

# **PRACTICUM MODULE COMPUTATIONAL PHYSICS**



By:  
**Dr. Warsono, M.Si**

**PHYSICS STUDY PROGRAM  
PHYSICS EDUCATION DEPARTMENT  
FACULTY OF MATHEMATICS AND NATURAL SCIENCES,  
YOGYAKARTA STATE UNIVERSITY  
2020**

## FOREWORD

Praise be to Allah swt who has given his grace and guidance so that we can complete the preparation of this Computational Physics Practicum Manual. As is known, Physics is one of the branches of science that plays a central role in understanding the behavior of the universe. With advances in technology and computing, the discipline has evolved rapidly, allowing scientists to explore physical phenomena with a level of detail never before imagined. This practicum module is designed to help you understand, master, and apply computational techniques in the context of physics.

The use of computing in physics not only broadens our insight into the universe, but also has profound applications in areas such as materials science, astronomy, nuclear physics, fluid dynamics, and more. In this module, you will learn how computers can be powerful tools for analyzing experimental data, simulating complicated physical phenomena, and solving differential equations that describe physical systems.

The module consists of a number of labs designed with clear steps, as well as real case examples. By following this module, you will acquire computational skills that you can apply in your own physics research and experiments.

We would like to thank all those who have contributed to the preparation of this module. Happy learning, and may this Computational Physics practicum module help you understand, appreciate, and delve into the fascinating world of computational physics.

Greetings,

Dr. Warsono

# I NUMERICAL DIFFERENTIATION

## 1.1. Destination

After attending this lecture, students are expected to have the ability to solve physical problems related to numerical differentiation through the steps of preparing numerical formulations, preparing algorithms, preparing flow charts and making computer programs.

## 1.2. Numerical Differentiation Formula

Solving physics problems often involves complicated and complex functions that are difficult or even impossible to find their differential values analytically (exact). In addition, experimental data is often not easily solved analytically. Numerical differentiation is one of the alternative methods to solve these problems. There are several ways to obtain numerical differential formulations, including the development (expansion) of the Tylor and Lagrange series. Tylor's formulation for the expansion of the function  $f(x \pm h)$  is expressed by the following series:

$$f(x \pm h) = f(x) \pm hf'(x) + \frac{1}{2}h^2 f''(x) \pm \frac{1}{6}h^3 f'''(x) + \dots \tag{1.1}$$

The development of the function  $f(x)$  into the Lagrange formulation is expressed by:

$$f(x) = \sum_{i=0}^n f(x_i) \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x - x_j)}{(x_i - x_j)} \tag{1.2}$$

where "Π" denotes "product of". For example for  $n = 1$  and  $n = 2$

the solution of  $f(x)$  is :

> For  $n = 1$

$$f(x) = \sum_{i=0}^1 f(x_i) \prod_{\substack{j=0 \\ j \neq i}}^1 \frac{(x - x_j)}{(x_i - x_j)} = f(x_0) \frac{(x - x_1)}{(x_0 - x_1)} + f(x_1) \frac{(x - x_0)}{(x_1 - x_0)} \tag{1.3}$$

> For  $n = 2$

$$f(x) = \sum_{i=0}^2 f(x_i) \prod_{\substack{j=0 \\ j \neq i}}^2 \frac{(x - x_j)}{(x_i - x_j)} = f(x_0) \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + f(x_1) \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}$$

$$+ f(x_2) \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} \quad (1.4)$$

The development of the Taylor series to obtain numerical differential formulas is done by manipulating the coefficients of the series terms then followed by addition or subtraction operations to obtain the desired formulation. The following is an example of deriving a 3-point *Differential-Centered* (DP3) formula

with Taylor series:

$$\begin{aligned}
 f(x+h) &= f(x) + hf'(x) + \frac{h^2}{2!} f''(x) + \frac{h^3}{3!} f^{(3)}(x) + \dots \\
 f(x-h) &= f(x) - hf'(x) + \frac{h^2}{2!} f''(x) - \frac{h^3}{3!} f^{(3)}(x) + \dots
 \end{aligned}$$


---


$$f(x+h) - f(x-h) = 2hf'(x) + \frac{2h^3}{3!} f^{(3)}(x) + \dots \quad (1.5)$$

Rearranging equation (1.5) obtained equation :

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{3!} f^{(3)}(x) + \dots \quad (1.6)$$

The first derivative approximation formula based on equation (1.6) is:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} \quad (1.7)$$

with truncation error term =  $-\frac{h^2}{3!} f^{(3)}(x) + \dots$

*Center-differential* means that the abscissa value for which the derivative will be sought is at the center or

in the center (see Figure 1).

