"On two occasions I have been asked [by members of Parliament], 'Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?' I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question."

Charles Babbage (1792-1871)

Chapter 8
Ordinary Differential Equations

Many ordinary differential equations encountered do not have easily obtainable closed form solutions, and we must seek other methods by which solutions can be constructed. Numerical methods provide an alternative way of constructing solutions to these sometimes difficult problems. In this chapter we present an introduction to some numerical methods which can be applied to a wide variety of ordinary differential equations. These methods can be programmed into a digital computer or even programmed into some hand-held calculators. Many of the numerical techniques introduced in this chapter are readily available in the form of subroutine packages available from the internet.

We consider the problem of developing numerical methods to solve a first order initial value problem of the form

$$\frac{dy}{dx} = f(x, y), \qquad y(x_0) = y_0 \tag{8.1}$$

and then consider how to generalize these methods to solve systems of ordinary differential equations having the form

$$\begin{aligned}
\frac{dy_1}{dx} &= f_1(x, y_1, y_2, \ldots, y_m), \qquad y_1(x_0) = y_{10} \\
\frac{dy_2}{dx} &= f_2(x, y_1, y_2, \ldots, y_m), \qquad y_2(x_0) = y_{20} \\
&\vdots \\
\frac{dy_m}{dx} &= f_m(x, y_1, y_2, \ldots, y_m), \qquad y_m(x_0) = y_{m0}
\end{aligned} \tag{8.2}$$

Coupled systems of ordinary differential equations are sometimes written in the vector form

$$\frac{d\vec{y}}{dx} = \vec{f}(x, \vec{y}), \qquad \vec{y}(x_0) = \vec{y}_0 \tag{8.3}$$

where $\vec{y}$, $\vec{y}(x_0)$ and $\vec{f}(x, \vec{y})$ are column vectors given by $\vec{y} = \text{col}(y_1, y_2, y_3, \ldots, y_m)$, $\vec{y}(x_0) = \text{col}(y_{10}, y_{20}, \ldots, y_{m0})$ and $\vec{f}(x, \vec{y}) = \text{col}(f_1, f_2, \ldots, f_m)$.

We start with developing numerical methods for obtaining solutions to the first order initial value problem (8.1) over an interval $x_0 \le x \le x_n$. Many of the techniques developed for this first order equation can, with modifications, also be applied to solve a first order system of differential equations.

Higher Order Equations By defining new variables, higher order differential equations can be reduced to a first order system of differential equations. As an example, consider the problem of converting a nth order linear homogeneous differential equation

$$\frac{d^n y}{dx^n} + a_1 \frac{d^{n-1} y}{dx^{n-1}} + a_2 \frac{d^{n-2} y}{dx^{n-2}} + \cdots + a_{n-1} \frac{dy}{dx} + a_n y = 0 \tag{8.4}$$

to a vector representation. To convert this equation to vector form we define new variables. Define the vector quantities

$$\begin{aligned} \vec{y} =& \text{col}(y_1, y_2, y_3, \cdots, y_n) = \text{col}\left(y, \frac{dy}{dx}, \frac{d^2 y}{dx^2}, \cdots, \frac{d^{n-1} y}{dx^{n-1}}\right) \\ \vec{f}(x, \vec{y}) =& A\vec{y}, \\ \text{where} \quad A =& \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & -a_{n-3} & \cdots & -a_2 & -a_1 \end{bmatrix}. \end{aligned} \tag{8.5}$$

Observe that the linear nth order differential equation (8.4) can now be represented in the form of equation (8.3). In this way higher order linear ordinary differential equations can be represented as a first order vector system of differential equations.

