CHAPTER 11

Numerical Differentiation

11.1 NEED AND SCOPE

Need for differentiation of a function arises quite often in engineering and scientific problems. If the function has a closed form representation in terms of standard calculus, then its derivatives can be found exactly. However, in many situations, we may not know the exact function. What we know is only the values of the function at a discrete set of points. For instance, we are given the distance travelled by a moving object at some regular time intervals and asked to determine its velocity at a particular time. In some other instances, the function is known but it is so complicated that an analytic differentiation is difficult (if not impossible). In both these situations, we seek the help of numerical techniques to obtain the estimates of function derivatives. The method of obtaining the derivative of a function using a numerical technique is known as *numerical differentiation*. There are essentially two situations where numerical differentiation is required. They are :

1. The function values are known but the function is unknown. Such functions are called *tabulated function*.
2. The function to be differentiated is complicated and, therefore, it is difficult to differentiate.

In this chapter, we discuss various numerical differentiation methods that could be applied to both tabulated and continuous functions.

Remember that while analytical methods give exact answers, the numerical techniques provide only approximations to derivatives. Numerical differentiation methods are very sensitive to roundoff errors, in

addition to the truncation error introduced by the methods themselves. Therefore, we also discuss the errors and ways to minimise them.

11.2 DIFFERENTIATING CONTINUOUS FUNCTIONS

We discuss here the numerical process of approximating the derivative $f'(x)$ of a function $f(x)$, when the function itself is available.

Forward Difference Quotient

Consider a small increment $\Delta x = h$ in x. According to Taylor's theorem, we have

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2} f''(\theta) \tag{11.1}$$

for $x \le \theta \le x + h$. By rearranging the terms, we get

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2} f''(\theta) \tag{11.2}$$

Thus, if h is chosen to be sufficiently small, $f'(x)$ can be approximated by

$$\boxed{f'(x) = \frac{f(x+h) - f(x)}{h}} \tag{11.3}$$

with a truncation error of

$$\boxed{E_t(h) = -\frac{h}{2} f''(\theta)} \tag{11.4}$$

Equation (11.3) is called the first order *forward difference quotient*. This is also known as *two-point formula*. The truncation error is in the order of h and can be decreased by decreasing h.

Similarly, we can show that the first-order *backward difference quotient* is

$$\boxed{f'(x) = \frac{f(x) - f(x-h)}{h}} \tag{11.5}$$

Example 11.1

Estimate approximate derivative of $f(x) = x^2$ at $x = 1$, for $h = 0.2$, 0.1, 0.05 and 0.01 using the first-order forward difference formula.

$$f'(x) = \frac{f(x+h) - f(x)}{h}$$

Therefore,

$$f'(1) = \frac{f(1+h) - f(1)}{h}$$

Derivative approximations are tabulated below:

h	$f'(1)$	*Error*
0.2	2.2	0.2
0.1	2.1	0.1
0.05	2.05	0.05
0.01	2.01	0.01

Note that the correct answer is 2. The derivative approximation approaches the exact value as h decreases. The truncation error decreases proportionally with decrease in h. There is no roundoff error.

Central Difference Quotient

Note that Eq. (11.3) was obtained using the linear approximation to $f(x)$. This would give large truncation errors if the functions were of higher order. In such cases, we can reduce truncation errors for a given h by using a quadratic approximation, rather than a linear one. This can be achieved by taking another term in Taylor's expansion, i.e.,

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(\theta_1) \tag{11.6}$$

Similarly,

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2!}f''(x) - \frac{h^3}{3!}f'''(\theta_1) \tag{11.7}$$

Subtracting Eq. (11.7) from Eq. (11.6), we obtain

$$f(x+h) - f(x-h) = 2hf'(x) + \frac{h^3}{3!}[f'''(\theta_1) + f'''(\theta_2)] \tag{11.8}$$

Thus, we have

$$\boxed{f'(x) = \frac{f(x+h) - f(x-h)}{2h}} \tag{11.9}$$

with the truncation error of

$$E_t(h) = -\frac{h^2}{12}[f'''(\theta_1) + f'''(\theta_2)] = -\frac{h^2}{6}f'''(\theta)$$

which is of order h^2. Equation (11.9) is called the second-order *central difference quotient*. Note that this is the average of the forward difference quotient and the backward difference quotient. This is also known as *three-point formula*. The distinction between the two-point and three-point formulae is illustrated in Fig. 11.1(a) and Fig. 11.1(b). Note that the approximation is better in the case of three-point formula.

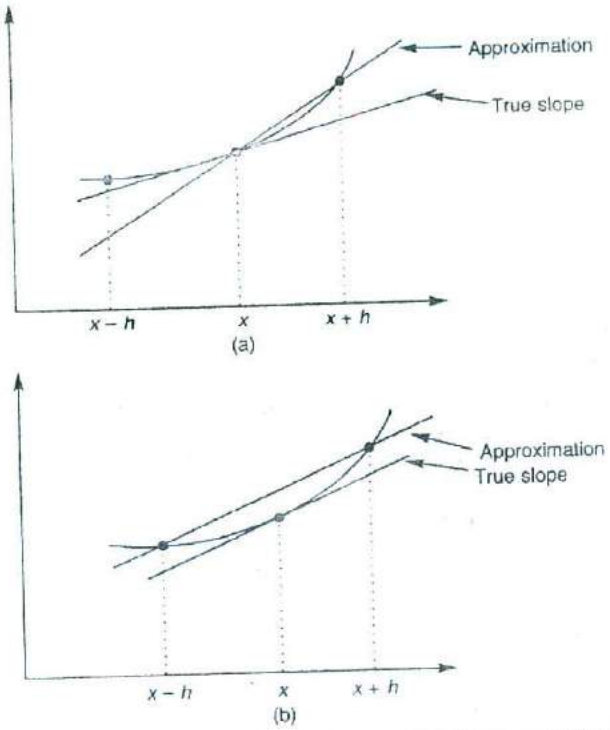


Fig. 11.1 Illustration of (a) Two-point formula and (b) Three-point formula

Example 11.2

Repeat the exercise given in Example 11.1 for the three-point formula.

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$$

Therefore,

$$f'(1) = \frac{f(1+h) - f(1-h)}{2h}$$

The derivative approximations are tabulated below:

h	$f'(1)$	*Error*
0.2	2.0	0
0.1	2.0	0
0.05	2.0	0

The derivative is exact for all values of h. This is because we have used quadratic approximation for a quadratic function. We can also derive further higher-order derivatives by using more points in the formula. For example, the *five-point central difference formula* is given by

$$f'(x) = \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h} \tag{11.10}$$

This is a fourth-order approximation and the truncation error is of order h^4. In this case, the truncation error will approach zero much faster compared to the three-point approximation. The derivation of Eq. (11.10) is left to the reader as an exercise. (*Hint: use step size 2h instead of h in Eq. (11.8) and use up to fifth derivative of Taylor's expansion*).

Error Analysis

As mentioned earlier, numerical differentiation is very sensitive to roundoff errors. If $E_r(h)$ is the roundoff error introduced in an approximate derivative, then the total error is given by

$$E(h) = E_r(h) + E_t(h)$$

Let us consider the two-point formula for the purpose of analysis. That is,

$$f'(x) = \frac{f(x+h) - f(x)}{h} = \frac{f_1 - f_0}{h}$$

If we assume the roundoff errors in f_1 and f_0 as e_1 and e_0, respectively, then

$$f'(x) = \frac{(f_1 + e_1) - (f_0 + e_0)}{h}$$

$$= \frac{f_1 - f_0}{h} + \frac{e_1 - e_0}{h}$$

If the errors e_1 and e_0 are of the magnitude e and of opposite sign (i.e., the worst case) then we get the bound for roundoff error as

$$|E_r(h)| \leq \frac{2e}{h}$$

We know that the truncation error for two-point formula is

$$|E_t(h)| = -\frac{h}{2} f''(\theta)$$

or

$$|E_t(h)| \leq \frac{M_2 h}{2}$$

where M_2 is the bound given by

$$M_2 = \max_{x \leq \theta \leq x+h} |f''(\theta)|$$

Thus, the bound for total error in the derivative is

$$|E(h)| \leq \frac{M_2 h}{2} + \frac{2e}{h} \tag{11.11}$$

Note that when the step size h is increased, the truncation error increases while the roundoff error decreases. This is illustrated in Fig. 11.2. For small values of h, roundoff error has an overriding influence on the total error. Therefore, while reducing the step size, we should exercise proper judgement in choosing the size. This argument applies to all the formulae discussed here.

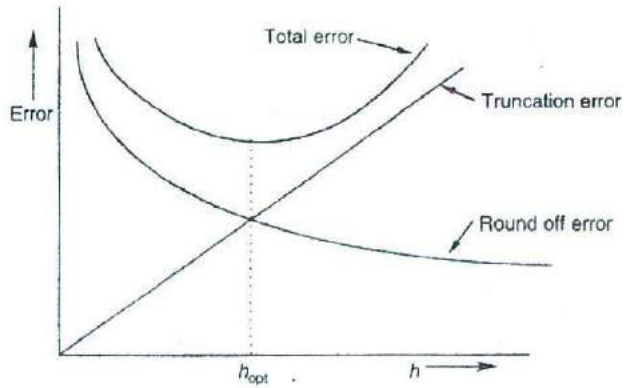


Fig. 11.2 Error in derivatives as a function of h

We can obtain a rough estimate of h that gives the minimum error. By differentiating Eq. (11.11) with respect to h, we obtain

$$E'(h) = \frac{M_2}{2} - \frac{2e}{h^2}$$

We know that $E(h)$ is minimum when $E'(h) = 0$. That is,

$$\frac{M_2}{2} - \frac{2e}{h^2} = 0$$

Solving for h, we obtain

$$h_{opt} = 2\sqrt{\frac{e}{M_2}} \tag{11.12}$$

Substituting this in Eq. (11.11), we get

$$E(h_{opt}) = 2\sqrt{eM_2} \tag{11.13}$$

Example 11.3

Compute the approximate derivatives of $f(x) = \sin x$, at $x = 0.45$ radians, at increasing values of h from 0.01 to 0.04, with a step size of 0.005. Analyse the total error. What is the optimum step size?

$$f(x) = \sin x$$

Using two-point formula

$$f'(x) = \frac{f(x+h) - f(x)}{h}$$

Given

$$x = 0.45 \text{ radians}$$

So, $f(x) = \sin(0.45) = 0.4350$ (rounded to four digits). Exact $f'(x) = \cos x = \cos(0.45) = 0.9004$.

Table below gives the approximate derivatives of $\sin x$ at $x = 0.45$ using various values of h.

h	$f(x+h)$	$f'(x)$	*Error*
0.010	0.4439	0.8900	0.0104
0.015	0.4484	0.8933	0.0071
0.020	0.4529	0.8950	0.0054
0.025	0.4573	0.8935	0.0069
0.030	0.4618	0.8933	0.0071
0.035	0.4662	0.8914	0.0090
0.040	0.4706	0.8900	0.0104

The table shows that the total error decreases from 0.0104 (at $h = 0.01$) till $h = 0.02$ and again increases when h is increased as illustrated in Fig. 11.2.

Since we have used four significant digits, the bound for roundoff error e is 0.5×10^{-4}. For the two-point formula, the bound M_2 is given by

$$M_2 = \max |f''(\theta)|$$

$$0.45 \le \theta \le 0.49$$

$$= |\sin(0.49)| = 0.4706$$

Therefore, the optimum step size is

$$h_{opt} = 2\sqrt{\frac{e}{M_2}} = 2\sqrt{\frac{0.5 \times 10^{-4}}{0.4706}}$$

$$= 0.0206$$

This agrees very closely with our results.

Higher-order Derivatives

We can also obtain approximations to higher-order derivatives using Taylor's expansion. To illustrate this, we derive here the formula for $f''(x)$. We know that

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + R_1$$

and

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2!}f''(x) - \frac{h^3}{3!}f'''(x) + R_2$$

Adding these two expansions gives

$$f(x+h) + f(x-h) = 2f(x) + h^2 f''(x) + R_1 + R_2$$

Therefore

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} - \frac{(R_1 + R_2)}{h^2}$$

Thus, the approximation to second derivative is

$$\boxed{f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}} \tag{11.14}$$

The truncation error is

$$E_t(h) = -\frac{R_1 + R_2}{h^2}$$

$$= -\frac{1}{h^2}\frac{h^4}{4!}(f^{(4)}(\theta_1) + f^{(4)}(\theta_2))$$

$$= -\frac{h^2}{12}f^{(4)}(\theta)$$

The error is of order h^2.

Similarly, we can obtain other higher-order derivatives with the errors of order h^3 and h^4.

Example 11.4

Find approximation to second derivative of cos (x) at $x = 0.75$ with $h = 0.01$. *Compare with the* true value.

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$$

$$f''(0.75) = \frac{f(0.76) - 2f(0.75) + f(0.74)}{0.0001} \quad (\text{at } h = 0.01)$$

$$= \frac{0.7248360 - 2(0.7316888) + 0.7384685}{0.0001}$$

$$= \frac{1.4633046 - 1.4633776}{0.0001}$$

$$= -0.7300000$$

Exact value of $f''(0.75) = -\cos(0.75)$

$$= -0.7316888$$

$$\text{Error} = -0.0016888$$

This error includes roundoff error as well.

11.3 DIFFERENTIATING TABULATED FUNCTIONS

Suppose that we are given a set of data points (x_i, f_i), $i = 0, 1, \ldots n$ which correspond to the values of an unknown function $f(x)$ and we wish to estimate the derivatives at these points. Assume that the points are equally spaced with a step size of h.

When function values are available in tabulated form, we may approximate this function by an interpolation polynomial $p(x)$ discussed in Chapter 9 and then differentiate $p(x)$. We will use here Newton's divided difference interpolation polynomial.

Let us first consider the linear equation

$$p_1(x) = a_0 + a_1 (x - x_0) + R_1$$

where R_1 is the remainder term used for estimation. Upon differentiation of this formula, we obtain

$$p_1'(x) = a_1 + \frac{dR_1}{dx}$$

Then the approximate derivative of the function $f(x)$ is given by

$$f'(x) = p_1'(x) = a_1$$

We know that

$$a_1 = f[x_0, x_1]$$

$$= \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

On substituting

$$h = x_1 - x_0$$

$$x_1 = x + h$$

$$x_0 = x$$

we get

$$\boxed{f'(x) = \frac{f(x+h) - f(x)}{h}} \qquad (11.15)$$

This is the familiar two-point forward difference formula.

Now, let us consider the quadratic approximation. Here, we need to use three points. Thus,

$$p_2(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + R_2$$

Then

$$p_2'(x) = a_1 + a_2[(x - x_0) + (x - x_1)] + \frac{dR_2}{dx}$$

Thus, we obtain

$$f'(x) = a_1 + a_2[(x - x_0) + (x - x_1)] \tag{11.16}$$

Let $x_0 = x$, $x_1 = x + h$, $x_2 = x + 2h$, Then

$$a_1 = \frac{f(x+h) - f(x)}{h}$$

$$a_2 = f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

$$= \frac{\dfrac{f(x_2) - f(x_1)}{x_2 - x_1} - \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0}$$

$$= \frac{f(x+2h) - 2f(x+h) + f(x)}{2h^2}$$

Substituting for a_1 and a_2 in Eq. (11.16) and after simplification, we get

$$\boxed{f'(x) = \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h}} \tag{11.17}$$

This is a three-point forward difference formula. We can obtain a three-point backward difference formula by replacing h by $-h$ in Eq. (11.17). Therefore, the three-point backward difference formula is given by

$$\boxed{f'(x) = \frac{3f(x) - 4f(x-h) - f(x-2h)}{2h}} \tag{11.18}$$

Similarly, we can obtain the three-point central difference formula by letting $x_0 = x$, $x_1 = x - h$, $x_2 = x + h$ in Eq. (11.16). Thus,

$$a_1 = \frac{f(x) - f(x-h)}{h}$$

$$a_2 = \frac{f(x+h) - 2f(x) + f(x-h)}{2h^2}$$

Substituting these values in Eq. (11.16) we get

$$\boxed{f'(x) = \frac{f(x+h) - f(x-h)}{2h}} \tag{11.19}$$

Error Analysis

Let us first take the linear case.

$$R_1 = f[x_0, x_1, x]\,(x - x_0)\,(x - x_1)$$

where

$$f[x_0, x_1, x] = \frac{f''(\theta)}{2!}$$

for some point q in the interval containing x_0, x_1 and x. Then

$$\frac{dR_1}{dx} = \frac{f''(\theta)}{2}\,[(x - x_0) + (x - x_1)]$$

Letting $x_0 = x$ and $x_1 = x + h$

$$\frac{dR_1}{dx} = -\frac{h}{2} f''(\theta), \qquad x \le \theta \le x + h$$

Therefore, the truncation error is of order h. This conforms with Eq. (11.4). Now, let us consider the quadratic approximation.

$$R_2 = f[x_0, x_1, x_2, x]\,(x - x_0)\,(x - x_1)\,(x - x_2)$$

$$f[x_0, x_1, x_2, x] = \frac{f'''(\theta)}{3!}$$

for some point q in the interval containing x_0, x_1, x_2 and x.

$$\frac{dR_2}{dx} = \frac{f'''(\theta)}{3!}\,[(x - x_0)\,(x - x_1) + (x - x_1)\,(x - x_2) + (x - x_2)\,(x - x_0)]$$

By setting $x_0 = x$, $x_1 = x + h$ and $x_2 = x + 2h$,

$$\frac{dR_2}{dx} = \frac{h^2}{3} f'''(\theta), \qquad x \le \theta \le x + h$$

This error equation holds good for both forward and backward three-point formulae.

For central difference formula, we must set $x_0 = x$, $x_1 = x - h$ and $x_2 = x + h$. Therefore,

$$\frac{dR_2}{dx} = -\frac{h^2}{6} f'''(\theta), \qquad x - h \le \theta \le x + h$$

Note that the error is of order h^2

Example 11.5

The table below gives the values of distance travelled by a car at various time intervals during the initial running

Time, t(s)	5	6	7	8	9
Distance travelled, $s(t)$ (km)	10.0	14.5	19.5	25.5	32.0

Estimate velocity at time $t = 5$, $t = 7$ and $t = 9$.

We know that velocity is given by the first derivative of $s(t)$. At $t = 5$, we use the three-point forward difference formula (11.17).

$$v(t) = \frac{-3s(t) + 4s(t+h) - s(t+2h)}{2h}$$

Then

$$v(5) = \frac{-3(10) + 4(14.5) - 19.5}{2(1)}$$

$$= 4.25 \text{ km/s}$$

At $t = 7$, we use the central difference formulae (11.19). Therefore,

$$v(7) = \frac{s(8) - s(6)}{2h}$$

$$= \frac{25.5 - 14.5}{2} = 5.5 \text{ km/s}$$

At $t = 9$, we use the backward-difference formulae (11.18)

$$v(9) = \frac{3s(9) - 4s(8) + s(7)}{2h}$$

$$= \frac{3(32) - 4(25.5) + 19.5}{2}$$

$$= 6.75 \text{ km/s}$$

Higher-order Derivatives

Formulae for approximating the second and higher derivatives can also be obtained from the Newton divided difference formula. The second-order derivatives are as follows:

Central

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \qquad (11.20)$$

$$\text{Error} = -\frac{h^2}{12} f^{(4)}(\theta)$$

Forward

$$f''(x) = \frac{2f(x) - 5f(x+h) + 4f(x+2h) - f(x+3h)}{h^2} \tag{11.21}$$

$$\text{Error} = \frac{11h^2}{12} f^{(4)}(\theta)$$

Backward

$$f''(x) = \frac{2f(x) - 5f(x-h) + 4f(x-2h) - f(x-3h)}{h^2} \tag{11.22}$$

$$\text{Error} = \frac{11h^2}{12} f^{(4)}(\theta)$$

Example 11.6

Use the table of data given in Example 11.5 to estimate acceleration at $t = 7$ s.

Acceleration is given by the second derivative of $s(t)$. Therefore

$$a(t) = s''(t) = \frac{s(t+h) - 2s(t) + s(t-h)}{h^2}$$

Therefore,

$$a(7) = \frac{25.5 - 2(19.5) + 14.5}{12}$$

$$= 1.0 \text{ km/s}^2$$

Example 11.7

The equation for deflection of a beam is given by

$$y''(x) - e^{x^2} = 0 \qquad y(0) = 0,\ y(1) = 0$$

Estimate, using a second-order derivative, the approximate deflections at $x = 0.25$, 0.5, and 0.75. Note that $y(x)$ is the deflection at x.

$$y''(x) = \frac{y(x+h) - 2y(x) + y(x-h)}{h^2} = e^{x^2}$$

$$h = 0.25$$

Then,

$$\frac{y(x+0.25) - 2y(x) + y(x-0.25)}{0.0625} = e^{x^2}$$

Consequently,

$$y(x + 0.25) - 2y(x) + y(x - 0.25) = 0.0625\, e^{x^2}$$

Substituting $x = 0.25$, 0.5 and 0.75 in the above equation in turn, we get

$$y(0.5) - 2y(0.25) + y(0) = 0.0665$$
$$y(0.75) - 2y(0.5) + y(0.25) = 0.0803$$
$$y(1) - 2y(0.75) + y(0.5) = 0.1097$$

Given $y(0) = y(1) = 0$. Denoting

$$y_1 = y(0.25), \quad y_2 = y(0.5) \quad \text{and} \quad y_3 = y(0.75)$$

We have

$$0 + y_2 - 2y_1 = 0.0665$$
$$y_3 - 2y_2 + y_1 = 0.0803$$
$$-2y_3 + y_2 + 0 = 0.1097$$

Solving these three equations for y_1, y_2 and y_3, we get

$$y_1 = y(0.25) = -0.1175$$
$$y_2 = y(0.5) = -0.1684$$
$$y_3 = y(0.75) = -0.1391$$

11.4 DIFFERENCE TABLES

Tables 11.1 to 11.3 list difference derivatives $f'(x)$ and $f''(x)$ and associated errors for forward, backward and central difference formulae. Following notations are used:

f_2 denotes $f(x + 2h)$
f_{-2} denotes $f(x - 2h)$

Table 11.1 Forward difference derivatives

Derivative	*Formula*	*Error*
$f'(x_0)$	$\dfrac{-f_0 + f_1}{h}$	$-\dfrac{h}{2} f''(\theta)$ $(2e/h)^{\#}$
	$\dfrac{-3f_0 + 4f_1 - f_2}{h}$	$+\dfrac{h^2}{3} f'''(\theta)$ $(4e/h)$
	$\dfrac{-11f_0 + 18f_1 - 9f_2 + 2f_3}{6h}$	$-\dfrac{h^3}{4} f^{(4)}(\theta)$ $(20e/3h)$

(Contd.)

Table 11.1*(Contd.)*

Derivative	*Formula*	*Error*
$f'(x_0)$	$\frac{-25f_0 + 48f_1 - 36f_2 + 16f_3 - 3f_4}{12h}$	$+\frac{h^4}{5} f^{(5)}(\theta)$ $(32e/h)$
$f''(x_0)$	$\frac{f_0 - 2f_1 + f_2}{h^2}$	$-hf''(\theta)$ $(4e/h^2)$
	$\frac{2f_0 - 5f_1 + 4f_2 - f_3}{h^2}$	$+\frac{11h^2}{12} f^{(4)}(\theta)$ $(12e/h^2)$

#Roundoff error

Table 11.2 Central difference derivatives

Derivative	*Formula*	*Error*
$f'(x_0)$	$\frac{-f_{-1} + f_1}{2h}$	$-\frac{h^2}{6} f^{(3)}(\theta)$ $(e/h)^{\#}$
	$\frac{f_{-2} - 8f_{-1} + 8f_1 - f_2}{12h}$	$+\frac{h^4}{30} f^{(5)}(\theta)$ $(3e/2h)$
	$\frac{f_{-3} + 9f_{-2} - 45f_{-1} + 45f_1 - 9f_2 + f_3}{6h}$	$-\frac{h^6}{140} f^{(7)}(\theta)$ $(11e/6h)$
$f''(x_0)$	$\frac{f_{-1} - 2f_0 + f_1}{h^2}$	$-\frac{h^2}{12} f^{(4)}(\theta)$ $(4e/h^2)$
	$\frac{-2f_{-2} + 16f_{-1} - 30f_0 + 16f_1 - f_2}{h^2}$	$+\frac{h^4}{90} f^{(6)}(\theta)$ $(16e/3h^2)$

#Roundoff error

Table 11.3 Backward difference derivatives

Derivative	*Formula*	*Error*
	$\frac{-f_{-1} + f_1}{h}$	$\frac{h}{2} f''(\theta)$ $(2e/h)^{\#}$

(Contd.)

Table 11.3*(Contd.)*

Derivative	*Formula*	*Error*
$f'(x_0)$	$\frac{f_{-2} - 4f_{-1} + 3f_0}{2h}$	$\frac{h^4}{3} f^{(5)}(\theta)$ $(4e/h)$
	$\frac{-2f_{-3} + 9f_{-2} - 18f_{-1} + 11f_0}{6h}$	$\frac{h^3}{4} f^{(4)}(\theta)$ $(20e/3h)$
	$\frac{3f_{-4} - 16f_{-3} + 36f_{-2} - 48f_{-1} + 25f_0}{12h}$	$\frac{h^3}{5} f^{(5)}(\theta)$ $(32e/h)$
$f''(x_0)$	$\frac{f_{-2} - 2f_{-1} + f_0}{h^2}$	$hf^{(3)}(\theta)$ $(4e/h^2)$
	$\frac{-f_{-3} + 4f_{-2} - 5f_{-1} + 2f_0}{h^2}$	$\frac{11h^2}{12} f^{(4)}(\theta)$ $(12e/h^2)$

#Roundoff error

11.5 RICHARDSON EXTRAPOLATION

Richardson extrapolation is based on a model for the error in a numerical process. This is used to improve the estimates of numerical solutions. Let us assume

$$x_k = x^* + M h^n \tag{11.23}$$

x_k is the kth estimate of solution x^* and $M h^n$ is the error term. Let us now replace h by rh and obtain another estimate for x^*.

$$x_{k+1} = x^* + M r^n h^n \tag{11.24}$$

Multiplying Eq. (11.23) by r^n and solving for x^*, we get

$$\boxed{x^* = x_R = \frac{x_{k+1} - r^n x_k}{1 - r^n}} \tag{11.25}$$

This is known as *Richardson extrapolation estimate*. Note that the error term has been eliminated.

This concept can be extended to the estimation of derivatives discussed so far. Using this, we can obtain a higher-order formula from a lower-order formula, thus improving the accuracy of the estimates. This

process is known as *extrapolation*. Let us consider the three-point central difference formula with its error term.

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{6} f'''(\theta)$$

$$= D(h) - \frac{h^2}{6} f'''(\theta) \tag{11.26}$$

where $D(h)$ is the estimate obtained using h as step size. Note that $f'(x)$ is the exact solution which is usually approximated by $D(h)$. If we remove the error term, then we can obtain a better approximation. Now, let us obtain another approximation for $f'(x)$ by replacing h by rh. Thus,

$$f'(x) = \frac{f(x+rh) - f(x-rh)}{2hr} - \frac{h^2 r^2}{6} f'''(\theta)$$

$$= D(rh) - \frac{h^2 r^2}{6} f'''(\theta) \tag{11.27}$$

We can eliminate the error term by multiplying Eq. (11.26) by r^2 and subtracting it from Eq. (11.27). The result would be

$$f'(x) = \frac{D(rh) - r^2 D(h)}{1 - r^2} \tag{11.28}$$

This would give a better estimate of $f'(x)$ as we have eliminated the error term h^2. For $r = 2$, Eq. (11.28) becomes

$$f'(x) = \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h} \tag{11.29}$$

Note that this is a five-point central difference formula which contains error only in the order of h^4. We can repeat this process further to eliminate the error term containing h^4 and so on.

One of the most common choices of r is 0.5. Letting $r = 1/2$, Eq. (11.28) becomes

$$f'(x) = \frac{f(x-h) - 8(x-h/2) + 8f(x+h/2) - f(x+h)}{6h} \tag{11.30}$$

Note that the use of this formula depends on the availability of function values at $x \pm h/2$ points. This will be a restriction when Richardson's extrapolation technique is applied to tabulated functions.

Example 11.8

Show that, using the data given below, Richardson's extrapolation technique can provide better estimates for derivatives.

x	−0.5	−0.25	0	0.25	0.5	0.75	1.0	1.25	1.5
$f(x) = e^x$	0.6065	0.7788	1.0000	1.2840	1.6487	2.1170	2.7183	3.4903	4.4817

Let us estimate $f'(x)$ at $x = 0.5$ and assume $h = 0.5$ and $r = 1/2$. Then, using three-point central formula, we have

$$D(h) = D(0.5) = \frac{f(1.0) - f(0)}{2 \times 0.5} = 1.7183$$

$$D(rh) = D(0.25) = \frac{f(0.75) - f(0.25)}{0.5} = 1.666$$

$$f'(x) = \frac{D(rh) - r^2 D(h)}{1 - r^2}$$

Therefore,

$$f'(0.5) = \frac{1.6660 - 0.25(1.7183)}{0.75}$$

$$= 1.6486$$

Note that the correct answer is 1.6487. The result is much better than the results obtained using three-point formula with $h = 0.5$ and $h = 0.25$.

Now, let us take $r = 2$. Again using the same three-point central formula.

$$D(rh) = D(1.0) = \frac{f(1.5) - f(-0.5)}{(2)(0.5)(2)} = 1.9376$$

$$f(x) = \frac{1.9376 - 4(1.7183)}{-3} = 1.6452$$

This shows that the estimate with $r = 1/2$ is better than the estimate with $r = 2$.

11.6 SUMMARY

In this chapter, we have seen how numerical differentiation techniques may be used to obtain the derivative of continuous as well as tabulated functions. We have used forward, backward and central difference quotients to obtain derivative equations. We have also seen how Richardson extrapolation is used to improve the estimates of numerical solutions.

The discussions in this chapter bring out the following points:

- If we are given $n + 1$ data points equally spaced, the interpolating polynomial will be of order n, and the nth derivative will be the highest that can be obtained.

- Better approximation of derivatives can be achieved by using more points in the formula.
- For a given number of data points, the central difference formula is more accurate than their forward or backward counter parts.
  Roundoff error grows when h gets small. We will always face the *step-size dilemma*. One way to overcome this problem is to use a formula of higher order so that a large value of h will produce the desired accuracy.
- The problem becomes more pronounced when working with experimental data which contain not only roundoff errors but also measurement errors. In such cases, we should first fit a curve to the data by using least-squares technique and compute derivatives for the curve.

Key Terms

Backward difference derivative
Backward difference quotient
Central difference derivative
Central difference quotient
Difference tables
Extrapolation
Five-point formula
Forward difference derivative
Forward difference quotient
Richardson extrapolation
Two-point formula
Three-point formula

REVIEW QUESTIONS

1. What is numerical differentiation?
2. Why do we need to use numerical techniques to obtain the estimates of function derivatives?
3. What are the three primitive numerical differentiation formulae? Compare their truncation errors.
4. What is three-point formula? How is it different from the two-point formula? Illustrate the difference using geometric interpretations.
5. Derive the five-point central difference formula

$$f'(x) = \frac{-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)}{12h}$$

   Also estimate the order of truncation error.
6. Describe the effect of step size h on
   (a) truncation error,
   (b) roundoff error, and
   (c) total error.
7. Using Taylor's expansion, derive a formula for computing second derivative of a function.
8. Derive a three-point difference formula for estimating the first derivative of a tabulated function.

9. Derive a formula to estimate the second derivative of a tabulated function.
10. What is Richardson extrapolation? How does it improve the estimates of derivatives?

1. Estimate the first derivative of $f(x) = \ln x$ at $x = 1$ using the first order.
   (a) first-order forward difference formula,
   (b) first-order backward difference formula, and
   (c) second-order central difference formula.
   Compare the results with the exact value 1.
2. Estimate the first derivative of $f(x) = \ln x$ using the five-point central difference formula. How does the result compare with the results obtained in Exercise 1.
3. Compute the approximate first derivatives of $f(x) = \cos x$ at $x = 0.75$ radians at increasing values of h from 0.01 to 0.05 with a step size of 0.005 (using four decimal digits). Analyze the variations of error in each step.
4. Apply the three-point central difference formula to obtain estimates of the first derivatives of the following functions at $x = 1$ with $h = 0.01$. Compare the results with true values.
   (a) $\cosh x$
   (b) $\exp(x) \sin x$
   (c) $\ln (1 + x^2)$
   (d) $x^2 + 2x + 1$
   (e) $\dfrac{1}{1+x^2}$
5. For each of the following functions
   (a) $\cos x \quad x = 1.5$
   (b) $\exp(x/2) \quad x = 2$
   (c) $\dfrac{1}{1+x^2} \quad x = 1$

   estimate the size of h that will minimize total error when using the three-point central difference formula.
6. Estimate the first derivatives of the functions given in Exercise 5 at the indicated points using the optimum size h obtained.
7. Use the three-point formula to estimate the second derivatives of the functions given in Exercise 4 at $x = 0.5$ with $h = 0.01$.
8. Given below the table of function values of $f(x) = \sin h(x)$. Estimate the second derivatives of $f(x)$ at $x = 1.2$, 1.3 and 1.4 using a suitable formula.

x	1.1	1.2	1.3	1.4	1.5
$f(x)$	1.3356	1.5095	1.6983	1.9043	2.1293

9. Current through a capacitor is given by

$$I(t) = C\frac{dv}{dt} = Cv'(t)$$

where $v(t)$ is the voltage across the capacitor at time t and C is the capacitance value of the capacitor. Estimate the current through the capacitor at $t = 0.5$ using the two-point forward formula with a step size $h = 0.2$. Assume the following:

$$v(t) = (t + 0.1)\, e^{\sqrt{t}} \text{ volts}$$

$$C = 2\ F$$

10. Using the function in Exercise 8, estimate the first derivative at $x = 1.3$ with $h = 0.1$ using the three-point centre formula. Compute an improved estimate using Richardson extrapolation. Exact value of $f'(x) = \cosh(1.3) = 1.9709$.
11. Evaluate the first derivative at $x = -3$ and $x = 0$ of the following table function:

x	-3	-2	-1	0	1	2	3
y	-33	-12	-3	0	3	12	33

12. Compute the first derivative for the following table of data at $x = 0.75$, 1.00 and 1.25. Use $h = 0.05$ and 0.1.

x	0.5	0.7	0.9	1.1	1.3	1.5
y	1.48	1.64	1.78	1.89	1.96	1.00

Compare the results with $h = 0.05$ and $h = 0.1$. Comment on the differences, if any.

