# Chapter1: Numerical Differentiation

## 1.1 Finite Difference Approximation of the Derivative

In finite difference approximations of the derivative, values of the function at different points in the neighborhood of the point $x=a$ are used for estimating the slope. It should be remembered that the function that is being differentiated is prescribed by a set of discrete points. Various finite difference approximation formulas exist. Three such formulas, where the derivative is calculated from the values of two points, are presented in this section.

### 1.1.1Forward, Backward, and Central Difference Formulas for the First Derivative

The forward, backward, and central finite difference formulas are the simplest finite difference approximations of the derivative. In these approximations, illustrated in Fig. 1-1, the derivative at point $x_i$ is calculated from the values of two points. The derivative is estimated as the value of the slope of the line that connects the two points.

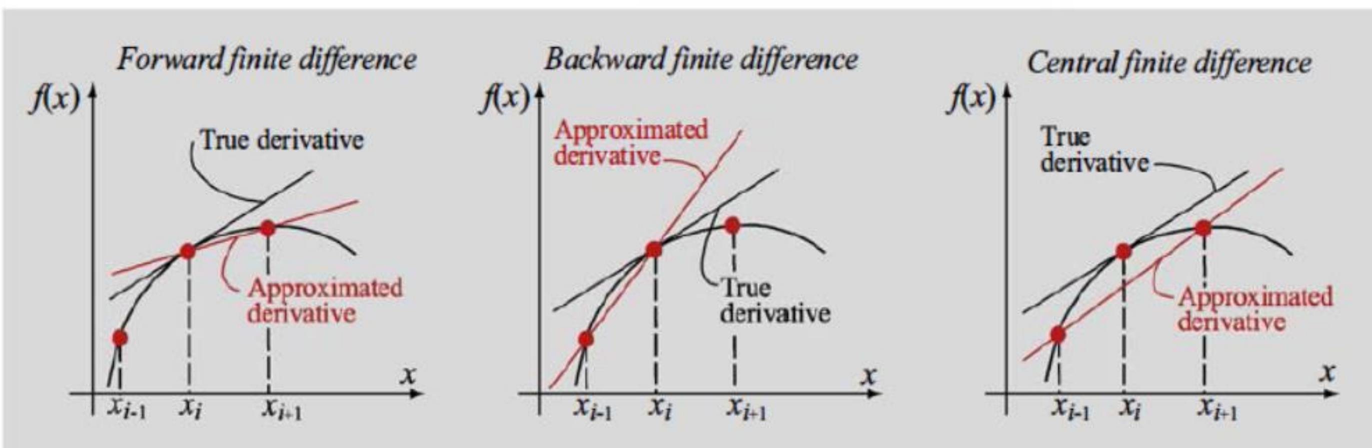


**Figure 1-1: Finite difference approximation of derivative.**

- **Forward difference** is the slope of the line that connects points $(x_i, f(x_i))$ and $(x_{i+1}, f(x_{i+1}))$:

$$\frac{df}{dx}\Big|_{x=x_i} = \frac{f(x_{i+1})-f(x_i)}{x_{i+1}-x_i} \qquad (1.1)$$

- **Backward difference** is the slope of the line that connects points $(x_{i-1}, f(x_{i-1}))$ and $(x_i, f(x_i))$:

$$\frac{df}{dx}\Big|_{x=x_i} = \frac{f(x_i)-f(x_{i-1})}{x_i-x_{i-1}} \qquad (1.2)$$

- **Central difference** is the slope of the line that connects points $(x_{i-1}, f(x_{i-1}))$ and $(x_{i+1}, f(x_{i+1}))$:

$$\frac{df}{dx}\Big|_{x=x_i} = \frac{f(x_{i+1})-f(x_{i-1})}{x_{i+1}-x_{i-1}} \qquad (1.3)$$

**Example 1-1: Comparing numerical and analytical differentiation.**

Consider the function $f(x) = x^3$ .Calculate its first derivative at point $x = 3$ numerically with the forward, backward, and central finite difference formulas and using:

(a) Points x = 2, x = 3, and x = 4.

(b) Points x = 2.75, x = 3, and x = 3.25.
Compare the results with the exact (analytical) derivative.

**SOLUTION**

***Analytical differentiation:*** The derivative of the function is $f'(x) = 3x^2$, and the value of the derivative at $x = 3$ is $f'(3) = 3(3^2) = 27$ .

***Numerical differentiation:***

(a) The points used for numerical differentiation are:

| | | | |
|---|---|---|---|
| *X* | 2 | 3 | 4 |
| *f(x)* | 8 | 27 | 64 |

Using Eqs. (1.1) through (1.3), the derivatives using the forward, backward, and central finite difference formulas are:

*Forward finite difference*:

$$\left.\frac{df}{dx}\right|_{x=3} = \frac{f(4)-f(3)}{4-3} = \frac{64-27}{1} = 37 \qquad error = \left|\frac{37-27}{27}\cdot 100\right| = 37.04\,\%$$

*Backward finite difference*:

$$\left.\frac{df}{dx}\right|_{x=3} = \frac{f(3)-f(2)}{3-2} = \frac{27-8}{1} = 19 \qquad error = \left|\frac{19-27}{27}\cdot 100\right| = 29.63\,\%$$

*Central finite difference*:

$$\left.\frac{df}{dx}\right|_{x=3} = \frac{f(4)-f(2)}{4-2} = \frac{64-8}{2} = 28 \qquad error = \left|\frac{28-27}{27}\cdot 100\right| = 3.704\,\%$$

(b)The points used for numerical differentiation are:

| | | | |
|---|---|---|---|
| *X* | 2.75 | 3 | 3.25 |
| *f(x)* | $2.75^3$ | $3^3$ | $3.25^3$ |

Using Eqs. (1.1) through (1.3), the derivatives using the forward, backward, and central finite difference formulas are:

*Forward finite difference*:

$$\left.\frac{df}{dx}\right|_{x=3} = \frac{f(3.25)-f(3)}{3.25-3} = \frac{3.25^3-27}{0.25} = 29.3125 \qquad error = \left|\frac{29.3125-27}{27}\right|\cdot 100 = 8.565\ \%$$

*Backward finite difference*:

$$\left.\frac{df}{dx}\right|_{x=3} = \frac{f(3)-f(2.75)}{3-2.75} = \frac{27-2.75^3}{0.25} = 24.8125 \qquad error = \left|\frac{24.8125-27}{27}\right|\cdot 100 = 8.102\ \%$$

*Central finite difference*:

$$\left.\frac{df}{dx}\right|_{x=3} = \frac{f(3.25)-f(2.75)}{3.25-2.75} = \frac{3.25^3-2.75^3}{0.5} = 27.0625 \qquad error = \left|\frac{27.0625-27}{27}\right|\cdot 100 = 0.2315\ \%$$

The results show that the central finite difference formula gives a more accurate approximation. This will be discussed further in the next section. In addition, smaller separation between the points gives a significantly more accurate approximation.

# 1.2 Finite Difference Formulas Using Taylor Series Expansion

The forward, backward, and central difference formulas, as well as many other finite difference formulas for approximating derivatives, can be derived by using Taylor series expansion. The formulas give an estimate of the derivative at a point from the values of points in its neighborhood. The number of points used in the calculation varies with the formula, and the points can be ahead, behind, or on both sides of the point at which the derivative is calculated. One advantage of using Taylor series expansion for deriving the formulas is that it also provides an estimate for the truncation error in the approximation.

## 1.2.1 Finite Difference Formulas of First Derivative

Several formulas for approximating the first derivative at point $x_i$ based on the values of the points near $x_i$are derived by using the Taylor series expansion. All the formulas derived in this section are for the case where the points are equally spaced.

***Two-point forward difference formula for first derivative***

The value of a function at point $x_{i+1}$can be approximated by a Taylor series in terms of the value of the function and its derivatives at point$x_i$:

$$\boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{i+1}}) = \boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{i}}) + \boldsymbol{f}'(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(x_i)}{3!}h^3 + \frac{f^{(4)}(x_i)}{4!}h^4 + \cdots \qquad (1.4)$$

where $h$=$x_{i+1} - x_i$; is the spacing between the points. By using two terms Taylor series expansion with a remainder can be rewritten as:

$$\boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{i+1}}) = \boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{i}}) + \boldsymbol{f}'(x_i)h + \frac{f''(\xi)}{2!}h^2 \qquad (1.5)$$

where $\xi$ is a value of *x* between $x_i$ and $x_{i+1}$. Solving Eq. (1.5) for $\boldsymbol{f}'(x_i)$ yields:

$$\boldsymbol{f}'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{f''(\xi)}{2!}h \qquad (1.6)$$

An approximate value of the derivative $\boldsymbol{f}'(x_i)$ can now be calculated if the second term on the right-hand side of Eq. (1.6) is ignored. Ignoring this second term introduces a truncation (discretization) error. Since this term is proportional to h, the truncation error is said to be on the order of h (written as O(h) ):

$$\boldsymbol{truncation\ error} = -\frac{f''(\xi)}{2!}h = O(h) \qquad (1.7)$$

Using the notation of Eq. (1.7), the approximated value of the first derivative is:

$$\boldsymbol{f}'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - O(h) \qquad (1.8)$$

The approximation in Eq. (1.8) is the same as the forward difference formula in Eq. (1.1).

***Two-point backward difference formula for first derivative***

The backward difference formula can also be derived by application of Taylor series expansion. The value of the function at point $x_{i-1}$ is approximated by a Taylor series in terms of the value of the function and its derivatives at point $x_i$:

$$\boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{i-1}}) = \boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{i}}) - \boldsymbol{f}'(x_i)h + \frac{f''(x_i)}{2!}h^2 - \frac{f'''(x_i)}{3!}h^3 + \frac{f^{(4)}(x_i)}{4!}h^4 - \cdots \qquad (1.9)$$

where $h$=$x_i - x_{i-1}$; is the spacing between the points. By using two terms Taylor series expansion with a remainder can be rewritten as:

$$\boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{i-1}}) = \boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{i}}) - \boldsymbol{f}'(x_i)h + \frac{f''(\xi)}{2!}h^2 \qquad (1.10)$$

where $\xi$ is a value of $x$ between $x_i$ and $x_{i+1}$. Solving Eq. (1.10) for $\boldsymbol{f}'(x_i)$ yields:

$$\boldsymbol{f}'(x_i) = \frac{\boldsymbol{f}(\boldsymbol{x_i}) - \boldsymbol{f}(\boldsymbol{x_{i-1}})}{h} + \frac{\boldsymbol{f}''(\xi)}{2!}h \qquad (1.11)$$

An approximate value of the derivative $\boldsymbol{f}'(x_i)$ can now be calculated if the second term on the right-hand side of Eq. (1.11) is ignored. This yileds:

$$\boldsymbol{f}'(x_i) = \frac{\boldsymbol{f}(\boldsymbol{x_i}) - \boldsymbol{f}(\boldsymbol{x_{i-1}})}{h} + O(h) \qquad (1.12)$$

The approximation in Eq. (1.12) is the same as the forward difference formula in Eq. (1.2).

**<u>Two-point central difference formula for first derivative</u>**

The central difference formula can be derived by using three terms in the Taylor series expansion and a remainder. The value of the function at point $x_{i+1}$ in terms of the value of the function and its derivatives at point $x_i$ is given by:

$$\boldsymbol{f}(\boldsymbol{x_{i+1}}) = \boldsymbol{f}(\boldsymbol{x_i}) + \boldsymbol{f}'(x_i)h + \frac{\boldsymbol{f}''(x_i)}{2!}h^2 + \frac{\boldsymbol{f}'''(\zeta_1)}{3!}h^3 \qquad (1.13)$$

where $\zeta_1$ is a value of $x$ between $x_i$ and $x_{i+1}$·The value of the function at point $x_{i-1}$ in terms of the value of the function and its derivatives at point $x_i$ is given by:

$$\boldsymbol{f}(\boldsymbol{x_{i-1}}) = \boldsymbol{f}(\boldsymbol{x_i}) - \boldsymbol{f}'(x_i)h + \frac{\boldsymbol{f}''(x_i)}{2!}h^2 - \frac{\boldsymbol{f}'''(\zeta_2)}{3!}h^3 \qquad (1.14)$$

where $\zeta_2$ is a value of $x$ between $x_{i-1}$ and $x_i$. In the last two equations, the spacing of the intervals is taken to be equal so that $h = x_{i+1}\text{-}x_i = x_i\text{-}x_{i-1}$. Subtracting Eq. (1.14) from Eq. (1.13) gives:

$$\boldsymbol{f}(\boldsymbol{x_{i+1}}) - \boldsymbol{f}(\boldsymbol{x_{i-1}}) = \boldsymbol{2f}'(x_i)h + \frac{\boldsymbol{f}'''(\zeta_1)}{3!}h^3 + \frac{\boldsymbol{f}'''(\zeta_2)}{3!}h^3 \qquad (1.15)$$

An estimate for the first derivative is obtained by solving Eq. (1.15) for $\boldsymbol{f}'(x_i)$ while neglecting the remainder terms, which introduces a truncation error, which is of the order of $h^2$ :

$$\boldsymbol{f}'(x_i) = \frac{\boldsymbol{f}(\boldsymbol{x_{i+1}}) - \boldsymbol{f}(\boldsymbol{x_{i-1}})}{2h} + O(h^2) \qquad (1.16)$$

The approximation in Eq. (1.16) is the same as the central difference formula Eq. (1.3) for equally spaced intervals.

## 1.2.2 Finite Difference Formulas for the Second Derivative

The same approach used in Section 1.2.1 to develop finite difference formulas for the first derivative can be used to develop expressions for higher-order derivatives. In this section, expressions based on central differences, one-sided forward differences, and one-sided backward differences are presented for approximating the second derivative at a point $x_i$.

***<u>Three-point central difference formula for the second derivative</u>***

Central difference formulas for the second derivative can be developed using any number of points on either side of the point $x_i$, where the second derivative is to be evaluated. The formulas are derived by writing the Taylor series expansion with a remainder at points on either side of $x_i$ in terms of the value of the function and its derivatives at point $x_i$. Then, the equations are combined in such a way that the terms containing the first derivatives are eliminated. For example, for points $x_{i+1}$, and $x_{i-1}$the four-term Taylor series expansion with a remainder is:

$$\boldsymbol{f(x_{i+1}) = f(x_i) + f'}(x_i)h + \frac{f''(x_i)}{2!}h^2 + \frac{f'''(x_i)}{3!}h^3 + \frac{f^{(4)}(\zeta_1)}{4!}h^4 \qquad (1.17)$$

$$\boldsymbol{f(x_{i-1}) = f(x_i) - f'}(x_i)h + \frac{f''(x_i)}{2!}h^2 - \frac{f'''(x_i)}{3!}h^3 + \frac{f^{(4)}(\zeta_2)}{4!}h^4 \qquad (1.18)$$

where $\zeta_1$ is a value of $x$ between $x_i$ and $x_{i+1}$. and $\zeta_2$ is a value of $x$ between $x_{i-1}$ and $x_i$. Adding Eq. (1.17) and Eq. (1.18) gives:

$$\boldsymbol{f(x_{i+1}) + f(x_{i-1}) = 2f(x_i) + 2}\frac{f''(x_i)}{2!}h^2 + + \frac{f^{(4)}(\zeta_1)}{4!}h^4 + \frac{f^{(4)}(\zeta_2)}{4!}h^4 \qquad (1.19)$$

An estimate for the second derivative can be obtained by solving Eq.(1.19) for $\boldsymbol{f''}(x_i)$ while neglecting the remainder terms. This introduces a truncation error of the order of $h^2$.

$$\boldsymbol{f''}(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2} + O(h^2) \qquad (1.20)$$

**Example 1-2: Comparing numerical and analytical differentiation.**

Consider the function $f(x) = \frac{2^x}{x}$. Calculate the second derivative at $x$ = 2 numerically with the three-point central difference formula using:

(a) Points $x$ = 1.8 , $x$ = 2 , and $x$ = 2.2 .