Numerical Solution

In our study of the scalar initial value problem (8.1) it is assumed that $f(x, y)$ and its partial derivative f_y both exist and are continuous in a rectangular region

about a point (x_0, y_0). If these conditions are satisfied, then theoretically there exists a unique solution of the initial value problem (8.1) which is a continuous curve $y = y(x)$, which passes through the point (x_0, y_0) and satisfies the differential equation. In contrast to the solution being represented by a continuous function $y = y(x)$, the numerical solution to the initial value problem (8.1) is represented by a set of data points (x_i, y_i) for $i = 0, 1, 2, \ldots, n$ where y_i is an approximation to the true solution $y(x_i)$. We shall investigate various methods for constructing the data points (x_i, y_i), for $i = 1, 2, \ldots, n$ which approximate the true solution. This data set is then called a numerical solution to the given initial value problem. The given rule or technique used to obtain the numerical solution is called a numerical method or algorithm. There are many numerical methods for solving ordinary differential equations. In this chapter we will consider only a select few of the more popular methods. The numerical methods considered can be classified as either single-step methods or multi-step methods. We begin our introduction to numerical methods for ordinary differential equations by considering single-step methods.

Single Step Methods

From calculus a function $y = y(x)$, which possesses derivatives of all orders, can be expanded in a Taylor series about a point $x = x_0$. The Taylor series expansion has the form

$$y(x_0 + h) = y(x_0) + y'(x_0)h + y''(x_0)\frac{h^2}{2!} + \ldots + y^{(n)}(x_0)\frac{h^n}{n!} + R_n, \tag{8.6}$$

where R_n is a remainder term. If the $(n+1)$st derivative of y is bounded such that $|y^{(n+1)}(x)| < K$ for $x \in (x_0, x_0 + h)$, then we can say that the remainder term satisfies

$$R_n = y^{(n+1)}(\xi)\frac{h^{n+1}}{(n+1)!} = O(h^{n+1}) \tag{8.7}$$

with ξ lying somewhere between x_0 and $x_0 + h$. We can use the Taylor series expansion to derive many numerical methods for solving initial value problems.

Example 8-1. Euler's Method (First-order Taylor series method)

Consider the specific initial value problem to solve

$$y'(x) = \frac{dy}{dx} = f(x, y) = x + y, \quad y(0) = 1, \quad 0 \le x \le 1. \tag{8.8}$$

We use the first and second term of the Taylor series expansion to approximate the value of the solution y at a nearby point $x_0 + h$. We calculate the slope of the curve at the initial point (x_0, y_0) directly from the given differential equation and find $y'(x_0) = f(x_0, y_0) = x_0 + y_0 = 1$. We then select a step size h, and use the value $y_1 = y(x_1) = y(x_0 + h)$ as an approximate value for the true solution. This gives $y_1 = y(x_0 + h) = y(x_0) + y'(x_0)h + O(h^2)$, where the error of the approximation is of order h^2. Letting $h = 0.1$ and substituting in the values for x_0 and y_0 we find $y_1 = 1.1$ at $x_1 = 0.1$. If we repeat this step-by-step process with (x_1, y_1) as the new initial point, we obtain the algorithm illustrated in figure 8-1, which is called Euler's method or a first-order Taylor series method. Notice the algorithm in figure 8-1 is a single-step method. That is, if we know the approximate value y_m of the solution curve $y(x)$ at $x = x_m$, then from the point (x_m, y_m) we can take a single-step to the next approximating value (x_{m+1}, y_{m+1}), where

$$x_{m+1} = x_m + h \qquad \text{and} \qquad y_{m+1} = y_m + y'_m h + O(h^2). \tag{8.9}$$