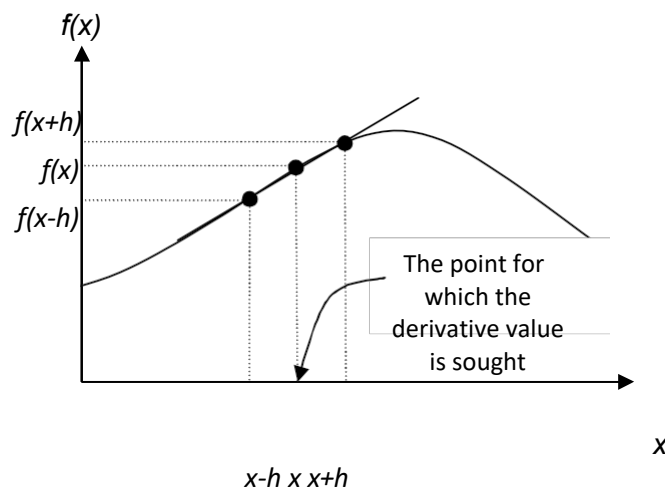


Figure 1. 3-point center-differential

The derivation of numerical differential formulas through the Lagrange expansion method

is done by directly deriving equation (1.2) according to the order of the derivative.

Expansion

Lagrange is usually used to derive forward or backward differential formulas.

Table 1 shows the numerical differential formulas for order one to order four obtained by the development of Tylor series and Lagrange series.

Table 1. Numerical differential formula

a. First Differential	
$f'(x) = \frac{f(x+h) - f(x-h)}{2h}$	Differential Center 3 Point (DP3)
$f'(x) = \frac{f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)}{12h}$	Differential Center 5 Point (DP5)
$f'(x) = \frac{3f(x) - 4f(x-h) + f(x+2h)}{2h}$	3-Point Backward Differential (DB3)
$f'(x) = \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h}$	Differential Forward 3 Point (DD3)
b. Higher-order differential	
$f''(x) = \frac{-f(x-2h) + 16f(x-h) - 30f(x) + 16f(x+h) - f(x+2h)}{12h^2}$	Differential Center 5 Points (DP5)
$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}$	Differential Center 3 Points (DP3)
$f''(x) = \frac{-f(x-2h) + 16f(x-h) - 30f(x) + 16f(x+h) - f(x+2h)}{12h^2}$	Differential Center 5 Points (DP5)
$f''(x) = \frac{-f(x+3h) + 4f(x+2h) - 5f(x+h) + 2f(x)}{2h^2}$	
$f''(x) = \frac{2f(x) - 5f(x-h) + 4f(x-2h) - f(x-3h)}{h^2}$	
$f^{(3)}(x) = \frac{-f(x+3h) + 8f(x+2h) - 13f(x+h) + 13f(x-h) - 8f(x-2h) + f(x-3h)}{8h^3}$	
$f^{(3)}(x) = \frac{-5f(x) + 18f(x+h) - 24f(x+2h) + 14f(x+3h) - 3f(x+4h)}{2h^3}$	
$f^{(3)}(x) = \frac{5f(x) - 18f(x-h) + 24f(x-2h) - 14f(x-3h) + 3f(x-4h)}{2h^3}$	
$f^{(4)}(x) = \frac{-f(x+3h) + 12f(x+2h) - 39f(x+h) + 56f(x) - 39f(x-h) + 12f(x-2h) - f(x-3h)}{6h^4}$	
$f^{(4)}(x) = \frac{3f(x) - 14f(x+h) + 26f(x+2h) - 24f(x+3h) + 11f(x+4h) - 2f(x+5h)}{4h^4}$	
$f^{(4)}(x) = \frac{3f(x) - 14f(x-h) + 26f(x-2h) - 24f(x-3h) + 11f(x-4h) - 2f(x-5h)}{4h^4}$	