(b) Points $x$=l.9, $x$=2, and $x$=2.1.

Compare the results with the exact (analytical) derivative.

**SOLUTION**

**Analytical differentiation**: The second derivative of the function $f(x) = \frac{2^x}{x}$ is:

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

$$f''(x) = \frac{2^x(2x) - x^2(\ln 2)2^x}{x^4} - \ln 2\left(\frac{2^x}{x^2} - \ln 2\frac{2^x}{x}\right)$$

$$\Rightarrow f''(x) = (\ln 2)^2\frac{2^x}{x} - 2(\ln 2)\frac{2^x}{x^2} + 2\frac{2^x}{x^3}$$

and the value of the derivative at $x$ = 2 is $f''$ (2) = 0.574617 .

**Numerical differentiation**

(a) The numerical differentiation is done by substituting the values of the points $x$ = 1.8, $x$ = 2, and $x$ = 2.2 in Eq. (1.20). The operations are done with MATLAB, in the Command Window:

```
>> xa = [1.8 2 2.2];
>> ya = 2.^xa./xa;
>> df = (ya(1) - 2*ya(2) + ya(3))/0.2^2
df =
   0.57748177389232
```

(b) The numerical differentiation is done by substituting the values of the points $x$ = 1.9, $x$ = 2, and $x$ = 2.1 in Eq. (1.20). The operations are done with MATLAB, in the Command Window:

```
>> xb = [1.9 2 2.1];
>> yb = 2.^xb./xb;
>> dfb = (yb(1) - 2*yb(2) + yb(3))/0.1^2
dfb =
   0.57532441566441
```

Error in part (*a*): $error = \frac{0.577482 - 0.574617}{0.574617} \cdot 100 = 0.4986\ \%$

Error in part (*b*): $error = \frac{0.575324 - 0.574617}{0.574617} \cdot 100 = 0.1230\ \%$

The results show that the three-point central difference formula gives a quite accurate approximation for the value of the second derivative.

## 1.3 Summary of Finite Difference Formulas for Numerical Differentiation

Table 3-1 lists difference formulas, of various accuracy, that can be used for numerical evaluation of first, second, third, and fourth derivatives. The formulas can be used when the function that is being differentiated is specified as a set of discrete points with the independent variable **equally spaced**.

**Table 1-1: Finite difference formulas.**

| First Derivative | | |
|---|---|---|
| **Method** | **Formula** | **Truncation Error** |
| **Two-point forward difference** | $f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h}$ | $O(h)$ |
| **Three-point forward difference** | $f'(x_i) = \frac{-3f(x_i) + 4f(x_{i+1}) - f(x_{i+2})}{2h}$ | $O(h^2)$ |
| **Two-point backward difference** | $f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h}$ | $O(h)$ |
| **Three-point backward difference** | $f'(x_i) = \frac{f(x_{i-2}) - 4f(x_{i-1}) + 3f(x_i)}{2h}$ | $O(h^2)$ |
| **Two-point central difference** | $f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h}$ | $O(h^2)$ |
| **Four-point central difference** | $f'(x_i) = \frac{f(x_{i-2}) - 8f(x_{i-1}) + 8f(x_{i+1}) - f(x_{i+2})}{12h}$ | $O(h^4)$ |
| **Second Derivative** | | |
| **Method** | **Formula** | **Truncation Error** |
| **Three-point forward difference** | $f''(x_i) = \frac{f(x_i) - 2f(x_{i+1}) + f(x_{i+2})}{h^2}$ | $o(h)$ |
| **Four-point forward difference** | $f''(x_i) = \frac{2f(x_i) - 5f(x_{i+1}) + 4f(x_{i+2}) - f(x_{i+3})}{h^2}$ | $o(h^2)$ |

| | | |
|---|---|---|
| **Three-point backward difference** | $f''(x_i)=\dfrac{f(x_{i-2})-2f(x_{i-1})+f(x_i)}{h^2}$ | $o(h)$ |
| **Four-point backward difference** | $f''(x_i)=\dfrac{-f(x_{i-3})+4f(x_{i-2})-5f(x_{i-1})+2f(x_i)}{h^2}$ | $o(h^2)$ |
| **Three-point central difference** | $f''(x_i)=\dfrac{f(x_{i-1})-2f(x_i)+f(x_{i+1})}{h^2}$ | $o(h^2)$ |
| **Five-point central difference** | $f''(x_i)=\dfrac{-f(x_{i-2})+16f(x_{i-1})-30f(x_i)+16f(x_{i+1})-f(x_{i+2})}{12h^2}$ | $o(h^4)$ |

# 1.4 DIFFERENTIATION FORMULAS USING LAGRANGE POLYNOMIALS

The differentiation formulas can also be derived by using Lagrange polynomials. For the first derivative, the two-point central, three-point forward, and three-point backward difference formulas are obtained by considering three points $(x_i, y_i)$, $(x_{i+1}, y_{i+1})$ , and $(x_{i+2}, y_{i+2})$. The polynomial, in Lagrange form, that passes through the points is given by:

$$f(x)=y_i\frac{(x-x_{i+1})(x-x_{i+2})}{(x_i-x_{i+1})(x_i-x_{i+2})}+y_{i+1}\frac{(x-x_i)(x-x_{i+2})}{(x_{i+1}-x_i)(x_{i+1}-x_{i+2})}+y_{i+2}\frac{(x-x_i)(x-x_{i+1})}{(x_{i+2}-x_i)(x_{i+2}-x_{i+1})} \qquad (1.21)$$

Differentiating Eq.(1.21) gives:

$$f'(x)=y_i\frac{2x-x_{i+1}-x_{i+2}}{(x_i-x_{i+1})(x_i-x_{i+2})}+y_{i+1}\frac{2x-x_i-x_{i+2}}{(x_{i+1}-x_i)(x_{i+1}-x_{i+2})}+y_{i+2}\frac{2x-x_i-x_{i+1}}{(x_{i+2}-x_i)(x_{i+2}-x_{i+1})} \qquad (1.22)$$

The first derivative at either one of the three points is calculated by substituting the corresponding value of x ( $x_i$, $x_{i+1}$ or $x_{i+2}$) in Eq. (1.22). This gives the following three formulas for the first derivative at the three points.

$$f'(x_i)=y_i\frac{2x_i-x_{i+1}-x_{i+2}}{(x_i-x_{i+1})(x_i-x_{i+2})}+y_{i+1}\frac{2x_i-x_i-x_{i+2}}{(x_{i+1}-x_i)(x_{i+1}-x_{i+2})}+y_{i+2}\frac{2x_i-x_i-x_{i+1}}{(x_{i+2}-x_i)(x_{i+2}-x_{i+1})} \qquad (1.23)$$

When the points are equally spaced, Eq. (1.23) reduces to the **three point forward difference formula**:

$$\boldsymbol{f'(x_i)=\frac{-3f(x_i)+4f(x_{i+1})-f(x_{i+2})}{2h}}$$

$$f'(x_{i+1})=y_i\frac{2x_{i+1}-x_{i+1}-x_{i+2}}{(x_i-x_{i+1})(x_i-x_{i+2})}+y_{i+1}\frac{2x_{i+1}-x_i-x_{i+2}}{(x_{i+1}-x_i)(x_{i+1}-x_{i+2})}+y_{i+2}\frac{2x_{i+1}-x_i-x_{i+1}}{(x_{i+2}-x_i)(x_{i+2}-x_{i+1})} \qquad (1.24)$$

When the points are equally spaced, Eq. (1.24) reduces to the **two point central difference formula**:

$$\boldsymbol{f'(x_{i+1})=\frac{f(x_{i+2})-f(x_i)}{2h}}$$

**Which is:**

$$\boldsymbol{f'(x_i)=\frac{f(x_{i+1})-f(x_{i-1})}{2h}}$$

$$f'(x_{i+2})=y_i\frac{2x_{i+2}-x_{i+1}-x_{i+2}}{(x_i-x_{i+1})(x_i-x_{i+2})}+y_{i+1}\frac{2x_{i+2}-x_i-x_{i+2}}{(x_{i+1}-x_i)(x_{i+1}-x_{i+2})}+y_{i+2}\frac{2x_{i+2}-x_i-x_{i+1}}{(x_{i+2}-x_i)(x_{i+2}-x_{i+1})} \qquad (1.25)$$

When the points are equally spaced, Eq. (1.24) reduces to the **three point backward difference formula**:

$$\boldsymbol{f'(x_i)=\frac{f(x_{i-2})-4f(x_{i-1})+3f(x_i)}{2h}}$$

# (1.4) First Derivatives From Interpolating Polynomials:

We begin with a Newton-Gregory forward polynomial:

$$f(x_t) = f_0 + t\Delta f_0 + \frac{t(t-1)}{2!}\Delta^2 f_0 + \frac{t(t-1)(t-2)}{3!}\Delta^3 f_0 + \cdots + \frac{t(t-1)\ldots(t-n+1)}{n!}\Delta^n f_0 + \cdots \qquad (1.26)$$

Differentiating Eq.(1.26) , remembering that $f_0$ and all the Δ-terms are constants (after all, they are just the numbers from the difference table), we have:

$$f'(x_t) = \frac{d}{dx}[f(x_t)] = \frac{d}{dt}[f(x_t)]\frac{1}{h}$$

$$= \frac{1}{h}\left[\Delta f_0 + \frac{(2t-1)}{2!}\Delta^2 f_0 + \frac{3t^2-6t+2}{3!}\Delta^3 f_0 + \cdots\right] \qquad (1.27)$$

If we let $t=0$, giving us the derivative corresponding to $x_0$, we have:

$$f'(x_0) = \frac{1}{h}\left[\Delta f_0 - \frac{1}{2}\Delta^2 f_0 + \frac{1}{3}\Delta^3 f_0 - \frac{1}{4}\Delta^4 f_0 \ldots\right] \qquad (1.28)$$

# 1.5 Use of MATLAB Built-In Functions for Numerical Differentiation

In general, it is recommended that the techniques described in this chapter be used to develop script files that perform the desired differentiation. MATLAB does not have built-in functions that perform numerical differentiation of an arbitrary function or discrete data. There is, however, a built-in function called **diff**, which can be used to perform numerical differentiation, and another built-in function called **polyder**, which determines the derivative of polynomial.

## 1.5.1 The diff command

The built-in function **diff** calculates the derivative of the functions:

```
>> syms x
>> diff(x^3+2*x^2-1)
ans =
3*x^2 + 4*x
>> diff(x^3+2*x^2-1,2)
ans =
6*x + 4
>> diff(x^3+2*x^2-1,3)
ans =
6
```

### 1.5.2 The polyder command

The built-in function **polyder** can calculate the derivative of a polynomial (it can also calculate the derivative of a product and quotient of two polynomials).

```
>> p=[4 0 2 5]
p =
   4    0    2    5
>> polyder(p)
ans =
   12    0    2
```

## 1.6 PROBLEMS

1. Given the following data:

| $x$ | 1 | 1.2 | 1.3 | 1.4 | 1.5 |
|---|---|---|---|---|---|
| $f(x)$ | 0.6133 | 0.7882 | 0.9716 | 1.1814 | 1.4117 |

Find the first derivative $f'(x)$ at the point $x$ = 1.3.
(a) Use the three-point forward difference formula.
(b) Use the three-point backward difference formula.
(c) Use the two-point central difference formula.

2. The following data is given for the stopping distance of a car on a wet road versus the speed at which it begins braking.

| *v(mi/h)* | 12.5 | 25 | 37.5 | 50 | 62.5 | 75 |
|---|---|---|---|---|---|---|
| d(ft) | 20 | 59 | 118 | 197 | 299 | 420 |

Calculate the rate of change of the stopping distance at a speed of 62.5 mph using:
(i) the two-point backward difference formula, and (ii) the three-point backward difference formula.
   a. Use Lagrange interpolation polynomials to find the finite difference formula for the second derivative at the point $x = x_i$ using the unequally spaced points $x = x_{i+1}$, and $x = x_{i+2}$ What is the second derivative at $x = x_{i+1}$ and at $x = x_{i+2}$?
3. Find the first derivative from backward polynomial approximated to the forth difference.
4. Find the second derivative from forward polynomial to the forth difference.
5. Use the data below to estimate the derivative of y at x=1.7:

| x | 1.3 | 1.5 | 1.7 | 1.9 | 2.1 | 2.3 | 2.5 |
|---|---|---|---|---|---|---|---|
| f(x) | 3.669 | 4.482 | 5.474 | 6.686 | 8.166 | 9.974 | 12.182 |

# Chapter2: Numerical Integration
## 2.1 Introduction to Quadrature:

We now approach the subject of numerical integration. The goal is to approximate the definite integral of f(x) over the interval [a,b] by evaluating f(x) at a finite number of sample points.

***Definition(2.1):*** Suppose that $a=x_0<x_1<\ldots<x_M=b$. A formula of the form:

$$Q[f]=\sum_{k=0}^{M} w_k f(x_k)=w_0 f(x_0)+w_1 f(x_1)+\cdots+w_M f(x_M) \qquad (2.1)$$

With the property that:

$$\int_a^b f(x)dx=Q[f]+E[f] \qquad (2.2)$$

is called a numerical integration or **quadrature** formula. The term *E[f]* is called the **truncation error** for integration. The values $\{x_k\}_{k=0}^{M}$ are called the **quadrature nodes** and $\{w_k\}_{k=0}^{M}$ are called **weights**.

***Definition (2.2):*** The **degree of precision** of a quadrature formula is the positive integer n such that $E[P_i]=0$ for all polynomials $P_i(x)$ of degree $i\leq n$, but for which $E[P_{n+1}]\neq 0$ for some polynomial $P_{n+1}(x)$ of degree n+1.

