a non-homogeneous second order differential equation.

These cases will usher the readers to use this valuable commercially available numerical analysis software package MatLAB in solving complicated engineering analysis problems.