Chapter XI

Solution of Ordinary Differential Equations

Several approaches are described for the solution of ODE (ordinary differential equations). These include:

- The Taylor series Method
- The Euler Method
- The Improved Euler Method
- The Runge-Kutta Methods.

We will begin our presentation on the numerical solution of a first-order ODE and later on extend the concepts to any order ODE.

11.1 Problem Statement

Given

$$\frac{dy}{dx} = f(x, y); \qquad y(x_0) = y_0 \tag{11.1}$$

where $y(x_0) = x_0$ is the initial condition needed to solve the problem. That is, $y = y_0$ at $x = x_0$.

Determine y as a function of x in either tabulated form or graph. Table 10.1 shows the tabulated solution required.

Table 11.1 Tabulated results.

x	x_0	x_1	...	x_{n-2}	x_{n-1}
y	y_0	y_1	...	y_{n-2}	y_{n-1}

Where $x_i = x_{i-1} + \Delta x$
and Δx is given.

10.2 The Taylor Series Method

Expand $y(x)$ about x_0 using Taylor series:

$$y(x) = y(x_0) + y'(x_0)(x - x_0) + \frac{y''(x_0)}{2!}(x - x_0)^2 + \frac{y'''(x_0)}{3!}(x - x_0)^3 + \frac{y^{(4)}}{4!}(x - x_0)^4 + \cdots$$

Let $x - x_0 = h$, therefore:

$$y(x) = y_0 + hf(x_0, y_0) + \frac{h^2}{2!} f'(x_0, y_0) + \frac{h^3}{3!} f''(x_0, y_0) + \frac{h^4}{4!} f'''(x_0, y_0) + \cdots \quad \textbf{(11.2)}$$

Note that

$$df(x_0, y_0) = \frac{\partial f(x_0, y_0)}{\partial x} dx + \frac{\partial f(x_0, y_0)}{\partial y} dy$$

We will drop the (x_0,y_0) from here on. Dividing by *dx* we get

$$\frac{df}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f = F_1 \quad \textbf{(11.3)}$$

Similarly one can derive:

$$\begin{aligned}
\frac{d^2 f}{dx^2} &= \frac{\partial F_1}{\partial x} + \frac{\partial F_1}{\partial y} f = F_2 \\
\frac{d^3 f}{dx^3} &= \frac{\partial F_2}{\partial x} + \frac{\partial F_2}{\partial y} f = F_3 \\
&\vdots \\
\frac{d^i f}{dx^i} &= \frac{\partial F_{i-1}}{\partial x} + \frac{\partial F_{i-1}}{\partial y} f = F_i \\
&\vdots
\end{aligned} \quad \textbf{(11.4)}$$

Example. Express the solution of

$$\frac{dy}{dx} = -2x - y \text{ with } y(0) = -1.$$

Solution.

From Equations (11.3) and (11.4) we can write:

$$\begin{aligned}
f &= -2x_0 - y_0 = F_1 \\
f' &= -2 - 1(-2x_0 - y_0) = -2 + 2x_0 + y_0 = F_2 \\
f'' &= 2 + 1(-2x_0 - y_0) = 2 - 2x_0 - y_0 = F_3 \\
f''' &= -2 - 1(-2x_0 - y_0) = -2 + 2x_0 + y_0 = F_4 \\
&\vdots
\end{aligned}$$

Substituting $x_0 = 0$ and $y_0 = -1$ we get:

$$f = -1, \quad f' = -3, \quad f'' = 3, \quad f''' = -3, \quad \cdots$$

That is:

$$y(h) = -1 - h - 3\frac{h^2}{2!} + 3\frac{h^3}{3!} - 3\frac{h^4}{4!} + \cdots$$

For small h we can neglect terms higher than h^4 and hence given h one can calculate $y(h)$. Setting $x_0 = h$ and $y_0 = y(h)$ we can repeat the above steps to calculate $y(2h)$ and so on.

11.3 Euler Method

In the Euler method the first two terms of the Taylor series are used.

$$y(x) = y(x_0) + hf(x_0, y_0)$$

or

$$\begin{aligned}
&y(x_{i+1}) = y(x_i) + hf(x_i, y_i) \\
&\text{where} \quad x_{i+1} = x_i + h, \quad i = 0,1,2,\cdots
\end{aligned}$$

Euler method can be graphically described as shown in Fig.11.1. The slope at (x_0, y_0) is used to calculate $y(x_1)$. Using x_1 and $y(x_1)$ calculate $f(x_1, y_1)$ and from there calculate $y(x_2)$ and so on.