***Theorem(2.1):*** (closed Newton-cotes Quadrature formula)

Assume that $x_k=x_0+kh$ are equally spaced nodes and $f_k=f(x_k)$. The first four closed Newton-Cotes quadrature formulas are

$$\int_{x_0}^{x_1} f(x)dx \approx \frac{h}{2}(f_0+f_1) \qquad (2.3)$$ **(the trapezoidal rule)**

$$\int_{x_0}^{x_2} f(x)dx \approx \frac{h}{3}(f_0+4f_1+f_2) \qquad (2.4)$$ **(Simpson rule)**

$$\int_{x_0}^{x_3} f(x)dx \approx \frac{3h}{8}(f_0+3f_1+3f_2+f_3) \qquad (2.5)$$ **(Simpson's $\frac{3}{8}$ rule)**

$$\int_{x_0}^{x_4} f(x)dx \approx \frac{2h}{45}(7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4) \quad (2.6)$$ **(Boole's rule)**

***<u>Corollary(2.1)</u>:*** (Newton-Cotes precision)

Assume that *f(x)* is sufficiently differentiable; then *E[f]* for Newton-Cotes quadrature involves an approximate higher derivative. The trapezoidal rule has degree of precision $n$=1. If $f \in C^2[a,b]$, then:

$$\int_{x_0}^{x_1} f(x)dx = \frac{h}{2}(f_0 + f_1) - \frac{h^3}{12} f^{(2)}(c) \qquad (2.7)$$

Simpson's rule has degree of precision $n=3$. If $f \in C^4[a,b]$, then:

$$\int_{x_0}^{x_2} f(x)dx = \frac{h}{3}(f_0 + 4f_1 + f_2) - \frac{h^5}{90} f^{(4)}(c) \qquad (2.8)$$

Simpson's $\frac{3}{8}$ rule has degree of precision $n$=3. If $f \in C^4[a,b]$, then:

$$\int_{x_0}^{x_3} f(x)dx = \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3) - \frac{3h^5}{80} f^{(4)}(c) \qquad (2.9)$$

Boole's rule has degree of precision $n$=5. If $f \in C^6[a,b]$, then:

$$\int_{x_0}^{x_4} f(x)dx = \frac{2h}{45}(7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4) - \frac{8h^7}{945} f^{(6)}(c) \qquad (2.10)$$

***<u>Proof of Theorem(2.1):</u>*** Start with the Lagrange polynomial $P_M(x)$ based on $x_0, x_1, \ldots, x_M$ that can be used to approximate *f(x)*:

$$f(x) \approx P_M(x) = \sum_{k=0}^{M} f(x_k) \prod_{\substack{j=0 \\ j\neq k}}^{M} \frac{(x-x_j)}{(x_k-x_j)} \qquad (2.11)$$

An approximate for the integral is obtained by replacing the integrand *f(x)* with the polynomial $P_M(x)$. This is the general method for obtaining a Newton-Cotes integration formula:

$$\int_{x_0}^{x_M} f(x)dx \approx \int_{x_0}^{x_M} P_M(x)dx = \int_{x_0}^{x_M}\left(\sum_{k=0}^{M} f_k \prod_{\substack{j=0\\ j\neq k}}^{M} \frac{(x-x_j)}{(x_k-x_j)}\right) \qquad (2.12)$$

The details for the general proof of the theorem are tedious. We shall give a Simpson's rule, which is the case M=2. This case involves the approximation polynomial

$$P_2(x) = f_0\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} + f_1\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} + f_2\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} \qquad (2.13)$$

Since $f_0$, $f_1$ and $f_2$ are constant with respect to integration, the relations in (2.12) lead to:

$$\int_{x_0}^{x_2} f(x)dx \approx \int_{x_0}^{x_2} f_0\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}dx + \int_{x_0}^{x_2} f_1\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}dx + \int_{x_0}^{x_2} f_2\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}dx \qquad (2.14)$$

We introduce the change of variable $x=x_0+th$ with $dx=hdt$ to assist with the evaluation of the integrals in (2.14). The new limits of integration are from $t=0$ to $t=2$. The equal spacing of the nodes $x_k=x_0+kh$ leads to $x_k-x_j=(k-j)=h$ and $x-x_k=(t-k)h$, which are used to simplify (2.14), and get:

$$\int_{x_0}^{x_2} f(x)dx \approx f_0\int_0^2 \frac{h(t-1)h(t-2)}{(-h)(-2h)}hdt + f_1\int_0^2 \frac{h(t-0)h(t-2)}{(h)(-h)}hdt + f_2\int_0^2 \frac{h(t-0)h(t-1)}{(2h)(h)}hdt$$

$$= f_0\frac{h}{2}\int_0^2 (t^2-3t+2)dt + f_1 h\int_0^2 (t^2-2t)dt + f_2\frac{h}{2}\int_0^2 (t^2-t)dt$$

$$= f_0\frac{h}{2}\left(\frac{t^3}{3}-\frac{3t^2}{2}+2t\right)\Big|_{t=0}^{t=2} - f_1 h\left(\frac{t^3}{3}-\frac{2t^2}{2}\right)\Big|_{t=0}^{t=2} + f_2\frac{h}{2}\left(\frac{t^3}{3}-\frac{t^2}{2}\right)\Big|_{t=0}^{t=2}$$

$$= f_0 \frac{h}{2}\left(\frac{2}{3}\right) - f_1 h\left(\frac{-4}{3}\right) + f_2 \frac{h}{2}\left(\frac{2}{3}\right)$$

$$= \frac{h}{3}(f_0 + 4f_1 + f_2)$$

and the proof is complete.

***Example(2.1):*** Consider the function $f(x)=1+e^{-x}\sin(4x)$, the equally spaced quadrature nodes $x_0=0$, $x_1=0.5$, $x_2=1$, $x_3=1.5$, $x_4=2$ and the corresponding function values $f_0=1$, $f_1=1.55152$, $f_2=0.72159$, $f_3=0.93765$ and $f_4=1.13390$. Apply the various quadrature formulas (2.3) through (2.6).

The step size is h=0.5, and the computations are:

$$\int_0^{0.5} f(x)dx \approx \frac{0.5}{2}(1 + 1.55152) = 0.63788$$

$$\int_0^{1} f(x)dx \approx \frac{0.5}{3}(1 + 4(1.55152) + 0.72159) = 1.32128$$

$$\int_0^{1.5} f(x)dx \approx \frac{3(0.5)}{8}(1 + 3(1.55152) + 3(0.72159) + 0.93765) = 1.64193$$

$$\int_0^{2} f(x)dx \approx \frac{2(0.5)}{45}\big(7(1) + 32(1.55152) + 12(0.72159) + 32(0.93765) + 7(1.1339)\big)$$

$$= 2.29444$$

***Examples (2.2):*** Consider the integration of the function $f(x)=1+e^{-x}\sin(4x)$ over the fixed interval [a,b]=[0,1]. Apply the various formulas (2.3) through (2.6).

For the trapezoidal rule, h=1 and

$$\int_0^1 f(x)dx \approx \frac{1}{2}\big(f(0)+f(1)\big) = \frac{1}{2}(1+0.72159) = 0.86079$$

For Simpson's rule, h=1/2, and we get:

$$\int_0^1 f(x)dx \approx \frac{1/2}{3}(f(0)+4f\left(\frac{1}{2}\right)+f(1) = \frac{1}{6}(1+4(1.55152)+0.72159) = 1.32128$$

For Simpson's $\frac{3}{8}$ rule, h=1/3, and we obtain:

$$\int_0^1 f(x)dx \approx \frac{3\left(\frac{1}{3}\right)}{8}(f(0)+3f\left(\frac{1}{3}\right)+3f\left(\frac{2}{3}\right)+f(1))$$

$$= \frac{1}{8}(1+3(1.69642)+3(1.23447)+0.72159) = 1.31440$$

For Boole's rule, h=1/4, and the result is:

$$\int_0^1 f(x)dx \approx \frac{2\left(\frac{1}{4}\right)}{45}(7f(0)+32f\left(\frac{1}{4}\right)+12f\left(\frac{1}{2}\right)+32f\left(\frac{3}{4}\right)+7f(1))$$

$$= \frac{1}{90}\big(7(1)+32(1.65534)+12(1.55152)+32(1.06666)+7(0.72159)\big)$$

$$=1.30859$$

The true value of the definite integral is:

$$\int_0^1 f(x)dx = 1.308\,250\,604$$

To make a fair comparison of quadrature methods, we must use the same number of function evaluations in each method. Our final example is concerned with comparing integration over a fixed interval [a,b] using exactly five function evaluation $f_k=f(x_k)$, for k=0,1,…,4 for each method. When the trapezoidal rule is applied on the four subintervals $[x_0,x_1]$, $[x_1,x_2]$, $[x_2,x_3]$ and $[x_3,x_4]$, it is called a **composite trapezoidal rule**:

$$\int_{x_0}^{x_4} f(x)dx = \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \int_{x_2}^{x_3} f(x)dx + \int_{x_3}^{x_4} f(x)dx$$

$$\approx \frac{h}{2}(f_0 + f_1) + \frac{h}{2}(f_1 + f_2) + \frac{h}{2}(f_2 + f_3) + \frac{h}{2}(f_3 + f_4)$$

$$= \frac{h}{2}(f_0 + 2f_1 + 2f_2 + 2f_3 + f_4) \qquad (2.15)$$

Simpson's rule can also be used in this manner. When Simpson's rule is applied on the two subintervals $[x_0,x_2]$ and $[x_2,x_4]$, it is called a **composite Simpson's rule**:

$$\int_{x_0}^{x_4} f(x)dx = \int_{x_0}^{x_2} f(x)dx + \int_{x_2}^{x_4} f(x)dx$$

$$\approx \frac{h}{3}(f_0 + 4f_1 + f_2) + \frac{h}{3}(f_2 + 4f_3 + f_4)$$

$$= \frac{h}{3}(f_0 + 4f_1 + 2f_2 + 4f_3 + f_4) \qquad (2.16)$$

***Example(2.3):*** Consider the integration of the function $f(x)=1+e^{-x}\sin(4x)$ over [a,b]=[0,1]. Use exactly five function evaluations and compare the results from the composite trapezoidal rule and composite Simpson's rule.

The uniform step size is h=1/4. The composite trapezoidal rule (2.15) produces:

$$\int_0^1 f(x)dx \approx \frac{1/4}{2}\left(f(0) + 2f\left(\frac{1}{4}\right) + 2f\left(\frac{1}{2}\right) + 2f\left(\frac{3}{4}\right) + f(1)\right)$$

$$= \frac{1}{8}(1 + 2(1.65534) + 2(1.55152) + 2(1.06666) + 0.72159)$$

$$=1.28358$$

Using the composite Simpson's rule (2.16), we get:

$$\int_0^1 f(x)dx \approx \frac{1/4}{3}\left(f(0) + 4f\left(\frac{1}{4}\right) + 2f\left(\frac{1}{2}\right) + 4f\left(\frac{3}{4}\right) + f(1)\right)$$

$$= \frac{1}{12}(1 + 4(1.65534) + 2(1.55152) + 4(1.06666) + 0.72159)$$

=1.30938

***<u>Example(2.4)</u>:*** Determine the degree of precision of Simpson's $\frac{3}{8}$ rule.

It will suffice to apply Simpson's $\frac{3}{8}$ rule over the interval [0,3] with the five test functions f(x)=1, x, $x^2$, $x^3$, and $x^4$. For the first four functions. Simpson's $\frac{3}{8}$ rule is exact.

$$\int_0^3 1dx = \frac{3}{8}(1 + 3(1) + 3(1) + 1) = 3$$

$$\int_0^3 xdx = \frac{3}{8}(0 + 3(1) + 3(2) + 3) = \frac{9}{2}$$

$$\int_0^3 x^2dx = \frac{3}{8}(0 + 3(1) + 3(4) + 9) = 9$$

$$\int_0^3 x^3dx = \frac{3}{8}(0 + 3(1) + 3(8) + 27) = \frac{81}{4}$$

the function f(x)=$x^4$ is the lowest power of x for which the rule is not exact.

$$\int_0^3 x^4dx = \frac{3}{8}(0 + 3(1) + 3(16) + 81) = \frac{99}{2}$$

Therefore, the degree of precision of Simpson's$\frac{3}{8}$ rule is n=3.

***Exercises:***

1. Consider a general interval [a,b]. Show that Simpson's rule produces exact results for the function $f(x)=x^2$ and $f(x)=x^3$, that is

   a. $\int_a^b x^2 dx = \frac{b^3}{3} - \frac{a^3}{3}$ b. $\int_a^b x^3 dx = \frac{b^4}{4} - \frac{a^4}{4}$

2. Integrate the Lagrange interpolation polynomial

   $$P_1(x) = f_0 \frac{(x-x_1)}{(x_0-x_1)} + f_1 \frac{(x-x_0)}{(x_1-x_0)}$$

   over the interval $[x_0,x_1]$ and establish the trapezoidal rule.

3. Determine the degree of precision of the trapezoidal rule.

## *2.2 Other Ways to Derive Integration Formulas Using Newton Forward Polynomial:*

During the integration we will need to change the variable of integration from x to t since our polynomials are expressed in terms of t. Observe that dx=hdt.

$$\int_{x_0}^{x_1} f(x)dx = h \int_{t=0}^{t=1} \left[ f_0 + t\Delta f_0 + \frac{t(t-1)}{2!}\Delta^2 f_{0+} \frac{t(t-1)(t-2)}{3!}\Delta^3 f_0 + \cdots \right] dt$$

$$= h \int_0^1 \left[ f_0 + t\Delta f_0 + \frac{t^2-t}{2}\Delta^2 f_0 + \frac{t^3-3t^2+2t}{6}\Delta^3 f_0 + \cdots \right] dt$$

$$= h \left[ f_0 t + \frac{t^2}{2}\Delta f_0 + \left(\frac{t^3}{6} - \frac{t^2}{4}\right)\Delta^2 f_0 + \left(\frac{t^4}{24} - \frac{t^3}{6} + \frac{t^2}{6}\right)\Delta^3 f_0 + \cdots \right]_{t=0}^{t=1}$$

$$= h \left[ f_0 + \frac{1}{2}\Delta f_0 - \frac{1}{12}\Delta^2 f_0 + \frac{1}{24}\Delta^3 f_0 + \cdots \right]$$

using first two terms only, we get:

$$\int_{x_0}^{x_1} f(x)dx = h\left[f_0 + \frac{1}{2}\Delta f_0\right] = h\left[f_0 + \frac{1}{2}(f_1 - f_0)\right] = \frac{h}{2}[f_0 + f_1]$$

***Exercise:***

Derive Simpson's formula using Newton Forward polynomial.

## *2.3 Composite Trapezoidal and Simpson's Rule:*

***Theorem(2.2):*** (Composite Trapezoidal Rule)

Suppose that the interval [a,b] is subdivided into subinterval $[x_k, x_{k+1}]$ of width h=(b-a)/M by using equally spaced nodes $x_k$=a+kh, for k=0,1,…,M. The **composite trapezoidal rule for M subintervals** can be expressed in:

$$\int_a^b f(x)dx \approx T(f,h) = \frac{h}{2}[f_0 + 2(f_1 + \cdots + f_{M-1}) + f_M]$$

$$= \frac{h}{2}[f(a) + f(b)] + h\sum_{k=1}^{M-1} f(x_k) \qquad (2.17)$$

***Proof:*** Apply the trapezoidal rule over each subinterval $[x_{k-1}, x_k]$. Use the additive property of the integral for subintervals:

$$\int_a^b f(x)dx = \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \cdots + \int_{x_{M-1}}^{x_M} f(x)dx$$

$$= \frac{h}{2}[f_0 + f_1] + \frac{h}{2}[f_1 + f_2] + \cdots + \frac{h}{2}[f_{M-1} + f_M]$$

$$= \frac{h}{2}[f_0 + 2(f_1 + f_2 + \cdots + f_{M-1}) + f_M].$$

***Example(2.5):*** Consider $f(x) = 2 + \sin(2\sqrt{x})$. Use the composite trapezoidal rule with 11 sample points to compute an approximation to the integral of f(x) taken over [1,6].