13. The following table gives the velocity of an object at various points in time

Time (seconds)	1	1.2	1.6	1.8	2.2	2.4	2.8	3.0
Velocity (m/sec)	9.0	9.5	10.2	11.0	13.2	14.7	18.7	22.0

Find the acceleration of the object at $T = 2.0$ seconds. Assume a suitable value for h.

14. The distances travelled by a vehicle at intervals of 2 minutes are given as follows:

Time (seconds)	0	2	4	6	8	10	12	14	16
Distance (km)	0	0.25	1	2.2	4	6.5	8.5	11	13

Evaluate the velocity and acceleration of the vehicle at $T = 5$, 10 and 13 seconds.

POGRAMMING PROJECTS

1. Write a program that will read on the values of x and $f(x)$, compute approximations to $f'(x)$ and $f''(x)$, and output x, $f(x)$, $f'(x)$ and $f''(x)$ in four columns.
2. Write a program that will compute the total error at increasing values of h at regular steps and then estimate that value of h for which the total error is minimum. Assume a formula of your choice.
3. Write a program to evaluate a given function at various points of interest and estimate its first and second derivatives at any specified point.

CHAPTER 12

Numerical Integration

12.1 NEED AND SCOPE

Like numerical differentiation, we need to seek the help of numerical integration techniques in the following situations:

1. Functions do not possess closed form solutions. Example:

$$f(x) = C \int_0^x e^{-t^2} dt$$

2. Closed form solutions exist but these solutions are complex and difficult to use for calculations.
3. Data for variables are available in the form of a table, but no mathematical relationship between them is known, as is often the case with experimental data.

We know that a definite integral of the form

$$I = \int_a^b f(x)\, dx \tag{12.1}$$

can be treated as the area under the curve $y = f(x)$, enclosed between the limits $x = a$ and $x = b$. This is graphically illustrated in Fig. 12.1. The problem of integration is then simply reduced to the problem of finding the shaded area.

One simple approach is to plot the function on a graph paper containing grids and find the area under the curve using the number grids covered under the desired boundaries. The accuracy of this rough estimate can be improved by using finer grids.

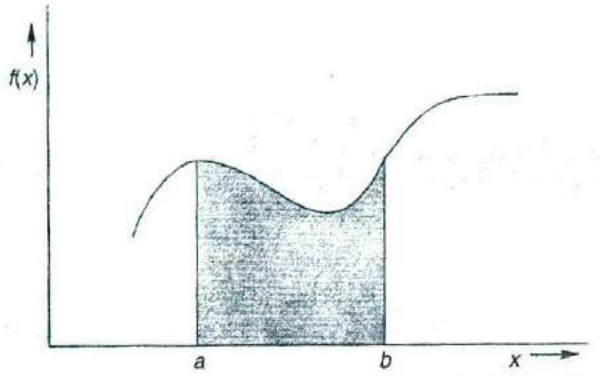


Fig. 12.1 Graphical representation of integral of a function

Although the grid method and other such graphical approaches can provide us rough estimates, they are cumbersome and time-consuming and the final results are far from satisfactory limits. A better alternative approach could be to use a technique that uses simple arithmetic operations to compute the area. Such an approach, if necessary, can be easily implemented on a computer. This approach is called *numerical integration* or *numerical quadrature*. Numerical integration techniques are similar in spirit to the graphical methods. Both of them use the concept of "summation" to find the area.

Numerical integration methods use an interpolating polynomial $p_n(x)$ in the place of $f(x)$. Thus

$$I = \int_a^b f(x)\,dx = \int_a^b p_n(x)\,dx \tag{12.2}$$

We know that the polynomial $p_n(x)$ can be easily integrated analytically. Equation (12.2) can be expressed in summation form as follows:

$$\int_a^b p_n(x)\,dx = \sum_{i=0}^{n} w_i\, p_n(x_i) \tag{12.3}$$

where $a = x_0 < x_1 < \ldots < x_n = b$

Since $p_n(x)$ coincides with $f(x)$ at all the points x_i, $i = 0, 1, \ldots n$, we can say that,

$$\boxed{I = \int_a^b f(x)\,dx \approx \sum_{i=0}^{n} w_i\,(x_i)} \tag{12.4}$$

The values x_i are called *sampling points* or *integration nodes* and the constants w_i are called *weighting coefficients* or simply *weights*.

Equation (12.4) provides the basic integration formula that will be extensively used in this chapter. Note that the interpolation polynomial

$p_n(x)$ was used only to derive the formula (12.4) and will not be used in the computation directly. Only the actual function values at sample points are used in numerical computation.

There are various methods of selecting the location and number of sampling points. There is a set of methods known as *Newton-Cotes rules* in which the sampling points are equally spaced. Another set of methods called *Gauss-Legendre rules,* uses sampling points that are not equally spaced, but are designed to provide improved accuracy. In this chapter, we discuss these two sets of methods in detail. We also discuss a method known as *Romberg integration* that is designed to improve the estimates of Newton-Cotes formulae.

In general, numerical integration methods yield much better results compared to the numerical differentiation methods discussed in the previous chapter. This is due to the fact that the errors introduced in separate subintervals tend to cancel each other. However, the estimates are still approximate and, therefore, we also consider the magnitude of errors in each of the methods discussed here.

12.2 NEWTON-COTES METHODS

Newton-Cotes formula is the most popular and widely used numerical integration formula. It forms the basis for a number of numerical integration methods known as *Newton-Cotes methods.*

The derivation of Newton-Cotes formula is based on polynomial interpolation. As pointed our earlier, an nth degree polynomial $p_n(x)$ that interpolates the values of $f(x)$ at $n + 1$ evenly spaced points can be used to replace the integrand $f(x)$ of the integral

$$I = \int_a^b f(x)\,dx$$

and the resultant formula is called $(n + 1)$ point *Newton-Cotes formula.* If the limits of integration a and b are in the set of interpolating points x_i, $i = 0, 1, \ldots n$, then the formula is referred to as *closed form.* If the points a and b lie beyond the set of interpolating points, then the formula is termed *open form.* Since open form formula is not used for definite integration, we consider here only the closed form methods. They include:

1. Trapezoidal rule (two-point formula)
2. Simpson's 1/3 rule (three-point formula)
3. Simpson's 3/8 rule (four-point formula)
4. Boole's rule (five-point formula)

All these rules can be formulated using either Newton or Lagrange interpolation polynomial for approximating the function $f(x)$. We use here the Newton-Gregory forward formula (Eq. (9.20)) which is given below:

$$p_n(s) = f_0 + \Delta f_0 s + \frac{\Delta^2 f_0}{2!} s(s-1) + \frac{\Delta^3 f_0}{3!} s(s-1)(s-2) + \ldots$$

$$= T_0 + T_1 + T_2 + \ldots + T_n \qquad (12.5)$$

where

$$s = (x - x_0)/h$$

and

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

12.3 TRAPEZOIDAL RULE

The trapezoidal rule is the first and the simplest of the Newton-Cotes formulae. Since it is a two-point formula, it uses the first order interpolation polynomial $p_1(x)$ for approximating the function $f(x)$ and assumes $x_0 = a$ and $x_1 = b$. This is illustrated in Fig. 12.2. According to Eq. (12.5), $p_1(x)$ consists of the first two terms T_0 and T_1. Therefore, the integral for trapezoidal rule is given by

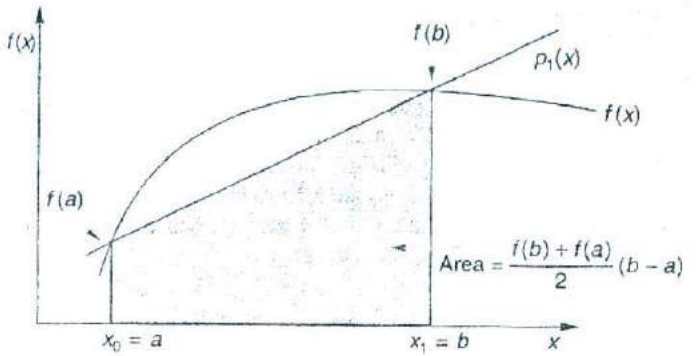


Fig. 12.2 Representation of trapezoidal rule

$$I_t = \int_a^b (T_0 + T_1)\,dx$$

$$= \int_a^b T_0\,dx + \int_a^b T_1\,dx = I_{t1} + I_{t2}$$

Since T_i are expressed in terms of s, we need to use the following transformation:

$$dx = h \times ds$$

$$x_0 = a, \quad x_1 = b \quad \text{and} \quad h = b - a$$

At $x = a, \quad s = (a - x_0)/h = 0$

At $x = b, \quad s = (b - x_0)/h = 1$

Then,

$$I_{t1} = \int_a^b T_0 \, dx = \int_0^1 h f_0 \, dx = h f_0$$

$$I_{t2} = \int_a^b T_1 \, dx = \int_0^1 \Delta f_0 \, sh \, ds = h \frac{\Delta f_0}{2}$$

Therefore,

$$I_t = h\left[f_0 + \frac{\Delta f_0}{2}\right] = h\left[\frac{f_0 + f_1}{2}\right]$$

Since $f_0 = f(a)$ and $f_1 = f(b)$, we have

$$\boxed{I_t = h\,\frac{f(a)+f(b)}{2} = (b-a)\,\frac{f(a)+f(b)}{2}} \qquad (12.6)$$

Note that the area is the *product of width of the segment* $(b - a)$ *and average height of the points* $f(a)$ *and* $f(b)$.

Error Analysis

Since only the first two terms of eq. (12.5) are used for I_t, the term T_2 becomes the remainder and, therefore, the truncation error in trapezoidal rule is given by

$$E_{tt} = \int_a^b T_2 \, dx = \frac{f''(\theta_s)}{2} \int_0^1 s(s-1)h \, ds$$

$$= \frac{f''(\theta_s)\,h}{2}\left[\frac{s^3}{3} - \frac{s^2}{2}\right]_0^1 = -\frac{f''(\theta_s)}{12}h$$

Since $dx/ds = h$,

$$f''(\theta_s) = h^2 f''(\theta_x),$$

we obtain

$$\boxed{E_{tt} = -\frac{h^3}{12} f''(\theta_x)} \qquad (12.7)$$

where $a < \theta_x < b$

Example 12.1 NUB

Evaluate the integral

$$I = \int_a^b (x^3 + 1)\,dx$$

for the intervals (a) (1, 2) and (b) (1, 1.5)
Also estimate truncation error in each case and compare the results with the exact answer.

Case a

$$a = 1, b = 2$$
$$h = 1$$

$$I_t = \frac{b-a}{2}\,[f(a) + f(b)]$$

$$= \frac{1}{2}(2 + 9) = 5.5$$

$$|E_{tt}| \le \frac{h^3}{12} \max_{1 \le x \le 2} |f''(x)|$$

$$f''(x) = 6x$$

$$\max_{1 \le x \le 2} |f''(x)| = f''(2) = 12$$

Therefore,

$$|E_{tt}| \le \frac{h^3}{12} f''(2) = 1$$

$$I_{\text{exact}} = 9.75$$

$$\text{True error} = I_t - I_{\text{exact}} = 0.75$$

Note that the error bound is an overestimate of the true error.

Case b

$$a = 1, b = 1.5$$
$$h = 0.5$$

$$I_t = \frac{0.5}{2}\,[f(1) + f(1.5)] = 1.59375$$

$$|E_{tt}| = \frac{(0.5)^3}{12} f''(1.5) = 0.09375$$

$$I_{\text{exact}} = 1.515625$$

$$\text{True error} = 0.078125$$

Composite Trapezoidal Rule

If the range to be integrated is large, the trapezoidal rule can be improved by dividing the interval (a, b) into a number of small intervals and applying the rule discussed above to each of these subintervals. The sum of areas of all the subintervals is the integral of the interval (a, b). This is known as *composite* or *multisegment approach*. This is illustrated in Fig. 12.3.

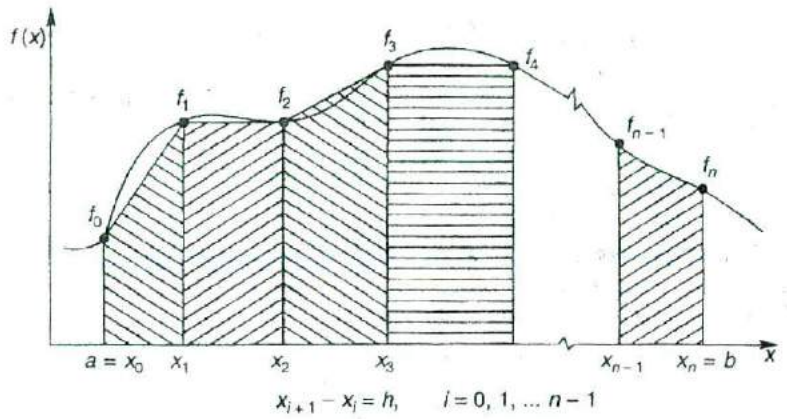


Fig. 12.3 Multisegment trapezoidal rule

As seen in Fig. 12.3, there are $n + 1$ equally spaced sampling points that create n segments of equal width h given by

$$h = \frac{b - a}{n}$$

$$x_i = a + ih, \qquad i = 0, 1, ..., n$$

From Eq. (12.6), area of the subinterval with the nodes x_{i-1} and x_i, is given by

$$I_i = \int_{x_{i-1}}^{x_i} p_1(x)\, dx = \frac{h}{2} [f(x_{i-1}) + (x_i)]$$

The total area of all the n segments is

$$I_{ct} = \sum_{i=1}^{n} \frac{h}{2} [f(x_{i-1}) + f(x_i)]$$

$$= \frac{h}{2} [f(x_0) + f(x_1)] + \frac{h}{2} [f(x_1) + f(x_2)]$$

$$+ ... + \frac{h}{2} [f(x_{n-1}) + f(x_n)]$$

Denoting $f_i = f(x_i)$ and regrouping the terms, we get

$$\boxed{I_{ct} = \frac{h}{2}\left[f_0 + 2\sum_{i=1}^{n-1} f_i + f_n\right]} \qquad (12.8)$$

Equation (12.8) is the general form of trapezoidal rule and is known as *composite trapezoidal rule*. Equation (12.8) can also be expressed as follows:

$$I_{ct} = \frac{h}{2}\,[f(a) + f(b)] + h \sum_{i=1}^{n-1} f(a + ih)$$

Similarly, we can estimate the error in the composite trapezoidal rule by adding the errors of individual segments. Thus

$$E_{ctt} = -\frac{h^3}{12} \sum_{i=1}^{n} f''(\theta_i) \tag{12.9}$$

We know that $f''(q_i)$ is the second derivative at q_i, $x_{i-1} < q_i < x_i$. If the maximum absolute value of the second derivative in the interval (a, b) is F, then we can say that the truncation error is

$$\boxed{E_{ctt} \le \frac{h^3}{12} nF = \frac{(b-a)^3}{12n^2} F} \tag{12.10}$$

Theoretically, we can say that it is always possible to increase accuracy by taking more and more segments. Unfortunately, this does not happen always. When the number of segments increases, error due to rounding off increases.

Example 12.2

Compute the integral

$$\int_{-1}^{1} e^x \, dx$$

using composite trapezoidal rule for (a) $n = 2$ and (b) $n = 4$.

Case a $n = 2$

$$h = \frac{b-a}{2} = \frac{2}{2} = 1$$

$$I_{ct} = \frac{h}{2}\,[f(a) + f(b)] + h \sum_{i=1}^{n-1} f(a + ih)$$

$$= \frac{1}{2}\,[\exp(-1) + \exp(1)] + \exp(0)$$

$$= 2.54308$$

Case b $n = 4$

$$h = \frac{b-a}{4} = 0.5$$

$$= \frac{0.5}{2} + [\exp(-1) + \exp(1)] + [\exp(-0.5) + \exp(0) + \exp(0.5)]\ 0.5$$

$$= 2.39917$$

Note that $I_{exact} = 2.35040$ and $n = 4$ gives better results.

Program TRAPE1

Trapezoidal rule is a simple algorithm and can be implemented by a few FORTRAN statements as shown in the program TRAPE1. Note that the major computation is done by just one statement in a DO loop. This statement calls a function subprogram to evaluate the given function at a specified value of x. We can use this program to integrate any function by simply changing the function definition statement

```
F = 1 - EXP(-X/2.0)
```

```
*-----------------------------------------------------*
      PROGRAM TRAPE1
*-----------------------------------------------------*
* Main program                                        *
*    This program integrates a given function         *
*    using the trapezoidal rule                       *
*-----------------------------------------------------*
* Functions invoked                                   *
*    F                                                *
*-----------------------------------------------------*
* Subroutines used                                    *
*    NIL                                              *
* Variables used                                      *
*    A - Lower limit of integration                   *
*    B - Upper limit of integration                   *
*    H - Segment width                                *
*    N - Number of segments                           *
*  ICT - Value of integral                            *
*-----------------------------------------------------*
* Constants used                                      *
*    NIL                                              *
*-----------------------------------------------------*
      INTEGER N
      REAL A, B, H, SUM, ICT
      EXTERNAL F

      WRITE(*, *) 'Give initial value of X'
      READ(*, *) A
      WRITE(*, *) 'Give final value of X'
      READ(*, *) B
```

```
      WRITE(*, *) 'What is the segment width?'
      READ(*, *) H
      N = (B-A)/H

      SUM = (F(A) + F(B))/2.0
      DO 10 I = 1, N-1
         SUM = SUM + F(A+I*H)
10    CONTINUE

      ICT = SUM * H

      WRITE(*, *)
      WRITE(*, *) 'INTEGRATION BETWEEN', A,' AND', B
      WRITE(*, *)
      WRITE(*, *) 'WHEN H =', H, ' IS', ICT
      WRITE(*, *)

      STOP
      END

* ---------------  End of main TRAPE1 ------------------ *
* ------------------------------------------------------ *
* Function subprogram F (X)                              *
* ------------------------------------------------------ *

      REAL FUNCTION F(X)
      REAL X

      F = 1-EXP(-X/2.0)

      RETURN
      END
* ---------------- End of function F(X) ---------------- *
```

Test Run Results Test run results shown below give the value of integration of the equation

$$f(x) = 1 - e^{-x/2}$$

from 0.0 to 10.0.

```
 Give initial value of X
 0.0
 Give final value of X
 10.0
 What is the segment width?
 0.5

 INTEGRATION BETWEEN  .0000000 AND 10.0000000

 WHEN H = 5.000000E-001 IS 8.0031400

 Stop - Program terminated.
```

12.4 SIMPSON'S 1/3 RULE

Another popular method is Simpson's 1/3 rule. Here, the function $f(x)$ is approximated by a second-order polynomial $p_2(x)$ which passes through three sampling points as shown in Fig. 12.4. The three points include the end points a and b and a midpoint between them, i.e., $x_0 = a$, $x_2 = b$ and $x_1 = (a + b)/2$. The width of the segments h is given by

$$h = \frac{b - a}{2}$$

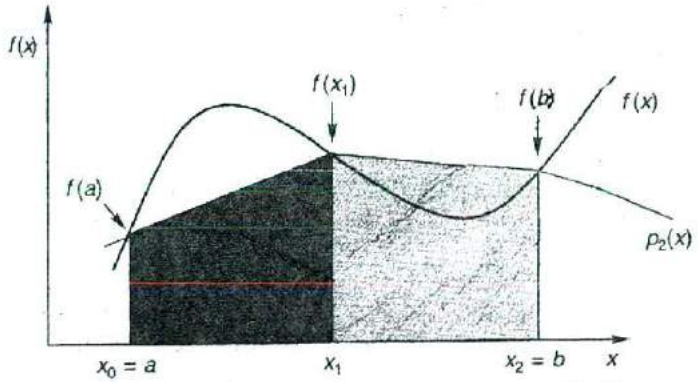


Fig. 12.4 Representation of Simpson's Three-point rule

The integral for Simpson's 1/3 rule is obtained by integrating the first three terms of equation (12.5), i.e.,

$$I_{s1} = \int_a^b p_2(x)\,dx = \int_a^b (T_0 + T_1 + T_2)\,dx$$

$$= \int_a^b T_0\,dx + \int_a^b T_1\,dx + \int_a^b T_2\,dx$$

$$= I_{s11} + I_{s12} + I_{s13}$$

where

$$I_{s11} = \int_a^b f_0\,dx$$

$$I_{s12} = \int_a^b \Delta f_0\, s\,dx$$

$$I_{s13} = \int_a^b \frac{\Delta^2 f_0}{2}\, s(s-1)\,dx$$

We know that $dx = h \times ds$ and s varies from 0 to 2 (when x varies from a to b). Thus,

$$I_{s11} = \int_0^2 f_0 h \, ds = 2hf_0$$

$$I_{s12} = \int_0^2 \Delta f_0 sh \, ds = 2h\Delta f_0$$

$$I_{s13} = \int_0^2 \frac{\Delta^2 f_0}{2} s(s-1)h \, ds = \frac{h}{3}\Delta^2 f_0$$

Therefore,

$$I_{s1} = h\left[sf_0 + 2\Delta f_0 + \frac{\Delta^2 f_0}{3}\right] \tag{12.11}$$

Since $\Delta f_0 = f_1 - f_0$ and $\Delta^2 f_0 = f_2 - 2f_1 + f_0$, equation (12.11) becomes

$$\boxed{I_{s1} = \frac{h}{3}[f_0 + 4f_1 + f_2] = \frac{h}{3}[f(a) + 4f(x_1) + f(b)]} \tag{12.12}$$

This equation is called *Simpson's 1/3 rule*. Equation (12.12) can also be expressed as

$$I_{s1} = (b-a)\frac{f(a) + 4f(x_1) + f(b)}{6}$$

This shows that the area is given by *the product of total width of the segments and weighted average of heights $f(a)$, $f(x_1)$ and $f(b)$.*

Error Analysis

Since we have used only the first three terms of Eq. (12.5), the truncation error is given by

$$E_{ts1} = \int_a^b T_3 \, dx$$

$$= \frac{f'''(\theta_s)}{6}\int_0^2 s(s-1)(s-2)h \, ds$$

$$= \frac{f'''(\theta_s)}{6}\left[\frac{s^4}{4} - s^3 + s^2\right]_0^2$$

Since the third-order error term turns out to be zero, we have to consider the next higher term for the error. Therefore,

$$E_{ts1} = \int_a^b T_4 \, dx$$

$$= \frac{f^{(4)}(\theta_s)}{4!} \int_0^2 s(s-1)(s-2)(s-3)h \, ds$$

$$= \frac{h \times f^{(4)}(\theta_s)}{24} \left[\frac{s^5}{5} - \frac{6s^4}{4} + \frac{11s^3}{3} - \frac{6s^2}{2} \right]_0^2$$

$$= -\frac{hf^4(\theta_s)}{90}$$

Since $f^4(\theta_s) = h^4 f^{(4)}(\theta_x)$, we obtain

$$\boxed{E_{ts1} = -\frac{h^5}{90} f^{(4)}(\theta_x)} \qquad (12.13)$$

where $a < \theta_x < b$. It is important to note that Simpson's 1/3 rule is exact up to degree 3, although it is based on quadratic equation.

Example 12.3

Evaluate the following integrals using Simpson's 1/3 rule

(a) $\int_{-1}^{1} e^x \, dx$ (b) $\int_0^{\pi} \sqrt{\sin x} \, dx$

Case (a)

$$I = \int_{-1}^{1} e^x \, dx$$

$$I_{s1} = \frac{h}{3} [f(a) + f(b) + 4f(x_1)]$$

$$h = \frac{b-a}{2} = 1$$

$$f(x_1) = f(a+b)$$

Therefore,

$$I_{s1} = \frac{e^{-1} + 4e^0 + e^{+1}}{3} = 2.36205$$

(Note that I_{s1} gives better estimate than I_{ct} when $n = 2$. This is because I_{s1} uses quadratic equation while I_{ct} uses a linear one)

Case (b)

$$I = \int_0^{\pi/2} \sqrt{\sin(x)} \, dx = \pi/4$$

$$I_{s1} = \frac{\pi}{12} [f(0) + 4f(\pi/4) + f(\pi/2)]$$

$$= 0.2617993(0 + 3.3635857 + 1)$$

$$= 1.1423841$$

Composite Simpson's 1/3 rule

Similar to the composite trapezoidal rule, we can construct a composite Simpson's 1/3 rule to improve the accuracy of the estimate of the area. Here again, the integration interval is divided into n number of segments of equal width, where n is an even number. Then the step size is

$$h = \frac{b-a}{n}$$

As usual, $x_i = a + ih$, $i = 0, 1, \ldots n$. Now, we can apply Eq. (12.12) to each of the $n/2$ pairs of segments or subintervals (x_{2i-2}, x_{2i-1}), (x_{2i-1}, x_{2i}). This gives

$$I_{cs1} = \frac{h}{3}\sum_{i=1}^{n/2}[f(x_{2i-2}) + 4f(x_{2i-1}) + f(x_{2i})]$$

$$= \frac{h}{3}[f(a) + 4f_1 + 2f_2 + 4f_3 + \ldots 2f_{n-2} + 4f_{n-1} + f(b)]$$

On regrouping terms, we get

$$\boxed{I_{cs1} = \frac{h}{3}\left[f(a) + 4\sum_{i=1}^{n/2} f(x_{2i-1}) + 2\sum_{i=1}^{(n/2)-1} f(x_{2i}) + f(b)\right]} \quad (12.14)$$

An analysis similar to the error analysis of composite trapezoidal rule can be performed to obtain the error due to truncation in composite Simpson's 1/3 rule.

$$\boxed{|E_{tcs1}| \le \frac{h^5}{180} nF = \frac{(b-a)^5}{180n^4} F} \quad (12.15)$$

where F is the maximum absolute value of the fourth derivative of $f(x)$ in the interval (a, b).

Example 12.4

Compute the integral

$$\int_0^{\pi/2} \sqrt{\sin(x)}\, dx$$

applying Simpson's 1/3 rule for $n = 4$ and $n = 6$ with an accuracy to five decimal places.

The composite Simpson's 1/3 rule is given by

$$I_{cs1} = \frac{h}{3}\left[(f(x_0) + f(x_a)) + 4\sum_{i=1}^{n/2} f(x_{2i-1}) + 2\sum_{i=1}^{n/2-1} f(x_{2i})\right]$$

For $n = 4$, $h = \pi/8$

There are five sampling points given by $x_k = k\pi/8$, $k = 0, 1, ..., 4$. Substituting the values of x_k in the composite rule, we get,

$$I_{cs1} = \frac{\pi}{24}\,[f(0) + f(\pi/2) + 4f(\pi/8) + 4f(3\pi/8) + 2f(\pi/4)]$$

$$= \frac{\pi}{24}\,[0 + 1.0 + 4(0.61861 + 0.96119) + 2\,(0.84090)]$$

$$= 1.17823$$

For $n = 6$, $h = \pi/12$

There are seven sampling points given by $x_k = k\pi/12$, $k = 0, 1, ..., 6$. Substituting these values in the above equation, we get

$$I_{cs1} = \frac{\pi}{36}[0 + 1.0 + 4(0.50874 + 0.84090 + 0.98282) + 2(0.70711 + 0.93060)]$$

$$= 1.18728$$

Program SIMS1

Program SIMS1 integrates a given function using the composite Simpson's 1/3 rule. Note that, unlike TRAPE1 which requests for segment width h, SIMS1 requests for number of segments n. Remember, n should be even. This program also uses a function subprogram which can be easily replaced for any other function without modifying the main program.

```
*-------------------------------------------------------*
      PROGRAM SIMS1
*-------------------------------------------------------*
* Main program                                          *
*   This program integrates a given function            *
*   using the Simpson's 1/3 rule                        *
*-------------------------------------------------------*
* Functions invoked                                     *
*   F                                                   *
*-------------------------------------------------------*
* Subroutines used                                      *
*   NIL                                                 *
*-------------------------------------------------------*
* Variables used                                        *
*   A - Lower limit of integration                      *
*   B - Upper limit of integration                      *
*   H - Segment width                                   *
*   N - Number of segments                              *
* ICS - Value of the integral                           *
*-------------------------------------------------------*
* Constants used                                        *
*   NIL                                                 *
*-------------------------------------------------------*
```

```
      INTEGER N,M
      REAL A,B,H,SUM,ICS,X,F1,F2,F3
      EXTERNAL F

      WRITE (*,*) 'Initial value of X'
      READ (*,*) A
      WRITE(*,*) 'Final value of X'
      READ(*,*) B
      WRITE(*,*) 'Number of segments (EVEN number)'
      READ(*,*) N

      H = (B-A)/N
      M = N/2

      SUM = 0.0
      X = A
      F1 = F(X)
      DO 10 I = 1,M
         F2 = F(X+H)
         F3 = F(X+2*H)
         SUM = SUM + F1 + 4*F2 + F3
         F1 = F3
         X = X + 2*H
10    CONTINUE

      ICS = SUM * H/3.0

      WRITE(*,*)
      WRITE(*,*) INTEGRAL FROM', A, '   TO', B
      WRITE(*,*)
      WRITE(*,*) 'WHEN H = ',H, '   IS', ICS
      WRITE(*,*)

      STOP
      END

------------------ End of main SIMS1 ------------------*
------------------------------------------------------*
Function subprogram F(X)                               *
------------------------------------------------------*

      REAL FUNCTION F(X)
      REAL X

      F = 1-EXP(-X/2.0)

      RETURN
      END
---------------- End of function F(X) ----------------*
```

Test Run Results Output of the program for integrating the function

$$f(x) = 1 - e^{-x/2}$$

from 0.0 to 10.0 is given below:

```
Initial value of X
0.0
Final value of X
10.0
Number of segments (EVEN number)
20

INTEGRAL FROM      .0000000   To   10.0000000
WHEN  H  =  5.000000E-001     IS        8.0134330
Stop - Program terminated.
```

12.5 SIMPSON'S 3/8 RULE

Simpson's 1/3 rule was derived using three sampling points that fit a quadratic equation. We can extend this approach to incorporate four sampling points so that the rule can be exact for $f(x)$ of degree 3. Remember, even Simpson's 1/3 rule, although it is based on three points, is third-order accurate. However, a formula based on four points can be used even when the number of segments is odd.

By using the first four terms of Eq. (12.5) and applying the same procedure followed in the previous case, we can show that

$$I_{s2} = \frac{3h}{8}\,[f(a) + 3f(x_1) + 3f(x_2) + f(b)] \tag{12.16}$$

where $h = (b - a)/3$. This equation is known as *Simpson's 3/8 rule*. This is also known as *Newton's three-eighths rule*.

Similarly, we can show that, using the fifth term of Eq. (12.5), the truncation error of Simpson's 3/8 rule is

$$E_{ts2} = -\frac{3h^5}{80} f^{(4)}(\theta_x) = -\frac{(b-a)^5}{6480} f^{(4)}(\theta_x) \tag{12.17}$$

where $a < \theta_x < b$.

For a given interval (a, b), the truncation error of Simpson's 1/3 rule is

$$E_{ts1} = -\frac{h^5}{90} f^{(4)}(\theta_x) = -\frac{(b-a)^5}{2880} f^{(4)}(\theta_x)$$

This shows that the 3/8 rule is slightly more accurate than the 1/3 rule.

Use Simpson's 3/8 rule to evaluate

$$\text{(a)} \int_1^2 (x^3 + 1)\,dx \qquad \text{(b)} \int_0^{\pi/2} \sqrt{\sin(x)}\,dx$$

Case (a)

Basic Simpson's 3/8 rule is based on four sampling points and, therefore, $n = 3$.

$$I_{s2} = \frac{3h}{8}[f(a) + 3f(x_1) + 3f(x_2) + f(b)]$$

$$h = \frac{b-a}{a} = \frac{1}{3}$$

$$x_1 = a + h = 1 + 1/3 = 4/3$$

$$x_2 = a + 2h = 1 + 2/3 = 5/3$$

on substitution of these values, we obtain

$$I_{s2} = \frac{1}{8}[f(1) + f(2) + 3f(4/3) + 3f(5/3)]$$

$$= 4.75$$

Note that the answer is exact. This is expected because Simpson's 3/8 rule is supposed to be exact for cubic polynomials.

Case (b)

$$I = \int_0^{\pi/2} \sqrt{\sin(x)}\,dx$$

Here again, $n = 3$ and the integral is given by

$$I_{s2} = \frac{3h}{8}[f(a) + 3f(x_1) + 3f(x_2) + f(b)]$$

$$h = \frac{b-a}{3} = \frac{\pi}{6}$$

$$x_1 = a + h = \frac{\pi}{6}$$

$$x_2 = a + 2h = \frac{\pi}{3}$$

on substitution of these values, we obtain

$$I_{s2} = \frac{\pi}{16}[f(0) + 3f(\pi/6) + f(\pi/3) + f(\pi/2)]$$

$$= \frac{\pi}{16}[0 + 2.12132 + 2.79181 + 1.0]$$

$$= 1.16104$$

12.6 HIGHER ORDER RULES

There is no limit to the number of sampling points that could be incorporated in the derivation of Newton-Cotes rule. For instance, we can use a five-point rule to fit exactly the function $f(x)$ of degree 9 and so on. Since the repeated use of lower-order rules provide sufficient accuracy of the estimates, higher-order methods are rarely used.

One more rule which is sometimes used is *Boole's rule* based on five sampling points. This is given by

$$\boxed{I_b = \frac{2h}{45}(7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_3)} \tag{12.18}$$

where $h = (b - a)/4$

The truncation error of Boole's rule is

$$\boxed{E_{tb} = -\frac{8h^7}{945} f^{(6)}(\theta_x)} \tag{12.19}$$

Example 12.6

Use Boole's five-point formula to compute

$$\int_0^{\pi/2} \sqrt{\sin(x)}\, dx$$

and compare the results with those obtained in previous examples.