### 1.3. Case Example

#### Case 1:

A lightly damped oscillatory motion has the equation of motion :

$$x(t) = 5e^{-0,5t} \cos(4\pi t)$$

with  $x(t)$  the deviation in cm and  $t$  the time in seconds. Determine the speed and acceleration of the oscillator every 0.01 second increment from 0 to 10 seconds. Express the results as graphs of  $v(t)$  vs  $t$  and  $a(t)$  vs  $t$ . Also graph the deviation as a function of time on one screen. Use the 5-point center differential method (DP5).

#### Solution:

➤ Numerical formulation:

Deviation  $\rightarrow x(t) = f(x)$  ; Velocity  $\rightarrow v(t) = f'(x)$  ; Acceleration  $\rightarrow a(t) = f''(x)$

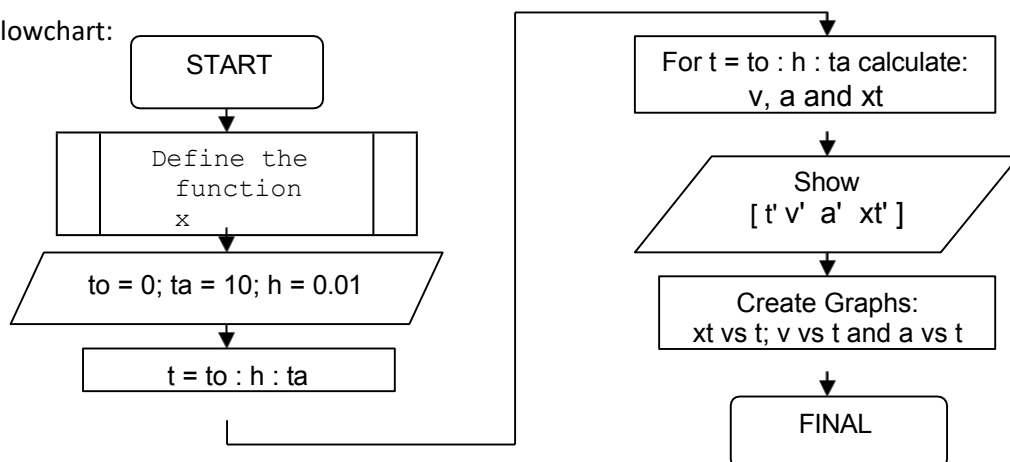
$$v(t) = \frac{x(t \otimes 2h) \otimes 8x(t \otimes h) + 8x(t + h) \otimes x(t \otimes 2h)}{12h}$$

$$a(t) = \frac{-x(t \otimes 2h) + 16x(t \otimes h) \otimes 30x(t) + 16x(t + h) \otimes x(t \otimes 2h)}{12h^2}$$

➤ Algorithm:

- define the function  $x(t)$  with the **inline** command
- give the value of start time  $t_0 = 0$ , end time  $t_a = 10$  and time increment  $h = 0.01$
- give time range value  $t = t_0 : h : t_a$
- for values  $t = t_0 : h : t_a$  calculate:  $v(t)$ ,  $a(t)$  and  $x(t)$
- display the values of  $t$ ,  $v(t)$ ,  $a(t)$  and  $x(t)$  in column matrix form
- graph the relationship between  $x(t)$  vs  $t$ ,  $v(t)$  vs  $t$  and  $a(t)$  vs  $t$  in one screen with the **subplot** command.