To generate 11 sample points, we use M=10 and h=(6-1)/10=1/2.

| *x* | *1* | *1.5* | *2* | *2.5* | *3* | *3.5* | *4* | *4.5* | *5* | *5.5* | *6* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *f(x)* | *2.909297* | *2.638157* | *2.308071* | *1.979316* | *1.683052* | *1.4353041* | *1.243197* | *1.108317* | *1.028722* | *1.000241* | *1.017357* |

$\int_1^6 f(x)dx = \frac{\frac{1}{2}}{2}[f(1) + 2\big(f(1.5) + f(2) + f(2.5) + f(3) + f(3.5) + f(4) + f(4.5) +$
$f(5) + f(5.5)\big) + f(6)]$=8.193854.

***Theorem(2.3):*** (Composite Simpson Rule)

Suppose that [a,b] is subdivided into 2M subintervals $[x_k, x_{k+1}]$ of equal width with h=(b-a)/(2M) by using $x_k$=a+kh for k=0,1,…,2M. The **composite Simpson rule for 2M subintervals** can be expressed in:

$$\int_a^b f(x)dx \approx S(f,h) = \frac{h}{3}[f_0 + 4f_1 + 2f_2 + 4f_3 + \cdots + 2f_{2M-2} + 4f_{2M-1} + f_{2M}]$$

$$= \frac{h}{3}[f(a) + f(b)] + \frac{2h}{3}\sum_{k=1}^{M-1} f(x_{2k}) + \frac{4h}{3}\sum_{k=1}^{M} f(x_{2k-1}) \qquad (2.18)$$

**proof**: (EXC)

***Example(2.6):*** Consider $f(x) = 2 + \sin(2\sqrt{x})$. Use the composite Simpson rule with 11 sample points to compute an approximation to the integral of f(x) taken over [1,6].

$\int_a^b f(x)dx = \frac{1/2}{3}[f(1) + f(6)] + \frac{1}{3}[f(2) + f(3) + f(4) + f(5)] + \frac{2}{3}[f(1.5) + f(2.5) +$
$f(3.5) + f(4.5) + f(5.5)]$=8.1830155

***Error Analysis:***

***Corollary(2.2):*** (Trapezoidal Rule: Error Analysis)

Suppose that [a,b] is subdivided into M subintervals $[x_k, x_{k+1}]$ of width h=(b-a)/M. The composite trapezoidal rule:

$$T(f,h) = \frac{h}{2}[f(a) + f(b)] + h\sum_{k=1}^{M-1} f(x_k) \qquad (2.19)$$

is an approximation to the integral:

$$\int_a^b f(x)dx = T(f,h) + E_T(f,h) \qquad (2.20)$$

Furthermore, if $f \in C^2[a,b]$, there exists a value c with a<c<b so that the error term $E_T(f,h)$ has the form:

$$E_T(f,h) = \frac{-(b-a)f^{(2)}(c)h^2}{12} = O(h^2) \qquad (2.21)$$

***Proof:*** We first determine the error term when the rule is applied over $[x_0, x_1]$. Integrating the Lagrange polynomial $P_1(x)$ and its remainder yields:

$$\int_{x_0}^{x_1} f(x)dx = \int_{x_0}^{x_1} P_1(x)dx + \int_{x_0}^{x_1} \frac{(x-x_0)(x-x_1)f^{(2)}(c(x))}{2!}dx \qquad (2.22)$$

The term $(x-x_0)(x-x_1)$ does not change sign on $[x_0, x_1]$, and $f^{(2)}(c(x))$ is continuous. Hence the second Mean value Theorem for integrals implies that there exists a value $c_1$ so that:

$$\int_{x_0}^{x_1} f(x)dx = \frac{h}{2}[f_0 + f_1] + f^{(2)}(c_1)\int_{x_0}^{x_1} \frac{(x-x_0)(x-x_1)}{2!}dx \qquad (2.23)$$

Use the change of variable $x=x_0+ht$ in the integral on the right side of (2.23)

$$\int_{x_0}^{x_1} f(x)dx = \frac{h}{2}[f_0 + f_1] + \frac{f^{(2)}(c_1)}{2}\int_0^1 h(t-0)h(t-1)hdt$$

$$= \frac{h}{2}[f_0 + f_1] + \frac{f^{(2)}(c_1)h^3}{2}\int_0^1 (t^2 - t)dt$$

$$= \frac{h}{2}[f_0 + f_1] - \frac{f^{(2)}(c_1)h^3}{12} \qquad (2.24)$$

Now we are ready to add up the error terms for all of the intervals $[x_k, x_{k+1}]$:

$$\int_a^b f(x)dx = \sum_{k=1}^{M}\int_{x_{k-1}}^{x_k} f(x)dx = \sum_{k=1}^{M}\frac{h}{2}[f(x_{k-1}) + f(x_k)] - \frac{h^3}{12}\sum_{k=1}^{M} f^{(2)}(c_k) \qquad (2.25)$$

The first sum is the composite trapezoidal rule T(f,h). In the second term, one factor of h is replaced with its equivalent h=(b-a)/M, and the result is:

$$\int_a^b f(x)dx = T(f,h) - \frac{(b-a)h^2}{12}\left(\frac{1}{M}\sum_{k=1}^{M} f^{(2)}(c_k)\right)$$

The term in parentheses can be recognized as an average of values for the second derivative and hence is replaced by $f^{(2)}(c)$. Therefore, we have established that:

$$\int_a^b f(x)dx = T(f,h) - \frac{(b-a)f^{(2)}(c)h^2}{12}$$

and the proof is complete.

***<u>Corollary(2.3)</u>:*** (Simpson's rule: Error analysis)

Suppose that [a,b] is subdivided into 2M subintervals $[x_k, x_{k+1}]$ of equal width h=(b-a)/(2M). The composite Simpson rule

$$S(f,h) = \frac{h}{3}(f(a)+f(b)) + \frac{2h}{3}\sum_{k=1}^{M-1} f(x_{2k}) + \frac{4h}{3}\sum_{k=1}^{M} f(x_{2k-1}) \quad (2.26)$$

is an approximation to the integral:

$$\int_a^b f(x)dx = S(f,h) + E_S(f,h) \quad (2.27)$$

Furthermore, if $f \in C^4[a,b]$, there exists a value c with a<c<b so that the error term $E_S(f,h)$ has the form:

$$E_S(f,h) = \frac{-(b-a)f^{(4)}(c)h^4}{180} = O(h^4) \quad (2.28)$$

***<u>Example(2.7)</u>:*** Consider $f(x) = \frac{1}{x}$. Investigate the error when the composite trapezoidal rule is used over [1,6] and the number of subintervals is 10.

h=(6-1)/10=0.5, since:

$$E_T(f,h) = \frac{-(b-a)f^{(2)}(c)h^2}{12} = O(h^2)$$

we first compute $f'(x) = \frac{-1}{x^2}$ and $f''(x) = \frac{2}{x^3}$,therefore:

$$f''(1) = 2, f''(2) = \frac{1}{4}, f''(6) = \frac{2}{6^3} = 0.009\,259$$

and hence f''(c)=2 and $E_T(f,h)=\frac{-(6-1)(2)(0.5)^2}{12}=\frac{-2.5}{12}=-0.208\,333$

***Example(2.8):*** Find the number M and the step size h so that the error $E_S(f,h)$ for the Simpson's rule is less than $5\times10^{-9}$ for the approximation $\int_2^7 dx/_x \approx S(f,h)$.

$$f(x)=\frac{1}{x}\xrightarrow{yields}f'(x)=\frac{-1}{x^2}\xrightarrow{yields}f''(x)=\frac{2}{x^3}\xrightarrow{yields}f^{(3)}(x)=\frac{-6}{x^4}\xrightarrow{yields}f^{(4)}(x)=\frac{24}{x^5}$$

the maximum value of $|f^{(4)}(x)|$ taken over [2,7] occurs at the end point x=2 and $f^{(4)}(2)=3/4$, then:

$$|E_S(f,h)|=\frac{\left|-(b-a)f^{(4)}(c)h^4\right|}{180}\le\frac{(7-2)\frac{3}{4}h^4}{180}=\frac{h^4}{48}$$

The step size h and number M satisfy the relation h=5/(2M), and this is used in the above equation to get the relation

$$|E_S(f,h)|\le\frac{625}{768M^4}\le5\times10^{-9}$$

$$\xrightarrow{yields}\frac{125}{768}\times10^9\le M^4\xrightarrow{yields}112.95\le M$$

since M must be integer, we chose M=113

and the corresponding step size h=5/226=0.022123

***Exercises:***

1. Approximate the integral $\int_{-1}^{1}\frac{dx}{1+x^2}$using the composite trapezoidal rule with M=10.
2. The length of the curve y=f(x) over the interval $a\le x\le b$ is L=$\int_a^b\sqrt{1+(f'(x)^2}$ approximate the length of the function $f(x)=x^3$ over [0,1] using composite Simpsons rule with M=5.

3. Verify that the trapezoidal rule (M=1, h=1) is exact for polynomials of degree≤1 of the form $f(x)=c_1x+c_0$ over [0,1].
4. Determine the number M and the interval width h so that the composite trapezoidal rule for M subintervals can be used to compute the integral $\int_0^2 xe^{-x}dx$ with an accuracy of $5 \times 10^{-9}$ .

## *2.4 Romberg Integration:*

The discussion here is based upon the trapezium rule. Let the integration domain [a,b] be divided by three equispaced nodes $x_0=a$, $x_1=(a+b)/2$ and $x_2=b$ at interval of size h. Two successive trapezium estimates using one and two subintervals respectively are:

$$T_1 = \frac{2h}{2}[f(x_0) + f(x_1)] \ \ and \ \ T_2 = \frac{h}{2}[f(x_0) + 2f(x_1) + f(x_2)]$$

On including the truncation error for this estimate we can write:

$$I = T_1 - \frac{(2h)^2}{12} f''(x_0) - G(2h)^4 - \cdots$$

$$I = T_2 - \frac{h^2}{12} f''(x_0) - Gh^4 - \cdots$$

where G is independent of the step size h. Four times the second estimate minus the first estimate gives:

$$I = \frac{1}{3}[4T_2 - T_1] + 4Gh^4 + O(h^6) \qquad (2.29)$$

Taken as an estimate to I, the values $(4T_2-T_1)/3$ has leading error of $O(h^4)$. Expand this estimate:

$$I \approx \frac{1}{3}[4T_2 - T_1] = \frac{1}{3}\left[4\left\{\frac{h}{2}(f_0 + 2f_1 + f_2)\right\} - \frac{2h}{2}(f_0 + f_2)\right]$$

$$= \frac{h}{3}[f_0 + 4f_1 + f_2]$$

Shows it to be the Simpson estimate $S_2$ using two sub-intervals of size h=(b-a)/2.

This process can be carried out for any two trapezium estimates $T_N$ and $T_{2N}$ to give the more accuracy Simpson's estimate $S_{2N}$.

| **Trapezoidal** | **Simpson** | |
|---|---|---|
| $T_1$ | | |
| $T_2$ | $S_2$ | |
| $T_4$ | $S_4$ | **In general** $S_{2N}=1/3\{4T_{2N}-T_N\}$ |
| $T_8$ | $S_8$ | |

In the same way we get:

$$I \approx \frac{1}{15}[16S_4 - S_2] + O(h^6) \qquad (2.30)$$

known as Boole's rule.

| **Trapezoidal** | **Simpson** | **Boole's** | |
|---|---|---|---|
| $T_1$ | | | |
| $T_2$ | $S_2$ | | |
| $T_4$ | $S_4$ | $B_4$ | **In general** $S_{2N}=1/3\{4T_{2N}-T_N\}$ |
| $T_8$ | $S_8$ | $B_8$ | **In general** $B_{4N}=1/15\{16S_{4N}-S_{2N}\}$ |

***Example(2.9):*** Estimate the value of $\int_0^1 e^{sinx}dx$ using Romberg integration

| N | Trapezium k=1 | Simpson k=2 | Boole k=3 | k=4 |
|---|---|---|---|---|
| 1 | 1.659 888 | | | |
| 2 | 1.637 517 | 1.630 060 | | |
| 4 | 1.633 211 | 1.631 776 | 1.631 891 | |
| 8 | 1.632 201 | 1.631 864 | 1.631 869 | 1.631 869 |

***Exercises:***

1. Use Romberg integration to estimate $\int_0^2 x^2e^{-x^2}dx$ as accurately as possible, working to four decimal places.

# Chapter3: Numerical Solution of Ordinary Differential Equations

## 3.1 Numerical Solution of a First-Order ODE

A numerical solution of a first order ODE formulated as

$$\frac{dy}{dx} = f(x, y) \text{ with the initial condition } y(x_1) = y_1 \qquad (3.1)$$

is a set of discrete points that approximate the function y(x). When a differential equation is solved numerically, the problem statement also includes the domain of the solution. For example, a solution is required for values of the independent variable from $x = a$ to $x = b$ (the domain is [a, b]). Depending on the numerical method used to solve the equation, the number of points between a and b at which the solution is obtained can be set in advance, or it can be decided by the method. For example, the domain can be divided into N subintervals of equal width defined by N + 1 values of the independent variable from $x_1 = a$ to $x_{N+1} = b$. The solution consists of values of the dependent variable that are determined at each value of the independent variable. The solution then is a set of points $(x_1, y_1), (x_2, y_2), \ldots, (x_{N+1}, Y_{N+1})$ that define the function y( x) .

### 3.1.1 Overview of Numerical Methods Used/or Solving a First-Order ODE

Numerical solution is a procedure for calculating an estimate of the exact solution at a set of discrete points. The solution process is incremental, which means that it is determined in steps. It starts at the point where the initial value is given. Then, using the known solution at the first point, a solution is determined at a second nearby point. This is followed by a solution at a third point, and so on.

There are procedures with a single-step and multistep approach. In a **single-step approach**, the solution at the next point, $x_{i+1}$, is calculated from the already known solution at the present point, $x_i$. In a **multi-step approach**, the solution at $x_{i+1}$ is calculated from the known solutions at several previous points. The idea is that the value of the function at several previous points can give a better estimate for the trend of the solution.

Also, two types of methods, explicit, and implicit, can be used for calculating the solution at each step. The difference between the methods is in the way that the solution is calculated at each step. Calculating the value of the dependent variable at the next value of the independent variable. In an **explicit formula**, the right-hand side of the equation only has known quantities. In other words, the next unknown value of the dependent variable, $y_{i+1}$, is calculated by evaluating an expression of the form:

$$y_{i+1} = F(x_i, x_{i+1}, y_i) \qquad (3.2)$$

where $x_i$, $y_i$, and $x_{i+1}$ are all known quantities. In **implicit methods**, the equation used for calculating $y_{i+1}$from the known $x_i$, $y_i$, and $x_{i+1}$ has the form:

$$y_{i+1} = F(x_i, x_{i+1}, y_{i+1}) \qquad (3.3)$$

Here, the unknown $y_{i+1}$ appears on both sides of the equation.

## 3.1.2 Errors in Numerical Solution of ODEs

Two types of errors, round-off errors and truncation errors, occur when ODEs are solved numerically. Round-off errors are due to the way that computers carry out calculations. **Truncation errors** are due to the approximate nature of the method used to calculate the solution. Since the numerical solution of a differential equation is calculated in increments (steps), the truncation error at each step of the solution consists of two parts. One, called **local truncation error**, is due to the application of the numerical method in a single step. The second part, called **propagated, or accumulated, truncation error**, is due to the accumulation of local truncation errors from previous steps. Together, the two parts are the **global (total) truncation error** in the solution.