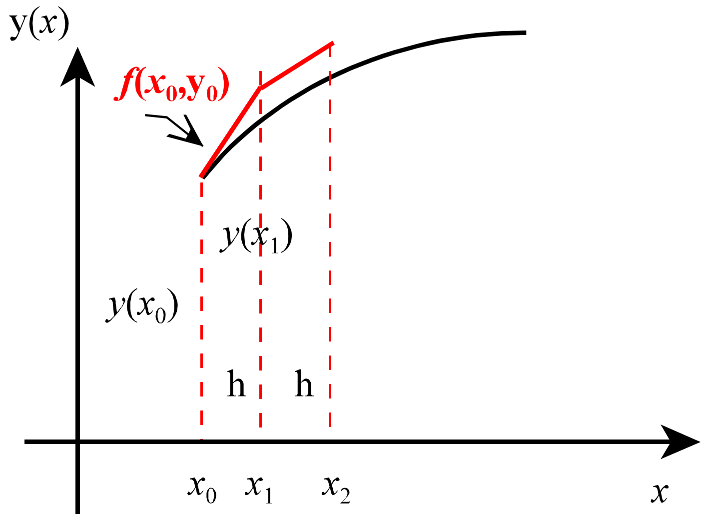

Figure 11.1 Euler Method.

11.4 Improved Euler Method

In the improved Euler method we use three terms of the Taylor series:

$$y(x_{i+1}) = y(x_i) + hf(x_i, y_i) + \frac{h^2}{2!} f'(x_i, y_i)$$
$$i = 0,1,2,3,\cdots$$

The derivative of $f(x_i, y_i)$ can be calculated using forward difference as follows:

$$f'(x_i, y_i) \approx \frac{f(x_{i+1}, y_{i+1}^*) - f(x_i, y_i)}{h}$$

where y_{i+1}^* is calculated using Euler method as follows:

$$y_{i+1}^* = y_i + hf(x_i, y_i)$$

Hence:

$$y(x_{i+1}) = y(x_i) + hf(x_i, y_i) + \frac{h^2}{2} \frac{f\big(x_{i+1}, y_i + hf(x_i, y_i)\big) - f(x_i, y_i)}{h}$$

which can be placed in the form:

$$y(x_{i+1}) = y(x_i) + \frac{h}{2}\left[f(x_i, y_i) + f\left(x_{i+1}, y_i + hf(x_i, y_i)\right)\right]$$
$$= y(x_i) + \frac{h}{2}(k_1 + k_2)$$

where

$$k_1 = f(x_i, y_i)$$
$$k_2 = f\left(x_i + h, y_i + hk_1\right)$$

Procedure:

5. Calculate $k_1 = f(x_i, y_i)$
6. Calculate $y_i^* = y_i + hk_1$
7. Calculate $k_2 = f(x_i + h, y_i^*)$
8. Calculate $y_{i+1} = y_i + \frac{h}{2}(k_1 + k_2)$

The improved Euler method can be described graphically as shown Figure 11.2.

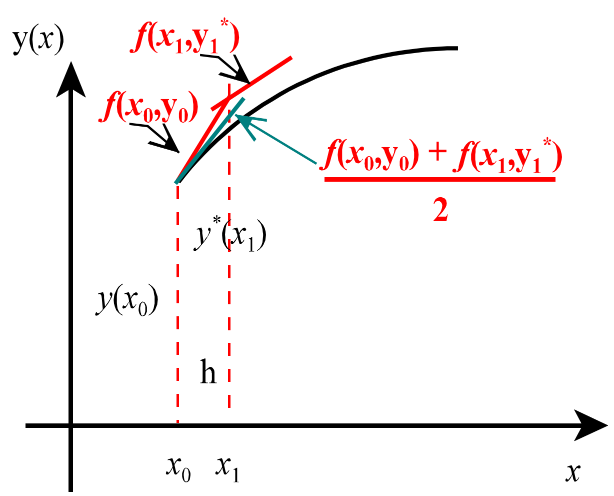

Figure 11.2 Improved Euler Method.

From Fig.11.2 the average slope $\bar{f} = \dfrac{f(x_0, y_0) + f(x_1, y_1^*)}{2}$ is used in Euler method to calculate the next value $y_{i+1} = y_i + h\bar{f}$, which is the same approach derived in this section for the Improved Euler method.

Example. Calculate *y*(0.1) and *y*(0.2) using *h*=0.1 for the following ODE

$$\frac{dy}{dx} = -2x - y \text{ with } y(0) = -1.$$

Solution. Using $f(x, y) = -2x - y$ and the initial conditions $x_0 = 0, \quad y_0 = -1$ we follow the procedure described above.

1. Calculate $k_1 = f(0, -1) = -2(0) - (-1) = 1$
2. Calculate $y^* = y_0 + hk_1$

   $= -1 + 0.1 * 1 = -0.9$
3. Calculate $k_2 = f(x_0 + 0.1, y^*) = f(0.1, -0.9) = 0.7$
4. Calculate $y_1 = y(0.1) = y_0 + \dfrac{0.1}{2}(k_1 + k_2) = -1 + \dfrac{0.1}{2}(1 + 0.7) = -0.915$

Repeating the procedure to obtain *y*(0.2).

