Chapter 8
Numerical Solution of Ordinary Differential Equations

8.0 Introduction

- This chapter concerns numerical problems involving ordinary differential equations.
- The central problem is to solve a single first-order equation when one point on the solution curve is given.
- Consider the following initial-value problem written in the form:

$$\begin{cases} y' = f(x, y) & (1.1) \\ y(x_0) = y_0 & (1.2) \end{cases}$$

Here y is an unknown function of x that we hope to constuct from the information given in Equations (1.1) and (1.2), where $y' = dy(x) / dx$. Equation (1.2) specifies one particular value of the function $y(x)$. Equation (1.1) gives the slope of the curve y at any point x. Of course, the function f must be specified.

8.1 The Existence and Uniqueness of Solutions

Theorem 3 Existence and Uniqueness Theorem, Initial-Value Problem

If f is continuous in the strip $a \le x \le b, -\infty < y < \infty$ and satisfies an inequality

$$\left| f(x, y_1) - f(x, y_2) \right| \le L \left| y_1 - y_2 \right| \quad (1.3)$$

Then the initial-value problem (1.1) (1.2) has a unique solution in the interval $[a, b]$.

Inequality (1.3) is called a **Lipschitz condition** in the second variable. For a function of one variable, such a condition would assert simply

$$\left| g(x_1) - g(x_2) \right| \le L \left| x_1 - x_2 \right|$$

8.2 Taylor-Series Method

- Euler's Method

In the numerical solution of differential equations, we rarely expect to obtain the solution directly as a formula giving $y(x)$ as a function of x. For a series of nodes

$$x_1 < x_2 < \cdots < x_n < x_{n+1} < \cdots$$

instead of finding the **exact** solution $y(x_i)$ at x_i, we usually compute the **approximate** value y_i. $h_n = x_{n+1} - x_n$ is called the **step**. Unless specified, we assume that $h_i = h \ (i = 1, 2, \cdots)$, then $x_n = x_0 + nh, n = 0, 1, 2, \cdots$.

From the initial point $P_0(x_0, y_0)$, we find the next point $P_1(x_1, y_1)$, according to equations (1.1) (1.2),

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

8.2 Taylor-Series Method

- Euler's Method

We use the slope of $\overline{P_0P_1}$ to approximate $y'(x_0)$, then,

$$\frac{y_1 - y_0}{x_1 - x_0} = f(x_0, y_0) \quad \Rightarrow \quad y_1 = y_0 + hf(x_0, y_0)$$

Generally, from the point $P_n(x_n, y_n)$, to find the next point $P_{n+1}(x_{n+1}, y_{n+1})$, similarly, we have

$$\frac{y_{n+1} - y_n}{x_{n+1} - x_n} = f(x_n, y_n)$$

$$y_{n+1} = y_n + hf(x_n, y_n) \qquad (2.1)$$

This is called the **Euler's formula**.

Given the initial y_0, according to (2.1),

$$y_1 = y_0 + hf(x_0, y_0)$$

$$y_2 = y_1 + hf(x_1, y_1)$$

$$\cdots$$

8.2 Taylor-Series Method

- Euler's Method

Example 1 Solve the following initial-value problem

$$\begin{cases} y' = y - \dfrac{2x}{y} \quad (0 < x < 1) \\ y(0) = 1 \end{cases} \qquad (2.2)$$

Solution : Using the Euler Method,

$$y_{n+1} = y_n + h(y_n - \frac{2x_n}{y_n})$$

Set $h = 0.1,$ the computed results are shown in Table 8.1.

The initial-value problem (2.2) has the solution $y = \sqrt{1+2x}$, according to this formula, we can compute the exact values $y(x_n),$ which are shown with the approximate values y_n in Table 8.1.

8.2 Taylor-Series Method

- Euler's Method

Table 8.2 comparison of computation results

x_n	y_n	$y(x_n)$	x_n	y_n	$y(x_n)$
0.1	1.1000	1.0954	0.6	1.5090	1.4832
0.2	1.1918	1.1832	0.7	1.5803	1.5492
0.3	1.2774	1.2649	0.8	1.6498	1.6125
0.4	1.3582	1.3416	0.9	1.7178	1.6733
0.5	1.4351	1.4142	1.0	1.7848	1.7321

From the comparison between y_n and $y(x_n)$, we can see that the precision of the Euler Method is not high.

8.2 Taylor-Series Method

- Euler's Method

To analyze the precision of the Euler Method, we write the Taylor-series of $y(x_{n+1})$ at x_n as

$$y(x_{n+1}) = y(x_n + h) = y(x_n) + y'(x_n)h + \frac{h^2}{2} y''(\xi_n), \ \xi_n \in (x_n, x_{n+1})$$

Assume that $y_n = y(x_n)$, then $f(x_n, y_n) = f(x_n, y(x_n)) = y'(x_n)$.

Thus, we can obtain the error of the Euler formula (2.1):

$$y(x_{n+1}) - y_{n+1} = \frac{h^2}{2} y''(\xi_n) \approx \frac{h^2}{2} y''(x_n), \quad (2.3)$$

which is called the **local truncation error**.