## 3.1.3 Single-step explicit methods

In a single-step explicit method, illustrated in Fig. 3-1,

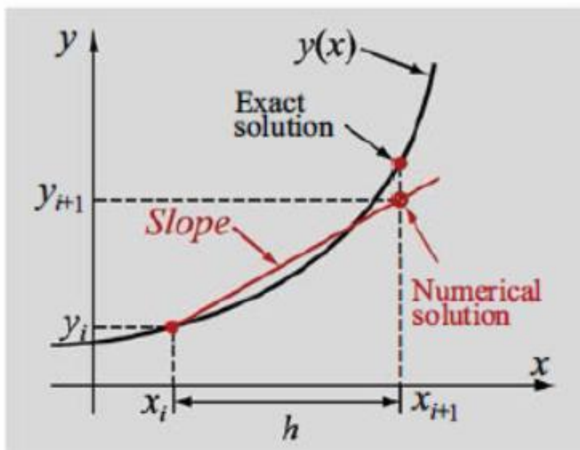


**Figure 3-1: Single-step explicit methods.**

The approximate numerical solution $(x_{i+1}, y_{i+1})$ is calculated from the known solution at point $(x_i, y_i)$ by:

$$x_{i+1} = x_i + h \qquad (3.4)$$

$$y_{i+1} = y_{i+1} + \text{Slope} \cdot h \qquad (3.5)$$

where h is the step size, and the Slope is a constant that estimates the value of $\frac{dy}{dx}$ in the interval from $x_i$to $x_{i+1}$. The numerical solution starts at the point where the initial value is known. This corresponds to $i = 1$ and point $(x_1, y_1)$. Then $i$ is increased to $i = 2$, and the solution at the next point, $(x_2, y_2)$, is calculated by using Eqs. (3.4) and (3.5). The procedure continues with $i = 3$ and so on until the points cover the whole domain of the solution.

# 3.2 EULER'S METHODS

Euler's method is the simplest technique for solving a first-order ODE of the form of Eq. (3.1):

$$\frac{dy}{dx} = f(x, y) \; with \; the \; initial \; condition \; y(x_1) = y_1$$

The method can be formulated as an explicit or an implicit method.

## 3.2.1 Euler's Explicit Method

Euler's explicit method (also called the forward Euler method) is a single-step, numerical technique for solving a first-order ODE. The method uses Eqs. (3.4) and (3.5), where the value of the constant Slope in Eq. (3.5) is the slope of *y(x)* at point $(x_i,\ y_i)$. This slope is actually calculated from the differential equation:

$$Slope = \frac{dy}{dx}\Big|_{x=x_i} = f(x_i, y_i) \qquad (3.6)$$

Euler's method assumes that for a short distance $h$ near $(x_i, y_i)$, the function *y(x)* has a constant slope equal to the slope at $(x_i, y_i)$. With this assumption, the next point of the numerical solution $(x_{i+1}, y_{i+1})$ is calculated by:

$$x_{i+1} = x_i + h \qquad (3.7)$$

$$y_{i+1} = y_i + f(x_i, y_i)h \qquad (3.8)$$

Equation (3.8) of Euler's method can be derived in several ways. Starting with the given differential equation:

$$\frac{dy}{dx} = f(x, y) \qquad (3.9)$$

An approximate solution of Eq. (3.9) can be obtained either by numerically integrating the equation or by using a finite difference approximation for the derivative.

**3.2.1.1 Deriving Euler's method by using finite difference approximation for the derivative**

Euler's formula, Eq. (3.8), can be derived by using an approximation for the derivative in the differential equation. The derivative $\frac{dy}{dx}$ in Eq. (3.8) can be approximated with the forward difference formula by evaluating the ODE at the point $x = x_i$:

$$\frac{dy}{dx}\Big|_{x=x_i} \approx \frac{y_{i+1} - y_i}{x_{i+1} - x_i} = f(x_i, y_i) \qquad (3.10)$$

Solving Eq. (3.10) for $y_{i+1}$ gives Eq. (3.8) of Euler's method. (Because the equation can be derived in this way, the method is also known as the **forward Euler method**.)

**Example 3-1:** Use Euler's explicit method to solve the ODE

$$\frac{dy}{dx} = -1.2y + 7e^{-0.3x}$$

from $x = 0$ to $x = 2.5$ with the initial condition $y = 3$ at $x = 0$.

(a) Solve by hand using $h = 0.5$.

( b) Write a MATLAB program in a script file that solves the equation using $h = 0.5$.

(c) Use the program from part (b) to solve the equation using h = 0.1.

In each part compare the results with the exact (analytical) solution:

$$y(x) = \frac{70}{9}e^{-0.3x} - \frac{43}{9}e^{-1.2x}$$

**Solution:**

***(a) Solution by hand:*** The first point of the solution is (0, 3), which is the point where the initial condition is given. For the first point $i = 1$. The values of $x$ and $y$ are $x_1 = 0$ and $y_1 = 3$. The rest of the solution is determined by using Eqs. (3.7) and (3.8). In the present problem these equations have the form:

$$x_{i+1} = x_i + h = x_i + 0.5 \quad (3.11)$$

$$y_{i+1} = y_i + f(x_i, y_i)h = y_i + (-1.2y_i + 7e^{-0.3x_i})0.5 \quad (3.12)$$

Equations (3.11) and (3.12) are applied five times with $i$ = 1, 2, 3, 4, and 5.

**First step:** For the first step $i = 1$. Equations (3.11) and (3.12) give:

$$x_2 = x_1 + h = 0 + 0.5 = 0.5$$

$$y_2 = y_1 + (-1.2y_1 + 7e^{-0.3x_1})0.5 = 4.7$$

The second point is (0.5, 4.7).

**Second step:** For the second step $i = 2$. Equations (3.11) and (3.12) give:

$$x_3 = x_2 + h = 0.5 + 0.5 = 1$$

$$y_3 = y_2 + (-1.2y_2 + 7e^{-0.3x_2})0.5 = 4.8924779$$

The third point is (1, 4.8924779).

**Third step:** For the third step $i = 3$. Equations (3.11) and (3.12) give:

$$x_4 = x_3 + h = 1 + 0.5 = 1.5$$

$$y_4 = y_3 + (-1.2y_3 + 7e^{-0.3x_3})0.5 = 4.5498549$$

The fourth point is (1.5, 4.5498549).

**Fourth step:** For the fourth step $i = 4$. Equations (3.11) and (3.12) give:

$$x_5 = x_4 + h = 1.5 + 0.5 = 2$$

$$y_5 = y_4 + (-1.2y_4 + 7e^{-0.3x_4})0.5 = 4.0516405$$

The fifth point is (2, 4.0516405).

**Fifth step:** For the fourth step $i = 5$. Equations (3.11) and (3.12) give:

$$x_6 = x_5 + h = 2 + 0.5 = 2.5$$

$$y_6 = y_5 + (-1.2y_5 + 7e^{-0.3x_5})0.5 = 3.5414969$$

The sixth point is (2.5, 3.5414969).

The values of the exact and numerical solutions, and the error, which is the difference between the two, are:

| $i$ | $x_i$ | $y_i$ numerical | $y(x_i)$ *exact* | Error |
|---|---|---|---|---|
| 1 | 0 | 3.0000000 | 3.0000000 | 0 |
| 2 | 0.5000 | 4.7000000 | 4.0722953 | 0.6277047 |
| 3 | 1.0000 | 4.8924779 | 4.3228804 | 0.5695975 |
| 4 | 1.5000 | 4.5498549 | 4.1695687 | 0.3802862 |
| 5 | 2.0000 | 4.0516405 | 3.8351047 | 0.2165358 |
| 6 | 2.5000 | 3.5414969 | 3.4360905 | 0.1054064 |

***(b) To solve the ODE with MATLAB:***

```
function d=euler(f,y1,a,b,n)
h=(b-a)/n;x(1)=a;y(1)=y1;
for k=1:n
    x(k+1)=x(k)+h;
    y(k+1)=y(k)+h*f(x(k),y(k));
end
d=[x' y']
```

### 3.2.2 Analysis of Truncation Error in Euler's Explicit Method

As mentioned in Section 3.1.2, when ODEs are solved numerically there are two sources of error, round-off and truncation. The round-off errors are due to the way that computers carry out calculations. The truncation error is due to the approximate nature of the method used for calculating the solution in each increment (step). In addition, since the numerical solution of a differential equation is calculated in increments (steps), the truncation error consists of a local truncation error and propagated truncation error. The truncation errors in Euler's explicit method are discussed in this section.

The discussion is divided into two parts. First, the **local truncation error** is analyzed, and then the results are used for determining an estimate of the **global truncation error**.

**Definition 3.1:** Assume that $\{(x_k,y_k),k=1,\dots,N\}$ is the set of discrete approximations and that $y=y(x)$ is the unique solution to the initial value problem. The ***global discretization error*** $\boldsymbol{e_k}$ is defined by:

$$e_k=y(x_k)-y_k \quad for\ k=1,\dots,N \tag{3.13}$$

The local discretization error $\in_{k+1}$ is defined by:

$$\in_{k+1}=y(x_{k+1})-y_k-h\emptyset(x_k,y_k) \quad for\ k=1,\dots,N-1 \tag{3.14}$$

for some function $\emptyset$ called an increment function.

**Theorem 3.1:** (Precision of Euler's Method)

Assume that $y(x)$ is the solution to the IVP given in (3.1).If $y(x)\in C^2[t_0,b]$ and $\{(x_k,y_k),k=1,\dots,N\}$ is the sequence of approximations generated by Euler's method, then:

$$|e_k|=|y(x_k)-y_k|=O(h) \tag{3.15}$$

$$|\in_{k+1}|=|y(x_{k+1})-y_k-hf(x_k,y_k)|=O(h^2) \tag{3.16}$$

The error at the end of the interval is called the ***final global error (FGE)***:

$$E(y(b),h)=|y(b)-y_M|=O(h) \tag{3.17}$$

### 3.2.3 Euler's Implicit Method

The form of Euler's implicit method is the same as the explicit scheme, except, for a short distance, $h$, near $(x_i, y_i)$ the slope of the function $y(x)$ is taken to be a constant equal to the slope at the endpoint of the interval $(x_{i+1}, y_{i+1})$. With this assumption, the next point of the numerical solution $(x_{i+1}, y_{i+1})$ is calculated by:

$$x_{i+1} = x_i + h \tag{3.18}$$

$$y_{i+1}= y_i + f(x_{i+1}, y_{i+1})h \tag{3.19}$$

Now, the unknown $y_{i+1}$ appears on both sides of Eq. (3.19), and unless $f(x_{i+1}, y_{i+1})$depends on $y_{i+1}$ in a simple linear or quadratic form, it is not easy or even possible to solve the equation for $y_{i+1}$ explicitly.

# 3.3 MODIFIED EULER'S METHOD

The modified Euler method is a single-step, explicit, numerical technique for solving a first-order ODE. The method is a modification of Euler's explicit method. (This method is sometimes called **Heun's method**). As discussed in Section 3.2.1, the main assumption in Euler's explicit method is that in each subinterval (step) the derivative (slope) between points $(x_i, y_i)$ and $(x_{i+1}, y_{i+1})$is constant and equal to the derivative (slope) of *y(x)* at point $(x_i, y_i)$. This assumption is the main source of error. In the modified Euler method the slope used for calculating the value of $y_{i+1}$ is modified to include the effect that the slope changes within the subinterval. The slope used in the modified Euler method is the average of the slope at the beginning of the interval and an estimate of the slope at the end of the interval. The slope at the beginning is given by:

$$\frac{dy}{dx}\Big|_{x=x_i} = f(x_i, y_i) \qquad (3.20)$$

The estimate of the slope at the end of the interval is determined by first calculating an approximate value for $y_{i+1}$ written as $y_{i+1}^{Eu}$using Euler's explicit method:

$$y_{i+1}^{Eu} = y_i + f(x_i, y_i) \qquad (3.21)$$

and then estimating the slope at the end of the interval by substituting the point $\left(x_{i+1}, y_{i+1}^{Eu}\right)$ in the equation for $\frac{dy}{dx}$ :

$$\frac{dy}{dx}\Big|_{\substack{x=x_{i+1} \\ y=y_{i+1}^{Eu}}} = f(x_{i+1}, y_{i+1}^{Eu}) \qquad (3.22)$$

The modified Euler method is summarized in the following algorithm.

***<u>Algorithm for the modified Euler method</u>***

1. Given a solution at point $(x_i, y_i)$, calculate the next value of the independent variable:

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

2. Calculate$f(x_i, y_i)$.
3. Estimate $y_{i+1}$ using Euler's method:

$$y_{i+1}^{Eu} = y_i + f(x_i, y_i)$$

4. Calculate $(x_{i+1}, y_{i+1}^{Eu})$ .
5. Calculate the numerical solution at $x = x_{i+1}$:

$$y_{i+1} = y_i + \frac{h}{2}\left[f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{Eu})\right]$$

**Example 10-2:**Use the modified Euler method to solve the ODE

$$\frac{dy}{dx} = -1.2y + 7e^{-0.3x}$$

from *x=0* to *x* = 2.5 with the initial condition *y(0) = 3*. Using *h = 0.5*. Compare the results with the exact (analytical) solution:

$y(x) = \frac{70}{9}e^{-0.3x} - \frac{43}{9}e^{-1.2x}$.

**Solution:**

The first point of the solution is (0, 3), which is the point where the initial condition is given. For the first point $i = 1$. The values of $x$ and $y$ are $x_1 = 0$ and $y_1 = 3$.

In the present problem these equations have the form:

$$x_{i+1} = x_i + h = x_i + 0.5$$

$$y_{i+1}^{Eu} = y_i + f(x_i, y_i)h = y_i + (-1.2y_i + 7e^{-0.3x_i})0.5$$

$$y_{i+1} = y_i + \frac{h}{2}\left[f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{Eu})\right] = y_i + \frac{0.5}{2}\left[(-1.2y_i + 7e^{-0.3x_i}) + \left(-1.2y_{i+1}^{Eu} + 7e^{-0.3x_{i+1}}\right)\right]$$

**First step:** For the first step $i = 1$:

$$x_2 = x_1 + h = 0 + 0.5 = 0.5$$

$$y_2^{Eu} = y_1 + (-1.2y_1 + 7e^{-0.3x_1})0.5 = 4.7$$

$$y_i + \frac{0.5}{2}[(-1.2y_1 + 7e^{-0.3x_1}) + (-1.2y_2^{Eu} + 7e^{-0.3x_2})] = 3.946238958743852$$

The second point is (0.5, 3.946238958743852).

The values of the exact and numerical solutions, and the error, which is the difference between the two, are:

| $i$ | $x_i$ | $y_i$ numerical | $y(x_i)$ exact | Error |
|---|---|---|---|---|
| 1 | 0 | 3.0000000 | 3.0000000 | 0 |
| 2 | 0.5000 | 3.946238958743852 | 4.0722953 | 0.126056374335137 |
| 3 | 1.0000 | 4.187746065761980 | 4.3228804 | 0.135134415959749 |
| 4 | 1.5000 | 4.063314737957255 | 4.1695687 | 0.106253975375624 |
| 5 | 2.0000 | 3.763482617314995 | 3.8351047 | 0.071622108811351 |
| 6 | 2.5000 | 3.393629530605291 | 3.4360905 | 0.042460997400584 |

Comparing the error values here with those in Example 3-1, where the problem was solved with Euler's explicit method using the same size subintervals, shows that the error with the modified Euler method is much smaller.