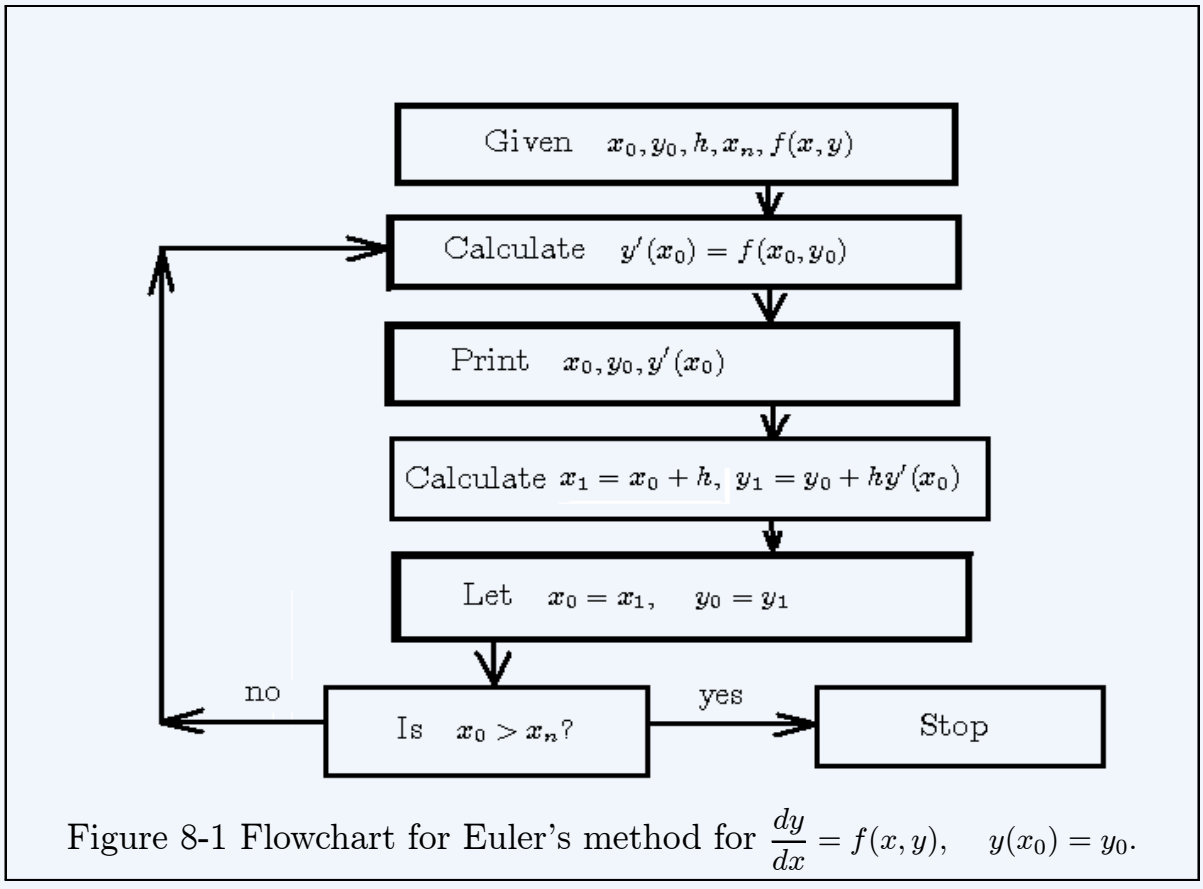

Figure 8-1 Flowchart for Euler's method for $\dfrac{dy}{dx} = f(x, y), \quad y(x_0) = y_0.$

Applying the Euler method single step algorithm illustrated in figure 8-1 to the initial value problem (8.8) and using a step size of $h = 0.1$, we obtain the numerical values in table 8.1. The fourth column in this table gives the exact solution for comparison purposes. Analysis of the error term associated with the Euler method is considered in the exercises at the end of this chapter.

Table 8.1
Numerical Results for Euler's method applied to
$$\frac{dy}{dx} = x + y, \quad y(0) = 1, \quad h = 0.1$$

x	$y(x)$	$y'(x)$	$y(x) = 2e^x - x - 1$	% error
0.0	1.000	1.000	1.000	0.00
0.1	1.100	1.200	1.110	0.93
0.2	1.220	1.420	1.243	1.84
0.3	1.362	1.662	1.400	2.69
0.4	1.528	1.928	1.584	3.51
0.5	1.721	2.221	1.797	4.25
0.6	1.943	2.543	2.044	4.95
0.7	2.197	2.897	2.328	5.61
0.8	2.487	3.287	2.651	6.19
0.9	2.816	3.716	3.019	6.73
1.0	3.187	4.187	3.437	7.26