➤ Flowchart:

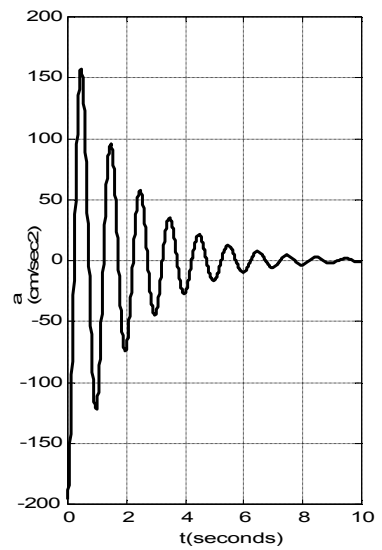
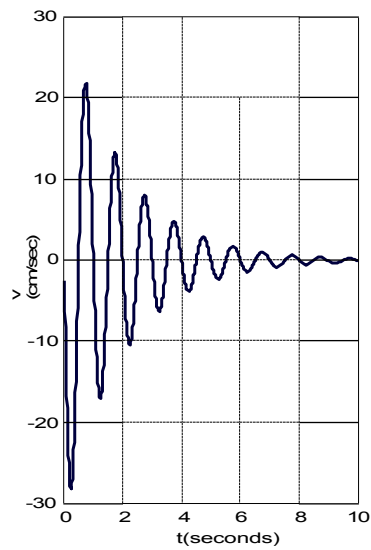
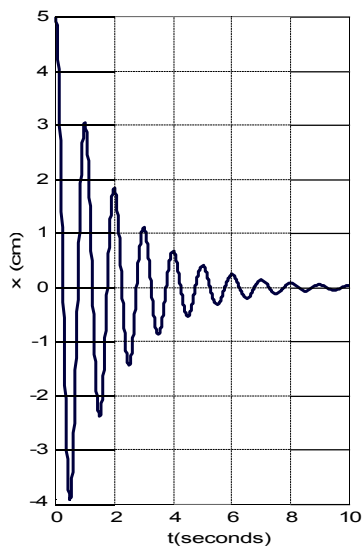




➤ Listing Program:

```
x=inline('5*exp(-0.5*t).*cos(2*pi*t)','t');
t0=0;
ta=10;
h=0.01;
t=t0:h:ta;
v=(x(t-2*h)-8*x(t-h)+8*x(t+h)-x(t+2*h))./(12*h);
a=(-x(t-2*h)+16*x(t-h)-30*x(t)+16*x(t+h)-x(t+2*h))./(12*h.^2);
xt=x(t);
[t' xt' v' a']
subplot(1,3,1),plot(t,xt,'b')
xlabel('t(sec)')
ylabel('x(cm)')
grid on
subplot(1,3,2),plot(t,v,'r')
xlabel('t(sec)')
ylabel('v(cm/sec)')
grid on
subplot(1,3,3),plot(t,a,'k')
xlabel('t(seconds)')
ylabel('a(cm/det2)')
grid on
```

➤ Program Outputs



## Case 2:

Data on the relationship between time  $t$  and position  $x$  of a moving object is expressed in the following table:

$n(\text{data})$	$t$ (seconds)	$x(t)$ in meters
1	0	0
2	0.1	0.4975
3	0.2	0.9806
4	0.3	1.4367
5	0.4	1.8570
6	0.5	2.2361
7	0.6	2.5725
8	0.7	2.8673
9	0.8	3.1235
10	0.9	3.3448
11	1.0	3.5355

Find the velocity and acceleration of the object at time  $t = 0, 0.10, 0.20, \dots, 1.00$  sec.

### Solution:

➤ Numerical formulation:

Position  $\rightarrow x(t) = f(x)$ ; Velocity  $\rightarrow v(t) = f'(x)$ ; Acceleration  $\rightarrow a(t) = f''(x)$

This case cannot be solved with center-differential, forward-differential and backward-differential formulas alone, but can be solved with a combination of forward and backward-differential formulas.

For the velocity solution, formulas DD3 and DB3 are used. DD3 formula to determine the velocity value from  $t=0$  to  $t=0.8$  and DB3 formula to determine the velocity from  $t=0.9$  to  $t=1.0$ .

$$v(t) = \frac{-3x(t) + 4x(t+h) - x(t+2h)}{2h} \rightarrow \text{DD3 for } t = 0 \text{ to } t = 0.8$$

$$v(t) = \frac{3x(t) - 4x(t-h) + x(t-2h)}{2h} \rightarrow \text{DB3 for } t = 1.0 \text{ to } t = 0.9$$