$n = 4,\ h = \pi/8$

$f_0 = 0$

$f_1 = f(\pi/8) = 0.61861$

$f_2 = f(\pi/4) = 0.84090$

$f_3 = f(3\pi/8) = 0.96119$

$f_4 = f(\pi/2) = 1.0$

$$I_b = \frac{\pi}{180}[0 + 32(0.61861 + 0.96119) + 12(0.84090) + 7(1.0)]$$

$$= 1.18062$$

The table below shows the results of

$$\int_0^{\pi/2} \sqrt{\sin(x)}\, dx$$

obtained by various Newton-Cotes rules.

Rule	n	Result
Trapezoidal (simple)	1	0.78540
Trapezoidal (composite)	2	1.05314
Simpson's 1/3 (simple)	2	1.14238
Simpson's 3/8	3	1.16104
Simpson's 1/3 (composite)	4	1.17823
Boole's rule	4	1.18062
Simpson's 1/3 (composite)	6	1.18728
Simpson's 1/3 (composite)	12	1.19429

The estimate can be further improved by using still more intervals.

Table 12.1 lists the basic Newton-Cotes rules.

Table 12.1 Basic Newton-Cotes rules

Name	*Intervals (n)*	*Formula*	*Error*
Trapezoidal	1	$\frac{h}{2}(f_0 + f_1)$	$-\frac{h^2}{12}f''(\theta)$
Simpson's 1/3	2	$\frac{h}{3}(f_0 + 4f_1 + f_2)$	$-\frac{h^5}{90}f^{(4)}(\theta)$
Simpson's 3/8	3	$\frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3)$	$-\frac{3h^5}{80}f^{(4)}(\theta)$
Boole's rule	4	$\frac{2h}{45}(7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4)$	$-\frac{8h^7}{945}f^{(6)}(\theta)$

12.7 ROMBERG INTEGRATION

It is clear from the discussions we had so far that the accuracy of a numerical integration process can be improved in two ways:

1. By increasing the number of subintervals (i.e. by decreasing h)—this decreases the magnitude of error terms. Here, the order of the method is fixed.
2. By using higher-order methods—this eliminates the lower-order error terms. Here, the order of the method is varied and, therefore, this method is known as *variable-order approach*.

The variable-order method can be implemented using Richardson's extrapolation technique discussed in the previous chapter. As we know, this technique involves combining two estimates of a given order to obtain a third estimate of higher order. The method that incorporates this process (i.e. Richardson's extrapolation) to the trapezoidal rule is called *Romberg integration*.

According to the Euler-Maclaurin formula, the error expansion for trapezoidal rule approximation to a definite integral is of the form

$$\int_a^b f(x)\mathrm{d}x - T(h) = a_2h^2 + a_4h^4 + a_6h^6 + \ldots \qquad (12.20)$$

$T(h)$ is the trapezoidal approximation with step size = $(b - a)/n = h$. Let us define

$$T(h, 0) = T(h)$$

to indicate that $T(h)$ is the trapezoidal rule with no Richardson's extrapolation being applied (zero level extrapolation). Thus, Eq. (12.20) can be written as

$$I = T(h, 0) + a_2h^2 + a_4h^4 + a_6h^6 + \ldots \tag{12.21}$$

Let us have another estimate with step size = $(b - a)/2n = h/2$ (at zero level extrapolation) as

$$I = T(h/2, 0) + \frac{a_2}{4}h^2 + \frac{a_4}{16}h^4 + \frac{a_6}{32}h^6 + \ldots \tag{12.22}$$

By multiplying Eq. (12.22) by 4 and then subtracting Eq. (12.21) from the resultant equation, we obtain (after rearranging terms),

$$I = \frac{4T(h/2, 0) - T(h, 0)}{4 - 1} + b_4h^4 + b_6h^6 + \ldots$$

$$= T(h/2, 1) + b_4h^4 + b_6h^6 + \ldots \tag{12.23}$$

where

$$T(h/2, 1) = \frac{4T(h/2, 0) - T(h, 0)}{3}$$

is the *corrected* trapezoidal formula using Richardson's extrapolation technique "once" (level 1). Note that its truncation error is of the order h^4, instead of h^2 which is the order in the "uncorrected" trapezoidal formula.

Now, we can apply Richardson's extrapolation technique once more to Eq. (12.23) to eliminate the error term containing h^4. The result would be

$$I = \frac{16T(h/4, 1) - T(h/2, 1)}{16 - 1} + C_6h^6 + \ldots$$

$$= T(h/4, 2) + C_6h^6 + \ldots \tag{12.24}$$

where

$$T(h/4, 2) = \frac{16T(h/4, 1) - T(h/2, 1)}{16 - 1}$$

is the estimate, refined again by applying Richardson's extrapolation a second time (level 2). Similarly, we can obtain an estimate with third-level correction as

$$T(h/8, 3) = \frac{64T(h/8, 2) - T(h/4, 2)}{64 - 1}$$

The entire process of repeated use of Richardson's extrapolation technique can be represented in general form as

$$T(h/2^i, j) = \frac{4^j T(h/2^i, j-1) - T(h/2^{i-1}, j-1)}{4^j - 1} \tag{12.25}$$

where $i = 0, 1, 2 \ldots$ denotes the depth of division and $j \leq i$ denotes the level of improvement.

We can further simplify the notation of Eq. (12.25) by defining

$$R_{ij} = T(h/2^i, j)$$

Thus, we have

$$R_{ij} = \frac{4^j R_{i,j-1} - R_{i-1,j-1}}{4^j - 1} \tag{12.26}$$

Equation (12.26) is known as *Romberg integration formula.* Note that this equation, when expanded, will form a lower-diagonal matrix. The elements of the matrix **R** are computed row by row in the order indicated in Fig. 12.5. The circled numbers indicate the order of computations and the arrows indicate the dependencies of elements. An element at the head end depends on the element at the tail end.

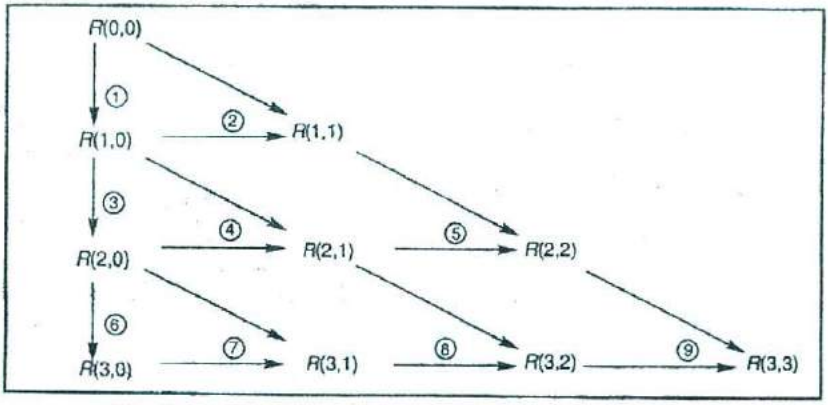


Fig. 12.5 Order of calculations for Romberg integration

Elements in the first column represent trapezoidal rule at h, $h/2$, $h/4$, etc. They can be evaluated recursively as follows:

$$h = b - a$$

$$R(0, 0) = \frac{h}{2}[f(a) + f(b)]$$

$$R(i, 0) = \frac{R(i-1, 0)}{2} + h_i \sum_{k=1}^{2^{i-1}} f(x_{2k-1}) \quad \text{for } i = 1, 2, \ldots \tag{12.27}$$

where

$$h_i = (b - a)/2^i$$

$$x_k = a + kh_i$$

Equation (12.27) is known as *recursive trapezoidal rule.*

Example 12.7

Compute Romberg estimate R_{22} for

$$\int_1^2 1/x \, dx$$

First we apply the basic trapezoidal rule to obtain $R(0,0)$

$$R(0, 0) = \frac{h}{2} [f(a) + f(b)]$$

$$= \frac{2-1}{2}(1 + 1/2) = 0.75$$

Now, we obtain $R(1, 0)$ and $R(2, 0)$ using equation (12.27)

$$R(1, 0) = \frac{R(0,0)}{2} + h_1 f(x_1)$$

$$= \frac{0.75}{2} + \frac{1}{2} \times \frac{1}{1.5} = 0.7083333$$

$$R(2, 0) = \frac{R(1,0)}{2} + h_2[f(x_1) + f(x_3)]$$

$$= \frac{0.7083333}{2} + \frac{1}{4}[f(1.25) + f(1.75)]$$

$$= 0.6970237$$

Now, Romberg approximations can be obtained using Eq. (12.26).

$$R(1, 1) = \frac{4R(1, 0) - R(0, 0)}{3}$$

$$= \frac{4(0.7083333) - 0.75}{3} = 0.6944444$$

$$R(2, 1) = \frac{4R(2, 0) - R(1, 0)}{3}$$

$$= \frac{4(0.6970237) - 0.7083333}{3} = 0.6932538$$

$$R(2, 2) = \frac{16R(2, 1) - R(1, 1)}{15}$$

$$= \frac{16(0.6932538) - 0.6944444}{15} = 0.6931744$$

Correct answer = $I_n(2) = 0.6931471$

Error = 0.0000273

Program ROMBRG

Computer algorithm for implementing Romberg integration is simple and straight-forward. Starting from the element $R(0, 0)$, all the other elements are calculated row by row. The elements in the first column are calculate using the recursive trapezoidal rule (Eq. (12.27)) and the remaining elements are calculated using the Romberg integration formula (Eq. (12.26)). The process is terminated when two diagonal elements $R(i-1, j-1)$ and $R(i, j)$ agree to the required level of accuracy.

Program ROMBRG implements the steps involved in Romberg integration.

```
*-----------------------------------------------------------*
      PROGRAM ROMBRG
*-----------------------------------------------------------*
* Main program                                              *
*   This program performs Romberg integration               *
*   by bisecting the intervals N times                      *
*-----------------------------------------------------------*
* Functions invoked                                         *
*   F,ABS                                                   *
*-----------------------------------------------------------*
* Subroutines used                                          *
*   NIL                                                     *
*-----------------------------------------------------------*
* Variables used                                            *
*   A - Starting point of the interval                      *
*   B - End point of the interval                           *
*   H - Width of the interval                               *
*   N - Number of times bisection is done                   *
*   M - Number of trapezoids                                *
*   R - Matrix of Romberg integral values                   *
*-----------------------------------------------------------*
* Constants used                                            *
*   EPS - Error bound                                       *
*-----------------------------------------------------------*
      REAL A,B,EPS,H,R,SUM,F,X,ABS
      INTEGER N,M
      INTRINSIC ABS
      EXTERNAL F
      PARAMETER( EPS = 0.00001 )
      DIMENSION R(10,10)
```

```
      WRITE(*,*) 'Input endpoints of the interval'
      READ(*,*) A,B
      WRITE(*,*) 'Input maximum number of times'
      WRITE(*,*) 'the subintervals are bisected'
      READ(*,*) N
*  Compute using entire interval as one trapezoid
      H = B-A
      R(1,1) = H * (F(A) + F(B))/2.0
      WRITE(*,*)
      WRITE(*,*) R(1,1)
      DO 30 I = 2, N+1
*        Determine number of trapezoids for I_th refinement
      M = 2**(I-2)
*        Reduce step size for I_th refinement
      H = H/2
*  Use recursive trapezoidal rule for M strips
      SUM = 0.0
      DO 10 K = 1,M
         X = A+(2*K-1)*H
         SUM = SUM + F(X)
         R(I,1) = R(I-1,1)/2.0+H*SUM
10    CONTINUE
*  Compute Richardson's improvements
      DO 20 L = 2,I
         R(I,L)= R(I,L-1)+(R(I,L-I) - R(I-1,L-1))/(4**(L-1)-1)
20    CONTINUE
*  Write the results of improvements for I_th refinement
      WRITE(*,*) (R(I,L),L = 1,I)
*  Test for desired accuracy
      IF(ABS(R(I-1,I-1) - R(I,I)) .LE. EPS) THEN
*        Stop further refinement
         WRITE(*,*)
         WRITE(*,*) 'ROMBERG INTEGRATION =', R(I,I)
         WRITE(*,*)
         GO TO 40
      ENDIF
*  Continue with the refinement process
30 CONTINUE
*  write the final result
      WRITE(*,*)
      WRITE(*,*) 'ROMBERG INTEGRATION = ', R(N+1,N+1)
      WRITE(*,*) '(Exit from loop)'
      WRITE(*,*)
```

```
40   STOP
     END

*  ----------------End of main ROMBRG ----------------  *
*  ------------------------------------------------------  *
*    Function subprogram F(X)                              *
*  ------------------------------------------------------  *

     REAL FUNCTION F(X)
     REAL X
     F = 1.0/X
     RETURN
     END

*  ---------------- End of function F(X) ----------------  *
```

Test Run Results

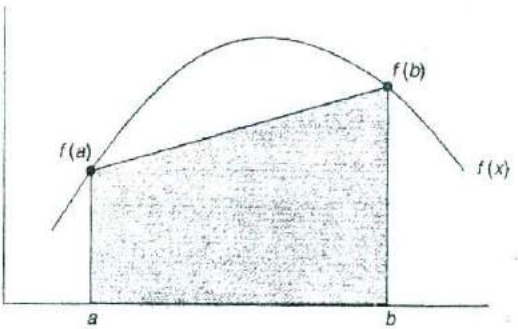
```
First run
      Input endpoints of the interval
      1 2
      Input maximum number of times
      the subintervals are bisected.
      1

           7.500000E-001
           7.083334E-001  6.944445E-001

      ROMBERG INTEGRATION = 6.944445E-001
      (Exit from loop)

Second run
      Input endpoints of the interval
      1 2
      Input maximum number of times
      the subintervals are bisected
      2

        7.500000E-001
        7.083334E-001    6.944445E-001
        6.970239E-001    6.932541E-001    6.931747E-001

      ROMBERG INTEGRATION = 6.931747E-001

      (Exit from loop)
```

GAUSSIAN INTEGRATION

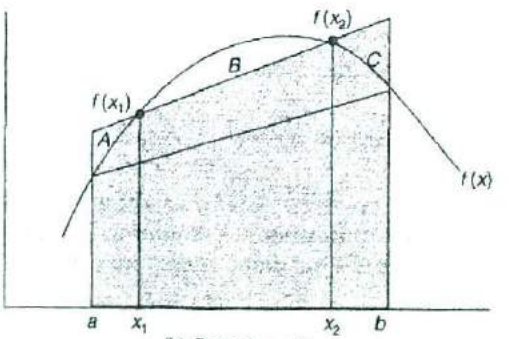
We have discussed so far a set of rules based on the Newton-Cotes formula. Recall that the Newton-Cotes formula was derived by integrat-

ing the Newton-Gregory forward difference interpolating polynomial. Consequently, all the rules were based on evenly faced sampling points (function values) within the range of integral.

Gauss integration is based on the concept that the accuracy of numerical integration can be improved by choosing the sampling points wisely, rather than on the basis of equal spacing. For example, consider a simple trapezoidal rule as shown in Fig. 12.6(a). Here, the end points of the integral lie on the function curve. Now, consider Fig. 12.6(b). Here, the straight line has been moved up such that area $B = A + C$. Notice that the sampling points are moved away from the end points. The function values at the end points are not used in computation. Rather, function values $f(x_1)$ and $f(x_2)$ are used to compute the shaded area. It is clear that the area obtained from Fig. 12.6(b) would be much closer to the actual area compared to the shaded area in Fig. 12.6(a). The problem is to compute the values of x_1 and x_2 given the values a and b and to choose appropriate "weights" w_1 and w_2. The method of implementing the strategy of finding appropriate values of x_i and w_i and obtaining the integral of $f(x)$ is called the *Gaussian integration* or *quadrature*.



(a) Trapezoidal rule



(b) Gaussian rule

Fig. 12.6 Gaussian integration

Gauss integration assumes an approximation of the form

$$I_g = \int_{-1}^{1} f(x)\,dx \simeq \sum_{i=1}^{n} w_i f(x_i) \tag{12.28}$$

Equation (12.28) contains $2n$ unknowns to be determined. These unknowns can be determined using the condition given in the integration formula (12.28). This should give the exact value of the integral for polynomials of as high a degree as possible.

Let us find the Gaussian quadrature formula for $n = 2$. In this case, we need to find the values of w_1, w_2, x_1 and x_2. Let us assume that the integral will be exact up to cubic polynomials. This implies that the functions 1, x, x^2 and x^3 can be numerically integrated to obtain exact results.

$$w_1 + w_2 = \int_{-1}^{1} dx = 2$$

$$w_1 x_1 + w_2 x_2 = \int_{-1}^{1} x\,dx = 0$$

$$w_1 x_1^2 + w_2 x_2^2 = \int_{-1}^{1} x^2\,dx = \frac{2}{3}$$

$$w_1 x_1^3 + w_2 x_2^3 = \int_{-1}^{1} x^3\,dx = 0$$

Solving these simultaneous equations, we obtain

$$w_1 = w_2 = 1$$

$$x_1 = -\frac{1}{\sqrt{3}} = -0.5113502$$

$$x_2 = \frac{1}{\sqrt{3}} = 0.5773502$$

Thus, we have the Gaussian quadrature formula for $n = 2$ as

$$\int_{-1}^{1} f(x)\,dx = f\left(-1/\sqrt{3}\right) + f\left(1/\sqrt{3}\right) \tag{12.29}$$

This formula will give correct value for integral of $f(x)$ in the range (–1, 1) for any function up to third-order. Equation (12.29) is also known as *Gauss-Legendre* formula. Two-point Gauss quadrature is illustrated in Fig. 12.7.

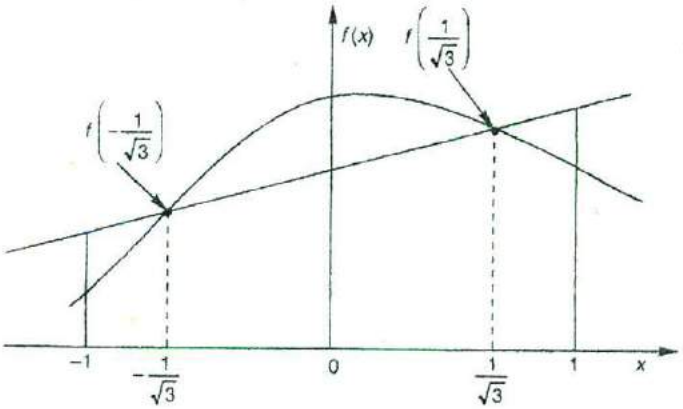


Fig. 12.7 Illustration of Gauss-Legendre formula

Example 12.8

Compute $\int_{-1}^{1} e^x \, dx$ using two-point Gauss-Legendre formula

$$I = \int_{-1}^{1} \exp(x)\, dx$$

$$= f(x_1) + f(x_2)$$

where x_1 and x_2 are Gaussian quadrature points and are given by

$$x_1 = -\frac{1}{\sqrt{3}} = -0.5773502$$

$$x_2 = +\frac{1}{\sqrt{3}} = 0.5773502$$

Therefore,

$$I = \exp(-0.5773502) + \exp(0.5773502)$$
$$= 0.5613839 + 1.7813122$$
$$= 2.3426961$$

Changing Limits of Integration

Note that the Gaussian formula imposes a restriction on the limits of integration to be from –1 to 1. This restriction can be overcome by using the technique of "interval transformation" used in calculus. Let

$$\int_a^b f(x)\, dx = C \int_{-1}^{1} g(z)\, dz$$

Assume the following transformation between x and the new variable z.

$$x = Az + B$$

This must satisfy the following conditions:

At

$$x = a, \quad z = -1 \quad \text{and } x = b, \quad z = 1$$

That is

$$B - A = a$$
$$A + B = b$$

Then

$$A = \frac{b-a}{2} \quad \text{and} \quad B = \frac{a+b}{2}$$

Therefore

$$x = \frac{b-a}{2} z + \frac{a+b}{2}$$

$$\mathrm{d}x = \frac{b-a}{2} \mathrm{d}z$$

This implies that

$$C = \frac{b-a}{2}$$

Then the integral becomes

$$\frac{b-a}{2} \int_{-1}^{1} g(z)\, \mathrm{d}z$$

The Gaussian formula for this integration is

$$\frac{b-a}{2} \int_{-1}^{1} g(z)\, \mathrm{d}z = \frac{(b-a)}{2} \sum_{i=1}^{n} w_i g(z_i)$$

where w_i and z_i are the weights and quadrature points for the integration domain (–1, 1)

Example 7.9

Compute the integral

$$I = \int_{-2}^{2} e^{-x/2} \mathrm{d}x$$

using Gaussian two-point formula.

n= 2 and therefore

$$I_g = \frac{b-a}{2} [w_1 g(z_1) + w_2 g(z_2)]$$

$$x = \frac{b-a}{2} z + \frac{b+a}{2} = 2z$$

Therefore,

$$g(z) = e^{-2z/2} = e^{-z}$$

For a two-point formula

$$w_1 = w_2 = 1$$

$$z_1 = -\frac{1}{\sqrt{3}}$$

$$z_2 = \frac{1}{\sqrt{3}}$$

Upon substitution of these values, we get

$$I_g = 2 [\exp(-1/\sqrt{3}) + \exp(1/\sqrt{3})]$$
$$= 4.6853922$$

Higher-Order Gaussian Formulae

By using a procedure similar to the one applied in deriving two-point formula, we can obtain the parameters w_i and z_i for higher-order versions of Gaussian quadrature. These parameters for formulae up to an order of six are tabulated in Table 12.2.

Table 12.2 Parameters for Gaussian integration

n	i	w_i	z_i
2	1	1.00000	−0.57735
	2	1.00000	0.57735
3	1	0.55556	−0.77460
	2	0.88889	0.00000
	3	0.55556	−0.77460
4	1	0.34785	−0.86114
	2	0.65215	−0.33998
	3	0.65215	+0.33998
	4	0.34785	0.86114
5	1	0.23693	−0.90618
	2	0.47863	−0.53847
	3	0.56889	0.00000
	4	0.47863	0.53847
	5	0.23693	0.90618

(Contd.)

Table 12.2 *(Contd.)*

n	i	w_i	z_i
6	1	0.17132	−0.93247
	2	0.36076	−0.66121
	3	0.46791	−0.23862
	4	0.46791	0.23862
	5	0.36076	0.66121
	6	0.17132	0.93247

Example 12.10

Use Gauss-Legendre three-point formula to evaluate

$$\int_2^4 (x^4 + 1)\,dx$$

Given $n = 3$, $a = 2$, and $b = 2$. Hence

$$I_g = \frac{b-a}{2} \sum_{i=1}^{3} w_i g(z_i)$$

$$= w_1 g(z_1) + w_2 g(z_2) + w_3 g(z_3)$$

$$x = \frac{(b-a)}{2} z + \frac{b+a}{2} = z + 3$$

Therefore,

$$g(z) = (z + 3)^4 + 1$$

For $n = 3$, we have

$$w_1 = 0.55556 \qquad z_1 = -0.77460$$

$$w_2 = 0.88889 \qquad z_2 = 0.0$$

$$w_3 = 0.55556 \qquad z_3 = 0.77460$$

Then

$$I_g = 0.55556\,[(-0.77460 + 3)^4 + 1]$$
$$+0.88889\,[(0+3)^4 + 1]$$
$$+0.55556[(0.77460 + 3)^4 + 1]$$
$$= 14.18140 + 72.88898 + 113.33105$$
$$= 200.40143$$

We can verify the answer with analytical solution which is 200.4. Note that three-point Gauss formula should give exact answer for a second order polynomial. The difference in the answer is due to roundoff errors. Roundoff error can be minimised by increasing the precision of Gaussian parameters.

Algorithm 12.1 gives a simple procedure for implementing Gauss-Legendre formula.

Gaussian Integration

1. Define the function, $f(x)$
2. Obtain integration limits (a, b)
3. Decide number of interpolating points (n)
4. Read the Gaussian parameters (w_i, z_i)
5. Compute x_i using

$$x_i = \frac{(b-a)}{2} z_i + \frac{b-a}{2}$$

6. Compute I_g

$$I_g = \frac{b-a}{2} \sum_{i=1}^{n} w_i f(x_i)$$

7. Write result

Algorithm 12.1

12.9 SUMMARY

In this chapter, we discussed the integration of definite integrals using numerical integration techniques. The following Newton-Cotes methods were considered in detail:

- Trapezoidal rule
- Simpson's 1/3 rule
- Simpson's 3/8 rule
- Boole's rule

We also presented a method known as Romberg integration to improve the accuracy of the results of the trapezoidal method.

We finally discussed another approach known as Gauss integration which is based on the concept that the accuracy can be improved by choosing the sampling points wisely, rather than equally.

FORTRAN programs were presented for the following methods:

- Trapezoidal rule
- Simpson's 1/3 rule
- Romberg integration

Key Terms

Boole's rule
Closed form
Newton's three-eighths rule
Newton-Cotes formula

(Contd.)

(Contd.)

Composite approach	*Newton-Cotes rules*
Composite Simpson's 1/3 rule	*Numerical integration*
Composite trapezoidal rule	*Numerical quadrature*
Extrapolation	*Open form*
Gaussian integration	*Recursive trapezoidal rule*
Gaussian quadrature	*Richardson's extrapolation*
Gauss-Legendre formula	*Romberg integration*
Gauss-Legendre rules	*Romberg integration formula*
Integration nodes	*Simpson's 1/3 rule*
Lagrange interpolation polynomial	*Simpson's 3/8 rule*
Multisegment approach	*Trapezoidal rule*
Newton interpolation polynomial	*Variable-order approach*

REVIEW QUESTIONS

1. What is numerical integration?
2. When do we need to use a numerical method instead of analytical method for integration?
3. Numerical integration is similar in spirit to the graphical method of finding area under the curve. Explain.
4. Explain the basic principle used in Newton-Cotes methods.
5. Describe the trapezoidal method of computing integrals.
6. What is composite trapezoidal rule? When do we use it?
7. In composite trapezoidal rule, the error is estimated to be inversely proportional to the square of number of segments. That is, we can decrease the error by taking more and more segments. But this does not happen always. Why? Explain.
8. Describe Simpson's method of computing integrals.
9. Prepare a flow chart for implementing the trapezoidal rule.
10. Prepare a flow chart for implementing Simpson's one-third rule.
11. Show that Simpson's one-third rule is exact up to degree 3.
12. Derive Simpson's three-eighth's rule using the first four terms of Newton-Gregory forward formula.
13. State formulae and error terms for the four basic Newton-Cotes rules.
14. How could we improve the accuracy of a numerical integration process?
15. What is Romberg integration? How does it improve the accuracy of integration?
16. Explain the concept used in Gaussian quadrature.

REVIEW EXERCISES

1. Evaluate analytically the following integrals:

   (a) $\int_0^2 (3x^2 + 2x - 5)\,dx$

   (b) $\int_0^2 (3x^3 + 2x^2 - 1)\,dx$

   (c) $\int_0^{\pi} (3\cos x + 5)\,dx$

2. Evaluate the integrals in Exercise 1 using the basic single segment trapezoidal rule.
3. Evaluate the integrals in Exercise 1 using the basic Simpson's 1/3 rule.
4. Evaluate the integrals in Exercise 1 using the basic Simpson's 3/8 rule.
5. Evaluate the integrals in Exercise 1 using multiple application of the following rules with $n = 4$.
   (a) Trapezoidal rule
   (b) Simpson's 1/3 rule
   (c) Simpson's 3/8 rule
6. Use the trapezoidal rule with $n = 4$ to estimate

$$\int_0^1 \frac{dx}{1+x^2}$$

   correct to five decimal places.
7. Use Simpson's method with $n = 4$ to estimate

$$\int_0^1 \frac{dx}{1+x^2}$$

   correct to five decimal places. Compare this with the result obtained in Exercise 6. How do they compare with the correct answer 0.785398.
8. Estimate the following integrals by (a) trapezoidal method and (b) Simpson's 1/3 method using the given n:

   (a) $\int_1^3 \frac{dx}{x}$, $n = 2, 4, 8$

   (b) $\int_1^2 \frac{e^x dx}{x}$, $n = 4$

(c) $\int_1^5 e^{-x^2}\,dx, \qquad n = 8$

(d) $\int_0^4 \cos^2 x\,dx, \qquad n = 6$

(e) $\int_0^{\pi} \sqrt{1+3\cos^2 x}\,dx, \qquad n = 6$

(f) $\int_0^2 (e^{x^2}-1)\,dx, \qquad n = 8$

9. The table below shows the temperature $f(t)$ as a function of time.

Time, t	1	2	3	4	5	6	7
Temperature, $f(t)$	81	75	80	83	78	70	60

(a) Use Simpson's 1/3 method to estimate

$$\int_1^7 f(t)\,dt$$

(b) Use the result in (a) to estimate the average temperature.

10. Use Romberg integration to evaluate

(a) $\int_0^{\pi/2} \frac{\cos x}{\sqrt{1+\sin x}}\,dx$

(b) $\int_0^{3\pi/2} e^x \sin x\,dx$

(c) $\int_1^e \frac{\sqrt{\ln x}}{x}\,dx$

(d) $\int_0^{2\pi} (5+2\sin x)\,dx$

(e) $\int_0^2 (e^{x^2}-1)\,dx$

11. Prove that if $f''(x) > 0$ and $a \le x \le b$, the value of the integral

$$\int_a^b (fxt)\,dx$$

by the trapezoidal rule will always be greater than the exact value of the integral. Verify your conclusion for the following functions:

(a) $f(x) = 2x^3$

(b) $f(x) = 1 + x + x^2$

12. The table below shows the speed of a car at various intervals of time. Find the distance travelled by the car at the end of 2 hours

Time, hr	0	0.5	1.0	1.5	2.0	2.5
Speed, km/hr	0	40	60	50	45	65

13. The velocity of a particle is governed by the law

$$v(t) = \frac{\sin(t)}{(t+1)^2 \exp(t)}$$

If the initial position of the particle is $x(0) = 0$, then estimate the position $x(2)$ using the integral

$$x(t) = \int_0^t v(t)\,dt$$

by applying a suitable Newton-Cotes formula.

14. Estimate the integral

$$I = \int_0^{10} \exp\left(\frac{-1}{1+x^2}\right) dx$$

by Gauss qudrature, with $n = 2$, 3, and 4.

15. Evaluate the integral

$$I = \int_0^{\pi/2} (1 - 0.25 \sin^2 x)^{1/2}\,dx$$

using Gaussian quadrature. Assume a suitable value of n.

16. In an electric circuit, the voltage across the capacitor is given by

$$v(T) = \frac{1}{C}\int_0^T i(t)\,dt \text{ volts}$$

$$= \frac{10}{C}\int_0^T \sin^2\left(\frac{t}{\pi}\right) \text{volts}$$

Assuming $C = 5F$, compute the value of voltage $v(T)$ for $T = 1$, 2, and 3 seconds.

17. The circumference of an ellipse is given by

$$\text{Circumference} = 4a\int_0^{\frac{\pi}{2}} \sqrt{1 - \left[\frac{(a+b)(a-b)}{a^2}\right]\sin^2\theta}\,d\theta$$

where $2a$ is the length of major axis and $2b$ is the length of minor axis. Find the circumference if $a = 30$ metres and $b = 20$ metres.

18. The viscous resistance of an object moving through a fluid with velocity v is given by

$$R = -v^{3/2}$$

The velocity is decreasing with time t. The time taken for the velocity to decrease from v_0 to v_1 is given by

$$T = \int_{v_0}^{v_1} \frac{m}{R}\,dv \text{ seconds} \qquad = -\int_{v_0}^{v_1} \frac{m}{v^{3/2}}\,dv \text{ seconds}$$

where m is the mass of the object. Estimate the time T required for an object with $m = 30$ kg to reach a velocity of 10 m/sec from an initial velocity of 20 m/sec.

PROGRAMMING PROJECTS

1. Write a modular program TRAPE2 that uses the following subprograms as modules to evaluate integrals using the composite trapezoidal rules.
   (a) Input module (Module1)
   (b) Evaluation module (Module2)
   (c) Output module (Module3)

   Use a function subprogram to evaluate the given function.
2. Develop a modular program SIMS2 (similar to TRAPE2) to evaluate integrals using multiple application (i.e. composite) of Simpson's 3/8 rule.
3. Modify the program SIMPS1 to include a test to determine whether n is even and stop the execution if n is odd.
4. Write a program that uses two-point Gaussian quadrature to estimate the integral

$$\int_a^b f(x)\,dx$$

Split the interval into n equal subintervals and apply the quadrature rule to each subinterval.

5. Show that the integral

$$I_n = \int_0^1 x^2\, e^{x-1}\,dx, \qquad n = 1, 2, \ldots$$

can be evaluated recursively by

$$I_n = 1 - n\, I_{n-1} \qquad n = 2, 3, \ldots$$

$$I_1 = \frac{1}{e}$$

Write a program to evaluate I_{10} using this recursive formula and compare the results by Simpson's rule.

6. Write a program to evaluate the integral of a table of points using Newton's three-eighth's rule.
7. Write a program to experiment with the integration problem

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

to see if you can determine an optimum value for h using Simpson's rule.

Note: The trapejoidal rule and the Simpson's 1/3 rule can be used to evaluate the integral of a table of points. Development of FORTRAN programs is left as an exercise to the readers. (C programs for evaluating the integral of tabulated data are given in the Appendix D).

CHAPTER 13

Numerical Solution of Ordinary Differential Equations

13.1 NEED AND SCOPE

Many of the laws in physics, chemistry, engineering, biology and economics are based on empirical observations that describe changes in the states of systems. Mathematical models that describe the state of such systems are often expressed in terms of not only certain system parameters but also their derivatives. Such mathematical models, which use differential calculus to express relationship between variables, are known as *differential equations*.

Examples of differential equations are many. A few of them are listed below to illustrate the nature of differential equations that occur in science and engineering.