Integrate (1.1) from x_n to x_{n+1}, then,

$$y(x_{n+1}) = y(x_n) + \int_{x_n}^{x_{n+1}} f(t, y(t))dt. \quad (2.4)$$

If we use $hf(x_n, y(x_n))$ to approximate the above integral, replace $y(x_n)$ and $y(x_{n+1})$ with y_n and y_{n+1}, then we can also obtain (2.1) with the same local truncation error in (2.3).

8.2 Taylor-Series Method

- Euler's Method

If we use $hf(x_{n+1}, y(x_{n+1}))$ to approximate the integral in (2.4), then we obtain another formula:

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}), \qquad (2.5)$$

which is called the **Backward Euler formula**.

Euler formula $y_{n+1} = y_n + hf(x_n, y_n)$ (2.1) ---- explicit

Backward Euler formula $y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$ (2.5) ---- implicit

The implicit equation is usually solved by the following iterative method.

Use the Euler formula to give the initial value $y_{n+1}^{(0)}$

$$y_{n+1}^{(0)} = y_n + hf(x_n, y_n),$$

substitute it to the right hand side of (2.5) to compute $y_{n+1}^{(1)}$

$$y_{n+1}^{(1)} = y_n + hf(x_{n+1}, y_{n+1}^{(0)}),$$

8.2 Taylor-Series Method

- Euler's Method

Again substitute it to the right hand side of (2.5) to compute $y_{n+1}^{(2)}$

$$y_{n+1}^{(2)} = y_n + hf(x_{n+1}, y_{n+1}^{(1)}),$$

then we can get the iterative formula

$$y_{n+1}^{(k+1)} = y_n + hf(x_{n+1}, y_{n+1}^{(k)}), \quad (k = 0,1,\cdots). \qquad (2.6)$$

Since $f(x, y)$ satisfies the Lipschitz condition (1.3) w.r.t. y, subtract (2.5) from (2.6), we get

$$\left|y_{n+1}^{(k+1)} - y_{n+1}\right| = h\left|f(x_{n+1}, y_{n+1}^{(k)}) - f(x_{n+1}, y_{n+1})\right| \le hL\left|y_{n+1}^{(k)} - y_{n+1}\right|.$$

Thus, if $hL < 1$, then the sequence $\left[y_{n+1}^{(k+1)}\right]$ obtained from (2.6) converges to the solution y_{n+1}.

8.2 Taylor-Series Method

- Trapezoidal Method

To improve the computation accuracy, we use the trapezoidal integration formula to approximate the integral in (2.4), replace $y(x_n)$ and $y(x_{n+1})$ with y_n and y_{n+1}, then:

$$y_{n+1} = y_n + \frac{h}{2}\left[f(x_n, y_n) + f(x_{n+1}, y_{n+1})\right], \qquad (2.7)$$

which is called the **trapezoidal method**.

The trapezoidal method is an impicit single-step method, and it can be solved by the iterative method. Similarly to the backward Euler method, the initial value is given by the Euler mehod, then the iterative formula of the trapezoidal method is as follows:

$$\begin{cases} y_{n+1}^{(0)} = y_n + hf(x_n, y_n), \\ y_{n+1}^{(k+1)} = y_n + \dfrac{h}{2}\left[f(x_n, y_n) + f(x_{n+1}, y_{n+1}^{(k)})\right], \quad (k = 0,1,\cdots). \end{cases} \qquad (2.8)$$

To analyze the convergence, subtract (2.7) from (2.8), we get

$$\left| y_{n+1}^{(k+1)} - y_{n+1} \right| = \frac{h}{2}\left| f(x_{n+1}, y_{n+1}^{(k)}) - f(x_{n+1}, y_{n+1}) \right| \le \frac{hL}{2}\left| y_{n+1}^{(k)} - y_{n+1} \right|.$$

If $hL/2 < 1$, then $y_{n+1}^{(k+1)} \to y_{n+1}, k \to \infty,$ which means the iteration (2.8) is convergent.

8.2 Taylor-Series Method

- The Local Truncation Error and Convergence Order of One-step Method

The single-step method of the inital-value problem (1.1) and (1.2) can be represented generally as:

$$y_{n+1} = y_n + h\varphi(x_n, y_n, y_{n+1}, h), \qquad (2.9)$$

where φ is associated with $f(x, y)$, when φ contains y_{n+1}, the method is implicit, else it is explicit. Thus the explicit single-step method can be given by

$$y_{n+1} = y_n + h\varphi(x_n, y_n, h), \qquad (2.10)$$

φ is called the increment function, for example, for the Euler method in (2.1),

$$\varphi(x, y, h) = f(x, y).$$

The local truncation error is given by (2.3).

$$y(x_{n+1}) - y_{n+1} = \frac{h^2}{2} y''(\xi_n) \approx \frac{h^2}{2} y''(x_n), \qquad (2.3)$$

For the general single-step method, we give the following definition.

Definition 1 Assume that $y(x)$ is the accurate solution of the inital-value problem (1.1) and (1.2), then

$$T_{n+1} = y(x_{n+1}) - y(x_n) - h\varphi(x_n, y(x_n), h) \qquad (2.11)$$

is called the **local truncation error** of the explicit single-step method (2.10).