1. Calculate $k_1 = f(0.1, -0.915) = 0.715$
2. Calculate $y^* = y_1 + hk_1 = -0.915 + 0.1(0.715) = -0.8435$
3. Calculate $k_2 = f(x_1 + 0.1, y^*) = f(0.2, -0.8435) = 0.4435$
4. Calculate

$$y_2 = y(0.2) = y(0.1) + \frac{0.1}{2}(k_1 + k_2)$$

$$= -0.915 + \frac{0.1}{2}(0.715 + 0.4435) = -0.857075$$

Hence:

$$y(0) = -1$$
$$y(0.1) = -0.915$$
$$y(0.2) = -0.857075$$

11.6 Runge Kutta Methods

The main reason for the development of alternative methods to the Taylor series approach is to avoid the differentiations required by the series. Neither Euler nor Improved methods require that we differentiate any function. The Runge Kutta Methods are basically generalizations of the Improved Euler Method. For second order Runge-Kutta method the following is assumed:

$$y(x+h) \approx y(x) + a_1 k_1 + a_2 k_2 \qquad \textbf{(11.5)}$$
$$\text{where} \quad k_1 = hf(x,y)$$
$$k_2 = hf(x+mh, y+mk_1)$$

The problem is now stated as follows: obtain the 3 unknowns $\{a_1, \quad a_2, \quad m\}$ such that $y(x+h)$ obtained using the above equation matches the Taylor Series expansion of $y(x+h)$ truncated after the first derivative.

The Taylor Series expansion with the terms truncated after the second derivative is:

$$y(x+h) = y(x) + hy'(x) + \frac{h^2}{2!} y''(x) \qquad \textbf{(11.6)}$$

Since the derivatives in the Taylor series expansion are in terms of $f(x,y)$ we need to expand $f(x+mh, y+mk_1)$ as a series in terms of $f(x,y)$. This calls for the use of Taylor series expansion in two variables which is given by:

$$f(x+\Delta x, y+\Delta y) = f(x,y) + \Delta x \frac{\partial f(x,y)}{\partial x} + \Delta y \frac{\partial f(x,y)}{\partial y}$$
$$+ \frac{1}{2!}\left[(\Delta x)^2 \frac{\partial^2 f(x,y)}{\partial x^2} + 2\Delta x \Delta y \frac{\partial^2 f(x,y)}{\partial x \partial y} + (\Delta y)^2 \frac{\partial^2 f(x,y)}{\partial y^2} \right]$$
$$+ \cdots$$

Hence we can write:

$$k_2 = h[f(x,y) + mhf_x(x,y) + mk_1 f_y(x,y) + \cdots] \quad \textbf{(11.7)}$$

Where $f_x(x,y) = \dfrac{\partial f(x,y)}{\partial x}$ and $f_y(x,y) = \dfrac{\partial f(x,y)}{\partial y}$

Replacing k_1 with $h\,f(x,y)$ and k_2 with the right-hand side of Eq.(11.7) in Equation (11.5) we get:

$$\begin{aligned} y(x+h) &= y(x) + a_1 hf(x,y) + a_2 h\Big[f(x,y) + mhf_x(x,y) + mk_1 f_y\Big] \\ &= y + a_1 hf + a_2 h\Big[f + mhf_x + mhff_y\Big] \\ &= y + h(a_1 + a_2)f + a_2 mh^2(f_x + ff_y) \end{aligned} \quad \textbf{(11.8)}$$

Notice that we dropped the (x,y) just for compactness sake.

In the Taylor series expansion (Eq.11.6) replace:

$$\begin{aligned} y'(x) &= f \\ y''(x) &= \frac{df(x,y)}{dx} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f = f_x + ff_y \end{aligned}$$

Hence

$$y(x+h) = y(x) + hf + \frac{h}{2}\Big(f_x + ff_y\Big) \quad \textbf{(11.9)}$$

Comparing (11.8) and (11.9) we get:

$$\begin{aligned} a_1 + a_2 &= 1 \\ a_2 m &= 1 \end{aligned}$$

These are two equations in three unknowns and hence we have the choice to select one of the variables. Choosing m=1, therefore

$$a_1 = a_2 = \frac{1}{2}$$

Hence a second order Runge-Kutta method is:

$$y(x+h) = y(x) + \frac{1}{2}(k_1 + k_2)$$

which turns out to be the Improved Euler Method. The Improved Euler method is actually a special case of the Runge-Kutta methods.