1. Law of cooling

The Newton's law of cooling states that the rate of loss of heat from a liquid is proportional to the difference of temperatures between the liquid and the surroundings. This can be stated in mathematical form as

$$\frac{dT(t)}{dt} = k\,(T_s - T(t)) \tag{13.1}$$

where T_s is the temperature of surroundings, $T(t)$ is the temperature of liquid at time t and k is the constant of proportionality.

2. Law of motion

The law governing the velocity $v(t)$ of a moving body is given by

$$m\frac{dv(t)}{dt} = F \tag{13.2}$$

where m is the mass of the body and F is the force acting on it.

3. Kirchhoff's law for an electric circuit

The voltage across an electric circuit containing an inductance L and a resistance R is given by

$$L\frac{di}{dt} + iR = V \tag{13.3}$$

4. Radioactive decay

The radioactive decay of an element is given by

$$\frac{dm}{dt} - km = 0 \tag{13.4}$$

where m is the mass, t is time and k is the constant rate of decay.

5. Simple harmonic motion

The equation to describe a simple harmonic motion is given by

$$m\frac{d^2y}{dt^2} + a\frac{dy}{dt} + ky = 0 \tag{13.5}$$

where y denotes displacement and m is the mass. Note that d^2y/dt^2 represents acceleration and dy/dt represents velocity of the moving weight.

6. Force on a moving boat

When a boat moves through water, the retarding force is proportional to the square of the velocity. The acceleration is given by

$$\frac{dv}{dt} = -\frac{k}{m}v^2 \tag{13.6}$$

where m is the mass and k is the drag coefficient.

7. Heat flow in a rectangular plate

The model for heat flow in a rectangle plate that is heated is given by

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{dy^2} = f(x, y) \tag{13.7}$$

where $u(x, y)$ denotes the temperature at point (x, y) and $f(x, y)$ is the heat source.

Note that all these examples contain the rate of change of a variable expressed as a function of variables and parameters. Although most of the differential equations may be solved analytically in their simplest form, analytical techniques fail when the models are modified to take into account the effect of other conditions of real-life situations. In all such cases, numerical approximation of the solution may be considered as a possible approach.

The main concern of this chapter is to present various methods of numerical solution of a class of differential equations known as *ordinary differential equations*. This implies that the differential equations may appear in different forms. This is evident from the examples. We, therefore, define some important classes of differential equations before considering the numerical solution of ordinary differential equations.

Number of Independent Variables

The quantity being differentiated is called the *dependent variable* and the quantity with respect to which the dependent variable is differentiated is called *independent variable*. If there is only one independent variable, the equation is called an *ordinary differential equation*. If it contains two or more independent variables, the derivatives will be partial and, therefore, the equation is called a *partial differential equation*. Eqs. (13.1) to (13.6) belong to the class of ordinary differential equations while Eq. (13.7) is a partial differential equation. In this chapter we shall consider only the ordinary differential equations.

Order of Equations

Differential equations are also classified according to their order. The order of a differential equation is the highest derivative that appears in the equation. When the equation contains only a first derivative, it is called a *first-order differential equation*. For example, Eqs (13.1) to (13.4) are first-order equations. On the other hand, if the highest derivative is a second derivative, the equation is called a *second-order differential equation*. For example, Eq. (13.5) is a second-order equation.

A first-order equation can be expressed in the form

$$\frac{dy}{dx} = f(x, y) \tag{13.8}$$

A second-order equation can be expressed in the form

$$y'' = f(x, y, y') \tag{13.9}$$

where y'' denotes the second derivative and y' is the first derivative. Higher-order equations can be reduced to a set of first-order equations by suitable transformations. For example, the equation

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

can be equivalently represented by

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

$$y' = u$$

Degree of Equations

Sometimes, the equations are referred to by their degree. The degree of

a differential equation is the power of the highest-order derivative. For example,

$$xy'' + y^2y' = 2y + 3$$

is a first-degree, second-order equation while,

$$(y''')^2 + 5y' = 0$$

is a second-degree, third-order equation.

Linear and Nonlinear Equations

A differential equation is known as a *linear* equation when it does not contain terms involving the products of the dependent variable or its derivatives. For example,

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

is a second-order, linear equation. The equations

$$y'' + (y')^2 = 1$$

$$y' = -ay^2$$

are nonlinear because the first one contains a product of y' and the second contains a product of y.

General and Particular Solutions

A solution to a differential equation is a relationship between the dependent and independent variables that satisfy the differential equation. For example,

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

is the solution of

$$y' = 6x + 1$$

Similarly,

$$y = e^x$$

is the solution of

$$y'' = y$$

Note that each of the solutions given above is only one of an infinite number of solutions. For example,

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

$$y = 3x^2 + x - 10$$

are also solutions of $y' = 6x + 1$. In general, $y' = 6x + 1$ has a solution of the form

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

where c is known as the constant of integration. Similarly, $y' = y$ has a solution of the form $y = ae^x$. The solution that contains arbitrary constants is not unique and is therefore known as the *general solution*.

If the values of the constants are known, then, on substitution of these values in the general solution, a unique solution known as *particular solution* can be obtained.

Initial Value Problems

In order to obtain the values of the integration constants, we need additional information. For example, consider the solution $y = ae^x$ to the equation $y' = y$. If we are given a value of y for some x, the constant a can be determined. Suppose $y = 1$ at $x = 0$, then,

$$y(0) = ae^0 = 1$$

Therefore,

$$a = 1$$

and the particular solution is

$$y = e^x$$

If the order of the equation is n, we will have to obtain n constants and, therefore, we need n conditions in order to obtain a unique solution. When all the conditions are specified at a particular value of the independent variable x, then the problem is called an *initial-value problem*.

It is also possible to specify the conditions at different values of the independent variable. Such problems are called the *boundary-value problems*. For example, if, instead of specifying only $y(0) = 1$, we also specify $y(0) + y(1) = 2$, then the problem will be a boundary-value problem. In this case,

$$y(0) + y(1) = a(1 + e) = 2$$

giving

$$a = 2/(1 + e)$$

One-step and Multistep Methods

All numerical techniques for solving differential equations involve a series of estimates of $y(x)$ starting from the given conditions. There are two basic approaches that could be used to estimate the values of $y(x)$. They are known as *one-step methods* and *multistep methods*.

In one-step methods, we use information from only one preceding point, i.e. to estimate the value y_i, we need the conditions at the previous point y_{i-1} only. Multistep methods use information at two or more previous steps to estimate a value.

Scope

In this chapter, we mainly concentrate on the solution of ordinary differential equations and discuss the following methods:

1. Taylor series method
2. Euler's method
3. Heun's method

4. Polygon method
5. Runge-Kutta method
6. Milne-Simpson method
7. Adams-Bashforth-Moulton method

The initial-value problem of an ordinary first-order differential equation has the form

$$\boxed{y'(x) = f(x, y(x)),\ y(x_0) = y_0} \tag{13.10}$$

In this chapter, we determine the solution of this equation on a finite interval (x_0, b), starting with the initial point x_0. For the sake of simplicity, in a number of places, we use f for $f(x, y)$, y for $y(x)$ and $y^{(k)}$ for $y^{(k)}(x)$.

TAYLOR SERIES METHOD

We can expand a function $y(x)$ about a point $x = x_0$ using Taylor's theorem of expansion

$$y(x) = y(x_0) + (x - x_0)\, y'(x_0) + (x - x_0)^2 \frac{y''(x_0)}{2!} + \ldots + (x - x_0)^n \frac{y^n(x_0)}{n!} \tag{13.11}$$

where $y^i(x_0)$ is the ith derivative of $y(x)$, evaluated at $x = x_0$. The value of $y(x)$ can be obtained if we know the values of its derivatives. This implies that if we are given the equation

$$y' = f(x, y) \tag{13.12}$$

we must then repeatedly differentiate $f(x, y)$ implicitly with respect to x and evaluate them at x_0.

For example, if $y' = f(x, y)$ then

$$y'' = \frac{d}{dx}\left(\frac{dy}{dx}\right) = \frac{d}{dx}[f(x, y)]$$

$$= \frac{\partial}{\partial x}[f(x, y)] + \frac{\partial}{\partial y}[f(x, y)]\frac{dy}{dx}$$

$$= \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f = f_x + f \times f_y \tag{13.13}$$

where f denotes the function $f(x, y)$ and f_x and f_y denote the partial derivatives of the function $f(x, y)$ with respect to x and y, respectively. Similarly, we can obtain

$$y''' = f_{xx} + 2f f_{xy} + f^2 f_{yy} + f_x f_y + f f_y^2 \tag{13.14}$$

Let us illustrate this through an example.

Consider the equation

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

under the condition $y(x) = 1$ when $x = 0$,

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

$$y'' = 2x + 2yy'$$

$$y''' = 2 + 2yy'' + 2(y')^2$$

at $x = 0$, $y(0) = 1$ and, therefore,

$$y'(0) = 1$$

$$y''(0) = 2$$

$$y'''(0) = 2 + (2)(1)(2) + (2)(1)^2 = 8$$

Substituting these values, the Taylor series becomes

$$y(x) = 1 + x + x^2 + \frac{8}{3!}x^3 + \ldots \tag{13.15}$$

The number of terms to be used depends on the accuracy of the solution needed.

Example 13.1

Use the Taylor method to solve the equation

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

for $x = 0.25$ and $x = 0.5$ given $y(0) = 1$

The solution of this equation is given by Eq. (13.15). That is,

$$y(x) = 1 + x + x^2 + 8\frac{x^3}{3!} + \ldots$$

Therefore,

$$y(0.25) = 1 + 0.25 + (0.25)^2 + \frac{8}{6}(0.25)^3 + \ldots$$

$$= 1.33333$$

Similarly,

$$y(0.5) = 1 + 0.5 + 0.5^2 + \frac{8}{6}(0.5)^3 + \ldots$$

$$= 1.81667$$

Improving Accuracy

The error in Taylor method is in the order of $(x - x_0)^{n+1}$. If $|x - x_0|$ is large, the error can also become large. Therefore, the result of this method in the interval (x_0, b) when $(b - x_0)$ is large, is often found unsatisfactory.

The accuracy can be improved by dividing the entire interval into subintervals (x_0, x_1), (x_1, x_2), (x_2, x_3), ... of equal length and computing $y(x_i)$, $i = 1, 2, ..., n$ successively, using the Taylor series expansion. Here, $y(x_i)$ is used as an initial condition for computing $y(x_{i+1})$. Thus,

$$y(x_{i+1}) = y(x_i) + \frac{y(x_i)}{1!}(x_{i+1} - x_i) + \frac{y''(x_i)}{2!}(x_{i+1} - x_i)^2 + \ldots$$

$$\ldots + \frac{y^{(m)}(x_i)}{m!}(x_{i+1} - x_i)^m \qquad (13.16)$$

If we denote the size of each subinterval as h, then,

$$x_{i+1} - x_i = h \qquad \text{for } i = 0, 1, ..., n-1$$

and Eq. (13.16) becomes

$$\boxed{y_{i+1} = y_i + \frac{y_i'}{1!}h + \frac{y_i''}{2!}h^2 + \ldots + \frac{y_i^{(m)}}{m!}h^m} \qquad (13.17)$$

The derivatives $y_i^{(k)}$ are determined using Eq. (13.12), (13.13) and (13.14) at $x = x_i$ and $y = y_i$. This formula can be used recursively to obtain y_i values.

Example 13.2

Use the Taylor method recursively to solve the equation

$$y' = x^2 + y^2, \qquad y(0) = 0$$

for the interval (0, 0.4) using two subintervals of size 0.2.

The derivatives of y are given by

$$y' = x^2 + y^2$$
$$y'' = 2x + 2yy'$$
$$y''' = 2 + 2(y')^2 + 2yy''$$
$$y^{(4)} = 6y'y'' + 2yy'''$$

Iteration 1

$$y_1 = y_0 + \frac{y_0'}{1!}h + \frac{y_0''}{2!}h^2 + \frac{y_0'''}{3!}h^3 + \frac{y^{(4)}}{4!}h^4 + \ldots$$

$$h = 0.2, \; y_0 = y(0) = 0$$

$$y_0' = y'(0) = 0 + y(0)^2 = 0$$

$$y_0'' = y''(0) = 2 \times 0 + 2 \times y(0) \times y'(0) = 0$$

Similarly,

$$y_0''' = 2$$

$$y_0^{(4)} = 0$$

Therefore,

$$y_1 = 0 + 0 + 0 + \frac{2}{3!}(0.2)^3 + 0$$

$$= 0.002667 \qquad (\text{at } x = 0.2)$$

Iteration 2

$$x_1 = 0.2$$

$$y_1 = 0.002667$$

$$y_1' = x_1^2 + y_1^2 = (0.2)^2 + (0.002667)^2 = 0.04$$

$$y_1'' = 2x_1 + 2y_1 y_1'$$

$$= 2(0.2) + 2(0.002667)\,(0.04)$$

$$= 0.400213$$

$$y_1''' = 2 + 2(y_1')^2 + 2y_1 y_1''$$

$$= 2 + 2(0.04)^2 + 2(0.002667)\,(0.400213)$$

$$= 2.005335$$

$$y_1^{(4)} = 6y_1'\, y_1'' + 2y_1 y_1'''$$

$$= 6(0.04)\,(0.400213) + 2(0.002667)\,(2.005335)$$

$$= 0.106748$$

$$y_2 = y_1 + y_1' h + \frac{y_1''}{2} h^2 + \frac{y_1'''}{6} h^3 + \frac{y^{(4)}}{24} h^4$$

$$= 0.002667 + 0.04(0.2) + \frac{0.400213}{2}(0.2)^2$$

$$+ \frac{2.005335}{6}(0.2)^3 + \frac{0.106748}{24}(0.2)^4$$

$$= 0.021352$$

That is

$$y(0.4) = 0.021352$$

If we use $h = (b - x_0) = 0.4$ (without subdividing), we obtain

$$y(0.4) = \frac{2}{6}(0.4)^3 = 0.021333$$

The correct answer to the accuracy shown is $y(0.4) = 0.021359$. It shows that the accuracy has been improved by using subintervals. The accuracy can be further improved by reducing h further, say, $h = 0.1$.

One major problem with the Taylor series method is the evaluation of higher-order derivatives. They become very complicated. All these derivatives must be evaluated at (x_i, y_i), $i = 0, 1, 2, \ldots$. This method is, therefore, generally impractical from a computational point of view. However, it illustrates the basic approach to numerical solution of differential equations.

Picard's Method

Consider the differential equation

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

We can integrate this to obtain the solution in the interval (x_0, x)

$$\int_{x_0}^{x} dy = \int_{x_0}^{x} f(x, y)\,dx$$

or

$$y(x) = y(x_0) + \int_{x_0}^{x} f(x, y)\,dx$$

Since y appears under the integral sign on the right, the integration cannot be formed. The dependent variable y should be replaced by either a constant or a function of x. Since we know the initial value of y(at $x = x_0$), we may use this as a first appropriation to the solution and the result can be used on the right-hand side to obtain the next appropriation. The iterative equation is written as

$$y^{i+1} = y_0 + \int_{x_0}^{x} f(x, y^{(i)})\,dx \qquad (13.18)$$

Equation (13.18) is known as *Picard's method*. Since this method involves actual integration, sometimes it may not be possible to carry out the integration. Example 13.3 illustrates the application of Picard's method.

It can be seen that Picard's method is not convenient for computer-based solutions. Like Taylor's series method, this is also a semi-numeric method.

Example 13.3

Solve the following equations by Picard's method

(i) $y'(x) = x^2 + y^2$, $\quad y(0) = 0$

(ii) $y'(x) = xe^y$, $\quad y(0) = 0$

and estimate $y(0.1)$, $y(0.2)$ and $y(1)$

(i) $$y'(x) = x^2 + y^2$$

$$y_0 = 0, \quad x_0 = 0$$

$$y^{(1)} = y_0 + \int_{x_0}^{x} (x^2 + (y^0)^2)\,dx$$

$$= 0 + \int_0^x x^2\,dx = \frac{x^3}{3}$$

$$y^{(2)} = 0 + \int_{x_0}^{x} (x^2 + (y^1)^2)\,dx$$

$$= \int_0^x \left(x^2 + \frac{x^6}{9}\right) dx = \frac{x^3}{3} + \frac{x^7}{63}$$

This process can be continued further although it may be a difficult task. If we stop at $y^{(2)}$, then

$$y(x) = \frac{x^3}{3} + \frac{x^7}{63}$$

$$y(0.1) = 0.00003333$$

$$y(0.2) = 0.0026667$$

$$y(1) = 0.3492063$$

(ii) $$y'(x) = xe^y$$

$$y_0 = 0, \quad x_0 = 0$$

$$y^{(1)} = 0 + \int_0^x xe^0\,dx = \frac{x^2}{2}$$

$$y^{(2)} = 0 + \int_0^x xe^{(x^2/2)}\,dx = e^{(x^2/2)} - 1$$

Note that further integrations will become more difficult and even impossible. Now, let us assume

$$y(x) = y^{(2)} = e^{(x^2/2)} - 1$$

$$y(0.1) = 0.0050125$$

$$y(0.2) = 0.0202013$$

$$y(1) = 0.6487213$$

We know that the exact solution of $y'(x) = xe^y$ is

$$y(x) = -\ln\left(1 - \frac{x^2}{2}\right)$$

Therefore

$$y(0.1)_{\text{exact}} = 0.0050125$$
$$y(0.2)_{\text{exact}} = 0.0202027$$
$$y(1)_{\text{exact}} = 0.6933147$$

Note that the error increases when $(x - x_0)$ increases. Better accuracy can be achieved by using the new initial value i.e. $y(0.1)$ can be used as the initial value for computing $y(0.2)$, instead of $y(0)$.

13.3 EULER'S METHOD

Euler's method is the simplest one-step method and has a limited application because of its low accuracy. However, it is discussed here as it serves as a starting point for all other advanced methods.

Consider the first two terms of the expansion (13.11)

$$y(x) = y(x_0) + y'(x_0)(x - x_0)$$

Given the differential equation

$$y'(x) = f(x, y) \text{ with } y(x_0) = y_0$$

we have

$$y'(x_0) = f(x_0, y_0)$$

and therefore

$$y(x) = y(x_0) + (x - x_0) f(x_0, y_0)$$

Then, the value of $y(x)$ at $x = x_1$ is given by

$$y(x_1) = y(x_0) + (x_1 - x_0) f(x_0, y_0)$$

Letting $h = x_1 - x_0$, we obtain

$$y_1 = y_0 + h f(x_0, y_0)$$

Similarly, $y(x)$ at $x = x_2$ is given by

$$y_2 = y_1 + h f(x_1, y_1)$$

In general, we obtain a recursive relation as

$$\boxed{y_{i+1} = y_i + h f(x_i, y_i)} \tag{13.19}$$

This formula is known as *Euler's method* and can be used recursively to evaluate $y_1, y_2, \ldots$ of $y(x_1), y(x_2), \ldots$, starting from the initial condition $y_0 = y(x_0)$. Note that this does not involve any derivatives.

A new value of y is estimated using the previous value of y as the initial condition. Note that the term $h f(x_i, y_i)$ represents the incremental value of y and $f(x_i, y_i)$ is the slope of $y(x)$ at (x_i, y_i), i.e. the new value is obtained by extrapolating linearly over the step size h using the slope at its previous value. That is

New value = old value + slope × step size

This is illustrated in Fig. 13.1. Remember that y_1 approximates $y(x_1)$ and y_2 approximates $y(x_2)$. The difference between them is the error introduced by the method.

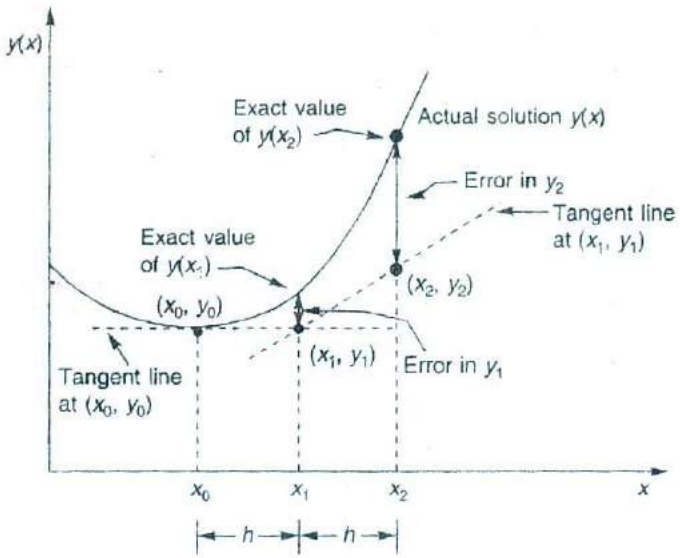


Fig. 13.1 Illustration of Euler's method for two steps

Example 13.4

Given the equation

$$\frac{dy}{dx} = 3x^2 + 1 \qquad \text{with } y(1) = 2$$

estimate $y(2)$ by Euler's method using (i) $h = 0.5$ and (ii) $h = 0.25$.

(i) $h = 0.5$

$$y(1) = 2$$
$$y(1.5) = 2 + 0.5[3(1.0)^2 + 1] = 4.0$$
$$y(2.0) = 4.0 + 0.5[3(1.5)^2 + 1] = 7.875$$

(ii) $h = 0.25$

$$y(1) = 2$$
$$y(1.25) = 2 + 0.25[3(1)^2 + 1] = 3.0$$
$$y(1.5) = 3 + 0.25[3(1.25)^2 + 1] = 5.42188$$
$$y(1.75) = 5.42188 + 0.25[3(1.5)^2 + 1] = 7.35938$$
$$y(2.0) = 7.35938 + 0.25[3(1.75)^2 + 1] = 9.90626$$

Notice the difference in answers of $y(2)$ in these two cases. The accuracy is improved considerably when h is reduced to 0.25 (true answer is 10.0).

Accuracy of Euler's Method

As usual, the accuracy is affected by two sources of error, namely, round-off error and truncation error. Roundoff error is always present in a computation and this can be minimised by increasing the precision of calculations.

The major cause is loss of accuracy is truncation error. This arises because of the use of a truncated Taylor series. Since Euler's method uses Taylor series iteratively, the truncation error introduced in an iteration is propagated to the following iterations. This means the total truncation error in any iteration step will consist of two components—the propagated truncation error and the truncation error introduced by the step itself.

The truncation introduced by the step itself is known as the *local truncation error* and the sum of the propagated error and the local error is called the *global truncation error*. Recall the Taylor series expansion used in Euler's method for estimating the values of y_i.

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

Since only the first two terms are used in Euler's formula, the local truncation error is given by

$$E_{t,i+1} = \frac{y_i''}{2} h^2 + \frac{y_i'''}{3!} h^3 + \frac{y_i^{(4)}}{4!} h^4 + \ldots$$

If the step size h is very small, the higher-order terms may be neglected and therefore

$$E_{t,i+1} = \frac{y_i''}{2} h^2$$

The above analysis assumes that the function $y' = f(x, y)$ has continuous derivatives. The local truncation error of Euler's method is of the order h^2. If the final estimation requires n steps, the total (global) truncation error at the target point b will be

$$|E_{tg}| = \sum_{i=1}^{n} c_i h^2 = (c_1 + c_2 + \ldots + c_n)h^2 = nch^2$$

where

$$c = (c_1 + c_2 + \ldots + c_n)/n$$

Since

$$n = (b - x_0)/h,$$

$$|E_{tg}| = (b - x_0)ch$$

Example 13.5

Compute the errors in the estimates of Example (13.4) when $h = 0.5$

Given

$$y' = 3x^2 + 1$$
$$y'' = 6x$$
$$y''' = 6$$

Step 1

$$x_0 = 1, \qquad y_0 = 2$$
$$y_1 = y(1.5) = 4.0 \qquad \text{(from example (13.4))}$$

$$E_{t,1} = \frac{y_0''}{2} h^2 + \frac{y_0'''}{6} h^3$$

$$= \frac{6(1)}{2}(0.5)^2 + \frac{6}{6}(0.5)^3 = 0.875$$

Step 2

$$x_1 = 1.5, \qquad y_1 = 4.0$$
$$y_2 = y(2.0) = 7.875 \qquad \text{(from example (13.4))}$$

$$E_{t,2} = \frac{6(1.5)}{2}(0.5)^2 + \frac{6}{6}(0.5)^3 = 1.25$$

Exact solution is

$$y(x) = x^3 + x$$

and therefore

$$\text{True } y(1.5) = 4.875$$
$$\text{True } y(2.0) = 10.000$$

The estimated and true values of y and corresponding errors are tabulated below.

x	*Estimated* y	*True* y	E_t	*Global error*
1.5	4.0	4.875	0.875	0.875
2.0	7.875	10.000	1.250	2.125

Note that the local and global errors are equal at the first step and they are different in the second step. The difference between them is the propagated truncation error that results from the first step.

Observe that

$$c_1(0.5)^2 = 0.875 \qquad \text{since } c_1 = 3.5$$
$$c_2(0.5)^2 = 1.250 \qquad \text{since } c_2 = 5.0$$
$$c = (c_1 + c_2)/2 = 4.25$$
$$E_{tg} = c(b - x_0)h = 4.25 \times 1 \times 0.5 = 2.125$$

This confirms the error equations for local and global truncation errors discussed in this section.

Program EULER

The program EULER estimates the solution of the first order differential equation $y' = f(x, y)$ at a given point using Euler's method (Eq. 13.19). The program is simple and self-explanatory. Note the use of an intrinsic function INT in the line

```
N = INT(XP - X)/(H + 0.5)
```

Given initial value of X and the point of solution XP, this statement computes the number of steps required for evaluation using the specified step-size H. The function INT returns the nearest integer of the real value

$$\frac{XP - X}{H}$$

```
*-----------------------------------------------------*
      PROGRAM EULER
*-----------------------------------------------------*
* Main program                                        *
*    This program estimates the solution of the first *
*    order differential equation y' = f(x, y) at a    *
*    given point using Euler's method                 *
*-----------------------------------------------------*
* Functions invoked                                   *
*      F, INT                                         *
*-----------------------------------------------------*
* Subroutines used                                    *
*      NIL                                            *
*-----------------------------------------------------*
* Variables used                                      *
*    X   - Initial value of independent variable      *
*    Y   - Initial value of dependent variable        *
*   XP   - Point of solution                          *
*    H   - Incremental step-size                      *
*    N   - Number of computational steps required     *
*   DY   - Incremental Y in each step                 *
*-----------------------------------------------------*
* Constants used                                      *
*      NIL                                            *
*-----------------------------------------------------*
      REAL X,Y,XP,H,DY,F
      INTEGER N,INT
      EXTERNAL F
      INTRINSIC INT
```

```
      WRITE(*,*)
      WRITE(*,*)' SOLUTION BY EULERS METHOD'
      WRITE(*,*)
C* Read values
      WRITE(*,*)  'Input initial values of x and y'
      READ(*,*) X,Y
      WRITE(*,*)  'Input x at which y is required'
      READ(*,*) XP
      WRITE(*,*)  'Input step-size h'
      READ(*,*) H
C* Compute number of steps required
      N = INT((XP-X)/H+0.5)
C* Compute Y recursively at each step
      DO 10 I = 1,N
         DY = H*F(X,Y)
         X = X+H
         Y = Y+DY
         WRITE(*,*) I,X,Y
10    CONTINUE
C* write the final result
      WRITE(*,*)
      WRITE(*,*)  'Value of Y at X  =',X,' is', Y
      WRITE(*,*)
      STOP
      END
C* ------------- End of main EULER ------------------*
C* ------------------------------------------------- *
C* Function subprogram                               *
C* ------------------------------------------------- *
      REAL FUNCTION F(X,Y)
      REAL X,Y
      F = 2.0 * Y/X
      RETURN
      END
C* ----------- End of function F(X,Y) ---------------*
```

Test Run Results

```
                SOLUTION BY EULERS METHOD
Input initial values of x and y
1.0 2.0
Input x at which y is required
2.0
Input step-size h
0.25
          1        1.2500000        3.0000000
          2        1.5000000        4.2000000
          3        1.7500000        5.6000000
          4        2.0000000        7.2000000
Value of Y at X =      2.0000000 is      7.2000000
Stop - Program terminated.
```

13.4 HEUN'S METHOD

Euler's method is the simplest of all one-step methods. It does not require any differentiation and is easy to implement on computers. However, its major weakness is large truncation errors. This is due to its linear characteristic. Recall that Euler's method uses only the first two terms of the Taylor series. In this section, we shall consider an improvement to Euler's method.

In Euler's method, the slope at the beginning of the interval is used to extrapolate y_i to y_{i+1} over the entire interval. Thus,

$$y_{i+1} = y_i + m_1 h$$

where m_1 is the slope at (x_i, y_i). As illustrated in Fig. 13.2, y_{i+1} is clearly an underestimate of $y(x_{i+1})$.

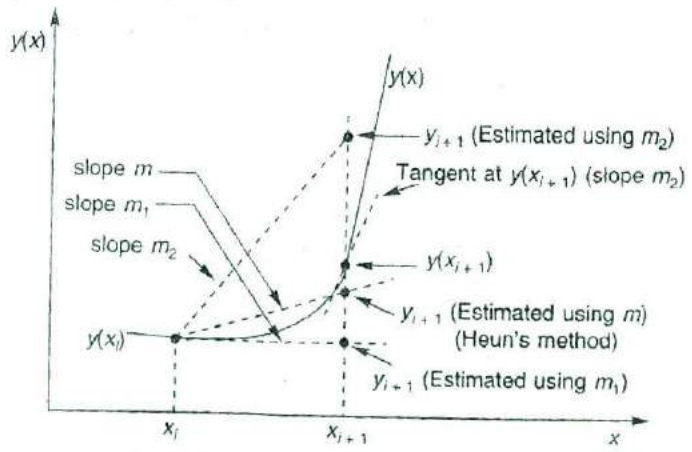


Fig. 13.2 Illustration of Heun's method

An alternative is to use the line which is parallel to the tangent at the point $(x_{i+1}, y(x_{i+1}))$ to extrapolate from y_i to y_{i+1} as shown in Fig. 13.2. That is

$$y_{i+1} = y_i + m_2 h$$

where m_2 is the slope at $(x_{i+1}, y(x_{i+1}))$. Note that the estimate appears to be overestimated.

A third approach is to use a line whose slope is the average of the slopes at the end points of the interval. Then

$$y_{i+1} = y_i + \frac{m_1 + m_2}{2} h \tag{13.20}$$

As shown in Fig. 13.2, this gives a better approximation to y_{i+1}. This approach is known as *Heun's method*.

The formula for implementing Heun's method can be constructed easily. Given the equation

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

we can obtain

$$m_1 = y'(x_i) \quad = f(x_i, y_i)$$
$$m_2 = y'(x_{i+1}) = f(x_{i+1}, y_{i+1})$$

and therefore

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

Equation (13.20) becomes

$$\boxed{y_{i+1} = y_i + \frac{h}{2}\,[f(x_i, y_i) + f(x_{i+1}, y_{i+1})]} \tag{13.21}$$

Note that the term y_{i+1} appears on both sides of Eq. (13.21) and, therefore, y_{i+1} cannot be evaluated until the value of y_{i+1} inside the function $f(x_{i+1}, y_{i+1})$ is available. This value can be predicted using the Euler's formula as

$$\boxed{y_{i+1} = y_i + h \times f(x_i, y_i)} \tag{13.22}$$

Then, Heun's formula becomes

$$\boxed{y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+2}, y_{i+1}^e)]} \tag{13.23}$$

Equation (13.23) is an improved version of Euler's method. Since it attempts to correct the values of y_{i+1} using the predicted value of y_{i+1} (by Euler's method), it is classified as a *one-step predictor-corrector*

method. Eq. (13.22) is known as the *predictor* and Eq. (13.23) is known as the *corrector*. Substituting Eq. (13.22) into Eq. (13.23), we obtain

$$y_{i+1} = y_i + \frac{h}{2}\left[f(x_i, y_i) + f(x_{i+1}, y_i + h\, f(x_i, y_i))\right] \qquad (13.24)$$

Example 13.6

Given the equation

$$y'(x) = 2y/x \qquad \text{with } y(1) = 2$$

estimate $y(2)$ using (i) Euler's method, and (ii) Heun's method, using $h = 0.25$, and compare the results with exact answers.

Given

$$y' = f(x, y) = 2y/x$$
$$x_0 = 1, \qquad y_0 = 2, \qquad h = 0.25$$

(i) *Euler's method*

$$y(1.25) = y_1 = y_0 + h.\, f(x_0, y_0)$$
$$= 2 + 0.25\,\frac{2 \times 2}{1} = 3.00$$

$$y(1.5) = 3.0 + 0.25\,\frac{2 \times 3.0}{1.25} = 4.2$$

$$y(1.75) = 4.2 + 0.25\,\frac{2 \times 4.2}{1.5} = 5.6$$

$$y(2.0) = 5.6 + 0.25\,\frac{2 \times 5.6}{1.75} = 7.2$$

(ii) *Heun's method*

Iteration 1

$$m_1 = \frac{2 \times 2}{1} = 4.0$$
$$y_e(1.25) = 2 + 0.25(4.0) = 3.0$$
$$m_2 = \frac{2 \times 3.0}{1.25} = 4.8$$
$$y(1.25) = 2 + \frac{0.25}{2}(4.0 + 4.8) = 3.1$$

Iteration 2

$$m_1 = \frac{2 \times 3.1}{1.25} = 4.96$$
$$y_e(1.5) = 3.1 + 0.25(4.96) = 4.34$$

$$m_2 = \frac{2 \times 4.34}{1.5} = 4.8$$

$$y(1.5) = 3.1 + \frac{0.25}{2}(4.96 + 5.79) = 4.44$$

Iteration 3

$$m_1 = \frac{2 \times 4.44}{1.5} = 5.92$$

$$y_e(1.75) = 4.44 + 0.25(5.92) = 5.92$$

$$m_2 = \frac{2 \times 5.92}{1.75} = 6.77$$

$$y(1.75) = 4.44 + \frac{0.25}{2}(5.92 + 6.77) = 6.03$$

Iteration 4

$$m_1 = \frac{2 \times 6.03}{1.75} = 6.89$$

$$y_e(2.0) = 6.03 + 0.25(6.89) = 7.75$$

$$m_2 = \frac{2 \times 7.75}{2} = 7.75$$

$$y(2.0) = 6.03 + \frac{0.25}{2}(6.89 + 7.75) = 7.86$$

Exact solution of the equation

$$y'(x) = 2y/x \text{ with } y(1) = 2$$

is obtained as

$$y(x) = 2x^2$$

The exact values of $y(x)$ and the estimated values by both the methods are tabulated below.

x	$y(x)$		
	Euler's method	Heun's method	Analytical
1.00	2.00	2.00	2.00
1.25	3.00	3.10	3.125
1.50	4.20	4.44	4.50
1.75	5.60	6.03	6.125
2.00	7.20	7.86	8.00

All estimated values are accurate to two decimal places. It is clear that Heun's method provides better results compared to Euler's method.