8.2 Taylor-Series Method

- The Local Truncation Error and Convergence Order of One-step Method

T_{n+1} is local, because we assume that there is no error before x_n. When $y_n = y(x_n)$, calculate for one step, then

$$y(x_{n+1}) - y_{n+1} = y(x_{n+1}) - \left[y_n + h\varphi(x_n, y_n, h)\right] = y(x_{n+1}) - y(x_n) - h\varphi(x_n, y(x_n), h) = T_{n+1}.$$

Thus, T_{n+1} is the error made in one step when using (2.10), which is the formula error when we replace the numerical solution with the accurate solution $y(x)$.

From the definition, the local truncation error of the Euler method is:

$$T_{n+1} = y(x_{n+1}) - y(x_n) - hf(x_n, y(x_n)) = y(x_n + h) - y(x_n) - hy'(x_n) = \frac{h^2}{2} y''(x_n) + O(h^3),$$

which is consistent with (2.3). Here, $\frac{h^2}{2} y''(x_n)$ is called the dominant term of the local truncation error. Obviously, $T_{n+1} = O(h^2)$.

8.2 Taylor-Series Method

- The Local Truncation Error and Convergence Order of One-step Method

Definition 2 Assume that $y(x)$ is the accurate solution of the inital-value problem (1.1) and (1.2), if there exists a largest integer such that

$$T_{n+1} = y(x+h) - y(x) - h\varphi(x, y, h) = O(h^{p+1}), \qquad (2.12)$$

then we say that the explicit single-step method (2.10) is **of order** p.

Rewrite (2.12) as

$$T_{n+1} = \psi(x_n, y(x_n))h^{p+1} + O(h^{p+2}),$$

Then $\psi(x_n, y(x_n))h^{p+1}$ is called the **dominant term of the local truncation error**.

The above definition is also applicable for the implicit single-step method (2.9).

For example, the local truncation error of the backward Euler method is:

$$T_{n+1} = y(x_{n+1}) - y(x_n) - hf(x_{n+1}, y(x_{n+1})) = hy'(x_n) + \frac{h^2}{2}y''(x_n) + O(h^3)$$

$$-h\left[y'(x_n) + hy''(x_n) + O(h^2)\right] = -\frac{h^2}{2}y''(x_n) + O(h^3).$$

Here, $p = 1$, the method is of order 1, and the dominant term of the local truncation error is $-\frac{h^2}{2}y''(x_n)$.

8.2 Taylor-Series Method

- The Local Truncation Error and Convergence Order of One-step Method

Similarly, for the trapezoidal method (2.7),

$$T_{n+1} = y(x_{n+1}) - y(x_n) - \frac{h}{2}\left[y'(x_n) + y'(x_{n+1})\right] = hy'(x_n) + \frac{h^2}{2}y''(x_n) + \frac{h^3}{3!}y'''(x_n)$$

$$-\frac{h}{2}\left[y'(x_n) + y'(x_n) + hy''(x_n) + \frac{h^2}{2}y'''(x_n)\right] + O(h^4) = -\frac{h^3}{12}y'''(x_n) + O(h^4).$$

Thus, the trapezoidal method is of order 2, and the dominant term of its local truncation error is $-\frac{h^3}{12}y'''(x_n)$.

Euler method	$p = 1$	order 1
Backward Euler method	$p = 1$	order 1
Trapezoidal method	$p = 2$	order 2

8.2 Taylor-Series Method

- Modified Euler Method

The trapezoidal method improves the computation accuracy compared with the Euler method, but its calculation is complex when using the iteration (2.8). In practice, the following simplified method is used.

Specifically, we first use the Euler formula to give an inital approximate value $\bar{y}_{n+1}$, which is called the **predictor**. The precision of $\bar{y}_{n+1}$ is low, then we use the trapezoidal formula (2.7) to correct it, that is, iterate (2.8) for only one time to obtain y_{n+1}, which is called the **corrector**. This predictor-corrector system is called the **modified Euler formula**:

$$\begin{cases} \text{prediction} \quad \bar{y}_{n+1} = y_n + hf(x_n, y_n), \\ \text{correction} \quad y_{n+1} = y_n + \dfrac{h}{2}\left[f(x_n, y_n) + f(x_{n+1}, \bar{y}_{n+1}) \right]. \end{cases} \tag{2.13}$$

This formula can also be represented as the following average form:

$$\begin{cases} y_p = y_n + hf(x_n, y_n), \\ y_c = y_n + hf(x_{n+1}, y_p), \\ y_{n+1} = \dfrac{1}{2}(y_p + y_c). \end{cases}$$

8.2 Taylor-Series Method

- Modified Euler Method

Example 2 Use the modified Euler method to solve the initial-value problem

$$\begin{cases} y' = y - \dfrac{2x}{y} \quad (0 < x < 1) \\ y(0) = 1 \end{cases} \qquad (2.2)$$

Solution: The modified Euler formula for this problem is

$$\begin{cases} y_p = y_n + h\left(y_n - \dfrac{2x_n}{y_n} \right), \\ y_c = y_n + h\left(y_p - \dfrac{2x_{n+1}}{y_p} \right), \\ y_{n+1} = \dfrac{1}{2}(y_p + y_c). \end{cases}$$

Set $h = 0.1,$ the computed results are shown in Table 8.2. Compared with the results of the Euler method in Table 8.1, it can be seen clearly that the modified Euler method improves the accuracy.