We will now tackle the fourth order Runge-Kutta method. In this case we assume the Runge-Kutta formulas are:

$$\begin{aligned} y(x+h) &\approx y(x) + ak_1 + bk_2 + ck_3 + dk_4 \\ k_1 &= hf(x, y) \\ k_2 &= hf(x+mh, y+mk_1) \\ k_3 &= hf(x+nh, y+mk_2) \\ k_4 &= hf(x+ph, y+pk_3) \end{aligned}$$

What is required is to find the coefficients $\{a, b, c, d, m, n, p\}$ such that the above formulas duplicate the Taylor series through the term in h^4. That is:

$$y(x+h) = y(x) + hy'(x) + \frac{h^2}{2!}y''(x) + \frac{h^3}{3!}y'''(x) + \frac{h^4}{4!}y^{(\text{iv})}(x)$$

Since $\quad \dfrac{dy}{dx} = y'(x) = f(x, y)$

then $y''(x) = f'(x) = \dfrac{\partial f}{\partial x} + \dfrac{\partial f}{\partial y} f = f_x + f_y f$

Let $\qquad F_1 = f_x + f_y f$

$\therefore \qquad y''(x) = F_1$

Similarly

$$y'''(x) = f_{xx} + 2ff_{xy} + f^2 f_{yy} + f_y(f_x + ff_y)$$

Let $\qquad F_2 = f_{xx} + 2ff_{xy} + f^2 f_{yy}$

$\therefore \quad y'''(x) = F_2 + f_y F_1$

and

$$y^{(\mathrm{iv})}(x) = f_{xxx} + 3ff_{xxy} + 3f^2 f_{xyy} + f^3 f_{yyy} + f_y(f_{xx} + 2ff_{xy} + f^2 f_{yy})$$
$$+ 3(f_x + ff_y)(f_{xy} + ff_{yy}) + f_y^2(f_x + ff_y)$$

Let $F_3 = f_{xxx} + 3ff_{xxy} + 3f^2 f_{xyy} + f^3 f_{yyy}$

$$\therefore \quad y^{(\mathrm{iv})}(x) = F_3 + f_y F_2 + 3F_1(f_{xy} + ff_{yy}) + f_y^2 F_1$$

which allows the Taylor series to be written as:

$$y(x+h) = y(x) + \frac{1}{2}h^2 F_1 + \frac{1}{6}h^3(F_2 + f_y F_1) +$$
$$\frac{1}{24}h^4\left[F_3 + f_y F_2 + 3(f_{xy} + ff_{yy})F_1 + f_y^2 F_1\right] + \cdots$$

Turning now to the various k values, similar computations produce

$$k_1 = hf$$
$$k_2 = h\left[f + mhF_1 + \frac{1}{2}m^2h^2F_2 + \frac{1}{6}m^3h^3F_3 + \cdots\right]$$

$$k_3 = h\left[f + nhF_1 + \frac{1}{2}h^2(n^2F_2 + 2mnf_yF_1) + \right.$$
$$\left.\frac{1}{6}h^3\left\{n^3F_3 + 3m^2nf_yF_2 + 6mn^2(f_{xy} + ff_{yy})F_1\right\} + \cdots\right]$$

$$k_4 = h\left[f + phF_1 + \frac{1}{2}h^2(p^2F_2 + 2npf_yF_1) + \right.$$
$$\left.\frac{1}{6}h^3\left\{p^3F_3 + 3n^2pf_yF_2 + 6np^2(f_{xy} + ff_{yy})F_1 + 6mnpf_y^2F_1\right\} + \cdots\right]$$

Combining these equations as suggested by the Runge-Kutta formula,

$$
\begin{aligned}
y(x+h) = {} & y(x) + (a+b+c+d)hf + (bm+cn+dp)h^2 F_1 \\
& + \frac{1}{2}(bm^2+cn^2+dp^2)h^3 F_2 + \frac{1}{6}(bm^3+cn^3+dp^3)h^4 F_3 + \\
& + (cmn+dnp)h^3 f_y F_1 + \frac{1}{2}(cm^2 n + dn^2 p)h^4 f_y F_2 \\
& + (cmn^2+dnp^2)h^4 (f_{xy} + ff_{yy})F_1 + dmnph^4 f_y^2 F_1 + \cdots
\end{aligned}
$$

Comparison with the Taylor series results in the eight conditions

$$
\begin{aligned}
a+b+c+d &= 1 & cmn+dnp &= \frac{1}{6} \\
bm+cn+dp &= \frac{1}{2} & cmn^2+dnp^2 &= \frac{1}{8} \\
bm^2+cn^2+dp^2 &= \frac{1}{3} & cm^2 n + dn^2 p &= \frac{1}{12} \\
bm^3+cn^3+dp^3 &= \frac{1}{4} & dmnp &= \frac{1}{24}
\end{aligned}
$$