Error Analysis

It can be easily shown that Heun's method is a second-order method

and, therefore, its local truncation is of the order h^3. Let us consider Eq. (13.24). Letting $i = 0$, for simplicity, we obtain

$$y_1 = y_0 + \frac{h}{2}[m_1 + f(x_0 + h, y_0 + m_1 h)] \tag{13.25}$$

Let us expand the term $f(x_{0+h}, y_0 + m_1 h)$ in a Taylor series form.

$$f(x_0 + h, y_0 + m_1 h) = f(x_0, y_0) + h\frac{\partial f}{\partial x} + m_1 h\frac{\partial f}{\partial y}$$

$$= m_1 + hf_x + m_1 hf_y$$

On substituting this into Eq. (13.25) we get

$$y_1 = y_0 + hm_1 + \frac{h^2}{2}(f_x + m_1 f_y) \tag{13.26}$$

We know that

$$m_1 = y'$$

$$(f_x + m_1 f_y) = y'' \text{ (see equation (13.13))}$$

and therefore Eq. (13.26) can be written as

$$y_1 = y_0 + \frac{y'}{1!}h + \frac{y''}{2!}h^2$$

This proves that Heun's method is of order h^2 and the local truncation error is of the order h^3. If the final estimate is obtained after n iterations, then the global truncation error is given by

$$|E_{tg}| = \sum_{i=1}^{n} c_i h^3 = nch^3$$

We know that

$$n = \frac{x_n - x_0}{h} = \frac{b - x_0}{h}$$

Therefore,

$$|E_{tg}| = (b - x_0)ch^2$$

That is, the global truncation error is of the order h^2.

Program HEUN

The algorithm of Heun's method is implemented by the program HEUN. Note that the algorithm is very similar to that of the Euler's method, except for the slope used in extrapolating the initial value.

```
*-----------------------------------------------------------*
    PROGRAM HEUN
*-----------------------------------------------------------*
* Main program                                              *
*    This program solves the first order differential       *
```

```
*     equation y' = f(x, y) using the Heun's method
* ------------------------------------------------------
* Functions invoked
*     F, INT
* ------------------------------------------------------
* Subroutines used
*     NIL
* ------------------------------------------------------
* Variables used
*     X - Initial value of independent variable
*     Y - Initial value of dependent variable
*    XP - Point of solution
*     H - Step-size
*     N - Number of steps
* ------------------------------------------------------
* Constants used
*     NIL
* ------------------------------------------------------
      REAL X,Y,XP,H,M1,M2,F
      INTEGER N,INT
      INTRINSIC INT
      EXTERNAL F

      WRITE(*,*)
      WRITE(*,*)'      SOLUTION BY HEUNS METHOD'
      WRITE(*,*)
* Input values
      WRITE(*,*) 'Input initial values of x and y'
      READ(*,*) X,Y
      WRITE(*,*) 'Input x at which y is required'
      READ(*,*) XP
      WRITE(*,*) 'Input step-size h'
      READ(*,*) H
* Compute number of steps required
      N = INT((XP-X)/H+0.5)
* Compute Y recursively at each step
      DO 20 I = 1,N
         M1 = F(X,Y)
         M2 = F(X+H,Y+M1*H)
         X = X+H
         Y = Y+0.5*H*(M1+M2)
         WRITE(*,*) I,X,Y
```

```
20    CONTINUE
* Write the final result
      WRITE(*,*)
      WRITE(*,*) 'Value of Y at X =', X, '  is', Y
      WRITE(*,*)
      STOP
      END
* ---------------- End of main HEUN ----------------- *
* ------------------------------------------------------ *
* Function subprogram                                    *
* ------------------------------------------------------ *
      REAL FUNCTION F(X,Y)
      REAL X,Y
      F = 2.0 * Y/X
      RETURN
      END
* -------------- End of function F(X,Y) --------------- *
```

Test Run Results

First run

```
                SOLUTION BY HEUNS METHOD
      Input initial values of x and y
      1.0 2.0
      Input x at which y is required
      2.0
      Input step-size h
      0.25
                1     1.2500000     3.1000000
                2     1.5000000     4.4433330
                3     1.7500000     6.0302380
                4     2.0000000     7.8608460
     Value of Y at X = 2.0000000 is     7.8608460
     Stop - Program terminated.
```

Second run

```
                SOLUTION BY HEUNS METHOD
      Input initial values of x and y
      1.0 2.0
      Input x at which y is required
      2.0
```

```
Input step-size h
0.125
          1     1.1250000     2.5277780
          2     1.2500000     3.1175920
          3     1.3750000     3.7694530
          4     1.5000000     4.4833640
          5     1.6250000     5.2593310
          6     1.7500000     6.0973560
          7     1.8750000     6.9974420
          8     2.0000000     7.9595900

Value of Y at X =   2.0000000 is7.9595900

Stop - Program terminated.
```

Third run

```
                SOLUTION BY HEUNS METHOD

Input initial values of x and y
1.0 2.0
Input x at which y is required
2.0
Input step-size h
0.1
          1     1.1000000     2.4181820
          2     1.2000000     2.8761710
          3     1.3000000     3.3739700
          4     1.4000000     3.9115800
          5     1.5000000     4.4890040
          6     1.6000000     5.1062420
          7     1.7000000     5.7632950
          8     1.8000000     6.4601640
          9     1.9000000     7.1968490
         10     2.0000000     7.9733510

Value of Y at X =     2.0000000 is 7.9733510

Stop - Program terminated.
```

13.5 POLYGON METHOD

Another modification of Euler's method is to use the slope of the function at the estimated midpoints of (x_i, y_i) and (x_{i+1}, y_{i+1}) to approximate y_{i+1}. Thus,

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

$$= y_i + f(x_i + h/2, y_i + \Delta y/2)h \qquad (13.27)$$

Δy is the estimated incremental value of y from y_i and can be obtained using Euler's formula as

$$\Delta y = h\, f(x_i, y_i)$$

Then, equation (13.27) can be written as

$$\begin{aligned} y_{i+1} &= y_i + hf\ (x_i + h/2,\ y_i + h/2\, f(x_i, y_i)) \\ &= y_i + hf\,(x_i + h/2,\ y_i + m_1 h/2) \\ &= y_i + m_2 h \end{aligned} \tag{13.28}$$

where

$$m_1 = f(x_i, y_i) \quad \text{and} \quad m_2 = f\left(x_i + \frac{h}{2},\ y_i + \frac{m_1 h}{2}\right)$$

Equation (13.28) is known as the *modified Euler's method* or *improved polygon method*. The method, also called the *midpoint method*, is illustrated in Fig. 13.3.

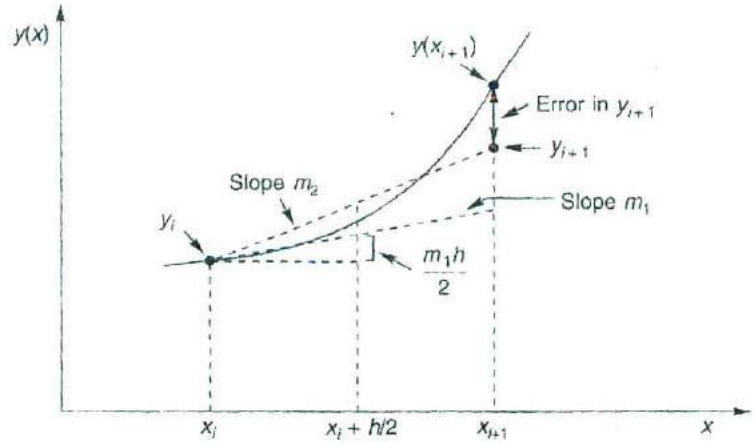


Fig. 13.3 Midpoint method

Like Heun's method, this method is also of the order h^2 and therefore, the local truncation error is of the order h^3 and the global truncation error is of the order h^2.

Example 13.7

Estimate $y(1.5)$ with $h = 0.25$ for the equation in Example 13.6 using polygon method.

$$y' = f(x, y) = \frac{2y}{x}$$

$$y(1) = 2.0$$

$$y(1.25) = 2.0 + 0.25\, f(1 + 0.125, 2 + 0.125 f(1, 2))$$

$$= 2.0 + 0.25\, f(1.125, 2.5) = 3.11$$

$$y(1.5) = 3.11 + 0.25\, f(1.25 + 0.125, 3.11 + 0.125\, f(1.25, 3.11))$$

$$= 3.11 + 0.25\, f(1.375, 3.732) = 4.47$$

Estimated values of $y(1.5)$ by various methods for the equation

$$y'(x) = 2y/x \quad \text{with } y(1) = 2.0$$

are given below:

Euler's method	:	4.20
Heun's method	:	4.44
Polygon method	:	4.47
Exact answer	:	4.50

Note that the polygon method yields better results compared to the Heun's method.

Program POLYGN

Program POLYGN implements the polygon algorithm to solve a differential equation of type $y' = f(x, y)$

```
*-----------------------------------------------------------*
      PROGRAM POLYGN
*-----------------------------------------------------------*
* Main program                                              *
*     This program solves the differential equation         *
*     of type y' = f(x, y) by polygon method                *
*-----------------------------------------------------------*
* Functions invoked                                         *
*     'F,INT                                                *
*-----------------------------------------------------------*
* Subroutines used                                          *
*     NIL                                                   *
*-----------------------------------------------------------*
* Variables used                                            *
*     X - Initial value of the independent variable         *
*     Y - Initial value of the dependent variable           *
*    XP - Point of solution                                 *
*     H - Incremental step-size                             *
*     N - Number of computational steps required            *
*-----------------------------------------------------------*
* Constants used                                            *
*     NIL                                                   *
*-----------------------------------------------------------*
```

```
      REAL X,Y,XP,H,M1,M2,F
      INTEGER N,INT
      INTRINSIC INT
      EXTERNAL F

      WRITE(*,*)
      WRITE(*,*)'          SOLUTION BY POLYGON METHOD'
      WRITE(*,*)

* Input values

      WRITE(*,*) 'Input initial values of x and y'
      READ(*,*) X,Y
      WRITE(*,*) 'Input x at which y is required'
      READ(*,*) XP
      WRITE(*,*) 'Input step-size h'
      READ(*,*) H

* Compute number of steps required

      N = INT((XP-X)/H+0.5)

* Compute Y at each step

      DO 30 I = 1,N
         M1 = F(X,Y)
         M2 = F(X+0.5*H,Y+0.5*H*M1)
         X = X+H
         Y = Y+M2*H
         WRITE(*,*) I,X,Y
30    CONTINUE

* Write the final value of Y

      WRITE(*,*)
      WRITE(*,*) 'Value of Y at X  =', X, '  is', Y
      WRITE(*,*)

      STOP
      END

* ---------------- End of main POLYGN ----------------- *
* ------------------------------------------------------ *
* Function subprogram                                    *
* ------------------------------------------------------ *

      REAL FUNCTION F(X,Y)
      REAL X,Y

      F = 2.0 * Y/X

      RETURN
      END
* ---------------- End of function F(X,Y) -------------- *
```

Test Run Results

```
          SOLUTION BY POLYGON METHOD
Input initial values of x and y
1.0 2.0
Input x at which y is required
2.0
Input step-size h
0.25
          1      1.2500000      3.1111110
          2      1.5000000      4.4686870
          3      1.7500000      6.0728310
          4      2.0000000      7.9235990

Value of Y at X =    2.0000000 is    7.9235990

Stop - Program terminated.
```

13.6 RUNGE-KUTTA METHODS

Runge-Kutta methods refer to a family of one-step methods used for numerical solution of initial value problems. They are all based on the general form of the extrapolation equation,

$$y_{i+1} = y_i + \text{slope} \times \text{interval size}$$
$$= y_i + mh$$

where m represents the slope that is weighted averages of the slopes at various points in the interval h. If we estimate m using slopes at r points in the interval (x_i, x_{i+1}), then m can be written as

$$m = w_1 m_1 + w_2 m_2 + \ldots + w_r m_r \tag{13.29}$$

where $w_1, w_2, \ldots, w_r$ are weights of the slopes at various points. The slopes $m_1, m_2, \ldots, m_r$ are computed as follows:

$$m_1 = f(x_i, y_i)$$
$$m_2 = f(x_i + a_1 h, y_i + b_{11} m_1 h)$$
$$m_3 = f(x_i + a_2 h, y_i + b_{21} m_1 h + b_{22} m_2 h)$$
$$\vdots$$
$$m_r = f(x_i + a_{r-1} h, y_i + b_{r-1,1} m_1 h + \ldots + b_{r-1,r-1} m_{r-1} h)$$

That is

$$m_1 = f(x_i, y_i) \qquad r = 1$$

$$m_r = f\left(x_i + a_{r-1} h, y_i + h \sum_{j=1}^{r-1} b_{r-1,j} m_j\right) \quad r \geq 2 \tag{13.30}$$

Note that the computation of slope at any point involves the slopes at all previous points. Slopes can be computed recursively using equation (13.30) starting from $m_1 = f(x_i, y_i)$.

Runge-Kutta (RK) methods are known by their order. For instance, an RK method is called the *r-order Runge-Kutta method* when slopes at r points are used to compute the weighted average slope m. Since Euler's method uses the slope at (x_i, y_i) to extrapolate linearly over the interval, therefore, Euler's method is a *first-order Runge-Kutta method*. Similarly, Heun's method is a *second-order Runge-Kutta method* because it employs slopes at two end points of the interval.

As it was demonstrated by the Euler and Heun methods, higher the order, better would be the accuracy of estimates. Therefore, selection of order for using RK methods depends on the problem under consideration.

Determination of Weights

For using a Runge-Kutta method, the first requirement is the determination of weights of slopes at various points. The number of points is equal to the order of the method chosen. For the purpose of demonstration, we consider here the second-order Runge-Kutta method and show how to evaluate various constants and weights.

The second-order RK method has the form

$$y_{i+1} = y_i + (w_1 m_1 + w_2 m_2)h \tag{13.31}$$

where

$$m_1 = f(x_i, y_i)$$

$$m_2 = f(x_i + a_1 h, y_i + b_{11}\, m_1 h) \tag{13.32}$$

The weights w_1 and w_2 and the constants a_1 and b_{11} are to be determined. The centre principle of the Runge-Kutta approach is that these parameters are chosen such that a power series expansion of the right side of Eq. (13.31) agrees with the Taylor series of expansion of y_{i+1} in terms of y_i and $f(x_i, y_i)$.

The second-order Taylor series expansion of y_{i+1} about y_i is

$$y_{i+1} = y_i + y'h + \frac{y''}{2!}\, h^2 \tag{13.33}$$

Given

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

$$y_i'' = \frac{df}{dx} = f_x + f_y f$$

Therefore, Eq. (13.33) becomes

$$y_{i+1} = y_i + fh + (f_x + f_y f)\, \frac{h^2}{2} \tag{13.34}$$

Now, consider the right side of Eq. (13.31). Since m_1 is already a function of x_i and y_i, we need to expand only m_2 as a power series in terms of $f(x_i, y_i)$. From Eq. (13.32)

$$m_2 = f(x_i + a_1 h, y_i + b_{11} m_1 h)$$

Expanding the function on the right-hand side using the Taylor series expansion, we get

$$m_2 = f(x_i, y_i) + a_1 h f_x + b_{11} m_1 h f_y + O(h^2)$$

Substituting this in Eq. (13.31) and replacing $m_1 = f(x_i, y_i)$ by f, we get

$$y_{i+1} = y_i + [w_1 f + w_2 f + w_2 a_1 h f_x + w_2 b_{11} h f f_y] h + O(h^3)$$

$$= y_i + (w_1 + w_2) hf + (w_2 a_1 f_x + w_2 b_{11} f f_y) h^2 + O(h^3) \qquad (13.35)$$

If Eqs (13.34) and (13.35) are to be equivalent, then they should agree term by term. This is possible only if

$$w_1 + w_2 = 1$$

$$w_2 a_1 = 1/2$$

$$w_2 b_{11} = 1/2$$

Note that we have four unknowns and only three equations. Therefore, there is no unique solution. However, we can assume a value for one of the constants and determine the others. This implies that there is an infinite family of second-order RK methods. For example, if we choose $w_1 = 1/2$, then we get

$$w_1 = 1/2, \qquad w_2 = 1/2, \qquad a_1 = 1, \qquad b_{11} = 1$$

With these values, Eq. (13.31) becomes

$$\boxed{y_{i+1} = y_i + \frac{m_1 + m_2}{2} h} \qquad (13.36)$$

where,

$$m_1 = f(x_i, y_i)$$

$$m_2 = f(x_i + h, y_i + m_1 h)$$

Note that this equation is the Heun's formula.

Similarly, if we choose $w_1 = 0$, then we get

$$w_1 = 0, \qquad w_2 = 1, \qquad a_1 = 1/2, \qquad b_{11} = 1/2$$

and Eq. (13.31) becomes

$$\boxed{y_{i+1} = y_i + m_2 h} \qquad (13.37)$$

where

$$m_1 = f(x_i, y_i)$$

$$m_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{m_1 h}{2}\right)$$

This results in the midpoint or polygon method.

Another strategy is to choose the parameters such that the bound on the truncation error is minimum. It has been shown by Ralston that $w_1 = 1/3$ and $w_2 = 2/3$ produces minimum truncation error. With these weights,

$$\boxed{y_{i+1} = y_i + \frac{m_1 + 2m_2}{3} h} \tag{13.38}$$

where,

$$m_1 = f(x_i, y_i)$$

$$m_2 = f\left(x_i + \frac{3}{4}h, y_i + \frac{3}{2}m_1 h\right)$$

Fourth-Order Runge-Kutta Methods

It is clear from Eq. (13.29) and the discussions above that it is possible to construct RK methods of different orders. However, the commonly used ones are the fourth-order methods. Although there are different versions of fourth-order RK methods, the most popular method is the *classical fourth-order Runge-Kutta* method given below:

$$\boxed{\begin{aligned} m_1 &= f(x_i, y_i) \\ m_2 &= f\left(x_i + \frac{h}{2}, y_i + \frac{m_1 h}{2}\right) \\ m_3 &= f\left(x_i + \frac{h}{2}, y_i + \frac{m_2 h}{2}\right) \\ m_4 &= f(x_i + h, y_i + m_3 h) \\ y_{i+1} &= y_i + \left(\frac{m_1 + 2m_2 + 2m_3 + m_4}{6}\right) h \end{aligned}} \tag{13.39}$$

Example 13.5

Use the classical RK method to estimate $y(0.4)$ when

$$y'(x) = x^2 + y^2 \quad \text{with} \quad y(0) = 0$$

Assume $h = 0.2$

$$f(x, y) = x^2 + y^2$$

$$m_1 = f(x_0, y_0) = 0$$

$$m_2 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{m_1 h}{2}\right) = f(0.1, 0) = 0.01$$

$$m_3 = f\left(x_0 + \frac{h}{2}, y_0 + \frac{m_2 h}{2}\right) = f\left(\frac{2}{2}, \frac{0.01 \times 0.2}{2}\right) = 0.01$$

$$m_4 = f(x_0 + h, y_0 + m_3 h) = f(0.2, 0.01 \times 0.2) = 0.04$$

$$y(0.2) = 0 + \frac{0 + 2 \times 0.01 + 2 \times 0.01 + 0.04}{6} 0.2 = 0.002667$$

Iteration 2

$$x_1 = 0.2$$

$$y_1 = 0.002667$$

$$m_1 = f(0.2, 0.002667) = 0.04$$

$$m_2 = f\left(0.3, 0.002667 + \frac{0.04 \times 0.2}{2}\right) = 0.090044$$

$$m_3 = f\left(0.3, 0.002667 + \frac{0.090044 \times 0.2}{2}\right) = 0.090136$$

$$m_4 = f(0.4, 0.002667 + (0.090136)(0.2)) = 0.160428$$

$$y(0.4) = 0.002667 + \frac{0.04 + 2(0.090044) + 2(0.090136) + 0.160428}{6} 0.2$$

$$= 0.021360 \text{ (correct to six decimals)}$$

The exact answer is 0.021359. If we use $h = 0.1$, then $y(0.4)$ will be 0.021359. Try and check.

Program RUNGE4

RUNGE4 is a program designed to compute the solution of a first order differential equation of type $y' = f(x, y)$ using the fourth-order Runge-Kutta method.

```
*-------------------------------------------------------*
    PROGRAM RUNGE4
*-------------------------------------------------------*
* Main program                                          *
*  This program computes the solution of first order    *
*  differential equation of type y' = f(x,y) using      *
*  the 4th order Runge-Kutta method                     *
*-------------------------------------------------------*
* Functions invoked                                     *
*     F, INT                                            *
*-------------------------------------------------------*
```

```
*  Subroutines used                                                    *
*      NIL                                                             *
*  ------------------------------------------------------------------- *
*  Variables used                                                      *
*      X - Initial value of independent variable                       *
*      Y - Initial value of dependent variable                         *
*     XP - Point of solution                                           *
*      H - Step-size                                                   *
*      N - Number of steps                                             *
*  ------------------------------------------------------------------- *
*  Constants used                                                      *
*      NIL                                                             *
*  ------------------------------------------------------------------- *

      REAL X,Y,XP,H,M1,M2,M3,M4,F
      INTEGER N,INT
      INTRINSIC INT
      EXTERNAL F

      WRITE(*,*)
      WRITE(*,*)'    SOLUTION BY 4_TH ORDER R-K METHOD'
      WRITE(*,*)

*  Input Values

      WRITE(*,*) 'Input initial values of x and y'
      READ(*,*) X,Y
      WRITE(*,*) 'Input x at which y is required'
      READ(*,*) XP
      WRITE(*,*) 'Input step-size h'
      READ(*,*) H

*  Compute number of steps required

      N = INT((XP-X)/H+0.5)

*  Compute Y at each step

      WRITE(*,*)'      ------------------------------ '
      WRITE(*,*)'      STEP         X           Y     '
      WRITE(*,*)'      ------------------------------ '

      DO 40 I = 1,N
         M1 = F(X,Y)
         M2 = F(X+0.5*H,Y+0.5*M1*H)
         M3 = F(X+0.5*H,Y+0.5*M2*H)
         M4 = F(X+H,Y+M3*H)
         X = X+H
         Y = Y+(M1+2.0*M2+2.0*M3+M4)*H/6.0
         WRITE(*,*) I,X,Y
```

```
40     CONTINUE
       WRITE(*,*)'    ----------------------------------'
* Write the final value of Y
       WRITE(*,*)
       WRITE(*,*) 'Value of Y at X =', X,'   is', Y
       WRITE(*,*)
       STOP
       END
* ---------------  End of main RUNGE4 ------------------*
* ----------------------------------------------------- *
* Function subprogram                                   *
* ----------------------------------------------------- *
       REAL FUNCTION F(X,Y)
       REAL X,Y
       F = 2.0 * Y/X
       RETURN
       END
* ---------------  End of function F(X,Y) -------------*
```

Test Run Results

```
        SOLUTION BY 4_TH ORDER R-K METHOD
Input initial values of x and y
1.0 2.0
Input x at which y is required
2.0
Input step-size h
0.25
```

STEP	X	Y
1	1.2500000	3.1246910
2	1.5000000	4.4993830
3	1.7500000	6.1240550
4	2.0000000	7.9986960

```
Value of Y at X = 2.000000 is 7.9986960
Stop - Program terminated.
```

ACCURACY OF ONE-STEP METHODS

How do we achieve the desired level of accuracy in one-step methods? One approach is to repeat the computations at decreasing values of h until the required accuracy is obtained. This may involve a large

number of repetitions if the initial value of h is far away from the optimal value. Another approach is to estimate the value of h that is likely to give the desired level of accuracy.

It is very difficult to have a formula in terms h for the global error. However, we know that the order of global truncation error is h^r if the order of local truncation error is h^{r+1}. We can use this information to study the sensitivity of the solution to the value of h and thereby estimate the size of h. Obtain estimates of $y(h)$ at two different values of h, say h_1 and h_2. Then

$$y_{\text{exact}} - y(b, h_1) = ch_1^r$$

$$y_{\text{exact}} - y(b, h_2) = ch_2^r$$

where c is the constant of proportionality. The above equations can be solved for c as

$$c = \left| \frac{y(b, h_2) - y(b, h_1)}{h_1^r - h_2^r} \right| \tag{13.40}$$

If we need the answer to an accuracy of d decimal places, then the error must not be greater than 0.5×10^{-d} and, therefore,

$$ch^r \leq 0.5 \times 10^{-d}$$

Thus

$$h_{\text{opt}}^r = \left| \frac{(h_1^r - h_2^r)\, 10^{-d}}{2\Delta y} \right| \tag{13.41}$$

where $\Delta y = y(b, h_2) - y(b, h_1)$. For example, for a second-order method, the global error is proportional to h^2 and therefore h_{opt} is given by

$$h_{\text{opt}} = \sqrt{\left| \frac{(h_1^2 - h_2^2)\, 10^{-d}}{2\Delta y} \right|}$$

If we choose $h_1 = 2h_2$, then

$$h_{\text{opt}} = h_1 \sqrt{\left| \frac{3 \times 10^{-d}}{8\Delta y} \right|}$$

Any value of $h < h_{\text{opt}}$ will give the desired accuracy.

Example 13.9

Two estimates of $y(0.8)$ of the equation

$$y'(x) = x^2 + y^2 \qquad \text{with} \qquad y(0) = 1$$

are obtained using fourth-order RK method at $h = 0.05$ and $h = 0.025$.

$$y(0.8, 0.05) = 5.8410870$$

$$y(0.8, 0.025) = 5.8479637$$

Estimate the value of h required to obtain the solution accurate to

(i) four decimal places and (ii) six decimal places

(i) $d = 4$

$$h_4 = \frac{(0.05^4 - 0.025^4)\,10^{-4}}{2(0.0068767)} = 4.26 \times 10^{-8}$$

$$h = 0.01437$$

(ii) $d = 6$

$$h^4 = \frac{(0.05^4 - 0.025^4)\,10^{-6}}{2(0.0068767)} = 4.26 \times 10^{-10}$$

$$h = 0.00454$$

We may use $h = 0.01$ to obtain a figure accurate to four decimal places and $h = 0.004$ to obtain a figure accurate to six decimal places.

13.8 MULTISTEP METHODS

So far we have discussed many methods for obtaining numerical solution of first-order initial-value problems. All of them use information only from the last computed point (x_i, y_i) to compute the next point (x_{i+1}, y_{i+1}). Therefore, all these methods are called *single-step methods*. They do not make use of the information available at the earlier steps, y_{i-1}, y_{i-2}, etc., even when they are available. It is possible to improve the efficiency of estimation by using the information at several previous points. Methods that use information from more than one previous points to compute the next point are called *multistep methods*. Sometimes, a pair of multistep methods are used in conjunction with each other, one for predicting the value of y_{i+1} and the other for correcting the predicted value of y_{i+1}. Such methods are termed *predictor-corrector methods*.

One major problem with multistep methods is that they are not self-starting. They need more information than the initial value condition. If a method uses four previous points, say y_0, y_1, y_2, and y_3, then all these values must be obtained before the method is actually used. These values, known as *starting values*, can be obtained using any of the single-step methods discussed earlier. *It is important to note that the degree of accuracy of the single-step method must match that of the multistep method to be used.* For instance, a fourth-order RK method is normally used to generate starting values for implementing a fourth-order multi-

step method. In this section, we consider the following popular multistep methods:

1. Milne-Simpson method
2. Adams-Bashforth-Moulton method

Both of them are fourth-order methods and use a pair of multistep methods in conjunction with each other.

Milne-Simpson Method

The Milne-Simpson method is a predictor-corrector method. It uses a Milne formula as a *predictor* and the popular Simpson's formula as a *corrector*. These formulae are based on the fundamental theorem of calculus.

$$y(x_{i+1}) = y(x_j) + \int_{x_j}^{x_{i+1}} f(x, y)\,dx \tag{13.42}$$

When $j = i - 3$, the Eq. becomes an open integration formula and produces the *Milne's formula*

$$\boxed{y_{i+1} = y_{i-3} + \frac{4h}{3}(2f_{i-2} - f_{i-1} + 2f_i)} \tag{13.43}$$

Similarly, when $j = i - 1$, Eq. (13.42) becomes a closed form integration and produces the two-segment *Simpson's formula*

$$\boxed{y_{i+1} = y_{i-1} + \frac{h}{3}(f_{i-1} + 4f_i + f_{i+1})} \tag{13.44}$$

Milne's formula is used to 'predict' the value of y_{i+1} which is then used to calculate f_{i+1} (in Eq. (13.44)) from the differential equation.

$$f_{i+1} = f(x_{i+1}, y_{i+1})$$

Then, Eq. (13.44) is used to correct the predicted value of y_{i+1}. The process is then repeated for the next value of i. Each stage involves four basic calculations, namely,

1. prediction of y_{i+1}
2. evaluation of f_{i+1}
3. correction of y_{i+1}
4. improved value of f_{i+1} (for use in next stage)

It is also possible to use the corrector formula repeatedly to refine the estimate of y_{i+1} before moving on to the next stage (see Fig. 13.4).

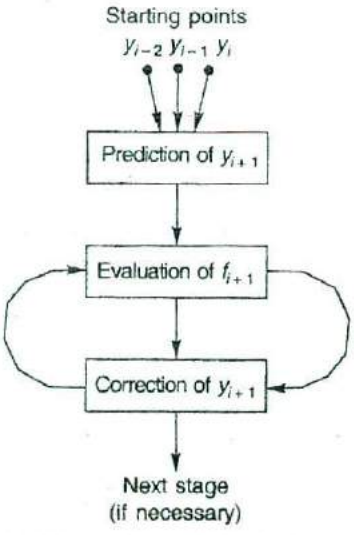


Fig.13.4 Implementation of predictor-corrector methods

Given the equation

$$y'(x) = \frac{2y}{x} \qquad \text{with } y(1) = 2$$

estimate $y(2)$ using the Milne-Simpson predictor-corrector method. Assume $h = 0.25$.

Milne's formula at $i = 3$ is

$$y_4^p = y_0 + \frac{4h}{3}(2f_1 - f_2 + 2f_3)$$

Simpson's formula at $i = 3$ is

$$y_4^c = y_2 + \frac{h}{3}\left(f_2 + 4f_3 + f_4^p\right)$$

where $f_i = f(x_i, y_i)$

To use these formulae, we require the estimates of y_1, y_2 and y_3 in addition to the initial condition y_0. These can be obtained using any of the single-step fourth-order methods.

Let us assume that they have been estimated using the fourth-order RK method as follows:

$$y_1 = y(1.25) = 3.13$$
$$y_2 = y(1.5\ ) = 4.50$$
$$y_3 = y(1.75) = 6.13$$

Then

$$f_1 = \frac{2 \times 3.13}{1.25} = 5.01$$

$$f_2 = \frac{2 \times 4.5}{1.5} = 6.00$$

$$f_3 = \frac{2 \times 6.13}{1.75} = 7.01$$

Substituting these in Milne's formula we predict the value of $y(2)$ as

$$y_4^p = 2.00 + \frac{4 \times 0.25}{3}(2 \times 5.01 - 6.00 + 2 \times 7.01) = 8.01$$

$$f_4^p = \frac{2 \times 8.01}{2} = 8.01$$

Now we obtain the corrected value of $y(2)$ using Simpson formula as

$$y_4^c = 4.50 + \frac{0.25}{3}(6.00 + 4 \times 7.01 + 8.01)$$

$$= 8.00$$

We can again use the corrector formula to refine the estimate

$$f_4 = \frac{2 \times 8.00}{2} = 8.00$$

$$y_4^c = 4.50 + \frac{0.25}{3}(6.00 + 4 \times 7.01 + 8.01)$$

$$= 8.00$$

Note that the exact solution is $y(2) = 8$.