# 3.4 RUNGE-KUTTA METHODS

Runge-Kutta methods are a family of single-step, explicit, numerical techniques for solving a first-order ODE. As was stated in Section 3.1, for a subinterval (step) defined by $[x_i, x_{i+1}]$, where h = $x_{i+1}$-$x_i$, the value of $y_{i+1}$ is calculated by:

$$y_{i+1} = y_i + slop.h \qquad (3.23)$$

where Slope is a constant. The value of Slope in Eq. (3.23) is obtained by considering the slope at several points within the subinterval. Various types of Runge-Kutta methods are classified according to their order. The order identifies the number of points within the sub interval that are used for determining the value of Slope in Eq. (3.23). Second order Runge-Kutta methods use the slope at two points, third-order methods use three points, and so on. The so-called classical Runge-Kutta method is of fourth order and uses four points. The order of the method is also related to the global truncation error of each method. For example, the second-order Runge-Kutta method is second-order accurate globally; that is, it has a local truncation error of $O(h^3)$ and a global truncation error of $O(h^2)$.

### 3.4.1 Second-Order Runge-Kutta Methods

The general form of second-order Runge-Kutta methods is:

$$\left.\begin{aligned} y_{i+1} &= y_i + \frac{h}{2}(k_1 + k_2) \\ k_1 &= f(x_i, y_i) \\ k_2 &= f(x_i + h, y_i + k_1 h) \end{aligned}\right\} \qquad (3.24)$$

**Example 3-3:** Solving by hand a first-order ODE using the second-order Runge-Kutta method to solve the ODE

$$\frac{dy}{dx} = -1.2y + 7e^{-0.3x}$$

from $x=0$ to $x = 2.5$ with the initial condition $y(0) = 3$. Using $h = 0.5$. Compare the results with the exact (analytical) solution:

$$y(x) = \frac{70}{9}e^{-0.3x} - \frac{43}{9}e^{-1.2x}.$$

**Solution:**

The first point of the solution is (0, 3), which is the point where the initial condition is given. For the first point $i = 1$. The values of $x$ and $y$ are $x_1 = 0$ and $y_1 = 3$.

The rest of the solution is done by steps. In each step the next value of the independent variable is given by:

$$x_{i+1} = x_i + h = x_i + 0.5 \qquad (3.25)$$

The value of the dependent variable $y_{i+1}$ is calculated by first calculating $k_1$ and $k_2$ using :

$$\left.\begin{aligned} k_1 &= f(x_i, y_i) \\ k_2 &= f(x_i + h, y_i + k_1 h) \end{aligned}\right\} \qquad (3.26)$$

and then substituting the $k$'s in :

$$y_{i+1} = y_i + \frac{h}{2}(k_1 + k_2) \qquad (3.27)$$

**First step:** In the first step $i = 1$. Equations (3. 25)-(3. 27) give:

$x_2 = x_1 + 0.5 = 0.5$

$k_1 = f(x_1, y_1) = f(0,3) = -1.2(3) + 7e^{-0.3(0)} = 3.4$

$k_2 = f(x_1 + h, y_1 + k_1 h) = f\big(0 + 0.5, 3 + 3.4(0.5)\big) = f(0.5, 1.7)$

$= -1.2(1.7) + 7e^{-0.3(0.5)} = 0.384955834975405$

$y_2 = y_1 + \frac{h}{2}(k_1 + k_2) = 3 + \frac{0.5}{2}(3.4 + 0.384955834975405) = 3.946238958743852$

**Second step:** In the first step $i = 2$. Equations (3. 25)-(3. 27) give:

$x_3 = x_2 + 0.5 = 1.0$

$k_1 = f(x_2, y_2) = f(0.5, 3.946238958743852)$

$= -1.2(3.946238958743852) + 7e^{-0.3(0.5)} = 1.289469084482783$

$k_2 = f(x_2 + h, y_2 + k_1 h)$

$= f\big(0.5 + 0.5, 3.946238958743852 + 1.289469084482783(0.5)\big)$

$= -0.323440656410266$

$y_3 = y_2 + \frac{h}{2}(k_1 + k_2)$ =4.187746065761980

**Third step:**

k1 = 0.160432265857648

k2 = -0.658157577076552

$y_4$ =4.063314737957255

**Fourth step:**

k1 = -0.412580624196292

k2 = -0.786747858372744

$y_5 = 3.763482617314995$

**Fifth step:**

k1 = -0.674497688119808

k2 = -0.804914658719007

$y_6 = 3.393629530605291$

The values of the exact and numerical solutions, and the error, which is the difference between the two, are:

| *i* | $x_i$ | $y_i$ numerical | *y*($x_i$) *exact* | Error |
|---|---|---|---|---|
| 1 | 0 | 3.0000000 | 3.0000000 | 0 |
| 2 | 0.5000 | 3.946238958743852 | 4.0722953 | 0.126056374335137 |
| 3 | 1.0000 | 4.187746065761980 | 4.3228804 | 0.135134415959749 |
| 4 | 1.5000 | 4.063314737957255 | 4.1695687 | 0.106253975375624 |
| 5 | 2.0000 | 3.763482617314995 | 3.8351047 | 0.071622108811351 |
| 6 | 2.5000 | 3.393629530605291 | 3.4360905 | 0.042460997400584 |

The solution obtained is obviously identical (except for rounding errors) to the solution in example 3-2.

### 3.4.2 Fourth-Order Runge-Kutta Methods

The general form of classical fourth-order Runge-Kutta method is:

$$\left.\begin{aligned} & y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ & with \\ & \qquad k_1 = f(x_i, y_i) \\ & \qquad k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{hk_1}{2}\right) \\ & \qquad k_3 = f\left(x_i + \frac{h}{2}, y_i + \frac{hk_2}{2}\right) \\ & \qquad k_4 = f(x_i + h, y_i + hk_3) \end{aligned}\right\} \qquad (3.28)$$

**Example 3-4:** Solving by hand a first-order ODE using the fourth-order Runge-Kutta method to solve the ODE

$$\frac{dy}{dx} = -1.2y + 7e^{-0.3x}$$

from *x=0* to *x* = 2.5 with the initial condition *y(0) = 3*. Using *h = 0.5*. Compare the results with the exact (analytical) solution:

$y(x)=\frac{70}{9}e^{-0.3x}-\frac{43}{9}e^{-1.2x}.$

**Solution:**

**First step:**

$k_1=f(x_1,y_1)=f(0,3)$ =3.40

$k_2=f\left(x_1+\frac{h}{2},y_1+\frac{hk_1}{2}\right)$ =1.874204404299870

$k_3=f\left(x_1+\frac{h}{2},y_1+\frac{hk_2}{2}\right)$ =2.331943083009909

$k_4=f(x_1+h,y_1+hk_3)$ = 1.025789985169459

$y_2=y_2+\frac{h}{6}(k_1+2k_2+2k_3+k_4)$ =4.069840413315752

**Second step:**

k1 = 1.141147338996503
k2 = 0.363460833637786
k3 = 0.596766785245403
k4 = -0.056141022354118
$y_3$ =4.320295542849815

**Third step:**

k1 =0.001372893352247
k2 =-0.373741567888647
k3 =-0.261207229516379
k4 = -0.564233252357536
$y_4$ =4.167565713365203

**Fourth step:**

k1 = -0.537681794685830
k2 =-0.698886767064788
k3 =-0.650525275351102
k4 =-0.769082238169397
$y_5$ =3.833766703557953

**Fifth step:**

k1 =-0.758838591611358
k2 =-0.808773522533291
k3 =-0.793793043256712
k4 =-0.817678349128413
$y_6$ =3.435295864197971

The values of the exact and numerical solutions, and the error, which is the difference between the two, are:

| $i$ | $x_i$ | $y_i$ numerical | $y(x_i)$ exact | Error |
|---|---|---|---|---|
| 1 | 0 | 3.000000000000000 | 3.0000000 | 0 |
| 2 | 0.5000 | 4.069840413315752 | 4.0722953 | 0.002454919763237 |
| 3 | 1.0000 | 4.320295542849815 | 4.3228804 | 0.002584938871915 |
| 4 | 1.5000 | 4.167565713365203 | 4.1695687 | 0.002002999967676 |
| 5 | 2.0000 | 3.833766703557953 | 3.8351047 | 0.001338022568394 |
| 6 | 2.5000 | 3.435295864197971 | 3.4360905 | 0.000794663807904 |

# 3.5 Predictor-Corrector Methods

Predictor-corrector methods refer to a family of schemes for solving ordinary differential equations using two formulae: ***predictor* and *corrector* formula**. In predictor-corrector methods, four prior values are required to find the value of $y$ at $x_n$. Predictor-corrector methods have the advantage of giving an estimate of error from successive approximations to $y_n$. The predictor is an explicit formula and is used first to determine an estimate of the solution $y_{n+1}$. The value $y_{n+1}$ is calculated from the known solution at the previous point $(x_n, y_n)$ using single-step method or several previous points (multi-step methods). If $x_n$ and $x_{n+1}$ are two consecutive mesh points such that :

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

then in Euler's method, we have:

$$y_{i+1} = y_i + h\, f(x_i, y_i),\ i = 0, 1, 2, 3, \ldots \qquad (3.29)$$

Once an estimate of $y_{i+1}$ is found, the corrector is applied. The corrector uses the estimated value of $y_{i+1}$ on the right-hand side of an otherwise implicit formula for computing a new, more accurate value for $y_{n+1}$ on the left-hand side. The modified Euler's method gives as:

$$y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, y_{i+1})] \qquad (3.30)$$

The value of $y_{i+1}$ is first estimated by Eq.(3.29) and then utilized in the right-hand side of Eq.(3.30) resulting in a better approximation of $y_{i+1}$. The value of $y_{i+1}$ thus obtained is again substituted in Eq.(3.30) to find a still better approximation of $y_{i+1}$. This procedure is repeated until two consecutive iterated values of $y_{i+1}$ are very close. Here, the corrector equation (3.30) which is an implicit equation is being used in an *explicit* manner since no solution of a non-linear equation is required.

In addition, the application of corrector can be repeated several times such that the new value of $y_{i+1}$ is substituted back on the right-hand side of the corrector formula to obtain a more refined value for $y_{i+1}$. The technique of refining an initially crude estimate of $y_{i+1}$ by means of a more accurate formula is known as ***predictor-corrector* method**. Equation (2.29) is called the ***predictor*** and Eq. (3.30) is called the ***corrector*** of $y_{n+1}$.

**Example 3.5:** Use the PC method on (2, 3) with $h = 0.1$ for the initial value problem

$$\frac{dy}{dx} = -xy^2,\ y(2) = 1.$$

Exact solution is $y(x) = \dfrac{2}{x^2 - 2}$.

***<u>Solution:</u>***

First, we use Euler method:

$y_1 = y_0 + hf(x_0, y_0)$=1+0.1(-2(1)$^2$)=0.8

Then, we use modified Euler:

$y_1 = y_0 + \frac{h}{2}[f(x_0, y_0) + f(x_1, y_1)]$=1+0.1/2*[-2*1$^2$+(-2.1)*(0.8)$^2$]=0.8328

Containing in the same manner, we obtain:

| $x_i$ | $y_i$ | $Y(x_i)$ |
|---|---|---|
| 2 | 1.000000000000000 | 1.000000000000000 |
| 2.1 | 0.832800000000000 | 0.829875518672199 |
| 2.2 | 0.708036878443888 | 0.704225352112676 |
| 2.3 | 0.611802381778826 | 0.607902735562310 |
| 2.4 | 0.535592749372665 | 0.531914893617021 |
| 2.5 | 0.473938067466517 | 0.470588235294118 |
| 2.6 | 0.423170282558423 | 0.420168067226891 |
| 2.7 | 0.380742913556783 | 0.378071833648393 |
| 2.8 | 0.344835715939071 | 0.342465753424658 |
| 2.9 | 0.314114751637895 | 0.312012480499220 |
| 3.0 | 0.287581256501905 | 0.285714285714286 |

**Example 3.6:** Approximate the *y* value at *x* = 0.4 of the following differential equation:

$$\frac{dy}{dx} = \frac{1}{2}y, \; y(0) = 1 \text{ and } 0 \le x \le 1.$$

using the PC method with h=0.1.

**Solution:**

| $x_i$ | $y_i$ |
|---|---|
| 0 | 1.000000000000000 |
| 0.1 | 1.051250000000000 |
| 0.2 | 1.105126562500000 |
| 0.3 | 1.161764298828125 |
| 0.4 | 1.221304719143066 |

# 3.6 Higher-Order Differential Equations:

Higher-order differential equations involve the higher derivatives $x''(t)$, $x'''(t)$, and so on. They arise in mathematical models for problems in physics and engineering. By solving for the second derivative, we can write a second-order initial value problem in the form:

$$x''(t)=f(t,x(t),x'(t)) \text{ with } x(t_0)=x_0 \text{ and } x'(t_0)=y_0 \qquad (3.31)$$

The second-order differential equation can be reformulated as a system of two first-order equations if we use the substitution:

$$x'(t)=y(t) \qquad (3.32)$$

Then $x''(t)=y'(t)$ and the differential equation in (3.31) becomes a system:

$$\frac{dx}{dt} = y$$

$$\frac{dy}{dt} = f(t,x,y) \qquad with \begin{cases} x(t_0) = x_0 \\ y(t_0) = y_0 \end{cases} \tag{3.33}$$

A numerical procedure such as Rung-Kutta method can be used to solve (3.33) and will generate two sequences $\{x_k\}$ and $\{y_k\}$. The first sequence is the numerical solution to (3.31).

Now, consider RK2 for the system of two differential equation :

$x'(t)=f(t,x,y)$

$y'(t)=g(t,x,y)$

as follows:

$x_{k+1}=x_k+1/2(k_1+k_2)$ , $y_{k+1}=y_k+1/2(p_1+p_2)$

where $k_1=hf(t_k,x_k,y_k)$, $p_1=hg(t_k,x_k,y_k)$

and $k_2=hf(t_k+h,x_k+k_1,y_k+p_1)$, $p_2=hg(t_k+h,x_k+k_1,y_k+p_1)$.

**Example 3.7:** Consider the second-order IVP

$x''(t)+4x'(t)+5x(t)=0$ with $x(0)=3$ and $x'(0)=-5$

(a) Write down the equivalent system of two first-order equation.
(b) Use The RK2 method to solve the reformulated problem over [0,1] using M=5.
(c) Compare the numerical solution with the true solution $x(t)=3e^{-2t}\cos(t)+e^{-2t}\sin(t)$.

First assume $x'(t)=y(t)$ then $x''(t)=y'(t)$ and we have:

$x'(t)=y(t)$

$y'(t)=-4y(t)-5x(t)$ with $x(0)=3$ and $y(0)=-5$, then $h=(1-0)/5=0.2$

| $t_k$ | $x_k$ | $x(t_k)$ |
|---|---|---|
| **0** | **3** | **3** |
| **0.2** | | |
| **0.4** | | |
| **0.6** | | |
| **0.8** | | |
| **1** | | |

**Exercises:**

Solve the system x'=3x-y, y'=4x-y with x(0)=0.2 and y(0)=0.5 using RK2 with h=0.5 in [0,1].