8.2 Taylor-Series Method

- Modified Euler Method

Table 8.2 comparison of computation results

x_n	y_n	$y(x_n)$	x_n	y_n	$y(x_n)$
0.1	1.0959	1.0954	0.6	1.4860	1.4832
0.2	1.1841	1.1832	0.7	1.5525	1.5492
0.3	1.2662	1.2649	0.8	1.6153	1.6125
0.4	1.3434	1.3416	0.9	1.6782	1.6733
0.5	1.4164	1.4142	1.0	1.7379	1.7321

8.3 Runge-Kutta Methods

- General Form of the Runge-Kutta Methods $y_{n+1} = y_n + h\varphi(x_n, y_n, h),$ (2.10)

We present the explicit single-step method in (2.10), and its local truncation error is given in (2.12). For the Euler method, $T_{n+1} = O(h^2)$, the method is of order $p = 1$. For the modified Euler method, $T_{n+1} = y(x_{n+1}) - y(x_n) - h\varphi(x_n, y(x_n), h),$ (2.12)

$$y_{n+1} = y_n + \frac{h}{2}\left[f(x_n, y_n) + f(x_n + h, y_n + hf(x_n, y_n)) \right]. \quad (3.1)$$

The increment function is

$$\varphi(x_n, y_n, h) = \frac{1}{2}\left[f(x_n, y_n) + f(x_n + h, y_n + hf(x_n, y_n)) \right]. \quad (3.2)$$

Compared with the increment function of the Euler method $\varphi(x_n, y_n, h) = f(x_n, y_n)$, one more value of f is added, and we expect that $p = 2$. If we want to obtain higher order p, φ must contain more values of f. Actually, from the equivalent integral form (2.4) of equation (1.1), that is $y' = f(x, y)$ (1.1)

$$y(x_{n+1}) - y(x_n) = \int_{x_n}^{x_{n+1}} f(x, y(x))dx, \quad (3.3)$$

to improve the order of the formula, it is necessary to improve the precision of the quadrature formula for the integral on the right hand side, which means that more nodes are needed.

8.3 Runge-Kutta Methods

- General Form of the Runge-Kutta Methods

Thus we approximate the integral in (3.3) with the following quadrature formula

$$\int_{x_n}^{x_{n+1}} f(x, y(x))dx \approx h\sum_{i=1}^{r} c_i f(x_n + \lambda_i h, y(x_n + \lambda_i h)).$$

Generally speaking, the larger the number r is, the higher the order p is. Similarly to the modified Euler method in (3.1) and (3.2), the formula can be represented as follows

$$y_{n+1} = y_n + h\varphi(x_n, y_n, h), \qquad (3.4)$$

$$\varphi(x_n, y_n, h) = \sum_{i=1}^{r} c_i K_i, \qquad (3.5)$$

$$K_1 = f(x_n, y_n),$$

$$K_i = f(x_n + \lambda_i h, y_n + h\sum_{j=1}^{r-1} \mu_{ij} K_j), \quad i = 2, \cdots, r,$$

where c_i, λ_i, μ_{ij} are constants. (3.4) and (3.5) are called the **Runge - Kutta (R - K) Method** of order r.

$r = 1$, $\varphi(x_n, y_n, h) = f(x_n, y_n)$ -- Euler method, $p = 1$

$r = 2$, (3.1) (3.2) -- modified Euler method is one kind of R-K method, $p = 2$ (proved later)

8.3 Runge-Kutta Methods

- Second-Order Runge-Kutta Method

Let $r = 2$, from (3.4) and (3.5) we can obtain

$$\begin{cases} y_{n+1} = y_n + h\left(c_1 K_1 + c_2 K_2\right) \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + \lambda_2 h, y_n + \mu_{21} h K_1) \end{cases} \qquad (3.6)$$

where c_1, c_2, λ_2, μ_{21} are coefficients to be determined. We choose appropriate values for them to make p as large as possible. From **Definition 1**, the local truncation error of (3.6) is

$$T_{n+1} = y(x_{n+1}) - y(x_n) - h\left[c_1 f(x_n, y_n) + c_2 f(x_n + \lambda_2 h, y_n + \mu_{21} h f_n)\right] \qquad (3.7)$$

where $y_n = y(x_n)$, $f_n = f(x_n, y_n)$. To obtain the order p, we need to give the Taylor series of the terms at (x_n, y_n).

$$y(x_{n+1}) = y_n + h y_n' + \frac{h^2}{2} y_n'' + \frac{h^3}{3!} y_n''' + O(h^4),$$

where

$$\begin{cases} y_n' = f(x_n, y_n) = f_n \\ y_n'' = \dfrac{d}{dx} f(x_n, y(x_n)) = f_x'(x_n, y_n) + f_y'(x_n, y_n) \cdot f_n \\ y_n''' = f_{xx}''(x_n, y_n) + 2 f_n f_{xy}''(x_n, y_n) + f_n^{\,2} f_{yy}''(x_n, y_n) + f_y'(x_n, y_n)\left[f_x'(x_n, y_n) + f_n f_y'(x_n, y_n)\right] \end{cases} \qquad (3.8)$$