Program MILSIM

An algorithm for evaluating the equation $y' = f(x, y)$ using Milne-Simpson method is illustrated in Fig. 13.4. Program MILSIM shows the implementation of the algorithm in details. The program does the following:

1. Computes the starting points using fourth-order RK method
2. Predicts the function value by Milne's formula
3. Corrects the value obtained using Simpson's method
4. Writes the results

```
*-----------------------------------------------------------
      PROGRAM MILSIM
*-----------------------------------------------------------*
* Main program                                              *
*    This program solves the first order differential       *
*    equation y' = f(x,y) using Milne-Simpson method        *
*-----------------------------------------------------------*
* Functions invoked                                         *
*      F,INT                                                *
*-----------------------------------------------------------*
* Subroutines used                                          *
*      NIL                                                  *
*-----------------------------------------------------------*
* Variables used                                            *
*      X(1)  - Initial value of independent variable        *
*      Y(1)  - Initial value of dependent variable          *
*        XP  - Point of solution                            *
*         N  - Number of steps                              *
*         H  - Step-size                                    *
*         X  - Array of independent variable                *
*         Y  - Array of dependent variable                  *
*-----------------------------------------------------------*
* Constants used                                            *
*      NIL                                                  *
*-----------------------------------------------------------*
```

```
      REAL X,Y,H,XP,M1,M2,M3,M4,SUM1,SUM2,F
      INTEGER N,INT
      INTRINSIC INT
      EXTERNAL F
      DIMENSION X(10), Y(10)
*  Read values
      WRITE(*,*) 'Input initial values of x and y'
      READ(*,*) X(1),Y(1)
      WRITE(*,*) 'Input x at which y is required'
      READ(*,*) XP
      WRITE(*,*) 'Input step-size h'
      READ(*,*) H
*  Compute number of computations involved
      N = INT((XP-X(1))/H + 0.5)
*  We need four starting points for Milne-Simpson method.
*  Initial values form the first point.  Remaining three
*  points are obtained using 4th order RK method
      WRITE(*,*)
      WRITE(*,*) 'INITIAL VALUES', X(1), Y(1)
      WRITE(*,*)
*  Computing three points by RK method
      WRITE(*,*) 'THREE VALUES BY RK METHOD'
      DO 10 I = 1,3
         M1 = F(X(I),Y(I))
         M2 = F(X(I)+0.5*H, Y(I)+0.5*M1*H)
         M3 = F(X(I)+0.5*H, Y(I)+0.5*M2*H)
         M4 = F(X(I)+H, Y(I)+M3*H)
         X(I+1) = X(I)+H
         Y(I+1) = Y(I)+(M1+2.0*M2+2.0*M3+M4)*H/6.0
         WRITE(*,*) I, X(I+1), Y(I+1)
10    CONTINUE
      WRITE(*,*)
      WRITE(*,*) 'VALUES OBTAINED BY MILNE-SIMPSON METHOD'
      DO 20 I = 4, N
         F2 = F(X(I-2), Y(I-2))
         F3 = F(X(I-1), Y(I-1))
         F4 = F(X(I), Y(I))
*  Predicted value of y (by Milne's formula)
         Y(I+1) = Y(I-3)+4.0*H/3.0*(2.0*F2-F3+2.0*F4)
```

```
            X(I+1) = X(I) + H
            F5 = F(X(I+1), Y(I+1))
*  Corrected value of Y (by Simpson's formula)
            Y(I+1) = Y(I-1) + H/3.0 *(F3+4.0*F4+F5)
            WRITE(*,*) I, X(I+1), Y(I+1)
20      CONTINUE
        WRITE(*,*)
        WRITE(*,*)'Value of Y at X  =', X(N+1),'  is',
              Y(N+1)
        WRITE(*,*)
        STOP
        END
*  ---------------End of main MILSIM ------------------ *
*  ------------------------------------------------------ *
*  Function subprogram                                    *
*  ------------------------------------------------------ *
        REAL FUNCTION F(X,Y)
        REAL X,Y
        F = 2.0 * Y/X
        RETURN
        END
*  -------------- End of function F(X,Y) --------------- *
```

Test Run Results

```
Input initial values of x and y
1.0 2.0
Input x at which y is required
2.0
Input step-size h
0.125

INITIAL VALUES   1.0000000     2.0000000
THREE VALUES BY RK METHOD
          1      1.1250000     2.5312380
          2      1.2500000     3.1249770
          3      1.3750000     3.7812150
VALUES OBTAINED BY MILNE-SIMPSON METHOD
          4      1.5000000     4.4999660
          5      1.6250000     5.2812040
          6      1.7500000     6.1249520
```

```
7          1.8750000      7.0311890
8          2.0000000      7.9999360
Value of Y at X = 2.0000000 is 7.9999360
Stop - Program terminated.
```

Adams-Bashforth-Moulton Method

Another popular fourth-order predictor-corrector method is the Adams-Bashforth-Moulton multistep method. The predictor formula is known as *Adams-Bashforth predictor* and is given by

$$y_{i+1} = y_i + \frac{h}{24}(55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}) \tag{13.45}$$

The corrector formula is known as *Adams-Moulton corrector* and is given by

$$y_{i+1} = y_i + \frac{h}{24}(f_{i-2} - 5f_{i-1} + 19f_i + 9f_{i+1}) \tag{13.46}$$

This pair of equations can be implemented using the procedure described for Milne-Simpson method.

Example 13.11

Repeat Example 13.10 using Adams-Bashforth-Moulton method.

By Adams predictor formula

$$y_4^p = y_3 + \frac{h}{24}(55f_3 - 59f_2 + 37f_1 - 9f_0)$$

$$= 6.13 + \frac{0.25}{24}(55 \times 7.01 - 59 \times 6.00 + 37 \times 5.01 - 9 \times 4)$$

$$= 8.0146$$

$$f_4^p = \frac{2 \times 8.0146}{2} \quad 8.0146$$

By Adams corrector formula

$$y_4^c = y_3 + \frac{h}{24}(f_1 - 5f_2 + 19f_3 + 9f_4^p)$$

$$= 6.13 + \frac{0.25}{24}(5.01 - 5 \times 6.00 + 19 \times 7.01 + 9 \times 8.0146)$$

$$= 8.0086$$

y_4^c (refined) = 8.0079

13.9 ACCURACY OF MULTISTEP METHODS

We know that for each differential equation, there is an optimum step size h. If h is too large, accuracy diminishes and if it is too small, round-off errors would dominate and reduce the accuracy.

By computing the predicted and corrected values of y_{i+1}, we can estimate the size and sign of the error. Let us denote the predicted value by y^p_{i+1}. Similarly, denote the truncation error in predicted value by E_{tp} and corrected value by E_{tc}. Then, we have,

$$E_{tp} = y - y^p_{i+1}$$

$$E_{tc} = y - y^c_{i+1}$$

where y denotes the exact value of $y(x_{i+1})$. The difference between the error is

$$E_{tp} - E_{tc} = y^c_{i+1} - y^p_{i+1} \tag{13.47}$$

A large difference indicates that the step size is too large. In such cases, we must reduce the size of h.

Milne-Simpson Method

Both the Milne and Simpson formulae are of order h^4 and their error terms are of order h^5.

The truncation error in Milne's formula is

$$E_{tp} = \frac{28}{90} y^{(5)}(\theta_1) h^5$$

The truncation error in Simpson's formula is

$$E_{tc} = -\frac{1}{90} y^{(5)}(\theta_2) h^5$$

If we assume that $y^{(5)}(\theta_1) = y^{(5)}(\theta_2)$ then

$$\frac{E_{tp}}{E_{tc}} = -28$$

or,

$$E_{tp} = -28E_{tc}$$

Substituting this in Eq. (13.47) we obtain,

$$E_{tc} = -\frac{y^c_{i+1} - y^p_{i+1}}{29}$$

If the answer is required to a precision of d decimal digits then

$$|E_{tc}| = \left| \frac{y^c_{i+1} - y^p_{i+1}}{29} \right| < 0.5 \times 10^{-d}$$

or

$$\boxed{\left| y^c_{i+1} - y^p_{i+1} \right| < 29/2 \times 10^{-d} = 15 \times 10^{-d}} \tag{13.48}$$

Adams Method

The truncation error in Adams-Bashforth predictor is

$$E_{tp} = \frac{251}{720} y^{(5)}(\theta_1) h^5$$

and the truncation error in Adams-Moulton corrector is

$$E_{tc} = -\frac{19}{720} y^{(5)}(\theta_2) h^5$$

Then, assuming $y^{(5)}(\theta_1) = y^{(5)}(\theta_2)$, we get

$$E_{tp} = -\frac{251}{19} E_{tc}$$

Substituting in Eq. (13.47) results in

$$E_{tc} = -\frac{19}{270} \left(y^c_{i+1} - y^p_{i+1} \right)$$

For achieving an accuracy of d decimal digits,

$$\boxed{\left| y^c_{i+1} - y^p_{i+1} \right| < 270/19 \times 0.5 \times 10^{-d} \approx 7 \times 10^{-d}} \tag{13.49}$$

According to Eqs (13.48) and (13.49), h should be reduced until the difference between the corrected and predicted values is within the specified limit. It should be noted, however, that if the step length is changed during the calculation, it will be necessary to recalculate the starting points at the new step value.

Modifiers

Using the error estimates, we can modify the estimates of y^c_{i+1} before proceeding to the next stage. That is

$$y_{i+1} = y^c_{i+1} + E_{tc}$$

For Milne's method

$$y_{i+1} = y^c_{i+1} - \frac{1}{29} \left(y^c_{i+1} - y^p_{i+1} \right)$$

For Adams meth d

$$y_{i+1} = y^c_{i+1} - \frac{19}{270}\left(y^c_{i+1} - y^p_{i+1}\right)$$

Example 12

Solve the differential equation

$$y'(x) = -y^2$$

for $y(2.0)$ using the Milne-Simpson method with the application of modifier to the corrector. The first four points are given under

i	x_i	$y_i = y(x_i)$
0	1.0	1.0000000
1	1.2	0.8333333
2	1.4	0.7142857
3	1.6	0.6250000

To estimate $y(2.0)$ with $h = 0.2$, we need two iterations.

$$f_i = -y_i^2$$

$$f_1 = -0.6944444$$

$$f_2 = -0.5102040$$

$$f_3 = -0.3906250$$

Iteration 1

$$y(1.8) = y^p_4 = y_0 + \frac{4h}{3}(2f_1 - f_2 + 2f_3)$$

$$= 1.0 + \frac{4 \times 0.2}{3}(-1.13888889 + 0.5102040 - 0.7812500)$$

$$= 0.5573506$$

$$f^p_4 = -\left(y^p_4\right)^2 = -0.3106396$$

$$y^c_4 = y_2 + \frac{h}{3}\left(f_2 + 4f_3 + f^p_4\right)$$

$$= 0.7142857 + \frac{0.2}{3}(-0.5102040 - 1.56250 - 0.3106396)$$

$$= 0.5553961$$

$$\text{Modifier } E_{tc} = -\frac{y^c_4 - y^p_4}{29} = 0.0000674$$

$$\text{Modified } y_4^c = y_4^c + E_{tc} = 0.5554635$$

Iteration 2

$$f_4 = -\left(y_4^c\right)^2 = -0.3085397$$

$$y(2.0) = y_5^p = y_1 + \frac{4h}{3}(2f_2 - f_3 + 2f_4)$$

$$= 0.5008366$$

$$f_5^p = -0.2508373$$

$$y_5^c = y_3 + \frac{h}{3}\left(f_3 + 4f_4 + f_5^p\right)$$

$$= 0.4999585$$

$$\text{Modifier } E_{tc} = 0.0000303$$

$$\text{Modified } y_5^c = 0.4999888$$

$$\text{Exact answer} = 0.5$$

$$\text{Error} = 0.0000112$$

13.10 SYSTEMS OF DIFFERENTIAL EQUATIONS

Mathematical models of many applications involve a system of several first-order differential equations. They may be represented as follows:

$$\frac{dy_1}{dx} = f_1(x, y_1, y_2, \ldots, y_m), \qquad y_1(x_0) = y_{10}$$

$$\frac{dy_2}{dx} = f_2(x, y_1, y_2, \ldots, y_m), \qquad y_2(x_0) = y_{20}$$

$$\vdots \qquad (13.50)$$

$$\frac{dy_m}{dx} = f_m(x, y_1, y_2, \ldots, y_m), \qquad y_m(x_0) = y_{m0}$$

These equations should be solved for $y_1(x)$, $y_2(x)$, ..., $y_m(x)$ over interval (a, b).

These equations can be solved by any of the methods discussed in this chapter. At each stage, all the equations are solved before proceeding to the next stage. For example, if $h = 0.5$ and $a = x_0 = 0$, then we must evaluate $y_1(0.5)$, $y_2(0.5)$,..., $y_m(0.5)$ before preceding to the stage $h = 1.0$. Let us consider a system of two equations for the purpose of illustration.

$$y_1'(x) = f_1(x, y_1, y_2), \qquad y_1(x_0) = y_{10}$$

$$y_2'(x) = f_2(x, y_1, y_2), \qquad y_2(x_0) = y_{20}$$

Assume that we want to use the Heun's method. The first stage would involve the following calculations.

$$m_1(1) = f_1(x_0, y_{10}, y_{20})$$

$$m_1(2) = f_2(x_0, y_{10}, y_{20})$$

$$m_2(1) = f_1(x_0 + h, y_{10} + hm_1(1), \quad y_{20} + hm_1(2))$$

$$m_2(2) = f_2(x_0 + h, y_{10} + hm_1(1), \quad y_{20} + hm_1(2))$$

$$m(1) = \frac{m_1(1) + m_2(1)}{2}$$

$$m(2) = \frac{m_1(2) + m_2(2)}{2}$$

$$y_1(x_1) = y_1(1) = y_1(x_0) + m(1)h = y_{10} + m(1)h$$

$$y_2(x_1) = y_2(1) = y_2(x_0) + m(2)h = y_{20} + m(2)h$$

The next stage uses $y_1(1)$ and $y_2(1)$ as initial values and, by following similar procedure, $y_1(2)$ and $y_2(2)$ are obtained.

Example 13.13

Given the equations

$$\frac{dy_1}{dx} = x + y_1 + y_2, \qquad y_1(0) = 1$$

$$\frac{dy_2}{dx} = 1 + y_1 + y_2, \qquad y_2(0) = -1$$

estimate the values of $y_1(0.1)$ and $y_2(0.1)$ using Heun's method.

Given $x_0 = 0, \quad y_{10} = 1, \quad y_{20} = -1$

$$m_1(1) = f_1(x_0, y_{10}, y_{20}) = 0 + 1 - 1 = 0$$

$$m_1(2) = f_2(x_0, y_{10}, y_{20}) = 1 + 1 - 1 = 1$$

$$m_2(1) = f_1(x_0 + h, y_{10} + hm_1(1), y_{20} + hm_1(2))$$

$$= f_1(0.1, 1 + 0.1 \times 0, -1 + 0.1 \times 1)$$

$$= f_1(0.1, 1, -0.9)$$

$$= 0.1 + 1 - 0.9 = 0.2$$

$$m_2(2) = f_2(x_0 + h, y_{10} + hm_1(1), y_{20} + hm_1(2))$$

$$= f_2(0.1, 1, -0.9)$$

$$= 1 + 1 - 0.9 = 1.1$$

$$m(1) = \frac{m_1(1) + m_2(1)}{2} = 0.1$$

$$m(2) = \frac{m_1(2) + m_2(2)}{2} = 1.05$$

$$y_1(0.1) = y_1(0) + hm(1) = 1 + (0.1)(0.1) = 1.01$$

$$y_2(0.1) = y_2(0) + h\,m(2) = -1 + (0.1)(1.05) = -0.895$$

13.11 HIGHER-ORDER EQUATIONS

We have seen in the introductory section of this chapter that many problems involve the solution of higher-order differential equations. A higher-order differential equation is in the form

$$\frac{d^m y}{d^m x} = f\left(x, y, \frac{dy}{dx}, \frac{d^2y}{dx^2}, \ldots, \frac{d^{m-1}y}{dx^{m-1}}\right) \tag{13.51}$$

with m initial conditions given as

$$y(x_0) = a_1, \quad y'(x_0) = a_2, \ldots, y^{m-1}(x_0) = a_m$$

We can replace Eq. (13.51) by a system of first-order equations as follows:
Let us denote

$$y = y_1, \quad \frac{dy}{dx} = y_2, \quad \frac{d^2y}{dx^2} = y_3, \ldots, \quad \frac{d^{m-1}y}{dx^{m-1}} = y_m$$

Then,

$$\begin{aligned}
&\frac{dy_1}{dx} = y_2 && y_1(x_0) = y_{10} = a_1\\
&\frac{dy_2}{dx} = y_3 && y_2(x_0) = y_{20} = a_2\\
&\quad\vdots\\
&\frac{dy_{m-1}}{dx} = y_m && y_{m-1}(x_0) = y_{m-1,0} = a_{m-1}\\
&\frac{dy_m}{dx} = f(x, y_1, y_2, \ldots, y_m) && y_m(x_0) = y_{m0} = a_m
\end{aligned} \tag{13.52}$$

This system is similar to the system of first-order Eq. (13.50) with the conditions

$$f_i = y_{i+1}, \qquad i = 1, 2, \ldots, m-1$$
$$f_m = f(x, y_1, y_2, \ldots, y_m)$$

and, therefore, can be solved using the procedure discussed in the previous section.

Example 13.14

Solve the following equation for $y(0.2)$

$$\frac{d^2y}{dx^2} + 2\frac{dy}{dx} - 3y = 6x$$

given $y(0) = 0$, $y'(0) = 1$. Use Heun's method

$$\frac{d^2y}{dx^2} = 6x + 3y - 2\frac{dy}{dx}$$

Let $$y = y_1, \frac{dy}{dx} = y_2$$

Then,

$$\frac{dy_1}{dx} = y_2, \qquad y_{10} = 0$$

$$\frac{dy_2}{dx} = 6x + 3y_1 - 2y_2, \qquad y_{20} = 1$$

Let $h = 0.2$

$$m_1(1) = y_{20} = 1$$
$$m_1(2) = 6x_0 + 3y_{10} - 2y_{20} = -2$$
$$m_2(1) = y_{20} + hm_1(2) = 1 + (0.2)(-2) = 0.6$$
$$m_2(2) = 6(0.2) + 3(0 + 0.2(1)) - 2(0.6) = 1.2 + 0.6 - 1.2 = 0.6$$

$$m(1) = \frac{1 + 0.6}{2} = 0.8$$

$$m(2) = \frac{-2 + 0.6}{2} = -0.7$$

$$y_1(0.2) = y_1(0) + 0.2m(1) = 0 + 0.2 \times 0.8 = 0.16$$
$$y_2(0.2) = y_2(0) + 0.2m(2) = 1 - (0.2)(0.7) = 0.86$$

$y(x)$ at $x = 0.2 = 0.16$

$y'(x)$ at $x = 0.2 = 0.86$

13.12 SUMMARY

We encounter differential equations in many different forms while attempting to solve real life problems in science and engineering. The most

common form of differential equations is known as ordinary differential equation. In this chapter, we considered various methods of numerical solution of ordinary differential equations. They include

- Taylor series method
- Euler's method
- Heun's method
- Polygon method
- Runge-Kutta method
- Milne-Simpson method
- Adams-Bashforth-Moulton method

We also discussed the accuracy (and techniques of improving the accuracy) of these methods. Finally, we discussed the solution of a system of differential equations as well as higher-order equations.

We presented FORTRAN programs and their test results for the following methods:

- Euler's method
- Heun's method
- Polygon method
- Runge-Kutta method
- Milne-Simpson method

Key Terms

Adams-Bashforth predictor
Adams-Moulton corrector
Boundary-value problem
Corrector
Differential equations
Euler's method
Global truncation error
Heun's method
Initial-value problem
Kirchhoff's law
Law of cooling
Law of motion
Local truncation error
Midpoint method
Milne's formula
Milne-Simpson method
Multistep methods
One-step methods
Ordinary differential equations
Partial differential equation
Picard's method
Polygon method
Predictor
Predictor-corrector method
Radioactive decay
Runge-Kutta method
Semi-numeric method
Simple harmonic motion
Simpson's formula
Starting values
Taylor series method

REVIEW QUESTIONS

1. What is a differential equation? Give two real-life examples of application of differential equations.
2. Distinguish between ordinary and partial differential equations.

3. State the degree and order of the following differential equations:
   (a) $(y'')^2 + 7y' = 0$
   (b) $y''' + 5(y')^2 = 1$
   (c) $y'' - y = 0$
   (d) $y'' + ay^2 = 0$
   (e) $y'' + 3y' - 2y = x^2$
   (f) $y(y')^2 - 2xy' + y = 0$
   (g) $xy' - (x + 1)y = 0$
   (h) $y^2(y')^2 + xy\,y' - 2x^2 = 0$
   (i) $(y'')^2 + y^2 = 0$
4. What is a nonlinear differential equation? State which of the equations given in Question 3 are nonlinear.
5. What is an initial-value problem? How is it different from a boundary-value problem?
6. Why do we need to use numerical computing techniques to solve differential equations?
7. State the basic two approaches used in estimating the solution of differential equations. How are they different?
8. Describe how Taylor's theorem of expansion can be used to solve a differential equation.
9. What are the limitations of Taylor's series method?
10. State the formula of Picard's method to solve the differential equation of type

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

    What are its limitations?
11. State the formula of Euler's method. Illustrate its concept graphically.
12. Comment on the accuracy of Euler's method.
13. Illustrate Heun's method of solution graphically.
14. How does the accuracy of Heun's method compare with that of Euler's method?
15. Heun's method is an improved version of Euler's method. Comment.
16. Why is Heun's method classified as one-step predictor-corrector method?
17. Why is the polygon method called the midpoint method? Illustrate graphically.
18. Describe the basic concept employed in Runge-Kutta methods.
19. What is meant by an r-order Runge-Kutta method? What is the order of the following methods?
    (a) Euler's method
    (b) Heun's method
    (c) Polygon method

20. What are multistep methods?
21. What are the merits and demerits of multistep methods?
22. State the formulae used in Milne-Simpson method. Describe the implementation scheme of these formulae.
23. State the predictor and corrector formulae used in Adams-Bashforth-Moulton method.
24. How is the accuracy? How is it controlled? How can we improve the accuracy of multistep methods?
25. A high-order differential equation can be solved by replacing it by a system of first-order equations. Discuss.

REVIEW EXERCISES

1. Use Taylor's expansion (with terms up to x^3) to solve the following differential equations:

(a) $\dfrac{dy}{dx} = x + y + xy, \qquad y(0) = 1$

for $x = 0.25$ and 0.5.

(b) $\dfrac{dy}{dx} = y(x^2 - 1), \qquad y(0) = 1$

for $x = 1.0$, 1.5 and 2.0

(c) $\dfrac{dy}{dx} = x + y, \qquad y(0) = 1$

for $x = 0.1$ and 0.5

(d) $\dfrac{dy}{dx} = \dfrac{2x}{y} - xy, \qquad y(0) = 1$

for $x = 0.25$ and 0.5

(e) $\dfrac{dy}{dx} = x^2y^2, \qquad y(1) = 0$

for $x = 2.0$ and 3.0

(f) $\left(\dfrac{dy}{dx}\right)^2 = xy, \qquad y(0) = 1$ and $y'(0) = 1$

for $x = 0.2$ and 0.4

(g) $10\dfrac{d^2y}{dt^2} + \left(\dfrac{dx}{dt}\right)^2 + 6x = 0, \qquad x(0) = 1$ and $x'(0) = 0$

for $t = 2$ and 5.

2. Solve the following equations by Picards method and estimate y at $x = 0.25$ and 0.5:

   (a) $\frac{dy}{dx} = x + (x + 1)y, \quad y(0) = 1$

   (b) $\frac{dy}{dx} = x^2y - y, \quad y(0) = 1$

   (c) $\frac{dy}{dx} = x + y, \quad y(0) = 1$

   (d) $\frac{dy}{dx} = x^2y^2, \quad y(1) = 0$

   (e) $\frac{dy}{dx} = \frac{x}{y}, \quad y(0) = 1$

3. Use the simple Euler's method to solve the following equations for $y(1)$ using $h = 0.5$, 0.25 and 0.1.

   (a) $y' = 2xy, \quad y(0) = 1$

   (b) $y' = x^2 + y^2, \quad y(0) = 2$

   (c) $y' = \frac{-y}{2y+1}, \quad y(0) = 1$

   (d) $y' = \frac{x}{y}, \quad y(0) = 1$

   (e) $y' = x + y + xy, \quad y(0) = 1$

4. Solve the differential equation

$$\frac{dy}{dx} = x + y, \quad y(0) = 1$$

   by the simple Euler's method to estimate $y(1)$ using $h = 0.5$ and $h = 0.25$. Compute errors in both the cases. How do they compare? Also, compare your results with the exact answer given the analytical solution as

$$y(x) = 2e^x - x - 1$$

5. Use Heun's method with $h = 0.5$ and $h = 0.25$ to solve the equations in Exercise 3 for $y(1)$. How do the results compare with these obtained using simple Euler's method.
6. Repeat Exercise 4 using the Polygon method instead of simple Euler's method. Analyse the results critically.
7. Use the polygon method to solve some of the equation in Exercise 3.
8. Use the classical RK method to estimate $y(0.5)$ of the following equations with $h = 0.25$.

   (a) $\frac{dy}{dx} = x + y, \quad y(0) = 1$

(b) $\frac{dy}{dx} = \frac{x}{y}$, $y(0) = 1$

(c) $\frac{dy}{dx} = y \cos x$, $y(0) = 1$

(d) $\frac{dy}{dx} = y + \sin x$, $y(0) = 2$

(e) $\frac{dy}{dx} = y + \sqrt{y}$, $y(0) = 1$

9. Solve the following initial value problems for $x = 1$ using the fourth-order Milne's method.

   (a) $\frac{dy}{dx} = y - x^2$, $y(0) = 1$

   Use a step size of 0.25 and fourth-order Runge-Kutta method to predict the starting values.
10. Repeat Exercise 9 using Adam's method instead of Milne's method. Compare the results.
11. Repeat Exercise 10 and 11 with the application of modifier to the corrector. Compare the results.
12. Solve the pair of simultaneous equations

$$\frac{dy_1}{dx} = y_2 \qquad y_1(0) = 0$$

$$\frac{dy_2}{dx} = y_1 y_2 + x^2 + 1 \qquad y_2(0) = 0$$

   to estimate $y_1(0.2)$ and $y_2(0.2)$ using any method of your choice.
13. Solve the following equation for $y(0.2)$:

$$10 \frac{d^2y}{dx^2} + \left(\frac{dy}{dx}\right)^2 + 6x = 0, \qquad y(0) = 1, \qquad y'(0) = 0$$

   Use Heun's method.
14. The general equation relating to current i, voltage V, resistance R, and inductance L of a serial electric circuit is given by

$$L \frac{di}{dt} + iR = V$$

   Find the value of current after 2 seconds, if resistance $R = 20$ ohms, inductance $L = 50$ H and voltage $V = 240$ volts. Current $I = 0$ when $t = 0$.

15. A tank contains a solution which is made dissolving 50 kg of salt in 100 gallons of water. A more concentrated solution of 3 kg of salt per gallon of water is pumped into the tank at a rate of 4 gallons per minute. The solution (which is stirred continuously to keep it uniform) is pumped out at a rate of 3 gallons per minute.
Find the amount of salt in the tank at $t = 15$ minutes. Note that initially at $t = 0$, the amount of salt $x = 50$ kg.

Hint: If x represents the amount of salt in the tank at time t, $\frac{dx}{dt}$ will represent the change in the amount of salt.

16. An object with a mass of 10 kg is falling under the influence of earth gravity. Find its velocity after 5 seconds if it starts from the rest. The object experiences a retarding force equal to 0.25 of its velocity.
Hint: The relationship between the various forces is given by

$$\text{mass} \times \frac{dv}{dt} = \text{mass} \times \text{gravity} - \text{retarding force}$$

where v is velocity at time t.

17. A body of mass 2 kg is attached to a spring with a spring constant of 10. The differential equation governing the displacement of the body y and time t is given by

$$\frac{d^2y}{dt^2} + 2\frac{dy}{dt} + 5y = 0$$

Find the displacement y at time $t = 1.5$ given that $y(0) = 2$ and $y'(0) = -4$.

PROGRAMMING PROJECTS

1. Rewrite the program RUNGE4 such that a subprogram implements the fourth order Runge-Kutta method and a main program receives input information, drives the subprogram to compute the solution, and prints the required output information.
2. Modify the program MILSIM to incorporate the following changes:
   (a) Computing the starting points by Runge-Kutta method using a subprogram
   (b) Implementing Milne-Simpson algorithm by a subprogram
   (c) Applying the modifier to the corrector with the help of a subprogram.
3. Develop a user-friendly modular program as suggested in Project 2 for the fourth-order Adams method with modifiers.
4. Develop a user-friendly program for solving systems of differential equations using Euler's method or Heun's method.
5. Repeat Project 4, but use the fourth-order Runge-Kutta method.

CHAPTER 14

Boundary Value and Eigenvalue Problems

14.1 NEED AND SCOPE

We have seen that we require m conditions to be specified in order to solve an m-order differential equation. In the previous chapter, all the m conditions were specified at one point, $x = x_0$, and, therefore, we call this problem as an *initial-value problem*. It is not always necessary to specify the conditions at one point of the independent variable. They can be specified at different points in the interval (a, b) and, therefore, such problems are called the *boundary value problems*. A large number of problems fall into this category.

In solving initial value problems, we move in steps from the given initial value of x to the point where the solution is required. In case of boundary value problems, we seek solutions at specified points within the domain of given boundaries, for instance, given

$$\frac{d^2y}{dx^2} = f(x, y, y') \qquad y(a) = y_a, \qquad y(b) = y_b \tag{14.1}$$

we are interested in finding the values of y in the range $a \le x \le b$.

There are two popular methods available for solving the boundary value problems. The first one is known as the *shooting method*. This method makes use of the techniques of solving initial value problems. The second one is called the *finite difference method* which makes use of the finite difference equivalents of derivatives.

Some boundary value problems, such as study of vibrating systems, structure analysis, and electric circuit system analysis, reduce to a system of equations of the form

$$\mathbf{Ax} = \lambda \mathbf{x} \tag{14.2}$$

Such problems are called the *eigenvalue problems*. We need to determine the values of λ and vector x which satisfy the Eq. (14.2). We have two simple methods available to solve this type of problems.

1. Polynomial method
2. Power method

In this chapter, we consider these two special categories of problems and discuss the following methods to solve them:

1. Shooting method
2. Finite difference method
3. Polynomial method
4. Power method

SHOOTING METHOD

This method is called the *shooting method* because it resembles an artillery problem. In this method, the given boundary value problem is first converted into an equivalent initial value problem and then solved using any of the methods discussed in the previous chapter. The approach is simple. Consider the equation

$$y'' = f(x, y, y') \qquad y(a) = A, \qquad y(b) = B$$

By letting $y' = z$, we obtain the following set of two equations:

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

In order to solve this set as an initial value problem, we need two conditions at $x = a$. We have one condition $y(a) = A$ and, therefore, require another condition for z at $x = a$. Let us assume that $z(a) = M_1$, where M_1 is a "guess". Note that M_1 represents the slope $y'(x)$ at $x = a$. Thus, the problem is reduced to a system two first-order equations with the initial conditions

$$y' = z \qquad y(a) = A$$
$$z' = f(x, y, z) \qquad z(a) = M_1 \ (= y' \ (a)) \tag{14.3}$$

Equation (14.3) can be solved for y and z using any one-step method using steps of h, until the solution at $x = b$ is reached. Let the estimated value of $y(x)$ at $x = b$ be B_1. If $B_1 = B$, then we have obtained the required solution. In practice, it is very unlikely that our initial guess $z(a) = M_1$ is correct.

If $B_1 \neq B$, then we obtain the solution with another guess, say $z(a) = M_2$. Let the new estimate of $y(x)$ at $x = b$ be B_2 (see Fig. 14.1). If B_2 is not equal to B, then the process may be continued until we obtain the correct estimate of $y(b)$. However, the procedure can be accelerated by using an improved guess for $z(a)$ after the estimates of B_1 and B_2 are obtained.

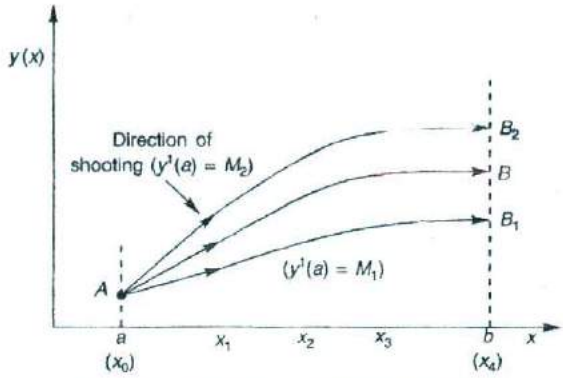


Fig. 14.1 Illustration of shooting method

Let us assume that $z(a) = M_3$ leads to the value $y(b) = B$. If we assume that the values of M and B are linearly related, then

$$\frac{M_3 - M_2}{B - B_2} = \frac{M_2 - M_1}{B_2 - B_1}$$

Then

$$M_3 = M_2 + \frac{B - B_2}{B_2 - B_1} \times (M_2 - M_1)$$

$$= M_2 - \frac{B_2 - B}{B_2 - B_1} \times (M_2 - M_1) \qquad (14.4)$$

Now with $z(a) = M_3$, we can again obtain the solution of $y(x)$.

Example 14.1

Using shooting method, solve the equation

$$\frac{d^2y}{dx^2} = 6x, \qquad y(1) = 2, \qquad y(2) = 9$$

in the interval (1,2)

By transformation we obtain the following:

$$\frac{dy}{dx} = z \qquad y(1) = 2$$

$$\frac{dz}{dx} = 6x$$

Let us assume that $z(1) = y'(1) = 2(M_1)$. Applying Heun's method, we obtain the solution as follows:

Iteration 1

$$h = 0.5$$

$$x_0 = 1, \quad y(1) = y_0 = 2, \quad z(1) = z_0 = 2$$

$$m_1(1) = z_0 = 2$$

$$m_1(2) = 6x_0 = 6$$

$$m_2(1) = z_0 + hm_1(2) = 2 + 0.5(6) = 5$$

$$m_2(2) = 6(x_0 + h) = 6(1.5) = 9$$

$$m(1) = \frac{m_1(1) + m_2(1)}{2} = \frac{2+5}{2} = 3.5$$

$$m(2) = \frac{m_1(2) + m_2(2)}{2} = \frac{6+9}{2} = 7.5$$

$$y(x_1) = y(1.5) = (1) + m(1)h = 2 + 3.5 \times 0.5 = 3.75$$

$$z(x_1) = z(1.5) = z(1) + m(2)h = 2 + 7.5 \times 0.5 = 5.75$$

Iteration 2

$$h = 0.5$$

$$x_1 = 1.5, \quad y_1 = 3.75, \quad z_1 = 5.75$$

$$m_1(1) = z_1 = 5.75$$

$$m_1(2) = 6x_1 = 9$$

$$m_2(1) = z_1 + hm_1(2) = 5.75 + 0.5 \times 9 = 10.25$$

$$m_2(2) = 6(x_1 + h) = 12$$

$$m(1) = \frac{5.75 + 10.25}{2} = 8$$

$$m(2) = \frac{9 + 12}{2} = 10.5$$

$$y(x_2) = (2) = y(1) + m(1)h = 3.75 + 8 \times 0.5 = 7.75$$

This gives $B_1 = 7.75$ which is less than $B = 9$

Now, let us assume $z(1) = y'(1) = 4(M_2)$ and again estimate $y(2)$.