■

Taylor Series Method

Other numerical methods for solving differential equations can be developed from the Taylor series equation (8.6). If we retain the first $(m+1)$ terms in the Taylor series expansion given by equation (8.6), one can write

$$y(x_0 + h) = y(x_0) + hT_m(x_0, y_0, h) + R_m \tag{8.10}$$

$$\text{with} \qquad T_m = T_m(x_0, y_0, h) = y'(x_0) + y''(x_0)\frac{h}{2!} + \ldots + y^{(m)}(x_0)\frac{h^{m-1}}{m!} \tag{8.11}$$

and $R_m = y^{(m+1)}(\xi)\dfrac{h^{m+1}}{(m+1)!} = O(h^{m+1})$ representing the error term of the approximation. Equation (8.10) represents an mth-order Taylor series approximation to the value $y_1 = y(x_0 + h)$. In order to use the above formula, it is necessary for us to obtain the various derivative terms $y^{(k)}(x_0)$, $k = 1, 2, 3, \ldots, m$. These derivatives

may be computed by differentiating the given differential equation (8.1) using the chain rule for differentiation. The first couple of derivatives are

$$\begin{aligned} y'(x_0) &= f\Big|_{(x_0,y_0)} & y'''(x_0) &= f_{xx} + 2f_{xy}f + f_x f_y + f_{yy}f^2 + f_y^2 f\Big|_{(x_0,y_0)} \\ y''(x_0) &= f_x + f_y f\Big|_{(x_0,y_0)} & y^{(iv)}(x_0) &= \frac{d}{dx}y'''(x)\Big|_{(x_0,y_0)} \end{aligned} \tag{8.12}$$

Higher derivatives become more difficult to calculate if $f = f(x, y)$ is a complicated expression. In equation (8.12) the subscripts denote partial derivatives. For example, $f_x = \frac{\partial f}{\partial x}$, $f_{xx} = \frac{\partial^2 f}{\partial x^2}$, $f_{xy} = \frac{\partial^2 f}{\partial x \partial y}$, etc. Of course the larger the value of m in the Taylor series expansion (8.10) (8.11), the more work is involved in calculating these higher derivatives. In most applications the Taylor series method should be used only when the derivatives of $f = f(x, y)$ are easily obtainable.

Example 8-2. (Second-order Taylor series method)

A second-order Taylor series algorithm for approximating solutions to the initial value problem (8.1) is given in figure 8-2. Applying the second-order Taylor series algorithm to the initial value problem (8.8) we obtain the results in table 8.2. Compare the results of table 8.1 with the entries in table 8.2 to see the difference in the errors between a first- and second-order Taylor series method.

Table 8.2

Numerical Results for second-order Taylor series method applied to

$$\frac{dy}{dx} = x + y, \quad y(0) = 1, \quad h = 0.1$$

x	$y(x)$	$y'(x)$	$y''(x)$	$y(x) = 2e^x - x - 1$	% error
0.0	1.000	1.000	2.000	1.000	0.000
0.1	1.100	1.210	2.210	1.110	0.031
0.2	1.242	1.442	2.442	1.243	0.061
0.3	1.398	1.698	2.698	1.400	0.089
0.4	1.582	1.982	2.982	1.584	0.117
0.5	1.795	2.295	3.295	1.797	0.142
0.6	2.041	2.641	3.641	2.044	0.165
0.7	2.323	3.023	4.023	2.328	0.187
0.8	2.646	3.446	4.446	2.651	0.298
0.9	3.012	3.912	4.912	3.019	0.227
1.0	3.428	4.428	5.428	3.437	0.244