8.3 Runge-Kutta Methods

- Second-Order Runge-Kutta Method

$$f(x_n+\lambda_2 h, y_n+\mu_{21}hf_n) = f_n + f_x'(x_n,y_n)\lambda_2 h + f_y'(x_n,y_n)\mu_{21}hf_n + O(h^2).$$

Substitue the above results into (3.7), then

$$\begin{aligned} T_{n+1} &= hf_n + \frac{h^2}{2}\left[f_x'(x_n,y_n) + f_y'(x_n,y_n) f_n \right] \\ &\quad - h\left[c_1 f_n + c_2\left(f_n + f_x'(x_n,y_n)\lambda_2 h + f_y'(x_n,y_n)\mu_{21}hf_n \right)\right] + O(h^3) \\ &= (1-c_1-c_2) f_n h + \left(\frac{1}{2}-c_2\lambda_2\right) f_x'(x_n,y_n)h^2 + \left(\frac{1}{2}-c_2\mu_{21}\right) f_y'(x_n,y_n) f_n h^2 + O(h^3). \end{aligned}$$

To make the order of (3.6) $p = 2$, the following equations should be satisfied

$$1-c_1-c_2=0, \quad \frac{1}{2}-c_2\lambda_2=0, \quad \frac{1}{2}-c_2\mu_{21}=0. \qquad (3.9)$$

Or

$$c_2\lambda_2 = \frac{1}{2}, \quad c_2\mu_{21} = \frac{1}{2}, \quad c_1+c_2=1.$$

The solution of (3.9) is not unique, let $c_2 = a \neq 0$, then

$$c_1 = 1-a, \quad \lambda_2 = \mu_{21} = \frac{1}{2a}.$$

8.3 Runge-Kutta Methods

- Second-Order Runge-Kutta Method

The obtained formula is called the **Second - Order R - K Method**.

$$c_2 = a \neq 0, \quad c_1 = 1 - a, \quad \lambda_2 = \mu_{21} = \frac{1}{2a}.$$

Let $a = 1/2$, then $c_1 = c_2 = 1/2$, $\lambda_2 = \mu_{21} = 1$, this is the modified Euler method (3.1).

Let $a = 1$, then $c_2 = 1$, $c_1 = 0$, $\lambda_2 = \mu_{21} = 1/2$, we obtain

$$\begin{cases} y_{n+1} = y_n + hK_2 \\ K_1 = f(x_n, y_n) \\ K_2 = f(x_n + \dfrac{h}{2}, y_n + \dfrac{h}{2}K_1) \end{cases} \qquad (3.10)$$

this is called the **Mid - Point Formula**, which is corresponding to the mid-rectangle formula in numerical integration. (3.10) can also be represented as

$$y_{n+1} = y_n + hf(x_n + \frac{h}{2}, y_n + \frac{h}{2} f(x_n, y_n)).$$

Question: Is it possible to improve the local error of the second-order R-K formula (3.6) to $O(h^4)$? Or can the order $p = 3$ when $r = 2$?

8.3 Runge-Kutta Methods

- Third-Order Runge-Kutta Method

Let $r = 3$, from (3.4) and (3.5) we can obtain

$$\begin{cases} y_{n+1} = y_n + h\left(c_1K_1 + c_2K_2 + c_3K_3\right) \\ K_1 = f\left(x_n, y_n\right) \\ K_2 = f\left(x_n + \lambda_2 h, y_n + \mu_{21}hK_1\right) \\ K_3 = f\left(x_n + \lambda_3 h, y_n + \mu_{31}hK_1 + \mu_{32}hK_2\right) \end{cases} \tag{3.11}$$

where c_1, c_2, c_3, λ_2, μ_{21}, λ_3, μ_{31}, μ_{32} are coefficients to be determined. The local truncation error of (3.11) is

$$T_{n+1} = y(x_{n+1}) - y(x_n) - h\left[c_1K_1 + c_2K_2 + c_3K_3\right].$$

Similarly to the second-order R-K method, give the Taylor series of K_2, K_3, let $T_{n+1} = O(h^4)$, we can derive that the coefficients satisfy the following equation system

$$\begin{cases} c_1 + c_2 + c_3 = 1 \\ \lambda_2 = \mu_{21} \\ \lambda_3 = \mu_{31} + \mu_{32} \\ c_2\lambda_2 + c_3\lambda_3 = \dfrac{1}{2} \\ c_2\lambda_2^{\,2} + c_3\lambda_3^{\,2} = \dfrac{1}{3} \\ c_3\lambda_2\mu_{32} = \dfrac{1}{6} \end{cases} \tag{3.12}$$

8.3 Runge-Kutta Methods

- Third-Order Runge-Kutta Method

The solution of (3.12) is not unique, either. We can obtain many formulas. The formulas (3.11) satisfying (3.12) are all called **Third - Order R - K Method**. We present a classical third-order R-K formula as follows

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{6}\left(K_1 + 4K_2 + K_3\right) \\ K_1 = f(x_n, y_n) \\ K_2 = f\left(x_n + \dfrac{h}{2}, y_n + \dfrac{h}{2}K_1\right) \\ K_3 = f(x_n + h, y_n - hK_1 + 2hK_2) \end{cases}$$

which is also called the **Third - Order Kutta Method**.