Iteration 1

$$h = 0.5$$

$$x_0 = 1, \quad y_0 = 2, \quad z_0 = 4$$

$$m_1(1) = z_0 = 4$$

$$m_1(2) = 6x_0 = 6$$

$$m_2(1) = z_0 + hm_1(2) = 4 + 0.5 \times 6 = 7$$

$$m_2(2) = 6(x_0 + h) = 6(1.5) = 9$$

$$m(1) = \frac{4+7}{2} = 5.5$$

$$m(2) = \frac{6+9}{2} = 7.5$$

$$y(x_1) = y(1.5) = 2 + 5.5 \times 0.5 = 4.75$$

$$z(x_1) = z(1.5) = 4 + 7.5 \times 0.5 = 7$$

Iteration 2

$$h = 0.5$$

$$x_1 = 1.5, \quad y_1 = 4.75, \quad z_1 = 7.75$$

$$m_1(1) = z_1 = 7.75$$

$$m_1(2) = 6x_1 = 9$$

$$m_2(1) = z_1 + hm_1(2) = 7.75 + 0.5 \times 9 = 12.25$$

$$m_2(2) = 6(x_1 + h) = 12$$

$$m(1) = \frac{7.75 + 12.25}{2} = 10$$

$$m(2) = \frac{9+12}{2} = 10.5$$

This gives $B_2 = 9.75$ which is greater than $B = 9$.

Now, let us have the third estimate of $z(1) = M_3$ using the relationship (14.4)

$$M_3 = M_2 - \frac{B_2 - B}{B_2 - B_1} \times (M_2 - M_1)$$

$$= 4 - \frac{(9.75 - 9)}{(9.75 - 7.75)} (4 - 2)$$

$$= 4 - 0.75 = 3.25$$

The new estimate for $z(1) = y'(1) = 3.25$

Iteration 1

$$h = 0.5$$

$$x_0 = 1, \quad y_0 = 2, \quad z_0 = 3.25$$

$$m_1(1) = z_0 = 3.25$$

$$m_1(2) = 6x_0 = 6$$

$$m_2(1) = z_0 + hm_1(2) = 3.25 + 0.5 \times 6 = 6.25$$

$$m_2(2) = 6(x_0 + h) = 9$$

$$m(1) = \frac{3.25 + 6.25}{2} = 4.75$$

$$m(2) = \frac{6+9}{2} = 7.5$$

$$y(1.5) = 2 + 4.75 \times 0.5 = 4.375$$
$$z(1.5) = 3.25 + 7.5 \times 0.5 = 7$$

Iteration 2

$$h = 0.5$$
$$x_1 = 1.5, \quad y_1 = 3.75, \quad z_1 = 7$$
$$m_1(1) = z_1 = 7$$
$$m_1(2) = 6x_1 = 6 \times 1.5 = 9$$
$$m_2(1) = z_1 + hm_1(2) = 7 + 0.5 \times 9 = 11.5$$
$$m_2(2) = 6(z_1 + h) = 12$$
$$m(1) = \frac{7 + 11.25}{2} = 9.25$$
$$m(2) = \frac{9 + 12}{2} = 10.5$$
$$y(2) = 4.375 + 9.25 \times 0.5 = 9$$

The solution is $y(1) = 2$, $y(1.5) = 4.375$, $y(2) = 9$ The exact solution is $y(x) = x^3 + 1$ and therefore $y(1.5) = 4.375$.

The sequence of procedures for implementir the shooting method is given in Algorithm 14.1.

Shooting Method

1. Convert the problem into an initial-value problem.
2. Initialise the variables including two guesses at the initial slope.
3. Solve the equations with these guesses using either a one-step or a multistep method.
4. Interpolate from these results to find an improved value of the slopes obtained.
5. Repeat the process until a specified accuracy in the final function value is obtained (or until a limit to the number of iterations is reached).

Algorithm 14.1

14.3 FINITE DIFFERENCE METHOD

In this method, the derivatives are replaced by their finite difference equivalents, thus converting the differential equation into a system of algebraic equations. For example, we can use the following "central difference" approximations:

$$y_i' = \frac{y_{i+1} - y_{i-1}}{2h} \tag{14.5}$$

$$y_i'' = \frac{y_{i+1} - 2y_i - y_{i-1}}{h^2} \tag{14.6}$$

These are second-order equations and the accuracy of estimates can be improved by using higher-order equations.

The given interval (a, b) is divided into n subintervals, each of width h. Then

$$x_i = x_0 + ih = a + ih$$
$$y_i = y(x_i) = y(a + ih)$$
$$y_0 = y(a)$$
$$y_n = y(a + nh) = y(b)$$

This is illustrated in Fig. 14.2. The difference equation is written for each of the internal points $i = 1, 2, ..., n - 1$. If the *DE* is linear, this will result with $(n - 1)$ unknowns $y_1, y_2, ..., y_{n-1}$. We can solve for these unknowns using any of the elimination methods.

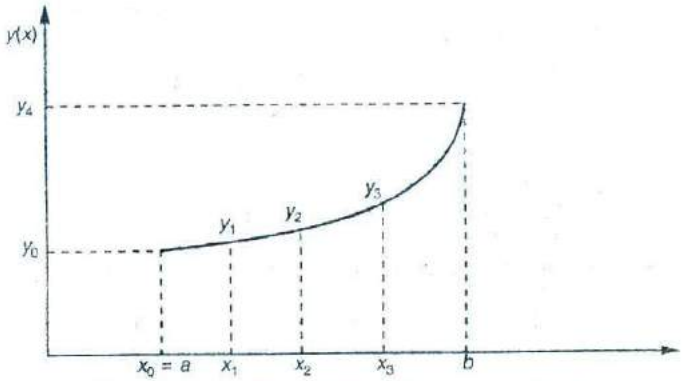


Fig. 14.2 Solution of DE by finite difference method

Note that smaller the size of h, more the subintervals and, therefore, more are the equations to be solved. However, a smaller h yields better estimates.

Example 14.2

Given the equation

$$\frac{d^2y}{dx^2} = e^{x^2} \quad \text{with} \quad y(0) = 0, \quad y(1) = 0$$

estimate the values of $y(x)$ at $x = 0.25$, 0.5 and 0.75.

We know that

$$y_0 = y(0) = 0$$
$$y_1 = y(0.25)$$
$$y_2 = y(0.5)$$
$$y_3 = y(0.75)$$
$$y_4 = y(1) = 0$$
$$h = 0.25$$

$$y_i'' = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = e^{x^2}$$

$i = 1,\ x = 0.25$

$$y_1'' = \frac{y_2 - 2y_1 + y_0}{0.0625} = e^{(0.25)^2} = 1.0645$$

$$\boxed{y_2 - 2y_1 + y_0 = 0.0665} \quad (1)$$

$i = 2,\ x = 0.50$

$$y_2'' = \frac{y_3 - 2y_2 + y_1}{0.0625} = e^{(0.5)^2} = 1.2840$$

$$\boxed{y_3 - 2y_2 + y_1 = 0.0803} \quad (2)$$

$i = 3,\ x = 0.75$

$$y_3'' = \frac{y_4 - 2y_3 + y_2}{0.0625} = e^{(0.75)^2} = 1.7551$$

$$\boxed{y_4 - 2y_3 + y_2 = 0.1097} \quad (3)$$

Letting $y_0 = 0$ and $y_4 = 0$, we have the following system of three equations.

$$-2y_1 + y_2 = 0.0665$$
$$y_1 - 2y_2 + y_3 = 0.0803$$
$$y_2 - 2y_3 = 0.1097$$

Solution of these equations results in

$$y_1 = y(0.25) = -0.1175$$
$$y_2 = y(0.50) = -0.1684$$
$$y_3 = y(0.75) = -0.1391$$

The major steps of finite difference method are given in Algorithm 14.2.

Finite Difference Method

1. Divide the given interval into n subintervals.
2. At each point of x, obtain difference equation using a suitable difference formula. This will result in $(n-1)$ equations with $(n-1)$ unknowns, $y_1, y_2, \ldots, y_n - 1$.
3. Solve for y_i, $i = 1, 2, \ldots, n-1$ using any of the standard elimination methods.

Algorithm 14.2

14.4 SOLVING EIGENVALUE PROBLEMS

As mentioned earlier, some boundary value problems, when simplified, may result in a set of homogeneous equation of the type

$$\begin{aligned}
(a_{11} - \lambda)x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &= 0 \\
a_{21}x_1 + (a_{22} - \lambda)x_2 + \ldots + a_{2n}x_n &= 0 \\
&\cdots \\
&\cdots \\
a_{n1}x_1 + a_{n2}x_2 + \ldots + (a_{nn} - \lambda)x_n &= 0
\end{aligned} \tag{14.7}$$

where λ is a scalar constant. Equation (14.7) may be expressed as

$$[\mathbf{A} - \lambda \mathbf{I}]\,[\mathbf{X}] = 0 \tag{14.8}$$

where $\mathbf{I}$ is the *identity matrix* and $[\mathbf{A} - \lambda \mathbf{I}]$ is called the *characteristic matrix* of the coefficient matrix $\mathbf{A}$.

The homogeneous Eq. (14.8) will have a non-trivial solution if, and only if, the characteristic matrix is singular. That is, the matrix $[\mathbf{A} - \lambda \mathbf{I}]$ is not invertible. Then, we have

$$|\mathbf{A} - \lambda \mathbf{I}| \tag{14.9}$$

Expansion of the determinant will result in a polynomial of degree n in λ.

$$\lambda^n - p_1\lambda^{n-1} p_2\lambda^{n-2} - \ldots - p_{n-1}\lambda - p_n = 0 \tag{14.10}$$

Equation (14.10) will have n roots $\lambda_1, \lambda_2, \ldots, \lambda_n$. The equation is known as the *characteristic polynomial* (or *characteristic equation*) and the roots are known as the *eigenvalues or characteristic values* of the matrix A. The solution vectors $X_1, X_2, \ldots, X_n$ corresponding to the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ are called the *eigenvectors*.

The roots representing the eigenvalues may be real distinct, real repeated, or complex, depending on the nature of the coefficient matrix A. The coefficients p_i of the characteristic polynomial are functions of the matrix elements a_{ij} and must be determined before the polynomial can be used.

Example 14.3 illustrates the procedure of evaluation of eigenvalues and eigenvectors of a simple system.

Example 14.3

Find the eigenvectors of the following system:

$$8x_1 - 4x_2 = \lambda x_1$$
$$2x_1 + 2x_2 = \lambda x_2$$

The characteristic equation of the given system is

$$\begin{vmatrix} 8-\lambda & -4 \\ 2 & 2-\lambda \end{vmatrix} = 0$$

That is

$$(8 - \lambda)(2 - \lambda) + 8 = 0$$

or

$$\lambda^2 - 10\lambda + 8 = 0$$

The roots are

$$\lambda_1 = 6$$
$$\lambda_2 = 4$$

For $\lambda = \lambda_1 = 6$, we get

$$2x_1 - 4x_2 = 0$$
$$2x_1 - 4x_2 = 0$$

Therefore $x_1 = 2x_2$ and the corresponding eigenvector is

$$X_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Similarly, for $\lambda = \lambda_2 = 4$, we get $x_1 = x_2$ and the eigenvector is

$$X_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

The process of finding the eigenvalues and eigenvectors of large matrices is complex and involves a multistep procedure. There are several methods available and a discussion on all these methods will be beyond the scope of this book. We consider, in the next two sections, the following two methods:

1. Polynomial method
2. Power method

14.5 POLYNOMIAL METHOD

The polynomial method consists of the following three steps:

1. Determine the coefficients p_i of the characteristic polynomial using the Fadeev-Leverrier method
2. Evaluate the roots (eigenvalues) of the characteristic polynomial using any of the root-finding techniques

3. Calculate the eigenvectors using any of the reduction techniques such as Gauss elimination

The Fadeev-Leverrier Method

The Fadeev-Leverrier method evaluate the coefficients p_i, $i = 1, 2, ..., n$, of the characteristic polynomial

$$\lambda^n - p_1\lambda^{n-1} - p_2\lambda^{n-2} - .... - p_n = 0$$

The method consists of generating a sequence of matrices A_i that can be employed to determine the p_i values. The process is as follows:

$$A_1 = \mathbf{A} \qquad (14.11)$$
$$p_1 = t_r A_1$$

Remaining values ($i = 2, 3, ..., n$) are evaluated from the recursive equations:

$$A_i = \mathbf{A}(A_{i-1} - p_{i-1}I)$$
$$p_i = \frac{t_r A_i}{i} \qquad (14.12)$$

where $t_r A_i$ is the *trace* of the matrix A_i. Remember, the trace is the sum of the diagonal elements of the matrix.

Example 14.4

Determine the coefficients of the characteristic polynomial of the system

$$(-1 - \lambda)x_1 = 0$$
$$x_1 + (-2 - \lambda)x_2 + 3x_3 = 0$$
$$2x_2 + (-3 - \lambda)x_3 = 0$$

using the Fadeev-Leverrier method.

The given system is a third-order one and, therefore, the characteristic polynomial takes the form

$$\lambda^3 - p_1\lambda^2 - p_2\lambda - p_3 = 0$$

The matrix A is given by

$$\mathbf{A} = \begin{bmatrix} -1 & 0 & 0 \\ 1 & -2 & 3 \\ 0 & 2 & -3 \end{bmatrix}$$

By using the Eq. (14.11),

$$A_1 = \mathbf{A}$$
$$p_1 = t_r A_1 = -6$$

By using the Eq. (14.12),

$$A_2 = \mathbf{A}(A_1 - p_1 I)$$

$$= \begin{bmatrix} -1 & 0 & 0 \\ 1 & -2 & 3 \\ 0 & 2 & -3 \end{bmatrix} \left\{ \begin{bmatrix} -1 & 0 & 0 \\ 1 & -2 & 3 \\ 0 & 2 & -3 \end{bmatrix} \right\} - \begin{bmatrix} -6 & 0 & 0 \\ 0 & -6 & 0 \\ 0 & 0 & -6 \end{bmatrix}$$

$$= \begin{bmatrix} -1 & 0 & 0 \\ 1 & -2 & 3 \\ 0 & 2 & -3 \end{bmatrix} \begin{bmatrix} 5 & 0 & 0 \\ 1 & 4 & 3 \\ 0 & 2 & 3 \end{bmatrix}$$

$$= \begin{bmatrix} -5 & 0 & 0 \\ -3 & -2 & 3 \\ 2 & 2 & -3 \end{bmatrix}$$

$$p_2 = \frac{t_r A_2}{2} = -5$$

Similarly,

$$A_3 = \mathbf{A}(A_2 - p_2 I)$$

$$= \begin{bmatrix} -1 & 0 & 0 \\ 1 & -2 & 3 \\ 0 & 2 & -3 \end{bmatrix} \left\{ \begin{bmatrix} -5 & 0 & 0 \\ -3 & -2 & 3 \\ 2 & 2 & -3 \end{bmatrix} \right\} - \begin{bmatrix} -5 & 0 & 0 \\ 0 & -5 & 0 \\ 0 & 0 & -5 \end{bmatrix}$$

$$= \begin{bmatrix} -1 & 0 & 0 \\ 1 & -2 & 3 \\ 0 & 2 & -3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ -3 & 3 & 3 \\ 2 & 2 & 2 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 \\ 6 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$p_3 = \frac{t_r A_3}{3} = 0$$

Therefore, the characteristic polynomial is

$$\lambda^3 + 6\lambda^2 + 5\lambda = 0$$

or

$$\lambda(\lambda^2 + 6\lambda + 5) = 0$$

Evaluating the Eigenvalues

Let us consider the characteristic polynomial obtained in Example 14.4.

$$\lambda(\lambda^2 + 6\lambda + 5) = 0$$

One of the roots is $\lambda_1 = 0$. The other two roots can be obtained using the familiar quadratic formula as

$$\lambda_2 = -1$$
$$\lambda_3 = -5$$

Since it is a quadratic equation, we can solve it using the quadratic formula. However, if the polynomial is of higher order, the roots may be evaluated using the techniques discussed in Chapter 6. We may use either the Newton-Raphson method with synthetic division or the Bairstow method.

Remember that the sum of the eigenvalues of a matrix is equal to the trace of that matrix. In the problem discussed above,

$$\text{trace of } \mathbf{A} = -1 - 2 - 3 = -6$$
$$\text{Sum of eigenvalues} = 0 - 1 - 5 = -6$$

Determining the Eigenvectors

Once eigenvalues are evaluated, eigenvectors corresponding to these eigenvalues may be obtained by applying Gauss elimination method to the homogeneous equations. Example 14.5 illustrates this.

Example 14.5

Determine the eigenvectors for the system discussed in Example 14.4.

The eigenvalues are:

$$\lambda_1 = 0, \quad \lambda_2 = -1, \quad \lambda_3 = -5$$

Eigenvector 1 ($\lambda_1 = 0$)

The system of equations for $\lambda_1 = 0$ is

$$-x_1 = 0$$
$$x_1 - 2x_2 + 3x_3 = 0$$
$$2x_2 - 3x_3 = 0$$

This is equivalent to the system of two equations

$$x_1 = 0$$
$$2x_2 - 3x_3 = 0$$

Choosing $x_2 = 1,$

$$x_3 = 2/3 = 0.6667$$

Eigenvector 2 ($\lambda_2 = -1$)

$$x_1 - x_2 + 3x_3 = 0$$
$$2x_2 - 2x_3 = 0$$

Choosing $x_2 = 1$, we get $x_3 = 1$ and $x_1 = -2$

Eigenvector 3 ($\lambda_3 = -5$)

$$4x_1 = 0$$
$$x_1 + 3x_2 + 3x_3 = 0$$
$$2x_2 + 2x_3 = 0$$

$x_1 = 0$ and choosing $x_2 = 1$, we get $x_3 = -1$

The three eigenvectors are:

$$X_1 = \begin{bmatrix} 0 \\ 1 \\ 0.6667 \end{bmatrix}$$

$$X_2 = \begin{bmatrix} -2 \\ 1 \\ 1 \end{bmatrix}$$

$$X_3 = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$$

Computing Algorithm

The computing algorithm for polynomial method combines three different algorithms discussed so far. Algorithm 14.3 lists the major steps involved in implementing the polynomial method of finding eigenvalues and associated eigenvectors.

Polynomial Method

1. Input order of the matrix and the elements of the matrix.
2. Determine the coefficients of the characteristic polynomial by using the Fadeev-Leverrier method.
3. Evaluate the roots of the polynomial using the Bairstow method (use Algorithm 6.10).
4. For each eigenvalue, construct the system of equations and then solve for the eigenvector using the Gauss elimination method (use Algorithm 7.2).
5. Print eigenvalues and the associated eigenvectors.

Algorithm 14.3

14.6 POWER METHOD

Power method is a 'single value' method used for determining the 'dominant' eigenvalue of a matrix. It is an iterative method implemented using an initial starting vector X. The starting vector can be arbitrary if no suitable approximation is available. Power method is implemented as follows:

$$Y = AX \tag{14.13}$$

$$X = \frac{1}{k}Y \tag{14.14}$$

The new value of X obtained from Eq. (14.14) is used in Eq. (14.13) to compute a new value of Y and the process is repeated until the desired level of accuracy is obtained. The parameter k, known as the *scaling factor*, is the element of Y with the largest magnitude.

Let us assume that the eigenvalues are $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n|$ and the corresponding eigenvectors are $X_1, X_2, \ldots, X_n$. After repeated applications of Eqs (14.13) and (14.14), the vector X converges to X_1 and k converges to λ_1.

Example 14.6

Find the largest eigenvalue λ_1 and the corresponding eigenvector V_1 of the matrix

$$\begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

using the power method.

Let us assume the starting vector as

$$X = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Equations (14.13) and (14.14) are repeatedly used as follows

Iteration 1

$$Y = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix}$$

$$X = \frac{1}{2} \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.5 \\ 0 \end{bmatrix}$$

Iteration 2

$$Y = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0.5 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 2.5 \\ 0 \end{bmatrix}$$

$$X = \frac{1}{2.5} \begin{bmatrix} 2 \\ 2.5 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.8 \\ 1 \\ 0 \end{bmatrix}$$

The process is continued and the results are tabulated below:

Iteration		*1*	*2*	*3*	*4*	*5*	*6*	*7*
		2.0	2.0	2.8	2.86	2.98	2.98	3.0
Y		1.0	2.5	2.6	2.93	2.96	2.99	3.0
		0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0	1.0	0.8	1.0	0.98	1.0	1.0	1.0
X	1	0.5	1.0	0.93	1.0	0.99	1.0	1.0
	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

The final entry in the table shows that $\lambda_1 = 3.0$ (element of $\boldsymbol{Y}$ with largest magnitude) and the corresponding eigenvector is the last $\boldsymbol{X}$. That is,

$$\boldsymbol{X}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

Algorithm 14.4 gives an implementation of the power method. Note that the stopping criterion is based on the successive values of the vector $\boldsymbol{X}$. There might be circumstances where the process will not converge at all (or where it will converge very slowly). Therefore, it is necessary to put a limitation on the number of iterations.

Power Method

1. Input matrix **A**, initial vector $\boldsymbol{X}$, error tolerance (EPS) and maximum iterations permitted (MAXIT).
2. Compute $\boldsymbol{Y} = \mathbf{A}\boldsymbol{X}$
3. Find the element k of $\boldsymbol{Y}$ that is largest in magnitude.
4. Compute $\boldsymbol{X} = \boldsymbol{Y}/k$
5. If $|\boldsymbol{X} - \boldsymbol{X}_{old}| <$ EPS or Iterations > MAXIT
   write k and $\boldsymbol{X}$
   else
   go to step 2

Algorithm 14.4

Engineers often come across cases where they are interested in the smallest eigenvalue of the system. The smallest eigenvalue can be determined by applying the power method to the matrix inverse of [A].

14.7 SUMMARY

We considered two classes of problems in this chapter:

- boundary value problems
- eigenvalue problems

We presented two methods for solving boundary-value problems:

- shooting method
- finite difference method

We also discussed in detail the nature and solution of eigenvalue problems and presented two methods for evaluating eigenvalues and eigenvectors:

- power method
- polynomial method

Key Terms

Boundary value problem
Characteristic equation
Characteristic matrix
Characteristic polynomial
Characteristic values
Eigenvalue problem
Eigenvalues
Eigenvector
Fadeev-Leverrier method
Finite difference method
Identity matrix
Initial-value problem
Polynomial method
Power method
Scaling factor
Shooting method
Trace of the matrix

REVIEW QUESTIONS

1. What is a boundary-value problem? How is it different from an initial-value problem?
2. State the two popular methods used for solving boundary-value problems.
3. What are eigenvalue problems? How are they different from boundary-value problems?
4. State at least two methods used for solving eigenvalue problems.
5. Describe the shooting method with graphical illustration.
6. Explain the concept employed in the finite difference method.
7. Define the following:
   (a) Identity matrix
   (b) Characteristic matrix
   (c) Characteristic polynomial or equation
   (d) Eigenvalues
   (e) Eigenvectors
   (f) Trace of a matrix
8. What is Fadeev-Leverrier method used for? Explain.
9. Describe the algorithm of polynomial method used for solving eigenvalue problems.
10. Describe the implementation of power method with the help of a flow chart.

REVIEW EXERCISES

1. Solve the following equations using the shooting method.

   (a) $\frac{d^2y}{dx^2} = 6x + 4, \quad y(0) = 2 \quad \text{and} \quad y(1) = 5$

   (b) $\frac{d^2y}{dx^2} = 12x^2, \quad y(1) = 2 \quad \text{and} \quad y(2) = 17$

2. Use the finite difference approach to solve the equations in Exercise 1 with $\Delta x = 0.2$.
3. Use the shooting method to solve the following differential equation:

$$\frac{d^2y}{dx^2} + 2\frac{dy}{dx} - \frac{y}{2} - 2.5 = 0$$

   given the boundary conditions $y(0) = 10$ and $y(10) = 6$.
4. Solve Exercise 3 using the finite difference method with $\Delta x = 2$.
5. Given the boundary-value problem

$$\frac{d^2y}{dx^2} = 3x + 4y, \quad y(0) = 1 \quad \text{and} \quad y(1) = 1$$

   obtain its solution in the range $0 \le x \le 1$ with $\Delta x = 0.25$ using
   (a) shooting method
   (b) finite difference method
6. Given the equation

$$x^2\frac{d^2y}{dx^2} - 2x\frac{dy}{dx} - 2y + x^2 \sin x = 0$$

   and boundary conditions $y(1) = 1$ and $y(2) = 2$, estimate $y(1.25)$, $y(1.5)$ and $y(1.75)$ using the shooting method.
7. Solve the following boundary-value problems using a suitable method.

   (a) $\frac{d^2y}{dx^2} - 6y^2 = 0, \quad y(1) = 1 \quad \text{and} \quad y(2) = 0.25$

   (b) $\frac{d^2y}{dx^2} + e^{-2y} = 0, \quad y(1) = 0 \quad \text{and} \quad y(2) = 1$

   (c) $\frac{d^2y}{dx^2} - 3\frac{dy}{dx} + 2y = 2, \quad y(0) = 1 \quad \text{and} \quad y(1) = 4$

   (d) $\frac{d^2y}{dx^2} - x\frac{dy}{dx} + y = -x^2, \quad y(0) = -2 \quad \text{and} \quad y(1) = 1$

8. Find the characteristic polynomials of the following systems using Fadeev-Leverrier method.

   (a) $2x_1 + 8x_2 + 10x_3 = \lambda x_1$

$$8x_1 + 3x_2 + 4x_3 = \lambda x_2$$
$$10x_1 + 4x_2 + 7x_3 = \lambda x_3$$

(b) $16x_1 - 24x_2 + 18x_3 = \lambda x_1$
$$3x_1 - 2x_2 = \lambda x_2$$
$$-9x_1 + 18x_2 - 17x_3 = \lambda x_3$$

(c) $2x_1 + 2x_2 + 2x_3 = \lambda x_1$
$$2x_1 + 5x_2 + 5x_3 = \lambda x_2$$
$$2x_1 + 5x_2 + 1x_3 = \lambda x_3$$

9. Evaluate the eigenvectors of the systems given in Exercise 8.
10. Find the largest eigenvalue and the corresponding eigenvector of the following matrices using the power method.

(a) $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$

(b) $B = \begin{bmatrix} -1 & 0 & 0 \\ 1 & -2 & 3 \\ 0 & 2 & -3 \end{bmatrix}$

(c) $C = \begin{bmatrix} -13 & 3 & -5 \\ 0 & -4 & 0 \\ 15 & -9 & 7 \end{bmatrix}$

PROGRAMMING PROJECTS

1. Develop a program to implement the shooting method algorithm for a linear, second-order ordinary differential equation.
2. Develop a program to implement the finite difference method of solving a linear, second-order ordinary differential equation.
3. Develop a modular program which uses the following subprograms to implement the polynomial method.
   (a) Subprogam to obtain the coefficients of the characteristic polynomial using Fadeev-Leverrier method.
   (b) Subprogram to evaluate the roots of the characteristic polynomial (use Bairstow method).
   (c) Subprogram to determine the eigenvectors (use Gauss elimination method).

   You may also use subprograms to receive input information and print output information.
4. Develop a user-friendly program to evaluate the largest eigenvalue and the corresponding eigenvector using the power method.
5. Develop a program to compute the smallest eigenvalue using the power method.

CHAPTER 15

Solution of Partial Differential Equations

15.1 NEED AND SCOPE

Many physical phenomena in applied science and engineering when formulated into mathematical models fall into a category of systems known as *partial differential equations*. A partial differential equation is a differential equation involving more than one independent variable. These variables determine the behaviour of the dependent variable as described by their *partial derivatives* contained in the equation. Some of the problems which lend themselves to partial differential equations include:

1. Study of displacement of a vibrating string,
2. Heat flow problems,
3. Fluid flow analysis,
4. Flectrical potential distribution,
5. Analysis of torsion in a bar subject to twisting,
6. Study of diffusion of matter, and so on.

Most of these problems can be formulated as second-order partial differential equations (with the highest order of derivative being the second). If we represent the dependent variable as f and the two independent variables as x and y, then we will have three possible second-order partial derivatives $\frac{\partial^2 f}{\partial x^2}$, $\frac{\partial^2 f}{\partial x\, \partial y}$ and $\frac{\partial^2 f}{\partial y^2}$ in addition to the two first-order partial derivatives $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$.

We can write a second-order equation involving two independent variables in general form as

$$a\frac{\partial^2 f}{\partial x^2} + b\frac{\partial^2 f}{\partial x\,\partial y} + c\frac{\partial^2 f}{\partial y^2} = F\left(x, y, f, \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) \tag{15.1}$$

where the coefficients a, b, and c may be constants or functions of x and y. Depending on the values of these coefficients, Eq. (15.1) may be classified into one of the three types of equations, namely, *elliptic*, *parabolic*, and *hyperbolic*.

Elliptic,	if $b^2 - 4ac < 0$
Parabolic,	if $b^2 - 4ac = 0$
Hyperbolic,	if $b^2 - 4ac > 0$

If a, b, and c are functions of x and y, then depending on the values of these coefficients at various points in the domain under consideration, an equation may change from one classification to another.

Solution of partial differential equations is too important to ignore but too difficult to cover in depth in an introductory book. Since the application of analytical methods becomes more complex, we seek the help of numerical techniques to solve partial differential equations. There are basically two numerical techniques, namely, *finite-difference* method and *finite-element* method that can be used to solve partial differential equations (PDEs). Although the finite-element method is very important for solving equations where regions are irregular, discussion on this technique is beyond the scope of this book. We will discuss here the application of finite-difference methods only, which are based on formulae for approximating the first and second derivatives of a function. We will also consider problems, only those where the coefficients a, b, and c are constants.

15.2 DERIVING DIFFERENCE EQUATIONS

In this section, we will discuss two-dimensional problems only. Consider a two-dimensional solution domain as shown in Fig. 15.1. The domain is split into regular rectangular grids of width h and height k. The pivotal values at the points of intersection (known as grid points or nodes) are denoted by f_{ij} which is a function of the two space variables x and y.

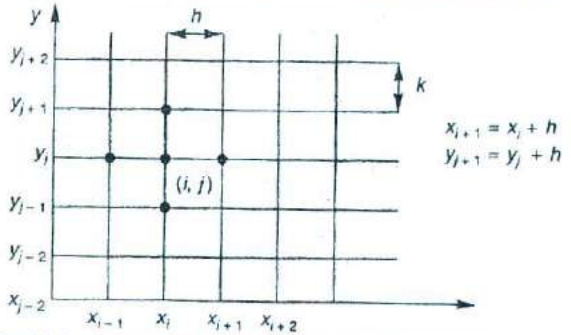


Fig. 15.1 Two-dimensional finite difference grid

In the finite difference method, we replace derivatives that occur in the PDE by their finite difference equivalents. We then write the difference equation corresponding to each "grid point" (where derivative is required) using function values at the surrounding grid points. Solving these equations simultaneously gives values for the function at each grid point.

We have discussed in Chapter 11 that, if the function $f(x)$ has a continuous fourth derivative, then its first and second derivatives are given by the following central difference approximations.

$$f'(x_i) = \frac{f(x_i + h) - f(x_i - h)}{2h}$$

or

$$f_i' = \frac{f_{i+1} - f_{i-1}}{2h} \tag{15.2}$$

$$f''(x_i) = \frac{f(x_i + h) - 2f(x_i) + f(x_i - h)}{h^2}$$

or

$$f_i'' = \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} \tag{15.3}$$

The subscript on f indicates the x-value at which the function is evaluated.

When f is a function of two variables x and y, the partial derivatives of f with respect x (or y) are the ordinary derivatives of f with respect to x (or y) when y (or x) does not change. We can use Eqs (15.2) and (15.3) in the x-direction to determine derivatives with respect to x and in the y-direction to determine derivatives with respect to y. Thus, we have

$$\frac{\partial f(x_i, y_j)}{\partial x} = f_x(x_i, y_j) = \frac{f(x_{i+1}, y_j) - f(x_{i-1}, y_j)}{2h}$$

$$\frac{\partial f(x_i, y_j)}{\partial y} = f_y(x_i, y_j) = \frac{f(x_i, y_{j+1}) - f(x_i, y_{j-1})}{2k}$$

$$\frac{\partial^2 f(x_i, y_j)}{\partial x^2} = f_{xx}(x_i, y_j) = \frac{f(x_{i+1}, y_j) - 2f(x_i, y_j) + f(x_{i-1}, y_j)}{h^2}$$

$$\frac{\partial^2 f(x_i, y_j)}{\partial y^2} = f_{yy}(x_i, y_j) = \frac{f(x_i, y_{j+1}) - 2f(x_i, y_j) + f(x_i, y_{j-1})}{k^2}$$

$$\frac{\partial^2 f(x_i, y_j)}{\partial x\, \partial y} = \frac{f(x_{i+1}, y_{j+1}) - f(x_{i+1}, y_{j-1}) - f(x_{i-1}, y_{j+1}) + f(x_{i-1}, y_{j-1})}{4hk}$$

It is convenient to use double subscripts i, j on f to indicate x and y values. Then, the above equations become

$$f_{x,ij} = \frac{f_{i+1,j} - f_{i-1,j}}{2h} \tag{15.4}$$

$$f_{y,ij} = \frac{f_{i,j+1} - f_{i,j-1}}{2k} \tag{15.5}$$

$$f_{xx,ij} = \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{h^2} \tag{15.6}$$

$$f_{yy,ij} = \frac{f_{i,j+1} - 2f_{ij} + f_{i,j-1}}{k^2} \tag{15.7}$$

$$f_{xy,ij} = \frac{f_{i+1,j+1} - f_{i+1,j-1} - f_{i-1,j+1} + f_{i-1,j-1}}{4hk} \tag{15.8}$$

We will use these finite difference equivalents of the partial derivatives to construct various types of differential equations.

ELLIPTIC EQUATIONS

Elliptic equations are governed by conditions on the boundary of closed domain. We consider here the two most commonly encountered elliptic equations, namely,

Laplace's equation, and Poisson's equation.