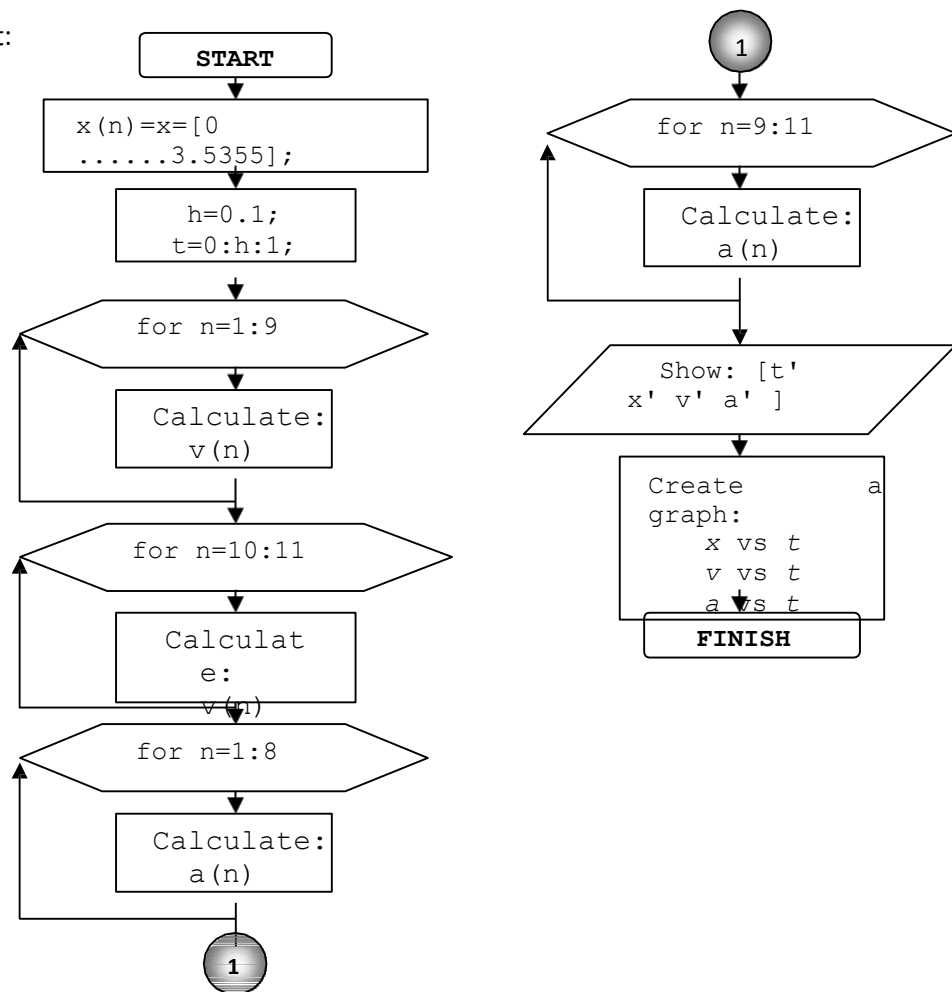
The acceleration solution uses formula DD4 for  $t = 0$  to  $t = 0.7$  and formula DB4 for  $t = 0.8$  to  $t = 1.0$ . The formulation is as follows:

$$a(t) = \frac{-x(t+3h) + 4x(t+2h) - 6x(t+h) + 3x(t)}{h^2} \rightarrow \text{DD40 } t \leq 0.7$$

$$a(t) = \frac{2x(t) - 5x(t-h) + 4x(t-2h) - x(t-3h)}{h^2} \rightarrow \text{DB40 } .8 \leq t \leq 1.0$$