8.3 Runge-Kutta Methods

- Fourth-Order Runge-Kutta Method

The higher-order R-K formulas are very tedious to derive, and we shall not do so. The formulas are rather elegant, however, and are easily programmed once they have been derived.

Here we give the formulas for one classical **Fourth - Order R - K Method**:

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{6}\left(K_1 + 2K_2 + 2K_3 + K_4\right) \\ K_1 = f(x_n, y_n) \\ K_2 = f\left(x_n + \dfrac{h}{2}, y_n + \dfrac{h}{2}K_1\right) \\ K_3 = f\left(x_n + \dfrac{h}{2}, y_n + \dfrac{h}{2}K_2\right) \\ K_4 = f(x_n + h, y_n + hK_3) \end{cases} \qquad (3.13)$$

For the fourth-order R-K method, in each step four values of the function f are computed. It can be proved that the error is $O(h^5)$.

8.3 Runge-Kutta Methods

- Fourth-Order Runge-Kutta Method

Example 3 Use the fourth-order R-K method to solve the initial-value problem

$$\begin{cases} y' = y - \dfrac{2x}{y} \quad (0 < x < 1) \\ y(0) = 1 \end{cases} \qquad (2.2)$$

Solution : The fourth-order R-K formula (3.13) for this problem becomes

$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{6}\left(K_1 + 2K_2 + 2K_3 + K_4\right) \\ K_1 = y_n - \dfrac{2x_n}{y_n} \\ K_2 = y_n + \dfrac{h}{2}K_1 - \dfrac{2x_n + h}{y_n + \dfrac{h}{2}K_1} \\ K_3 = y_n + \dfrac{h}{2}K_2 - \dfrac{2x_n + h}{y_n + \dfrac{h}{2}K_2} \\ K_4 = y_n + hK_3 - \dfrac{2\left(x_n + h\right)}{y_n + hK_3} \end{cases}$$

8.3 Runge-Kutta Methods

- Fourth-Order Runge-Kutta Method

Table 8.3 comparison of computation results

x_n	y_n	$y(x_n)$
0.2	1.1832	1.1832
0.4	1.3417	1.3416
0.6	1.4833	1.4832
0.8	1.6125	1.6125
1.0	1.7321	1.7321

Set $h = 0.2,$ the computed results are shown in Table 8.3.

Comparing with the results between **Example** 2 and **Example** 3, it can be seen clearly that the fourth R-K method is more accurate than the modified Euler method.

8.4 Multistep Methods

- General Form of the Multistep Methods

The Taylor-series method and the Runge-Kutta method for solving the initial-value problem are **single - step methods** because they don't use any knowledge of prior values of $y(x)$ when the solution is advanced from x to $x+h$. If $x_0, x_1, \cdots, x_n$ are steps along the x-axis, then y_{n+1} (the approximate value of $y(x_{n+1})$) depends only on y_n, and knowledge of the approximate values $y_{n-1}, y_{n-2}, \cdots, y_0$ are is not used.

More efficient procedures can be devised if some prior values of the solution are utilized at each step.

When computing y_{n+k}, besides $y_{n+k-1}, y_{n+i} (i = 0,1,\cdots,k-2)$ are also used, then the method is called **linear multistep methods**. The general formulas can be represented as

$$y_{n+k} = \sum_{i=0}^{k-1} \alpha_i y_{n+i} + h \sum_{i=0}^{k} \beta_i f_{n+i} \qquad (5.1)$$

where y_{n+i} is the approximate value of $y(x_{n+i})$, $f_{n+i} = f(x_{n+i}, y_{n+i})$, $x_{n+i} = x_n + ih$, α_i, β_i are constants, α_0, β_0 can not be 0 at the same time, then (5.1) is called a k-step method.

If β_k=0, then (5.1) is explicit, y_{n+k} can be computed directly; If $\beta_k \neq 0$, then (5.1) is implicit, y_{n+k} can be computed by the iterative method similar to (2.7).

8.4 Multistep Methods

- General Form of the Multistep Methods

The coefficients α_i, β_i can be determined according to the local truncation error and order of the method, which are defined as follows.

Definition 7 Assume that $y(x)$ is the accurate solution of the inital-value problem (1.1) and (1.2), then the **local truncation error** of the linear multistep method (5.1) is

$$T_{n+k} = L\left[y(x_n); h\right] = y(x_{n+k}) - \sum_{i=0}^{k-1} \alpha_i y\left(x_{n+i}\right) - h\sum_{i=0}^{k} \beta_i \, y'\left(x_{n+i}\right). \qquad (5.2)$$

If $T_{n+k} = O(h^{p+1})$, then we say that (5.1) is of order p.