Since we have eight equations in seven unknowns we can select one of the unknowns and solve for the rest. A classical solution set is

$$
m = n = \frac{1}{2} \quad p = 1 \quad a = d = \frac{1}{6} \quad b = c = \frac{1}{3}
$$

leading to the Runge-Kutta formulas

$$
\begin{aligned}
y(x+h) &= y(x) + \frac{1}{6}(k_1+k_2+k_3+k_4) \\
k_1 &= hf(x,y) \\
k_2 &= hf(x+\frac{h}{2}, y+\frac{1}{2}k_1) \\
k_3 &= hf(x+\frac{h}{2}, y+\frac{1}{2}k_2) \\
k_4 &= hf(x+h, y+k_3)
\end{aligned}
$$

11.7 Solving Higher Order ODE

An n^{th} order ODE can be broken into n simultaneous first order ODE using a state variable approach which will be explained through an example.

Example. Using h=0.1 determine y at t=0.1 and 0.2 for the following ODE:

$$\frac{d^2y}{dt^2} + \frac{dy}{dt} + y = 1.0$$
$$\text{given} \quad y(0) = y'(0) = 0.0$$

Solve using (a) Euler. (b) Improved Euler. (c) Fourth order Runge-Kutta.

Solution. Let $x_1 = y$

$x_2 = y'$

$$x_1' = y' = x_2$$
$$x_2' = y'' = 1.0 - y - y' = 1.0 - x_1 - x_2$$

or in matrix form,

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1.0 \\ 1.0 - x_1 - x_2 \end{bmatrix}$$

$$\text{or} \qquad \frac{d\mathbf{X}}{dt} = \mathbf{G}(t, \mathbf{X})$$

$$\mathbf{X}(0) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{t=0} = \begin{bmatrix} y(0) \\ y'(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

This approach applies to any order ODE as follows. For an n^{th} order ODE

$$\frac{d^n y}{dt^n} + a_1\frac{d^{n-1}y}{dt^{n-1}} + a_2\frac{d^{n-2}y}{dt^{n-2}} + \cdots + a_n y = f(t, y)$$
$$\text{given} \quad y(0) = \alpha_1, \quad y'(0) = \alpha_2, \quad \cdots \quad y^{(n-1)}(0) = \alpha_n$$

To place in the state variable form let

$$\begin{aligned} x_1 &= y \\ x_2 &= y_1' \\ x_3 &= y'' \\ &\vdots \\ x_n &= y^{(n-1)} \end{aligned}$$

We can therefore write:

$$\begin{aligned} x_1' &= x_2 \\ x_2' &= x_3 \\ x_3' &= x_4 \\ &\vdots \\ x_n' &= f(t, x_1) - a_n x_1 - a_{n-1} x_2 - \cdots - a_1 x_{n-1} \end{aligned}$$

The coefficients of the ODE can be either constants or functions of *t* and/or *y*.

(a) <u>Euler</u>

<u>Procedure</u>
At $t = t_i$
1. Calculate $\mathbf{K} = \mathbf{G}(t_i, \mathbf{X}_i)$
2. Calculate $\mathbf{X}_{i+1} = \mathbf{X}_i + h\mathbf{G}(t_i, \mathbf{X}_i)$

Hence at t=0

$$1. \quad \mathbf{K} = \begin{bmatrix} x_2 \\ 1.0 - x_1 - x_2 \end{bmatrix}_{t=0} = \begin{bmatrix} 0 \\ 1 - 0 - 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$2. \quad \mathbf{X}_1 = \mathbf{X}_0 + h\mathbf{K} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 0.1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} \begin{matrix} \leftarrow x_1 = y(0.1) \\ \leftarrow x_2 = y'(0.1) \end{matrix}$$

$$\therefore \quad y(0.1) = 0.0$$

At t=0.1

$$1.\quad \mathbf{K} = \begin{bmatrix} x_2 \\ 1 - x_1 - x_2 \end{bmatrix}_{t=0.1} = \begin{bmatrix} 0.1 \\ 1 - 0 - 0.1 \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix}$$

$$2.\quad \mathbf{X}_2 = \mathbf{X}_1 + h\mathbf{K} = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} + 0.1\begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix} = \begin{bmatrix} 0.01 \\ 0.9 \end{bmatrix} \begin{matrix} \leftarrow x_1 = y(0.2) \\ \leftarrow x_2 = y'(0.2) \end{matrix}$$

$$\therefore \quad y(0.2) = 0.01$$

(b) Improved Euler Method

Procedure

At $t = t_i$

1. Calculate $\mathbf{K}_1 = \mathbf{G}(t_i, \mathbf{X}_i)$
2. Calculate $\mathbf{XEST} = \mathbf{X}_i + h\mathbf{K}_1$
3. Calculate $\mathbf{K}_2 = \mathbf{G}(t_{i+1}, \mathbf{XEST})$
4. Calculate $\mathbf{X}_{i+1} = \mathbf{X}_i + \frac{h}{2}\left[\mathbf{K}_1 + \mathbf{K}_2\right]$