Laplace's Equation

Equation (15.1), when $a = 1$, $b = 0$, $c = 1$, and $F(x, y, f, f_x, f_y) = 0$, becomes

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = \nabla^2 f = 0 \tag{15.9}$$

The operator

$$\nabla^2 = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right)$$

is called the *Laplacian operator* and Eq. (15.9) is called *Laplace's equation*. (Many authors use u in place of f.)

To solve the Laplace equation on a region in the xy-plane, we subdivide the region as shown in Fig. 15.1. Consider the portion of the region near (x_i, y_i). We have to approximate

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0$$

Replacing the second-order derivatives by their finite difference equivalents (from Eqs (15.6) and (15.7)) at the point (x_i, y_i), we obtain,

$$\nabla^2 f_{ij} = \frac{f_{i+1,j} - 2f_{ij} + f_{i-1,j}}{h^2} + \frac{f_{i,j+1} - 2f_{ij} + f_{i,j-1}}{k^2}$$

If we assume, for simplicity, $h = k$, then we get

$$\nabla^2 f_{ij} = \frac{1}{h^2}(f_{i+1,j} + f_{i-1,j} - 4f_{ij} + f_{i,j+1} + f_{i,j-1}) = 0 \qquad (15.10)$$

Note that Eq. (15.10) contains four neighbouring points around the central point (x_i, y_j) (on all the four sides) as shown in Fig. 15.2. Equation (15.10) is known as the *five-point difference formula* for Laplace's equation.

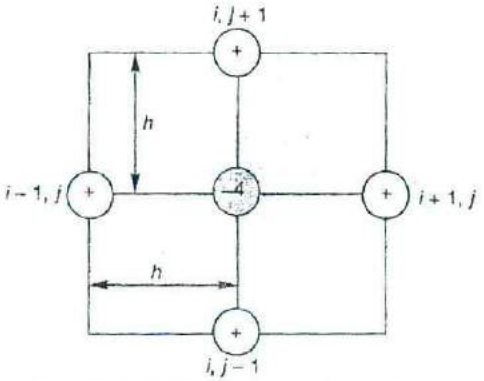


Fig. 15.2 Grid for Laplace's equation

We can also represent the relationship of pivotal values pictorially as in Eq. (15.11).

$$\nabla^2 f_{ij} = \frac{1}{h^2}\begin{Bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{Bmatrix} f_{ij} = 0 \qquad (15.11)$$

From Eq. (15.10) we can show that the function value at the grid point (x_i, y_j) is the average of the values at the four adjoining points. That is,

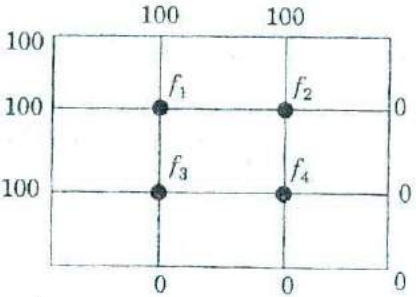
$$f_{ij} = \frac{1}{4}(f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}) \qquad (15.12)$$

To evaluate numerically the solution of Laplace's equation at the grid points, we can apply Eq. (15.12) at the grid points where f_{ij} is required (or unknown), thus obtaining a system of linear equations in the pivotal values f_{ij}. The system of linear equations may be solved using either direct methods or iterative methods.

Example 15.4

Consider a steel plate of size 15 cm × 15 cm. If two of the sides are held at 100°C and the other two sides are held at 0°C, what are the steady-state temperature at interior points assuming a grid size of 5 cm × 5 cm.

A problem with the values known on each boundary is said to have *Dirichlet boundary conditions*. The problem is illustrated below.



The system of equations is as follows:

At point 1: $f_2 + f_3 - 4f_1 + 100 + 100 = 0$

At point 2: $f_1 + f_4 - 4f_2 + 100 + 0 = 0$

At point 3: $f_1 + f_4 - 4f_3 + 100 + 0 = 0$

At point 4: $f_2 + f_3 - 4f_4 + 0 + 0 = 0$

That is,

$$-4f_1 + f_2 + f_3 + 0 = -200$$
$$f_1 + -4f_2 + 0 + f_4 = -100$$
$$f_1 + 0 - 4f_3 + f_4 = -100$$
$$0 + f_2 + f_3 - 4f_4 = 0$$

Solution of this system are

$$f_1 = 75 \qquad f_2 = 50$$
$$f_3 = 50 \qquad f_4 = 25$$

Note that there is a symmetry in the temperature distribution, i.e. it can be stated that

$$f_2 = f_3$$

and therefore the number of equations in Example 15.1 may be reduced to three with three unknowns as shown below.

$$\begin{aligned} -4f_1 + 2f_2 &= -200 \\ f_1 - 4f_2 + f_4 &= -100 \\ f_2 - 2f_4 &= 0 \end{aligned} \tag{15.13}$$

Liebmann's Iterative Method

We know that a diagonally dominant system of linear equations can be solved by iteration methods such as Gauss-Seidel method. When such an iteration is applied to Laplace's equation, the iterative method is called *Liebmann's iterative method.*

To obtain the pivotal values of f by Liebmann's iterative method, we solve for f_{ij} the equations obtained from Eq. (15.10). That is,

$$f_{ij} = \frac{1}{4}(f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}) \tag{15.14}$$

The value f_{ij} at the point ij is the average of the values of f at the four adjoining points. If we know the "initial values" of the functions at the right-hand side of Eq. (15.13), we can estimate the value f at the point ij. We can substitute the values thus obtained into the right-hand side to achieve improved approximations. This process may continue till the values f_{ij} converge to constant values.

Initial values may be obtained by either taking *diagonal average* or *cross average* of the adjoining four points.

Example 15.2

Solve the problem in Example 15.1 using Liebmann's iterative method correct to one decimal place.

By applying Eq. (15.14) to every grid point, we obtain

$$\begin{aligned} f_1 &= \frac{f_2 + f_3 + 200}{4} \\ f_2 &= \frac{f_1 + f_4 + 100}{4} \\ f_3 &= \frac{f_1 + f_4 + 100}{4} \\ f_4 &= \frac{f_2 + f_3}{4} \end{aligned} \tag{15.15}$$

Appropriate initial values for the iterative solution are obtained by taking diagonal average at 1 and cross average at other points, assuming first $f_4 = 0$.

$$f_1 = \frac{1}{4}(100 + 100 + 100 + 0) = 75.00 \text{ (average)}$$

$$f_2 = \frac{1}{4}(75 + 100 + 0 + 0) = 43.75$$

$$f_3 = \frac{1}{4}(100 + 75 + 0 + 0) = 43.75$$

$$f_4 = \frac{1}{4}(43.75 + 43.75 + 0 + 0) = 21.88$$

Note that f_2, f_3, and f_4 are computed using the latest values on the right-hand side.

Using these initial values in Eq. (15.15) and performing iterations gives the values as shown in Table 15.1.

Table 15.1

f_i	*Initial Values*	*Iterations*			
		1	*2*	*3*	*4*
f_1	75.00	71.88	74.22	74.81	74.95
f_2	43.75	48.44	49.61	40.90	49.98
f_3	43.75	48.44	49.61	49.90	49.98
f_4	21.88	24.22	24.81	24.95	24.99

The process may be continued till we get identical values in the last two columns. Note that the values are approaching to correct answers obtained in Example 15.1.

Poisson's Equation

Equation (15.1), when $a = 1$, $b = 0$, $c = 1$ and $F(x, y, f, f_x, f_y) = g(x, y)$, becomes

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = g(x, y) \tag{15.16}$$

or

$$\nabla^2 f = g(x, y)$$

Equation (15.16) is called *Poisson's equation*. Using the notation $g_{ij} = g(x_i, y_j)$, Eq. 15.10 used for Laplace's equation may be modified to solve Eq. 15.16. The finite difference formula for solving Poisson's equation then takes the form

$$\boxed{f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1} - 4f_{ij} = h^2 g_{ij}} \tag{15.17}$$

By applying the replacement formula to each grid point in the domain of consideration, we will get a system of linear equations in terms of f_{ij}. These equations may be solved either by any of the elimination methods or by any iteration techniques as done in solving Laplace's equation.

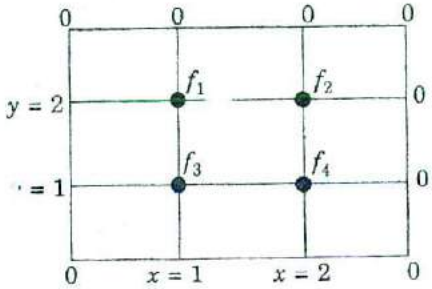
Example 15.6

Solve the Poisson equation

$$\nabla^2 f = 2x^2y^2$$

over the square domain $0 \le x \le 3$ and $0 \le y \le 3$ with $f = 0$ on the boundary and $h = 1$.

The domain is divided into squares of one unit size as illustrated below:



By applying Eq. (15.17) at each grid point, we get the following set of equations:

Point 1: $0 + 0 + f_2 + f_3 - 4f_1 = 2(1)^2(2)^2$

i.e. $f_2 + f_3 - 4f_1 = 8$ (a)

Point 2: $0 + 0 + f_1 + f_4 - 4f_2 = 2(2)^2(2)^2$

i.e. $f_1 - 4f_2 + f_4 = 32$ (b)

Point 3: $0 + 0 + f_1 + f_4 - 4f_3 = 2(1)^2(1)^2$

i.e. $f_1 - 4f_3 + f_4 = 2$ (c)

Point 4: $0 + 0 + f_2 + f_3 - 4f_4 = 2(2)^2(1)^2$

i.e. $f_2 + f_3 - 4f_4 = 8$ (d)

Rearranging the equations (a) to (d), we get

$$\begin{aligned} -4f_1 + f_2 + f_3 &= 8 \\ f_1 - 4f_2 + f_4 &= 32 \\ f_1 - 4f_3 + f_4 &= 2 \\ f_2 + f_3 - 4f_4 &= 8 \end{aligned}$$

Solving these equations by elimination method, we get the answers.

$$f_1 = -\frac{22}{4}, \qquad f_2 = -\frac{43}{4}$$

$$f_3 = -\frac{13}{4}, \qquad f_4 = -\frac{22}{4}$$

Example 15.4

Solve the problem in Example 15.3 by Gauss-Seidel iteration method.

By rearranging the equations used in Example 15.3, we have

$$f_1 = \frac{1}{4}(f_2 + f_3 - 8)$$

$$f_2 = \frac{1}{4}(f_1 + f_4 - 32)$$

$$f_3 = \frac{1}{4}(f_1 + f_4 - 2)$$

$$f_4 = \frac{1}{4}(f_2 + f_3 - 8)$$

Note that $f_1 = f_4$
Therefore,

$$f_1 = \frac{1}{4}(f_2 + f_3 - 8)$$

$$f_2 = \frac{1}{4}(2f_1 - 32)$$

$$f_3 = \frac{1}{4}(2f_1 - 2)$$

Assume starting values as $f_2 = 0 = f_3$

Iteration 1

$$f_1 = -2, \quad f_2 = -9, \quad f_3 = -1$$

Iteration 2

$$f_1 = -\frac{18}{4}, \quad f_2 = -\frac{41}{4}, \quad f_3 = -\frac{11}{4}$$

Iteration 3

$$f_1 = -\frac{22}{4}, \quad f_2 = -\frac{43}{4}, \quad f_3 = -\frac{13}{4}$$

Iteration 4

$$f_1 = -\frac{22}{4}, \quad f_2 = -\frac{43}{4}, \quad f_3 = -\frac{13}{4}$$

15.4 PARABOLIC EQUATIONS

Elliptic equations studied previously describe problems that are time-independent. Such problems are known as steady-state problems. But we come across problems that are not steady-state. This means that the function is dependent on both space and time. Parabolic equations, for which

$$b^2 - 4ac = 0$$

describe the problems that depend on space and time variables.

A popular case for parabolic type of equation is the study of heat flow in one-dimensional direction in an insulated rod. Such problems are governed by both boundary and initial conditions.

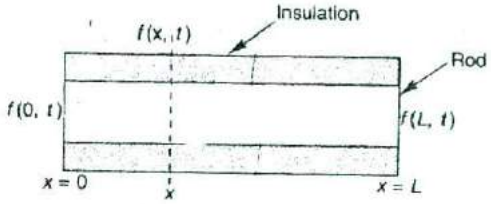


Fig. 15.3 Heat flow in a rod

Let f represent the temperature at any point in rod (Fig. 15.3) whose distance from the left end is x. Heat is flowing from left to right under the influence of temperature gradient. The temperature $f(x, t)$ in the rod at the position x and time t, is governed by the *heat equation*

$$k_1 \frac{\partial^2 f}{\partial x^2} = k_2 k_3 \frac{\partial f}{\partial t} \tag{15.18}$$

where k_1 = Coefficient of thermal conductivity; k_2 = Specific heat; and k_3 = Density of the material.

Equation (15.18) may be simplified as

$$\boxed{k f_{xx}(x, t) = f_t(x, t)} \tag{15.19}$$

where

$$k = \frac{k_1}{k_2 k_3}$$

The initial condition will be the initial temperatures at all points along the rod.

$$f(x, 0) = f(x), \qquad 0 \le x \le L$$

The boundary conditions $f(0, t)$ and $f(L, t)$ describe the temperature at each end of the rod as functions of time. If they are held constant, then

$$f(0, t) = c_1, \qquad 0 \le t < \infty$$
$$f(L, t) = c_2, \qquad 0 \le t < \infty$$

Solution of Heat Equation

We can solve the heat equation given by Eq. (15.19) using the finite difference formulae given below:

$$f_t(x, t) = \frac{f(x, t + \tau) - f(x, t)}{\tau}$$

$$= \frac{1}{\tau}(f_{i,j+1} - f_{i,j}) \tag{15.20}$$

$$f_{xx}(x, t) = \frac{f(x - h, t) - 2f(x, t) + f(x + h, t)}{h^2}$$

$$= \frac{1}{h^2}(f_{i-1,j} - 2f_{i,j} + f_{i+1,j}) \tag{15.21}$$

Substituting (15.20) and (15.21) in (15.19), we obtain

$$\frac{1}{\tau}(f_{i,j+1} - f_{i,j}) = \frac{k}{h^2}(f_{i-1,j} - 2f_{i,j} + 2f_{i+1,j}) \tag{15.22}$$

Solving for $f_{i,j+1}$

$$f_{i,j+1} = \left(1 - \frac{2\tau k}{h^2}\right) f_{ij} + \frac{\tau k}{h^2}(f_{i-1,j} + f_{i+1,j})$$

$$= (1 - 2r)f_{ij} + r(f_{i-1,j} + f_{i+1,j}) \tag{15.23}$$

where $$r = \frac{\tau k}{h^2}$$

Bender-Schmidt Method

The recurrence Eq. (15.23) allows us to evaluate f at each point x and at any time t. If we choose step sizes Δt and Δx such that

$$1 - 2r = 1 - \frac{2\tau k}{h^2} = 0 \tag{15.24}$$

Equation 15.23 simplifies to

$$\boxed{f_{i,j+1} = \frac{1}{2}(f_{i+1,j} + f_{i-1,j})} \tag{15.25}$$

Equation 15.25 is known as the *Bender-Schmidt recurrence equation*. This equation determines the value of f at $x = x_i$, at time $t = t_j + \tau$, as the average of the values right and left of x_i at time t_j.

Note that the step size in time Δt obtained from Eq. (15.24)

$$\tau = \frac{h^2}{2k}$$

gives the Eq. (15.25). Equation (15.23) is stable, if and only if the step size τ satisfies the condition $\tau \leq \frac{h^2}{2k}$.

Example 15.5

Solve the equation

$$2f_{xx}(x, t) = f_t(x, t), \qquad 0 < t < 1.5 \quad \text{and} \quad 0 < x < 4$$

given the initial condition

$$f(x, 0) = 50(4 - x), \qquad 0 \leq x \leq 4$$

and the boundary conditions

$$f(0, t) = 0, \quad 0 \le t \le 1.5$$
$$f(4, t) = 0, \quad 0 \le t \le 1.5$$

If we assume $\Delta x = h = 1$, $\Delta t = \tau$ must be

$$\tau \le \frac{1^2}{2 \times 2} = 0.25$$

Taking $\tau = 0.25$, we have

$$f_{i,j+1} = \frac{1}{2}(f_{i-1,j} + f_{i+1,j})$$

Using this formula, we can generate successfully $f(x, t)$. The estimates are recorded in Table 15.2. At each interior point, the temperature at any single point is just average of the values at the adjacent points of the previous time value.

Table 15.2

t \ x	0.0	1.0	2.0	3.0	4.0	
0.00	0.0	150.0	100.0	50.0	0.0	← $f(x,0) = 50(4 - x)$
0.25	0.0	50.0	100.0	50.0	0.0	
0.50	0.0	50.0	50.0	50.0	0.0	
0.75	0.0	25.0	25.0	25.0	0.0	
1.00	0.0	12.5	25.0	12.5	0.0	
1.25	0.0	12.5	12.5	12.5	0.0	
1.50	0.0	6.25	12.5	6.25	0.0	

The Crank-Nicholson Method

Solution of the parabolic equation given in Eq. (15.19) was solved using a forward difference formula in Eq. (15.20) for the time derivative and a central difference formula in Eq. (15.21) for the space derivative. This is called *explicit method* because all starting values are directly available from initial and boundary conditions and each new value is obtained from the values that are already known.

Accuracy of the explicit method may be improved if we use central difference formulae for both time and space derivatives. The forward difference quotient used for the time derivative (Eq. (15.20))

$$\frac{f_{i,j+1} - f_{i,j}}{\tau}$$

may be treated as central difference if we consider it to represent the *midpoint of the time interval* $(j, j + 1)$. We can also use the central difference quotient for the second derivative with distance, corresponding to the midpoint in time. Then,

$$f_t\left(x, t+\frac{\tau}{2}\right)=\frac{1}{\tau}(f_{i,j+1}-f_{i,j}) \tag{15.26}$$

The central difference quotient for the second-order space derivative is obtained by taking the average of difference quotients at the beginning and end of the time step, i.e.

Central difference at $t_{j+1/2}$ = $\frac{1}{2}$ (Central difference at t_j + Central difference at t_{j+1}).

Then,

$$k f_{xx}(x, t+\tau/2)=\frac{k}{2}\left(\frac{f_{i-1,j}+f_{i+1,j}-2f_{i,j}}{h^2}+\frac{f_{i-1,j+1}+f_{i+1,j+1}-2f_{i,j+1}}{h^2}\right) \tag{15.27}$$

Then equating Eqs (15.26) and (15.27) and substituting

$$r=\frac{\tau k}{h^2}$$

We get

$$\boxed{-rf_{i-1,j+1}+(2+2r)f_{i,j+1}-rf_{i,j+1}=rf_{i-1,j}+(2-2r)f_{i,j}+rf_{i+1,j}} \tag{15.28}$$

Equation (15.28) is called the *Crank-Nicholson formula*. If we let r =1, then Eq (15.28) simplifies to

$$\boxed{-f_{i-1,j+1}+4f_{i,j+1}-f_{i+1,j+1}=f_{i-1,j}+f_{i+1,j}} \tag{15.29}$$

The terms on the right-hand side are all known. Hence Eq. (15.29) forms a system of linear equations. The points used in the Crank-Nicholson formula are shown in Fig. 15.4. The boundary conditions are used in the first and last equations, i.e.

$$f_{1,j}=f_{1,j+1}=c_1$$
$$f_{n,j}=f_{n,j+1}=c_2$$

The Crank-Nicholson formula is called an *implicit method* because the values to be computed are not just a function of values at the previous time step, but also involve the values at the same time step which are not readily available. This requires us to solve a set of simultaneous equations at each time step.

Referring to Fig. 15.4,

$$-f_A+4f_B-f_C=f_D+f_E$$

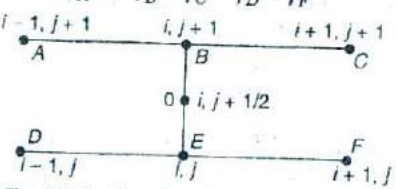


Fig. 15.4 The Crank-Nicholson grid

Example 15.6

Solve the problem in Example 15.5 by the Crank-Nicholson implicit method.

Let us use a table as shown in Table 15.3 for recording the function values at various time steps. The values for the first time step ($t = 0$) are obtained from the initial condition

$$f(x, 0) = 50(4 - x)$$

and the values for f_1 and f_5 are obtained from the boundary conditions.

Table 15.3

t_j	$x = 0$ f_1	$x = 1$ f_2	$x = 2$ f_3	$x = 3$ f_4	$x = 4$ f_5
$t_1 = 0.00$	0.0	150.00	100.00	50.00	0.0
$t_2 = 0.25$	0.0	56.25	75.00	43.75	0.0
$t_3 = 0.50$	0.0				0.0
$t_4 = 0.75$	0.0				0.0
$t_5 = 1.00$	0.0				0.0

Now, for the second time step ($t = 0.25$), we write equations at each point using Eq. (15.29) and solve for unknowns. Thus, for the second row in the table, we have

$$\begin{aligned} -0.0 + 4f_2 - f_3 &= 0.0 + 150 \\ -f_2 + 4f_3 - f_4 &= 150 + 150 \\ -f_3 + 4f_4 - 0.0 &= 100 + 0.0 \end{aligned}$$

Solving these three equations for three unknowns, we obtain

$$\begin{aligned} f_2 &= 56.25 \\ f_3 &= 75.00 \\ f_4 &= 43.75 \end{aligned}$$

This process may be continued for each time step. Students may complete the table.

15.5 HYPERBOLIC EQUATIONS

Hyperbolic equations model the vibration of structures such as buildings, beams and machines. We consider here the case of a vibrating string that is fixed at both the ends as shown in Fig. 15.5.

The lateral displacement of string f varies with time t and distance x along the string. The displacement $f(x, t)$ is governed by the *wave equation*

$$T \frac{\partial^2 f}{\partial x^2} = \rho \frac{\partial^2 f}{\partial t^2}$$

where T is the tension in the string and ρ is the mass per unit length.

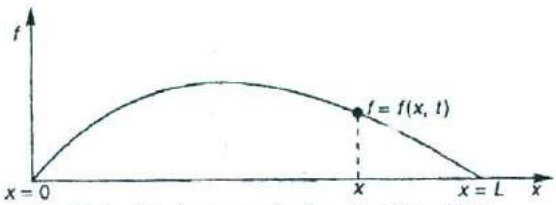


Fig. 15.5 Displacement of a vibrating string

Hyperbolic problems are also governed by both boundary and initial conditions, if time is one of the independent variables. Two boundary conditions for the vibrating string problem under consideration are

$$f(0, t) = 0 \qquad 0 \le t \le b$$

$$f(L, t) = 0 \qquad 0 \le t \le b$$

Two initial conditions are

$$f(x, 0) = f(x) \qquad 0 \le x \le a$$

$$f_t(x, 0) = g(x) \qquad 0 \le x \le a$$

Solution of Hyperbolic Equations

The domain of interest, $0 \le x \le a$ and $0 \le t \le b$, is partitioned as shown in Fig. 15.6. The rectangles of size $\Delta x = h$ and $\Delta t = \tau$

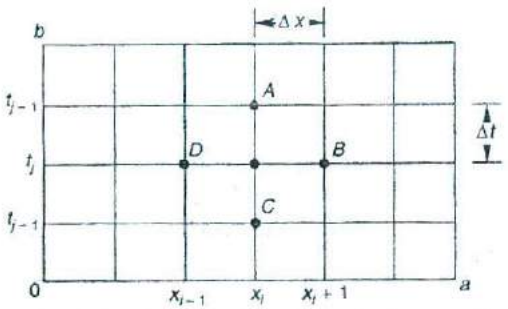


Fig. 15.6 Grid for solving hyperbolic equation

The difference equations for $f_{xx}(x, t)$ and $f_{tt}(x, t)$ are:

$$f_{xx}(x, t) = \frac{f(x-h, t) - 2f(x, t) + f(x+h, t)}{h^2}$$

$$f_{tt}(x, t) = \frac{f(x, t-\tau) - 2f(x, t) + f(x, t+\tau)}{\tau^2}$$

This implies that,

$$T\frac{f_{i-1,j}-2f_{i,j}+f_{i+1,j}}{h^2}=\rho\frac{f_{i,j-1}-2f_{i,j}+f_{i,j+1}}{\tau^2}$$

Solving this for $f_{i,j+1}$, we obtain

$$f_{i,j+1}=-f_{i,j-1}+2\left(1-\frac{T\tau^2}{\rho h^2}\right)f_{ij}+\frac{T\tau^2}{\rho h^2}(f_{i+1,j}+f_{i-1,j})$$

If we can make

$$1-\frac{T\tau^2}{\rho h^2}=0$$

then, we have

$$\boxed{f_{i,j+1}=-f_{i,j-1}+f_{i+1,j}+f_{i-1,j}} \qquad (15.30)$$

The value of f at $x = x_i$ and $t = t_j + \tau$ is equal to the sum of the values of f at the point $x = x_i - h$ and $x = x_i + h$ at the time $t = t_j$ (previous time) minus the value of f at $x = x_i$ at time $t = t_j - \tau$. From Fig. 15.6, we can say that,

$$f_A = f_B + f_D - f_C$$

Starting Values

We need two rows of starting values, corresponding to $j = 1$ and $j = 2$ in order to compute the values at the third row. First row is obtained using the condition

$$f(x, 0) = f(x)$$

The second row can be obtained using the second initial condition as follows:

$$f_t(x, 0) = g(x)$$

We know that

$$f_t(x, 0) = \frac{f_{i,0+1}-f_{i,0-1}}{2\tau}=g_i$$

$$f_{i,-1} = f_{i,1} - 2\,\tau g_i \qquad \text{for } t = 0 \text{ only}$$

Substituting this in Eq. (15.30), we get for $t = t_1$

$$\boxed{f_{i,1}=\frac{1}{2}(f_{i+1,0}+f_{i-1,0})+\tau g_i} \qquad (15.31)$$

In many cases, $g(x_i) = 0$. Then, we have

$$f_{i,1}=\frac{1}{2}(f_{i+1,0}+f_{i-1,0})$$

Example 15.7

Solve numerically the wave equation

$$f_{tt}(x, t) = 4f_{xx}(x, t), \quad 0 \le x \le 5$$

with the boundary conditions

$$f(0,t) = 0 \quad \text{and} \quad f(5, t) = 0$$

and initial values

$$f(x, 0) = f(x) = x(5 - x)$$

$$f_t(x,0) = g(x) = 0$$

Let $h = 1$

Given,

$$\frac{T}{\rho} = 4$$

and assuming

$$1 - 4\frac{\tau^2}{1^2} = 0$$

We get,

$$\tau = \frac{1}{2}$$

The values estimated using Eqs (15.30) and (15.31) are tabulated in Table 15.4.

Table 15.4

t \ x	0	1	2	3	4	5	
0.0	0.0	4	6	6	4	0.0	$x(5 - x)$
0.5	0.0	3	5	5	3	0.0	Equation (15.31)
1.0	0.0	1	2	2	1	0.0	Equation (15.30)
1.5	0.0	−1	−2	−2	−1	0.0	
2.0	0.0	−3	−5	−5	−3	0.0	
2.5	0.0	−4	−6	−6	−4	0.0	

15.6 SUMMARY

In this chapter, we discussed the solution of an important class of differential equations called partial differential equations. Due to complexity and limited scope of this book, we considered only the finite-difference method of solving the PDE problems where the coefficients a, b, and c are constants. We presented the following in this chapter:

- Definition and classification of partial differential equations.
- Derivation of difference equations for PDEs.
- Solution of Laplace's equation by the method of elimination.

- Liebmann's iterative method for solving Laplace's equation.
- Solution of Poisson's equation by both direct and iterative methods.
- Solution of parabolic type heat equation using the explicit Bender-Schmidt recurrence equation and the implicit Crank-Nicholson formula.
- Solution of hyperbolic type wave equation by iterative procedure.

Key Terms

Bender-Schmidt equation
Crank-Nicholson formula
Cross average
Diagonal average
Dirichlet boundary condition
Elliptic equation
Explicit method
Finite-difference method
Finite-element method
Gauss-Seidel iteration
Heat equation
Hyperbolic equation
Implicit method
Laplace's equation
Laplacian operator
Liebmann's method
Parabolic equation
Partial derivatives
Partial differential equation
Poisson's equation
Wave equation

REVIEW QUESTIONS

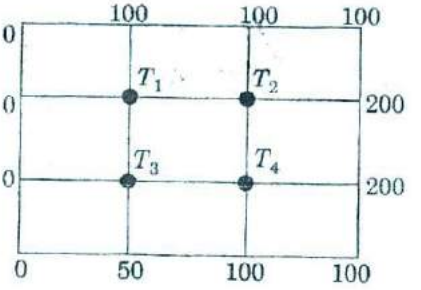
1. What is a partial differential equation? Give two examples.
2. State two real-life problems where partial differential equations are required to construct mathematical models.
3. How are the partial differential equations classified? Give an example from real-life situations for each type.
4. What are the various methods available to solve differential equations?
5. Explain how difference quotients are applied to solve partial differential equations.
6. What is Poisson's equation? How does it differ from Laplace's equation?
7. What is Liebmann's iteration method? What are its advantages?
8. What is meant by Dirichlet boundary conditions?
9. What is diagonal-averaging? When do we use it?
10. Derive a difference equation to represent a Poisson's equation.
11. Derive the five-point formula for Laplace's equation.
12. What is Crank-Nicholson method? Why is it known as implicit method?
13. What is Bender-Schmidt recurrence equation? Derive the formula.
14. Discuss the impact of size of the incremental width ΔT for the time variable t on the solution of a heat-flow equation.
15. Outline the argument that demonstrates the stability of the finite-difference procedure for solving a hyperbolic equation.

REVIEW EXERCISES

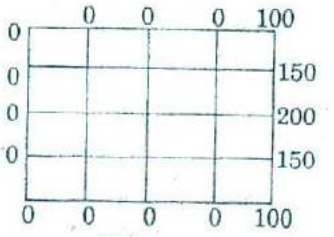
1. Determine which of the following equations are elliptic, parabolic, and hyperbolic.
   (a) $3f_{xx} + 4f_{yy} = 0$
   (b) $f_{xx} - f_{yy} = 0$
   (c) $f_{xx} - 2f_{xy} + 2f_{yy} = 2x + 5y$
   (d) $f_{xx} + 2f_{xy} + 4f_{yy} = 0$
   (e) $f_{xy} - f_y = 0$
   (f) $f_{xx} + 6f_{xy} + 9f_{yy} = 0$
2. The steady-state two-dimensional heat-flow in a metal plate is by

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

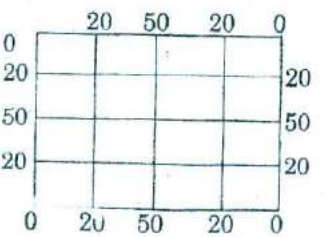
Given the boundary conditions as shown in the figure below, find the temperatures T_1, T_2, T_3, and T_4.



3. Solve for the steady-state temperatures in a rectangular plate 8 cm × 10 cm, if one 10 cm side is held at 50°C, and the other 10 cm side is held at 30°C and the other two sides are held at 10°C. Assume square grids of size 2 cm × 2 cm.
4. Repeat Exercise 3 by Leibmann's method.
5. Evaluate $f(x, y)$ at the internal grid points of the given domain governed by Laplace's equation. Use Liebmann's iteration method.



(a)



(b)

6. Torsion on a rectangular bar subject to twisting is governed by

$$\nabla^2 T = -4$$

Given the condition $T = 0$ on boundary, find T over a cross section of a bar of size 9 cm × 12 cm. Use a grid size of 3 cm × 3 cm.

7. Solve the equation

$$\nabla^2 f = F(x, y)$$

with $F(x, y) = xy$ and $f = 0$ on boundary. The domain is a square with corners at (0, 0) and (4, 4). Use $h = 1$.

8. Estimate the values at grid points of the following equations using Bender-Schmidt recurrence equation. Assume $h = 1$

(a) $f_{xx} - 0.5 f_t = 0$

Given,

$f(0, t) = 0, f(5, t) = 0$

$f(x, 0) = x(5 - x)$

(b) $9 f_{xx} = f_t$

Given,

$f(0, t) = -5, f(5, t) = 5$

$$f(x, 0) = \begin{cases} -5 \text{ for } 0 \le x \le 2.5 \\ 5 \text{ for } 2.5 < x \le 5 \end{cases}$$

9. Initial temperatures within an insulated cylindrical metal rod of 5 cm long are given by

$$T = 20x \text{ for } 0 \le x \le 5$$

where x is the distance from one face. Both the ends are maintained at 0°C. Find the temperatures as a function of x and t if the heat flow is governed by

$$4T_{xx} - T_t = 0$$

10. Solve the following equation using Crank-Nicholson method.

$$\frac{\partial^2 f}{\partial x^2} = 8 \frac{\partial f}{\partial t}$$

Given,

$$f(0, t) = 0, \quad f(20, t) = 10$$

$$f(x, 0) = 2.0$$

Assume $\Delta x = h = 5$ and $r = 1$

11. Solve Exercise 9 with the Crank-Nicholson method with $r = 1$.
12. Solve the following hyperbolic equations using finite difference method.

(a) $f_{tt} = 4 f_{xx}$

Given,

$f(0, t) = 0$ and $f(5, t) = 0$

$f(x, 0) = 100x^2 (5 - x)$

$f_t(x, 0) = 0$

(b) $f_{tt} = 4 f_{xx}$

Given,

$f(0, t) = 0$ and $f(1, t) = 0$

$f(x, 0) = f(x) = \sin(\pi x) + \sin(2\pi x)$

$f_t(x,0) = 0$

PROGRAMMING PROJECTS

1. Develop a program to solve Laplace's equation with Dirichlet conditions.
2. Write a program to solve Poisson's equation.
3. Develop a program using forward-difference method to solve the heat equation.
4. Write a program to solve the heat equation using Crank-Nicholson method.
5. Write a program for finite-difference solution of the wave equation.