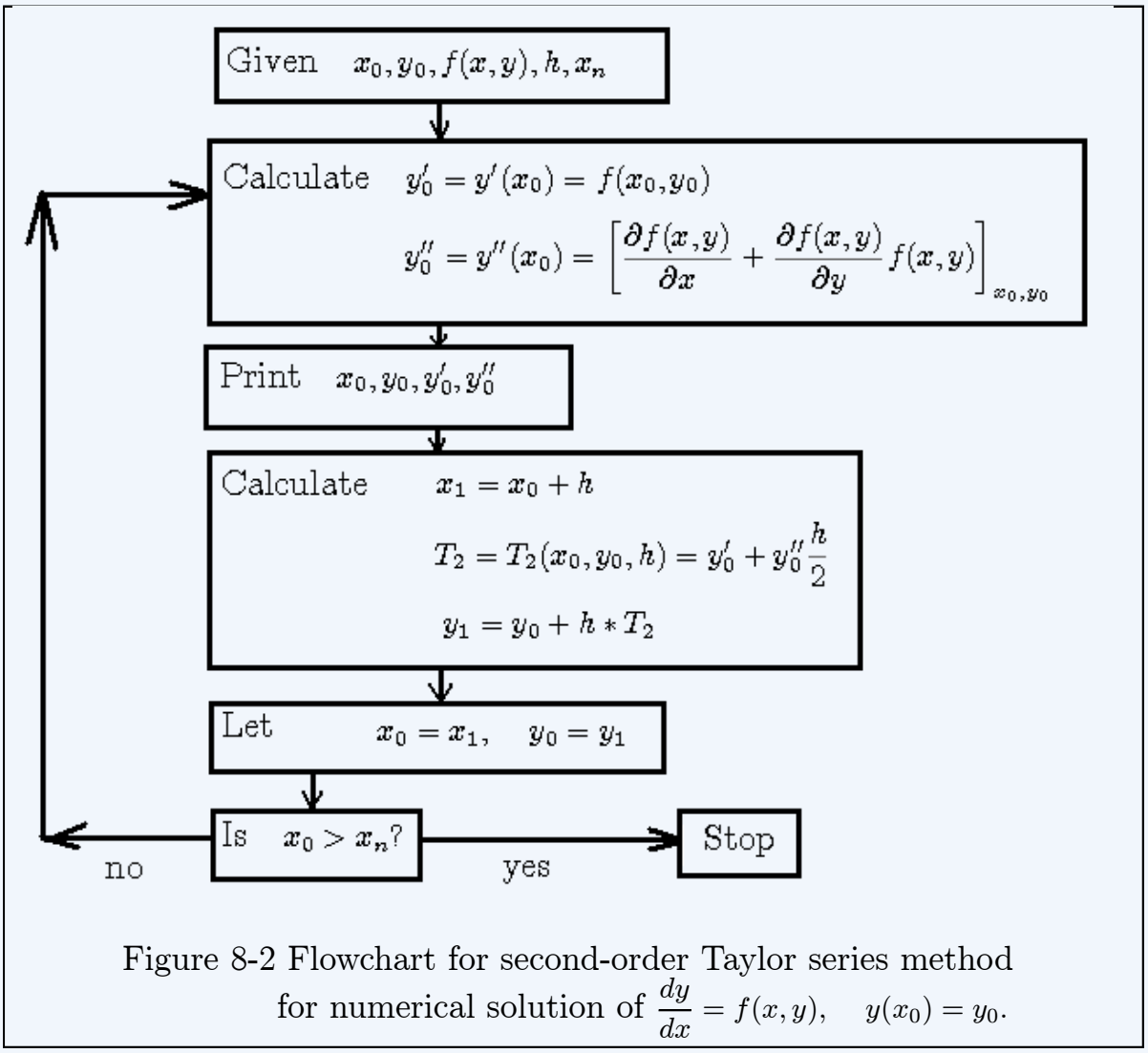

Figure 8-2 Flowchart for second-order Taylor series method for numerical solution of $\frac{dy}{dx} = f(x,y), \quad y(x_0) = y_0$.

■

Example 8-3. (Fourth-order Taylor series method)

Set up a fourth-order Taylor series algorithm to solve the initial value problem

$$y'(x) = \frac{dy}{dx} = f(x,y) = x + y, \quad y(0) = 1, \quad 0 \le x \le 1.$$

Solution: We differentiate the given differential equation to obtain derivatives through order four. There results the following equations:

$$y'(x) = x + y \qquad y'''(x) = 1 + y' = 1 + x + y$$
$$y''(x) = 1 + y' = 1 + x + y \qquad y^{(iv)}(x) = 1 + y' = 1 + x + y$$

Substituting the above derivatives into the Taylor series gives us the fourth-order

single step approximation

$$x_1 = x_0 + h \qquad\qquad y_1 = y(x_0 + h) = y_0 + hT_4,$$

where $y_0 = y(x_0)$ and

$$T_4 = (x_0 + y_0) + (1 + x_0 + y_0)\frac{h}{2!} + (1 + x_0 + y_0)\frac{h^2}{3!} + (1 + x_0 + y_0)\frac{h^3}{4!}.$$

This gives the fourth-order Taylor series algorithm illustrated in the figure 8-3.