Hence at t=0

$$1.\quad \mathbf{K}_1 = \mathbf{G}(t_0, \mathbf{X_0}) = \begin{bmatrix} x_2 \\ 1 - x_1 - x_2 \end{bmatrix}_{t=0} = \begin{bmatrix} 0 \\ 1 - 0 - 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$2.\quad \mathbf{XEST} = \mathbf{X_0} + h\mathrm{K}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 0.1\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{matrix} \leftarrow x_1 \\ \leftarrow x_2 \end{matrix}$$

$$3.\quad \mathbf{K}_2 = \mathbf{G}(\mathbf{t} = \underbrace{\mathbf{t_0} + h}_{0.1}, \mathbf{XEST}) = \begin{bmatrix} 0 \\ 1 - 0 - 0 \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix}$$

$$4.\quad \mathbf{X}_i = \mathbf{X}_0 + \frac{h}{2}\left(\mathbf{K}_1 + \mathbf{K}_2\right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \frac{0.1}{2}\left[\begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0.1 \\ 0.9 \end{pmatrix}\right] = \begin{bmatrix} 0.005 \\ 0.095 \end{bmatrix} \begin{matrix} \leftarrow x_1 \\ \leftarrow x_2 \end{matrix}$$

$$\therefore \quad y(0.1) = 0.005$$

At t=0.1

1. $\mathbf{K}_1 = \mathbf{G}(t_1, \mathbf{X}_1) = \mathbf{G}(0.1, \begin{bmatrix} 0.005 \\ 0.095 \end{bmatrix}) = \begin{bmatrix} 0.095 \\ 1 - 0.005 - 0.095 \end{bmatrix} = \begin{bmatrix} 0.095 \\ 0.9 \end{bmatrix}$

2. $\mathbf{XEST} = \mathbf{X}_1 + h\mathbf{K}_1 = \begin{bmatrix} 0.005 \\ 0.095 \end{bmatrix} + 0.1 \begin{bmatrix} 0.095 \\ 0.9 \end{bmatrix} = \begin{bmatrix} 0.0145 \\ 0.185 \end{bmatrix}$

3. $\mathbf{K}_2 = \mathbf{G}(\underbrace{t_1 + h}_{0.2}, \mathbf{XEST}) = \mathbf{G}(0.2, \begin{bmatrix} 0.0145 \\ 0.185 \end{bmatrix} = \begin{bmatrix} 0.185 \\ 1 - 0.0145 - 0.185 \end{bmatrix}$

$$= \begin{bmatrix} 0.185 \\ 0.8005 \end{bmatrix}$$

4. $\mathbf{X}_2 = \mathbf{X}_1 + \frac{h}{2}\left(\mathbf{K}_1 + \mathbf{K}_2\right) = \begin{bmatrix} 0.005 \\ 0.095 \end{bmatrix} + \frac{0.1}{2}\left[\begin{pmatrix} 0.095 \\ 0.9 \end{pmatrix} + \begin{pmatrix} 0.185 \\ 0.8005 \end{pmatrix}\right]$

$$= \begin{bmatrix} 0.19 \\ 0.180025 \end{bmatrix}$$

$$\therefore \quad y(0.2) = 0.19$$

(c) <u>Runge-Kutta 4th order Method</u>

<u>Procedure</u>

At $t = t_i$

1. Calculate $\mathbf{K}_1 = \mathbf{G}(t_i, \mathbf{X}_i)$
2. Calculate $\mathbf{XEST} = \mathbf{X}_i + \frac{h}{2}\mathbf{K}_1$
3. Calculate $\mathbf{K}_2 = \mathbf{G}(t_i + \frac{h}{2}, \mathbf{XEST})$
4. Calculate $\mathbf{XEST} = \mathbf{X}_i + \frac{h}{2}\mathbf{K}_2$
5. Calculate $\mathbf{K}_3 = \mathbf{G}(t_i + \frac{h}{2}, \mathbf{XEST})$

6. Calculate $\mathbf{XEST} = \mathbf{X}_i + h\mathbf{K}_3$

7. Calculate $\mathbf{K}_4 = \mathbf{G}(t_i + h, \mathbf{XEST})$

8. Calculate $\mathbf{X}_{i+1} = \mathbf{X}_i + \frac{h}{6}\left[\mathbf{K}_1 + 2\mathbf{K}_2 + 2\mathbf{K}_3 + \mathbf{K}_4\right]$

The calculations follow a similar procedure to the Improved Euler method and are left to the reader as an exercise.