# 3.7 Boundary Value Problems:

Another type of differential equation has the form:

$$x''=f(t,x,x') \quad \text{for } a\le t\le b \tag{3.34}$$

with the boundary conditions

$$x(a)=\alpha \ \text{ and } x(b)=\beta \tag{3.35}$$

This is called ***a boundary value problem (BVP)***.

## <u>Finite-difference Method:</u>

Methods involving difference quotient approximations for derivatives can be used for solving second-order BVP. Consider the linear equation:

$$x''=p(t)x'(t)+q(t)x(t)+r(t) \tag{3.36}$$

over [a,b] with x(a)= $\alpha$ and x(b)= $\beta$. Form a partition of [a,b] using the points $a=t_0<t_1<\dots<t_N=b$, where h=(b-a)/N and $t_j=a+jh$ for j=0,1,…N. The central-difference formulas discussed in chapter two are used to approximate the derivatives:

$$x'(t_j)=\frac{x(t_{j+1})-x(t_{j-1})}{2h}+O(h^2) \tag{3.37}$$

$$x''(t_j)=\frac{x(t_{j+1})-2x(t_j)-x(t_{j-1})}{h^2}+O(h^2) \tag{3.38}$$

To start derivation, we replace each term $x(t_j)$ on the right side of (3.37) and (3.38) with $x_j$ and the resulting equations are substituted into (3.36), to obtain the relation:

$$\frac{x_{j+1}-2x_j+x_{j-1}}{h^2}=p_j\left(\frac{x_{j+1}-x_{j-1}}{2h}\right)+q_jx_j+r_j \tag{3.39}$$

which is used to compute numerical approximation to the differential equation(3.36). This is carried out by multiplying each side of (3.39) by $h^2$ and then collecting terms involving $x_{j-1}$, $x_j$ and $x_{j+1}$ and arranging them in a system of linear equations:

$$\left(\frac{-h}{2}p_j - 1\right)x_{j-1} + \left(2 + h^2 q_j\right)x_j + \left(\frac{h}{2}p_j - 1\right)x_{j+1} = -h^2 r_j \qquad (3.40)$$

for j=1,2,…,N-1, where $x_0 = \alpha \; and \; x_N = \beta$.

**Example 3.8** Solve the boundary value problem

$$x''(t) = \frac{2t}{1+t^2}x'(t) - \frac{2}{1+t^2}x(t) + 1$$

with x(0)=1.25 and x(4)=-0.95 over the interval [0,4] with h=1.

since h=1 we get N=4 and $t_0=0$, $t_1=1$, $t_2=2$, $t_3=3$ and $t_4=4$

In the same way:

$$\frac{x_{j+1} - 2x_j + x_{j-1}}{h^2} = \frac{2t_j}{1+t_j^2}\left(\frac{x_{j+1} - x_{j-1}}{2h}\right) - \frac{2}{1+t_j^2}x_j + 1$$

then, we get:

$$\left(-\frac{h}{2}\frac{2t_j}{1+t_j^2} - 1\right)x_{j-1} + \left(2 - \frac{2h^2}{1+t_j^2}\right)x_j + \left(\frac{h}{2}\frac{2t_j}{1+t_j^2} - 1\right)x_{j+1} = -h^2$$

$$\left(-\frac{ht_j}{1+t_j^2} - 1\right)x_{j-1} + \left(2 - \frac{2h^2}{1+t_j^2}\right)x_j + \left(\frac{ht_j}{1+t_j^2} - 1\right)x_{j+1} = -h^2$$

for j=1,2,3 and $x_0=1.25$, $x_4=-0.95$

so for j=1, we get

$$\left(-\frac{ht_1}{1+t_1^2} - 1\right)x_0 + \left(2 - \frac{2h^2}{1+t_1^2}\right)x_1 + \left(\frac{ht_1}{1+t_1^2} - 1\right)x_2 = -h^2$$

for j=2

$$\left(-\frac{ht_2}{1+t_2^2} - 1\right)x_1 + \left(2 - \frac{2h^2}{1+t_2^2}\right)x_2 + \left(\frac{ht_2}{1+t_2^2} - 1\right)x_3 = -h^2$$

and for j=3

$$\left(-\frac{ht_3}{1+t_3^2} - 1\right)x_2 + \left(2 - \frac{2h^2}{1+t_3^2}\right)x_3 + \left(\frac{ht_3}{1+t_3^2} - 1\right)x_4 = -h^2$$

therefore, we hence the algebraic system of three equations

$$\left.\begin{array}{r}\left(2-\frac{2}{1+1}\right)x_1+\left(\frac{1}{1+1}-1\right)x_2=-1-\left(-\frac{1}{1+1}-1\right)(1.25)\\ \left(-\frac{2}{1+4}-1\right)x_1+\left(2-\frac{2}{1+4}\right)x_2+\left(\frac{2}{1+4}-1\right)x_3=-1\\ \left(-\frac{3}{1+9}-1\right)x_2+\left(2-\frac{2}{1+9}\right)x_3=-1-\left(\frac{3}{1+9}-1\right)(-0.95)\end{array}\right\}$$

$$\left.\begin{array}{r}x_1-\frac{1}{2}x_2=-1+\frac{3}{2}(1.25)\\ -\frac{7}{5}x_1+\frac{8}{5}x_2-\frac{3}{5}x_3=-1\\ -\frac{13}{10}x_2+\frac{18}{10}x_3=-1+\frac{7}{10}(-0.95)\end{array}\right\}$$

then after solving this system, we obtain:

$x_1$=0.52143, $x_2$-0.70714and $x_3$=-1.4357

## Problems:

1. Consider the following first-order ODE:

$$\frac{dy}{dx}=x^2/y \ \ from\ x=0\ to\ x=2.1\ with\ y(0)=2$$

(a) Solve with Euler's explicit method using $h$ = 0.7.
(b) Solve with the modified Euler method using $h$ = 0.7.
(c) Solve with the classical fourth-order Runge-Kutta method using $h$ = 0.7.

The analytical solution of the ODE is $y=\sqrt{\frac{2x^3}{3}+4}$. In each part, calculate the error between the true solution and the numerical solution at the points where the numerical solution is determined.

2. Write the following second-order ODE as a system of two first-order ODEs:

$$\frac{d^2y}{dt^2}+5\left(\frac{dy}{dt}\right)^2-6y+e^{sint}=0$$

3. Consider the following second-order ODE:

$$\frac{d^2y}{dx^2}+x\frac{dy}{dx}+y=2xy \quad for\ 0\le x\le 1\ ,with\ y(0)=1\ and\ y(1)=1$$

Using the difference formulas for approximating the derivatives, discretize the ODE (rewrite the equation in a form suitable for solution with the finite difference method).

# Chapter 4: Numerical Solution of Partial Differential Equations

## 4.1 Classification of Partial Differential Equations:

A partial differential equation (PDE) is an equation that involves an unknown function (the dependent variable) and some of its partial derivatives with respect to two or more independent variables. The classification of PDEs is important for the numerical solution you choose. Consider the general, second-order, linear partial differential equation in two variables :

$$A(x, y)U_{xx} + 2B(x, y)U_{xy} + C(x, y)U_{yy} = F(x,y ,U_x, U_y, U) \quad (4.1)$$

### 4.1.1 Elliptic

$$AC > B^2$$

For example, Laplace's equation:

$$U_{xx} + U_{yy} = 0$$

$A = C = 1, B = 0$

### 4.1.2 Hyperbolic

$$AC < B^2$$

For example the 1-D wave equation:

$$U_{xx} = \frac{1}{c^2} U_{tt}$$

A = 1, C = $\frac{1}{c^2}$, B = 0

### 4.1.3 Parabolic

$$AC = B^2$$

For example, the heat or diffusion Equation

$$U_t = Uxx$$

$A = 1; B = C = 0$

## 4.2 Finite Difference Solution of Partial Differential Equations:

### 4.2.1 Parabolic Equations

Consider the boundary-initial value problem (BIVP):

$$\left.\begin{array}{c} \boldsymbol{u_{xx} = \frac{1}{c} u_t \ , u = u(x,t), 0 < x < 1 \ , t > 0} \\ \boldsymbol{u(0,t) = u(1,t) = 0 \quad (boundary\ conditions)} \\ \boldsymbol{u(x,0) = f(x)\ (initial\ condition)} \end{array}\right\} \quad (4.2)$$

Where c is a constant. This problem represents transient heat conduction in a rod with the ends held at zero temperature and an initial temperature profile *f(x)*.

To solve this problem numerically, we discretize *x* and *t* such that:

$$x_i = i * h \, , i = 0,1,2, \ldots$$
$$t_j = jk \, , j = 0,1,2, ..$$

### 4.2.1.1 Explicit Finite Difference Method

Let $u_{ij}$ be the numerical approximation to $u(x_i \, , t_j)$. We approximate $u_t$ with the forward finite difference:

$$u_t \approx \frac{u_{i,j+1} - u_{i,j}}{k} \qquad (4.3)$$

and $u_{xx}$ with the central finite difference:

$$u_{xx} \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \qquad (4.4)$$

The finite difference approximation to the PDE is then:

$$\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} = \frac{1}{c}\frac{u_{i,j+1} - u_{i,j}}{k} \qquad (4.5)$$

Define the parameter *r* as

$$r = \frac{ck}{h^2}$$

in which case Eq. 4.5 becomes:

$$r\left(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}\right) = \left(u_{i,j+1} - u_{i,j}\right)$$

therefore,

$$u_{i,j+1} = ru_{i+1,j} + (1 - 2r)u_{i,j} + ru_{i-1,j} \qquad (4.6)$$

The domain of the problem and the mesh are illustrated in Fig. 4.1.

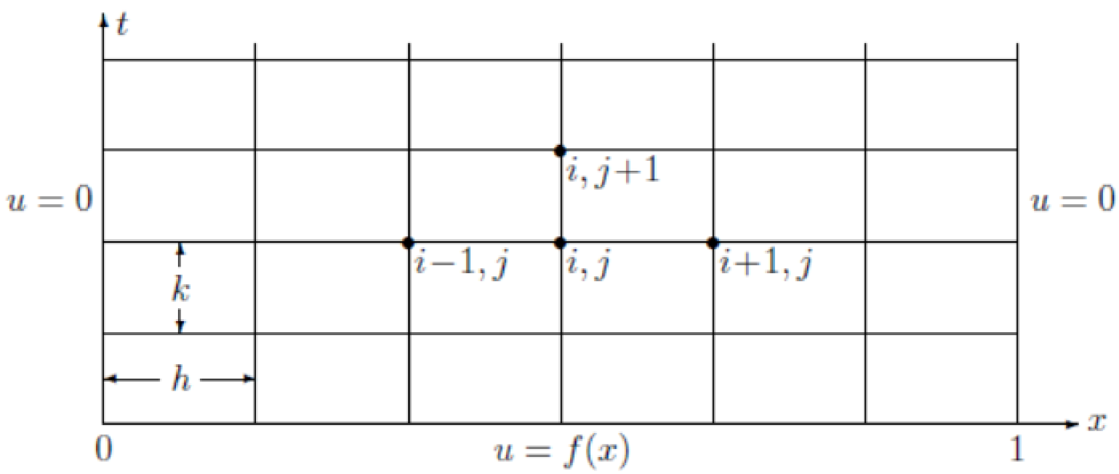


Figure 4.1: Mesh for 1-D Heat Equation.

Eq. 4.6 is a recursive relationship giving u in a given row (time) in terms of three consecutive values of u in the row below (one time step earlier). This equation is referred to as an explicit formula since one unknown value can be found directly in terms of several other known values.

We can write out the matrix system of equations we will solve numerically for the temperature $u$. Suppose we use 5 grid points $x_0$ , $x_1$ , $x_2$ , $x_3$ $and$ $x_4$ .

Now, for *i=1* eq.(4.6) becomes:

$$u_{1,j+1} = ru_{2,j} + (1-2r)u_{1,j} + ru_{0,j}$$

and for *i=2* eq.(4.6) becomes:

$$u_{2,j+1} = ru_{3,j} + (1-2r)u_{2,j} + ru_{1,j}$$

and for *i=3* eq.(4.6) becomes:

$$u_{3,j+1} = ru_{4,j} + (1-2r)u_{3,j} + ru_{2,j}$$

Using boundary condition in eq.(4.2), we get:

$$u_{1,j+1} = ru_{2,j} + (1-2r)u_{1,j}$$
$$u_{2,j+1} = ru_{3,j} + (1-2r)u_{2,j} + ru_{1,j}$$
$$u_{3,j+1} = (1-2r)u_{3,j} + ru_{2,j}$$

Equation above in matrix form becomes:

$$\begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ u_{3,j+1} \end{bmatrix} = \begin{bmatrix} 1-2r & r & 0 \\ r & 1-2r & r \\ 0 & r & 1-2r \end{bmatrix} \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ u_{3,j} \end{bmatrix} \qquad (4.7)$$

where

$$r = \frac{ck}{h^2}$$

Now, for the system of eq's (4.7) substitute j=0,1,2:
for j=0

$$\begin{bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \end{bmatrix} = \begin{bmatrix} 1-2r & r & 0 \\ r & 1-2r & r \\ 0 & r & 1-2r \end{bmatrix} \begin{bmatrix} u_{1,0} \\ u_{2,0} \\ u_{3,0} \end{bmatrix}$$

where $u_{k,0} = u(x_k, 0) = f(x_k)$ (by using initial condition)
for j=1

$$\begin{bmatrix} u_{1,2} \\ u_{2,2} \\ u_{3,2} \end{bmatrix} = \begin{bmatrix} 1-2r & r & 0 \\ r & 1-2r & r \\ 0 & r & 1-2r \end{bmatrix} \begin{bmatrix} u_{1,1} \\ u_{2,1} \\ u_{3,1} \end{bmatrix}$$

for j=2

$$\begin{bmatrix} u_{1,3} \\ u_{2,3} \\ u_{3,3} \end{bmatrix} = \begin{bmatrix} 1-2r & r & 0 \\ r & 1-2r & r \\ 0 & r & 1-2r \end{bmatrix} \begin{bmatrix} u_{1,2} \\ u_{2,2} \\ u_{3,2} \end{bmatrix}$$

# Chapter 5: Numerical Solution of Integral Equations

## 5.1 Classification of Integral Equations:

An integral equation is an equation in which the unknown function u(x) appears under an integral sign. The most general linear integral equation in u(x) can be presented as:

$$h(x)u(x) = f(x) + \int_a^{b(x)} k(x,t)u(t)dt \qquad (5.1)$$

where k(x,t) is a function of two variables called the **kernel** of the integral equation.