Give the Taylor-series of T_{n+k} at x_n, because

$$y(x_{n+i}) = y(x_n + ih) = y(x_n) + ihy'(x_n) + \frac{(ih)^2}{2!} y''(x_n) + \frac{(ih)^3}{3!} y'''(x_n) + \cdots$$

$$y'\left(x_{n+i}\right) = y'(x_n + ih) = y'(x_n) + ihy''(x_n) + \frac{(ih)^2}{2!} y'''(x_n) + \cdots \qquad (i = 0,1,\cdots,k)$$

Substitute them in (5.2), then

$$T_{n+k} = c_0 y(x_n) + c_1 hy'(x_n) + c_2 h^2 y''(x_n) + \cdots + c_p h^p y^{(p)}(x_n) + \cdots, \qquad (5.3)$$

8.4 Multistep Methods

- General Form of the Multistep Methods

where

$$\begin{cases} c_0 = 1-(\alpha_0+\cdots+\alpha_{k-1}) \\ c_1 = k-[\alpha_1+2\alpha_2+\cdots+(k-1)\alpha_{k-1}]-(\beta_0+\beta_1+\cdots+\beta_k) \\ c_q = \dfrac{1}{q!}\left[k^q-\left(\alpha_1+2^q\alpha_2+\cdots+(k-1)^q\alpha_{k-1}\right)\right]-\dfrac{1}{(q-1)!}\left[\beta_1+2^{q-1}\beta_2+\cdots+k^{q-1}\beta_k\right], q=2,3,\cdots \end{cases} \tag{5.4}$$

If the coefficients α_i, β_i are chosen to satisfy

$$c_0 = c_1 = \cdots = c_p = 0, \quad c_{p+1} \neq 0,$$

by **Definition 7**, the constructed multistep method is of order p, and

$$T_{n+k} = c_{p+1}h^{p+1}y^{(p+1)}(x_n) + O(h^{p+2}). \tag{5.5}$$

The first term is called the **dominant term of the local truncation error**, c_{p+1} is called the **error constant**.

8.4 Multistep Methods

- General Form of the Multistep Methods

When $k=1$, if $\beta_1 = 0$, then let $c_0 = c_1 = 0$ to determine α_0, β_0:

$$\begin{cases} \alpha_0 = 1 \\ \beta_0 + \beta_1 = 1 \end{cases} \Rightarrow \alpha_0 = 1, \beta_0 = 1.$$

Then (5.1) becomes

$$y_{n+1} = y_n + hf_n,$$

which is the Euler method. It can be obtained that $c_2 = 1/2 \neq 0,$ thus the Euler method is of order 1, and the local truncation error is

$$T_{n+1} = \frac{h^2}{2} y''(x_n) + O(h^3).$$

When $k=1$, if $\beta_1 \neq 0$, then let $c_0 = c_1 = c_2 = 0$ to determine $\alpha_0, \beta_0, \beta_1$:

$$\alpha_0 = 1, \beta_0 = \beta_1 = 1/2.$$

Then (5.1) becomes

$$y_{n+1} = y_n + \frac{h}{2}(f_n + f_{n+1}),$$

which is the trapezoidal method. It can be obtained that $c_3 = -1/12 \neq 0,$ thus the trapezoidal method is of order 2, and the local truncation error is

$$T_{n+1} = -\frac{h^3}{12} y'''(x_n) + O(h^4).$$

When $k \geq 2$, the coefficients α_i, β_i can be determined according to (5.4),and the local truncation error can be given by (5.5).

8.4 Multistep Methods

- Adams Methods

For a special case of (5.1), the following k-step method

$$y_{n+k} = y_{n+k-1} + h\sum_{i=0}^{k} \beta_i f_{n+i} \qquad (5.7)$$

is called the **Adams methods**.

If β_k =0, (5.7) is explicit and is called **Adams - Bashforth formula**;

If $\beta_k \neq 0$, (5.7) is implicit and is called **Adams - Monlton formula**.

We present several Adams explicit and implicit formulas in Table 8.5 and Table 8.6 when $k = 1, 2, 3, 4,$ where k is the number of steps, p is the order, c_{p+1} is the error constant.

For example, when $k = 3$, the Adams explicit formula is

$$y_{n+3} = y_{n+2} + \frac{h}{12}\left(23 f_{n+2} - 16 f_{n+1} + 5 f_n\right), \qquad (5.8)$$

the local truncation error is

$$T_{n+3} = \frac{3}{8} h^4 y^{(4)}(x_n) + O(h^5),$$

the order is 3.

8.4 Multistep Methods

- Adams Methods

Table 8.5 Adams explicit formulas

k	p	*Formulas*	c_{p+1}
1	1	$y_{n+1} = y_n + hf_n$	$\frac{1}{2}$
2	2	$y_{n+2} = y_{n+1} + \frac{h}{2}(3f_{n+1} - f_n)$	$\frac{5}{12}$
3	3	$y_{n+3} = y_{n+2} + \frac{h}{12}(23f_{n+2} - 16f_{n+1} + 5f_n)$	$\frac{3}{8}$
4	4	$y_{n+4} = y_{n+3} + \frac{h}{24}(55f_{n+3} - 59f_{n+2} + 37f_{n+1} - 9f_n)$	$\frac{251}{720}$