Exercise. Use the Runge-Kutta method to solve the following ODE for the points $y(0.1)$, $y(0.2)$ and $y(0.3)$. Use h=0.1.

$$\frac{d^2y}{dt^2} + ty^2\frac{dy}{dt} + e^{-t}y = 4y^2e^{-t}$$
$$y(0) = 0.5 \quad y'(0) = 1.0$$

Now we return back to program development. The following is a C++ program for solving ODE using either Euler or Fourth order Runge-Kutta method. To verify that the program works it is tested against an ODE with known analytical solution. The ODE used is:

$$\frac{d^2y}{dt^2} + \frac{dy}{dt} + y = 0.0, \quad y(0) = y'(0) = 1.0$$

Its analytical solution is given by:

$$y(t) = e^{-0.5}\left(\cos(\frac{\sqrt{3}}{2}t) + \sqrt{3}\sin(\frac{\sqrt{3}}{2}t)\right)$$

```
#include <iostream.h>
#include <iomanip.h> //input/output manipulation for formatting using setw, etc.
#include <conio.h>
#include <math.h>

class ODE
  {
```

```
// The default member classification in classes is private.
//Private variables
 double *x; //state space vector x[0],x[1],...

 //The k's in the Runge-Kutta method.
 // each k is a vector of dimension = order of ODE
 double *k1;
 double *k2;
 double *k3;
 double *k4;
 double *xest;
 void(*G)(double tp,double *xp,double *xdot); //function for calculating the k's
 double t; //independent variable
 int neqns; //number of equations=order of ODE
 double h;  //step length

public:
    ODE(void (*GP)(double, double *,double *), int n)
     {
    G=GP;
    neqns=n;
    x=new double[neqns];
    k1=new double[neqns];
    k2=new double[neqns];
    k3=new double[neqns];
    k4=new double[neqns];
    xest=new double[neqns];
     }

  ~ODE()
     {
     delete []x;
     delete []k1;
     delete []k2;
     delete []k3;
     delete []k4;
     delete []xest;
     }

   void Init(double *xp, double tp, double dt);

   double get_t()
     {
```

```
    return t;
    }

  double* getx()
   {
   return x;
   }

   void Euler();
   void RK4();  // 4th order Runge-Kutta
   };

void ODE::Init(double *xp, double tp, double dt)
    {
    for(int i=0; i<neqns; i++)
      x[i]=xp[i];
    t=tp;
    h=dt;
      }

void ODE::Euler()
  {
  (*G)(t,x,k1);

  for(int i=0; i<neqns; i++)
    x[i]=x[i]+h*k1[i];
  t=t+h;
    }

//Fourth order Runge-Kutta
void ODE::RK4()
    {
    int i;

    (*G)(t,x,k1);

    for(i=0; i<neqns; i++)
      xest[i]=x[i]+0.5*h*k1[i];

    (*G)(t+0.5*h,xest,k2);

    for(i=0; i<neqns; i++)
      xest[i]=x[i]+ 0.5*h*k2[i];
```

```
    (*G)(t+0.5*h,xest,k3);

    for(i=0; i<neqns; i++)
      xest[i]=x[i]+h*k3[i];

    (*G)(t+h,xest,k4);

    for(i=0; i<neqns; i++)
      x[i]=x[i]+ h*(k1[i]+2.0*k2[i]+2.0*k3[i]+k4[i])/6.0;

    t=t+h;
      }

//User supplied routine
// ODE y"(t)+y'(t)+y = 0, intitial conditions y'(0)=y(0)=1
// Analytical solution y=exp(-0.5t)*(cos(sqrt(3)*t/2)+sqrt(3)*sin(sqrt(3)*t/2))
 void Vector( double t, double *x, double *xdot)
    {
   xdot[0]=x[1];
   xdot[1]=0.0-x[0]-x[1];
      }

//-----------------------------------------------------------------------

int main()
{
ODE eqn(Vector, 2);  //Replace (*G) with Vector and set neqns=2
double *x, t, h, yp[10], tp[10];

x=new double[2];
//initial conditions
x[0]=1.0;
x[1]=1.0;
t=0.0;
h=0.1;

eqn.Init(x,t,h); //set initial conditions
tp[0]=0.0;
yp[0]=x[0];

int i;
for(i=1; i<10; i++)
 {
```

```
 eqn.RK4();
 x=eqn.getx();

 yp[i]=x[0];
 tp[i]=eqn.get_t();
 }

// print solution
cout << "From Runge Kutta" << "  " << "t" << "           "
    << "From Analytical solution" << endl;
for(i=0; i<10; i++)
  {
  cout << setprecision(7) << setw(10) << yp[i]
        << setprecision(3) << setw(10) << tp[i]
       << setprecision(7) << setw(20)  <<
       (exp(-0.5*tp[i])*(cos(0.866*tp[i])+1.732*sin(0.866*tp[i]))<< endl;
    }

getch();
 return 1;
   }
```