This equation is called a ***Volterra integral equation*** when b(x)=x,

$$h(x)u(x) = f(x) + \int_a^{x} k(x,t)u(t)dt \qquad (5.2)$$

when h(x)=0 it is called a ***Volterra equation of the first kind***,

$$-f(x) = \int_a^{x} k(x,t)u(t)dt \qquad (5.3)$$

and is called a ***Volterra equation of the second kind*** when h(x)=1,

$$u(x) = f(x) + \int_a^{x} k(x,t)u(t)dt \qquad \ldots(5.4)$$

The integral equation (5.1) is called a ***Fredholm integral equation*** when b(x)=b, where b constant,

$$h(x)u(x) = f(x) + \int_a^{b} k(x,t)u(t)dt \qquad \ldots(5.5)$$

It is also called a ***Fredholm equation of the first and second kinds*** when h(x)=0 and h(x)=1, respectively:

$$-f(x) = \int_a^{b} k(x,t)u(t)dt \qquad \ldots(5.6)$$

$$u(x) = f(x) + \int_a^{b} k(x,t)u(t)dt \qquad \ldots(5.7)$$

## 5.2 Numerical Solution of Volterra Integral Equations:

Let us consider the Volterra equation of the second kind:

$$u(x) = f(x) + \int_a^{x} k(x,t)u(t)dt$$

we will subdivide the interval of integration (a,x) into n equal subintervals of width $h=(x_n-a)/n$, $n\geq 1$, where $x_n$ is the end point we choose for x, we shall set $t_0=a$ and $t_j=a+jh$. Note that the particular value $u(x_0)=f(a)$, so if we use the trapezoidal rule with n subintervals to approximate the integral in the Volterra integral equation of the second kind (5.4), we have:

$$\int_a^x \boldsymbol{k(x,t)u(t)dt} \approx \frac{h}{2}\begin{bmatrix} \boldsymbol{k(x,t_0)u(t_0)+2k(x,t_1)u(t_1)+\cdots+2k(x,t_{n-1})u(t_{n-1})} \\ \boldsymbol{+k(x,t_n)u(t_n)} \end{bmatrix} \tag{5.8}$$

and the integral equation (5.4) is then approximated by the sum:

$$u(x)=f(x)+\frac{h}{2}\left[k(x,t_0)u(t_0)+2\sum_{j=1}^{n-1}k(x,t_j)u(t_j)+k(x,t_n)u(t_n)\right] \tag{5.9}$$

If we consider n+1 sample values of $u(x)$, $u(x_i), i=0,1,\ldots,n$, equation (5.9) will become a set of n+1 equations in $u(x_i)$ (or $u_i$)[note that $u(x_0)=f(x_0)$ since the integral in (5.4) vanishes for $x=x_0=a$].

$$\left.\begin{array}{c} u_0=f_0 \\ u_i=f_i+\frac{h}{2}\left[k_{i0}u_0+2\sum_{j=1}^{m-1}k_{i,j}u_j+k_{i,m}u_m\right], \\ i=1,2,\ldots,n, k_{ij}=k(x_i,t_j),\ \ j\leq i \end{array}\right\} \tag{5.10}$$

which are n+1 equations in $u_i$, the approximation to the solution $u(x)$ of (5.4) at $x_i=a+ih$ for $i=0,1,\ldots,n$.

**Example 5.1:** Use trapezoidal method to find an approximate values to the solution for the following Volterra integral equation $u(x)=x-\int_0^x(x-t)u(t)dt$ at x=0,1,2,3,and 4.

Here, f(x)=x, k(x,t)=t-x for t≤x=4 and is zero for t>x=4, and a=0 with u(0)=0. We also have n=4 and hence h=(4-0)/4=1. So using (5.10) to obtain:

$u_0=f_0=0$

$u_1=f_1+\frac{h}{2}[k_{10}u_0+k_{11}u_1]=1+\frac{1}{2}[(0-1)(0)+(1-1)u_1]=1$

$u_2=f_2+\frac{h}{2}[k_{20}u_0+2k_{21}u_1+k_{22}u_2]$

$=2+\frac{1}{2}[(0-2)(0)+2(1-2)(1)+(2-2)u_2]=1$

$u_3 = f_3 + \frac{h}{2}[k_{30}u_0 + 2k_{31}u_1 + 2k_{32}u_2 + k_{33}u_3] = 3 + \frac{1}{2}[(0-3)(0) + 2(1-3)(1) + 2(2-3)(1) + (3-3)u_3] = 3 + \frac{1}{2}[-4-2] = 0$

$u_4 = f_4 + \frac{h}{2}[k_{40}u_0 + 2k_{41}u_1 + 2k_{42}u_2 + 2k_{43}u_3 + k_{44}u_4] = 4 + \frac{1}{2}[(0-4)(0) + 2(1-4)(1) + 2(2-4)(1) + 2(3-4)(0) + (4-4)u_4] = 4 + \frac{1}{2}[-6-4] = -1$

| | | | | | |
|---|---|---|---|---|---|
| $x_k$ | 0 | 1 | 2 | 3 | 4 |
| $u_k$ | 0 | 1 | 1 | 0 | -1 |

## *5.3 Numerical Solution of Fredholm Integral Equations:*

Let us consider the Fredholm equation of the second kind:

$$u(x) = f(x) + \int_a^b k(x,t)u(t)dt \tag{5.11}$$

we will subdivide the interval of integration (a,b) into n equal subintervals of width h=(b-a)/n, n≥1, we shall set $t_0$=a,$t_n$=b and $t_j$=a+jh. Note that the particular value , so if we use the trapezoidal rule with n subintervals to approximate the integral in the Fredholm integral equation of the second kind (5.11), we have:

$$\int_a^b \boldsymbol{k(x,t)u(t)dt} \approx \frac{\boldsymbol{h}}{\boldsymbol{2}}\begin{bmatrix} \boldsymbol{k(x,t_0)u(t_0) + 2k(x,t_1)u(t_1) + \cdots + 2k(x,t_{n-1})u(t_{n-1})} \\ \boldsymbol{+k(x,t_n)u(t_n)} \end{bmatrix} \tag{5.12}$$

and the integral equation (5.11) is then approximated by the sum:

$$u(x) = f(x) + \frac{h}{2}\left[k(x,t_0)u(t_0) + 2\sum_{j=1}^{n-1} k(x,t_j)u(t_j) + k(x,t_n)u(t_n)\right] \tag{5.13}$$

If we consider n+1 sample values of *u(x)*, *u($x_i$)*,*i=0,1,…,n*, equation (5.13) will become a set of n+1 equations in *u($x_i$)* (or $u_i$).

$$\left.\begin{array}{l} u_i = f_i + \frac{h}{2}\left[k_{i0}u_0 + 2\sum_{j=1}^{m-1} k_{i,j}u_j + k_{i,m}u_m\right], \\ \quad i = 1,2,\dots,n, k_{ij} = k(x_i,t_j), \;\; j \le i \end{array}\right\} \tag{5.14}$$

which are n+1 equations in $u_i$, the approximation to the solution *u(x)* of (5.11) at $x_i = a+ih$ for *i=0,1,…,n*.

**Example 5.2*:*** Use trapezoidal method to find an approximate values to the solution for the integral equation u(x)=$x^2+\frac{1}{4}-\frac{1}{3}x+\int_0^1(x-t)u(t)dt$ with h=0.25 notice that the real solution is u(x)=$x^2$

We have f(x)= $x^2+\frac{1}{4}-\frac{1}{3}x$ and k(x,t)=x-t.

Since h=0.25, we have $x_0=t_0=0$,$x_1=t_1=0.25$,$x_2=t_2=0.5$, $x_3=t_3=0.75$ and $x_4=t_4=1$

for i=0,1,2,3 and 4, we have:

$$u_0=f_0+\frac{h}{2}[k_{00}u_0+2k_{01}u_1+2k_{02}u_2+2k_{03}u_3+k_{04}u_4]$$

$$u_1=f_1+\frac{h}{2}[k_{10}u_0+2k_{11}u_1+2k_{12}u_2+2k_{13}u_3+k_{14}u_4]$$

$$u_2=f_2+\frac{h}{2}[k_{20}u_0+2k_{21}u_1+2k_{22}u_2+2k_{23}u_3+k_{24}u_4]$$

$$u_3=f_3+\frac{h}{2}[k_{30}u_0+2k_{31}u_1+2k_{32}u_2+2k_{33}u_3+k_{34}u_4]$$

$$u_4=f_4+\frac{h}{2}[k_{40}u_0+2k_{41}u_1+2k_{42}u_2+2k_{43}u_3+k_{44}u_4]$$

therefore, we hence:

$$u_0=0.25+\frac{0.25}{2}[(0-0)u_0+2(0-0.25)u_1+2(0-0.5)u_2+2(0-0.75)u_3+(0-1)u_4]$$

$$u_1=0.22917+\frac{0.25}{2}[(0.25-0)u_0+2(0.25-0.25)u_1+2(0.25-0.5)u_2+2(0.25-0.75)u_3+(0.25-1)u_4]$$

$$u_2=0.33333+\frac{0.25}{2}[(0.5-0)u_0+2(0.5-0.25)u_1+2(0.5-0.5)u_2+2(0.5-0.75)u_3+(0.5-1)u_4]$$

$$u_3 = 0.5625 + \frac{0.25}{2}[(0.75-0)u_0 + 2(0.75-0.25)u_1 + 2(0.75-0.5)u_2 + 2(0.75-0.75)u_3 + (0.75-1)u_4]$$

$$u_4 = 0.91667 + \frac{0.25}{2}[(1-0)u_0 + 2(1-0.25)u_1 + 2(1-0.5)u_2 + 2(1-0.75)u_3 + (1-1)u_4]$$

then,

$8u_0 + 0.5u_1 + u_2 + 1.5u_3 + u_4 = 2$

$-0.25u_0+8u_1+0.5u_2+u_3+0.75u_4=1.8333$

$-0.5u_0-0.5u_1+8u_2+0.5u_3+0.5u_4=2.6667$

$-0.75u_0-u_1-0.5u_2+8u_3+0.25u_4=4.5$

$-u_0-1.5u_1-u_2-0.5u_3+8u_4=7.3333$

solving this system, we get:

$u=[-0.010417 \quad 0.052083 \quad 0.23958 \quad 0.55208 \quad 0.98958]^T$

| $x_k$ | $u_k$ | $u(x_k)$ |
|---|---|---|
| 0 | -0.010417 | 0 |
| 0.25 | 0.052083 | 0.0625 |
| 0.5 | 0.23958 | 0.25 |
| 0.75 | 0.55208 | 0.5625 |
| 1 | 0.98958 | 1 |

## **Exercise:**

1. Use trapezoidal method to find an approximate values to the solution for the integral equation $u(x)=x-\frac{x^3}{3}+\int_0^x tu(t)dt$, $x\in[0,1]$, with h=0.25.(note that u(x)=x)
2. Use trapezoidal method to find an approximate values to the solution for the integral equation $u(x)=e^x - xe^1 + x + \int_0^1 xu(t)dt$, with h=0.5 ( note that $u(x)=e^x$).

# Chapter 6: Eigenvalues and Eigenvectors

**Definition 6.1:** If A is an n×n real matrix, then its n eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the real and complex roots of the characteristic polynomial

$$p(\lambda) = \det(a - \lambda I) \qquad (6.1)$$

**Definition 6.2*:*** If $\lambda$ is an eigenvalue of A and the nonzero vector V has the property that

$$AV = \lambda V \qquad (6.2)$$

then V is called an eigenvector of A corresponding to the eigenvalue $\lambda$.

**Example 6.1*:*** Find the eigenvalues $\lambda_j$ for the matrix

$$A=\begin{bmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{bmatrix}$$

The characteristic equation det(A- $\lambda$I)=0 is

$$\begin{vmatrix} 3-\lambda & -1 & 0 \\ -1 & 2-\lambda & -1 \\ 0 & -1 & 3-\lambda \end{vmatrix} = -\lambda^3 + 8\lambda^2 - 19\lambda + 12 = 0$$

which can be written as $-(\lambda-1)(\lambda-3)(\lambda-4)=0$

Therefore, the eigenvalues are $\lambda_1=1$, $\lambda_2=3$ and $\lambda_3=4$.

## Power Method:

**Definition 6.3*:*** If $\lambda_1$ is an eigenvalue of A that is larger in absolute value than any other eigenvalue, it is called the dominant eigenvalue.

**Definition 6.4*:*** An eigenvector V is said to be normalized if the coordinate of largest magnitude is equal to unity. (i.e. the largest coordinate in the vector V is the number 1).

It is easy to normalize an eigenvector $[v_1\ v_2\ \ldots\ v_n]^T$, by forming a new vector $V=(1/c)[v_1\ v_2\ \ldots\ v_n]^T$, where $c=v_j$ and $v_j=\max_{1\le i\le n}\{|v_i|\}$.

Suppose that the matrix A has a dominant eigenvalues $\lambda$ and that there is a unique normalized eigenvector V that corresponds to $\lambda$. This eigenpair $\lambda$, V can be found by the following iterative procedure called ***power method***. Start with the vector

$$X_0=[1 \ \ 1 \ \ \dots \ \ 1]^T \tag{6.3}$$

Generate the sequence $\{X_k\}$ recursively, using

$$Y_k=AX_k \tag{6.4}$$

$$X_{k+1}=\frac{1}{c_{k+1}}Y_k \tag{6.5}$$

where $c_{k+1}$ is the coordinate of $Y_k$ of largest magnitude. The sequences $\{X_k\}$and $\{c_k\}$ will converge to V and $\lambda$, respectively:

$$\lim_{k\to\infty} X_k = V \qquad and \qquad \lim_{k\to\infty} c_k = \lambda \tag{6.6}$$

**Example 6.2:** Use the power method to find the dominant eigenvalue and eigenvector for the matrix

$$A=\begin{bmatrix} 0 & 11 & -5 \\ -2 & 17 & -7 \\ -4 & 26 & -10 \end{bmatrix}$$

Start $X_0=[1 \ \ 1 \ \ 1]^T$ and use the formulas in (6.4) and (6.5) to generate the sequence of vectors $\{X_k\}$ and constants $\{c_k\}$. The first iteration produces

$$\begin{bmatrix} 0 & 11 & -5 \\ -2 & 17 & -7 \\ -4 & 26 & -10 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 8 \\ 12 \end{bmatrix} = 13\begin{bmatrix} \frac{1}{2} \\ \frac{2}{3} \\ 1 \end{bmatrix} = c_1X_1$$

The second iteration produces

$$\begin{bmatrix} 0 & 11 & -5 \\ -2 & 17 & -7 \\ -4 & 26 & -10 \end{bmatrix}\begin{bmatrix} \frac{1}{2} \\ \frac{2}{3} \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{7}{3} \\ \frac{10}{3} \\ \frac{16}{3} \end{bmatrix} = \frac{16}{3}\begin{bmatrix} \frac{7}{16} \\ \frac{5}{8} \\ 1 \end{bmatrix} = c_2X_2$$

Iteration generate the sequence $\{X_k\}$ (where $X_k$ is a normalized vector):

$$12\begin{bmatrix}\frac{1}{2}\\\frac{2}{3}\\1\end{bmatrix},\frac{16}{3}\begin{bmatrix}\frac{7}{16}\\\frac{5}{8}\\1\end{bmatrix},\frac{9}{2}\begin{bmatrix}\frac{5}{12}\\\frac{11}{18}\\1\end{bmatrix},\frac{38}{9}\begin{bmatrix}\frac{31}{76}\\\frac{23}{38}\\1\end{bmatrix},\frac{78}{19}\begin{bmatrix}\frac{21}{52}\\\frac{47}{78}\\1\end{bmatrix}$$

the sequence of vectors converges to V=$\begin{bmatrix}\frac{2}{3} & \frac{3}{5} & 1\end{bmatrix}^T$, and the sequence of constants converges to $\lambda$=4.

# Exercises:

Find the dominant eigenpair of the following matrices:

$$A=\begin{bmatrix}7 & 6 & -3\\-12 & -20 & 24\\-6 & -12 & 16\end{bmatrix}\quad ,B=\begin{bmatrix}-14 & -30 & 42\\24 & 49 & -66\\12 & 24 & -32\end{bmatrix}$$

(do two iteration).