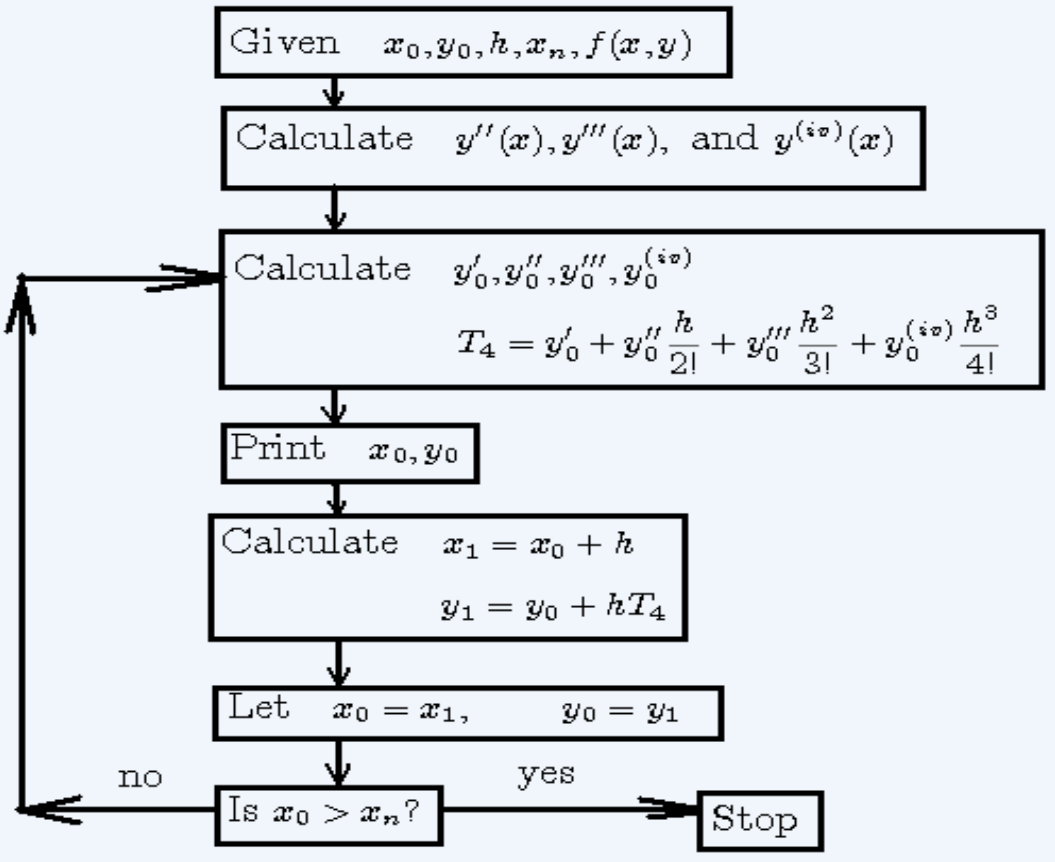

Figure 8-3 Flowchart for fourth-order Taylor series method for numerical solution of $\frac{dy}{dx} = f(x,y), \quad y(x_0) = y_0$.

■

Runge-Kutta Methods

There are various types of Runge-Kutta algorithms for the numerical solution of the initial value problem

$$y' = f(x,y), \quad y(x_0) = y_0, \quad x_0 \le x \le x_n. \tag{8.13}$$

These methods are single step methods with step size h. A p-stage Runge-Kutta algorithm to solve the initial value problem (8.13) is a stepping procedure from a point (x_i, y_i) to the next point (x_{i+1}, y_{i+1}) given by

$$\begin{aligned} x_{i+1} &= x_i + h \\ y_{i+1} &= y_i + \omega_1 k_1 + \omega_2 k_2 + \omega_3 k_3 + \ldots + \omega_p k_p, \end{aligned} \tag{8.14}$$