Print-out

From Runge Kutta	t	From Analytical solution
1	0	1
1.090171	0.1	1.090163
1.161395	0.2	1.16138
1.214798	0.3	1.214778
1.251569	0.4	1.251544
1.27294	0.5	1.272912
1.280174	0.6	1.280143
1.274547	0.7	1.274516
1.257337	0.8	1.257305
1.22981	0.9	1.229778

Problems

1. Solve the following ODE's in $0 \le t \le 5$ using the improved Euler method with h=0.1 (write your own program)

a. $$\frac{d^2y}{dt^2} + 8y = 0, \quad y(0) = 1,\ \dot{y}(0) = 0$$

b. $$\frac{d^2y}{dt^2} - 0.01\left(\frac{dy}{dt}\right)^2 + 2y = \sin(t), \quad y(0) = 0, \;\; \dot{y}(0) = 1$$

c. $$(e^t + y)\frac{d^2y}{dt^2} = t, \quad y(0) = 1, \quad \dot{y}(0) = 0$$

2. The number of bacterial cells (P) in a given reactor is related to time in days (t) as described by the following mathematical model

$$\frac{dP}{dt} = 0.3P - 0.0000007P^2$$

If at time t=0, P=10^6 determine the number of cells when t=30 days. Use the fourth-order Runge-Kutta method and a time increment of 1 day.

3. When a vehicle travels over a rough road, a vertical displacement will result. Consider the following highly idealized model:

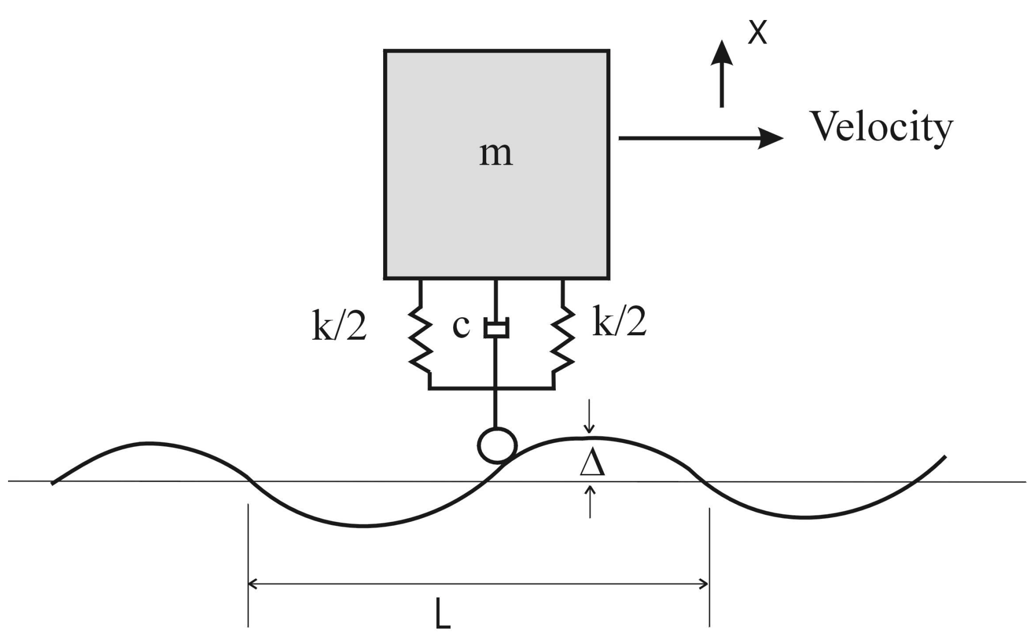

Figure 11.3 Problem 3.

Suppose that a manufacturer is interested in determining the largest vertical displacement x for a vehicle traveling at a velocity of 55 miles per hour over a sinusoidal surface, given

m=10 lb-s^2/in.
k=1210 lb/in.
c=88 lb-s/in.
L=50 ft.
Δ=1 in.

The mathematical model describing this behavior is given by:

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = m\varpi^2\Delta \sin\varpi t$$

where $\varpi = 2\pi v/L$ where v is the velocity of the vehicle. Determine the maximum displacement x using the Runge-Kutta 4th order method and Δt=0.01 s. Assume that the vehicle was initially at rest ($x(0) = \dot{x}(0) = 0$).

4. Use Taylor series to solve the following second-order ordinary differential equation at t=0.1:

$$4\frac{d^2x}{dt^2} + 3\frac{dx}{dt} + 16x = 0$$

with the initial conditions $x(0) = 1, \quad \dot{x}(0) = 1$.