Table 8.6 Adams implicit formulas

k	p	*Formulas*	c_{p+1}
1	2	$y_{n+1} = y_n + \frac{h}{2}(f_{n+1} + f_n)$	$-\frac{1}{12}$
2	3	$y_{n+2} = y_{n+1} + \frac{h}{12}(5f_{n+2} + 8f_{n+1} - f_n)$	$-\frac{1}{24}$
3	4	$y_{n+3} = y_{n+2} + \frac{h}{24}(9f_{n+3} + 19f_{n+2} - 5f_{n+1} + f_n)$	$-\frac{19}{720}$
4	5	$y_{n+4} = y_{n+3} + \frac{h}{720}(251f_{n+4} + 646f_{n+3} - 264f_{n+2} + 106f_{n+1} - 19f_n)$	$-\frac{3}{160}$

8.4 Multistep Methods

- Adams Methods

On the other hand, when $k = 3$, the Adams implicit formula is

$$y_{n+3} = y_{n+2} + \frac{h}{24}(9f_{n+3} + 19f_{n+2} - 5f_{n+1} + f_n), \qquad (5.9)$$

the local truncation error is

$$T_{n+3} = -\frac{19}{720}h^5 y^{(5)}(x_n) + O(h^6),$$

and the order is 4.

Example 6 Use the fourth-order Adams explicit and implicit methods to solve the initial-value problem

$$\begin{cases} y' = -y + x + 1 \\ y(0) = 1 \end{cases}$$

with $h = 0.1$.

Solution : For this problem, $f_n = -y_n + x_n + 1$, $x_n = nh = 0.1n$. The fourth-order Adams explicit formula is

$$y_{n+4} = y_{n+3} + \frac{h}{24}(55f_{n+3} - 59f_{n+2} + 37f_{n+1} - 9f_n)$$

$$= \frac{1}{24}(18.5y_{n+3} + 5.9y_{n+2} - 3.7y_{n+1} + 0.9y_n + 0.24n + 3.24).$$

8.4 Multistep Methods

- Adams Methods

The fourth-order Adams implicit formula is

$$y_{n+3} = y_{n+2} + \frac{h}{24}(9f_{n+3} + 19f_{n+2} - 5f_{n+1} + f_n)$$

$$= \frac{1}{24}(-0.9y_{n+3} + 22.1y_{n+2} + 0.5y_{n+1} - 0.1y_n + 0.24n + 3).$$

Instead of using the iterative method, y_{n+3} can be solved directly as

$$y_{n+3} = \frac{1}{24.9}(22.1y_{n+2} + 0.5y_{n+1} - 0.1y_n + 0.24n + 3)$$

The computed results are shown in Table 8.7. The initial values y_0, y_1, y_2, y_3 in the explicit method and y_0, y_1, y_2 in the implicit method are given by the true solution $y(x) = e^{-x} + x$.
For a general equation, the initial values can be approximated with the fourth-order R-K method.

Comparing the results of the explicit and implicit methods , it can be seen clearly that with the same order, the error of the Admas implicit method is less than that of the Admas explicit method. This can be explained from the error constants c_{p+1} of the two methods, which are $251/720$ and $-19/720$,respectively.

8.4 Multistep Methods

- Adams Methods

Table 8.7 comparison of computation results

x_n	true solution	Adams explicit method		Adams implicit method	
	$y(x_n)$	y_n	$\|y(x_n)-y_n\|$	y_n	$\|y(x_n)-y_n\|$
0.3	1.040 818 22			1.040 818 01	$2.1*10^{-7}$
0.4	1.070 320 05	1.070 322 92	$2.87*10^{-6}$	1.070 319 66	$3.9*10^{-7}$
0.5	1.106 530 66	1.106 535 48	$4.82*10^{-6}$	1.106 530 14	$5.2*10^{-7}$
0.6	1.148 811 64	1.148 818 41	$6.77*10^{-6}$	1.148 811 01	$6.3*10^{-7}$
0.7	1.196 585 30	1.196 593 40	$8.10*10^{-6}$	1.196 584 59	$7.1*10^{-7}$
0.8	1.249 328 96	1.249 338 16	$9.20*10^{-6}$	1.249 328 19	$7.7*10^{-7}$
0.9	1.306 569 66	1.306 579 62	$9.96*10^{-6}$	1.306 568 84	$8.2*10^{-7}$
1.0	1.367 879 44	1.367 889 96	$1.05*10^{-5}$	1.367 878 59	$8.5*10^{-7}$

Exercises

*Ex*1. Use the Euler method to solve the initial-value problem

$$y' = x^2 + 100y^2, \quad y(0) = 0.$$

Given that $h = 0.1$, compute results up to $x = 0.3$ (precise to four decimals).

*Ex*2. Use the trapezoidal method method to solve the initial-value problem

$$y' + y = 0, \quad y(0) = 1.$$

Given that $h = 0.1$, compute results up to $x = 0.5$ (precise to five decimals).

*Ex*3. Use the fourth-order R-K method ($h = 0.2$) to solve the initial-value problem

$$\begin{cases} y' = x + y, \ (0 < x < 1) \\ y(0) = 1. \end{cases}$$

(precise to nine decimals)

*Ex*4. Use the second-order Adams explicit and implicit methods to solve the initial-value problem

$$y' = 1 - y, \quad y(0) = 0.$$

Given that $h = 0.2, y_0 = 0, y_1 = 0.181$, compute $y(1.0)$ and compare the results with the true solution $y(x) = 1 - e^{-x}$. (precise to nine decimals)