where $\omega_1,\ \omega_2, \ldots, \omega_p$ are called weighting constants and

$$\begin{aligned} k_1 &= hf(x_i, y_i) \\ k_2 &= hf(x_i + c_2 h, y_i + a_{21} k_1) \\ k_3 &= hf(x_i + c_3 h, y_i + a_{31} k_1 + a_{32} k_2) \\ &\cdots \\ k_p &= hf(x_i + c_p h, y_i + a_{p1} k_1 + a_{p2} k_2 + \ldots + a_{p,p-1} k_{p-1}), \end{aligned} \tag{8.15}$$

are scaled slope calculations at specified points. In the equation (8.15) the quantities $c_2, c_3, \ldots, c_p, a_{21}, a_{31}, a_{32}, \ldots$ are given constants. The $k_i,\ \ i = 1, \ldots, p$ values require p function evaluations for the slope $f(x, y)$ to be evaluated at specified (x, y) points. In general, a p-stage Runge-Kutta method is a stepping method, with step size h, which requires p function evaluations for each step taken.

There is an array notation for representing Runge-Kutta methods which was introduced by J.C. Butcher around 1965. The array is called a Butcher array and has the form

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdots & a_{1p} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_p & a_{p1} & a_{p2} & \cdots & a_{pp} \\ \hline & \omega_1 & \omega_2 & \cdots & \omega_p \end{array} \qquad \begin{array}{c|c} \vec{c} & \mathbf{A} \\ \hline & \vec{\omega} \end{array} \tag{8.16}$$

where $\vec{c} = (c_1, c_2, \ldots, c_p)^T$ and $\vec{\omega} = (\omega_1, \omega_2, \ldots, \omega_p)$ are vectors and $\mathbf{A} = (a_{ij})$ for $i = 1, \ldots, p$ and $j = 1, \ldots, p$ is a $p \times p$ matrix array. The Runge-Kutta method given

by equations (8.14) and (8.15) is denoted by the Butcher array

$$\begin{array}{c|cccccc} 0 & 0 & & & & & \\ c_2 & a_{21} & 0 & & & & \\ c_3 & a_{31} & a_{32} & 0 & & & \\ c_4 & a_{41} & a_{42} & a_{43} & 0 & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \\ c_p & a_{p1} & a_{p2} & a_{p3} & \cdots & a_{p,p-1} & 0 \\ \hline & \omega_1 & \omega_2 & \omega_3 & \cdots & \omega_{p-1} & \omega_p \end{array} \tag{8.17}$$

where the matrix $\mathbf{A}$ is lower triangular. Whenever the matrix $\mathbf{A}$ is strictly lower triangular, the Runge-Kutta method is called an explicit stepping method. If the matrix $\mathbf{A}$ is not lower triangular, then the Runge-Kutta method is called an implicit stepping method.

The p-stage Runge-Kutta method requires that the weighting constants $\omega_1,\ \omega_2, \ldots, \omega_p$ and the constants

$$\begin{array}{ll} c_2 & a_{21} \\ c_3 & a_{31}\ a_{32} \\ c_4 & a_{41}\ a_{42}\ a_{43} \\ & \cdots \\ c_p & a_{p1}\ a_{p2}\ a_{p3} \ldots a_{p,p-1}, \end{array}$$

are chosen such that y_{n+1} of equation (8.14) agrees with a Taylor series expansion through some order. It is known that if equation (8.14) agrees with a Taylor series expansion of order m, for $m = 2, 3$ or 4, then $p = m$ and for $m > 4$ it is known that $p > m$. It is also known that for consistency it is required that

$$c_i = \sum_{j=1}^{i-1} a_{ij}, \quad \text{for} \quad i = 2, 3, \ldots, p. \tag{8.18}$$

Note that p-function evaluations are required if one uses a p-stage Runge-Kutta method. The Runge-Kutta methods for $p = 2, 3, 4$ have the same order of accuracy as the Taylor series methods of order $m = 2, 3, 4$ and they have the advantage that higher ordered derivative terms do not have to be calculated. In order to illustrate the general character of Runge-Kutta methods we begin by developing second-order Runge-Kutta methods in detail.