- Algorithm:
- Express the  $x$  data in row matrix form
  - Rate the time interval with the symbol  $h = 0.1$
  - Give a time value from 0 to 1.0 with interval  $h \rightarrow t=0:h:1.0$
  - For  $n = 1: 9$  calculate the speed:
    - $v(n)=(-3*x(n)+4*x(n+1)-x(n+2))./(2*h)$ ;  $\rightarrow$  formula DD3
  - For  $n = 10: 11$  calculate the speed:
    - $v(n)=(3*x(n)-4*x(n-1)+x(n-2))./(2*h)$ ;  $\rightarrow$  DB3 formula
  - For  $n = 1: 8$  calculate the acceleration:
    - $a(n)=(-x(n+3)+4*x(n+2)-5*x(n+1)+2*x(n))./(h^2)$ ;
  - For  $n = 9: 11$  calculate the acceleration:
    - $a(n)=(-x(n-3)+4*x(n-2)-5*x(n-1)+2*x(n))./(h^2)$ ;
  - Display the values of  $t, v, a$  and  $x$  as a column matrix  $\rightarrow [t' x' v' a' ]$
  - Graph the relationship between  $x$  vs  $t, v$  vs  $t$  and  $a$  vs  $t$  in one screen with the **subplot** command.

➤ Flow Chart:



➤ Listing Program

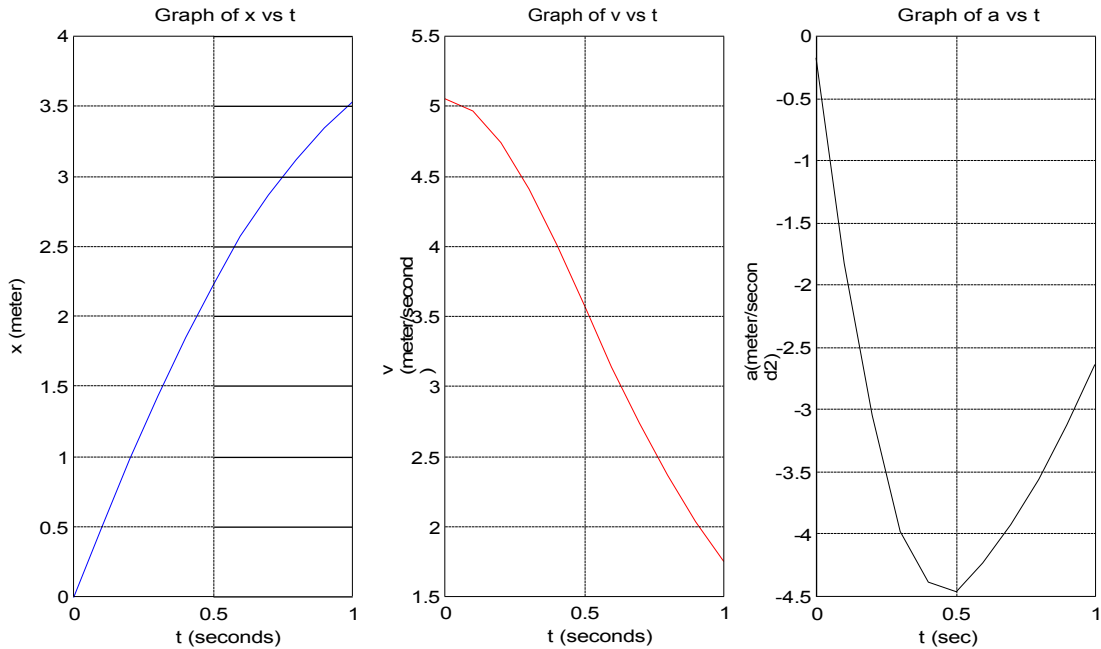
```
x=[0 0.4975 0.9806 1.4367 1.8570 2.2361 2.5725 2.8673 3.1235 3.3448 3.5355];
h=0.1;
t=0:h:1;
for n=1:9
    v(n)=(-3*x(n)+4*x(n+1)-x(n+2))./(2*h);% DD3 formula
end
for n=10:11
    v(n)=(3*x(n)-4*x(n-1)+x(n-2))./(2*h);% DB3 formula
end
for n=1:8
    a(n)=(-x(n+3)+4*x(n+2)-5*x(n+1)+2*x(n))./(h^2);% DD4 formula
end
for n=9:11
    a(n)=(-x(n-3)+4*x(n-2)-5*x(n-1)+2*x(n))./(h^2);% DB4 formula
end
[t' x' v' a' ]
subplot(1,3,1),plot(t,x,'b')
xlabel('t (sec)')
ylabel('x (meters)')
title('Graph of x vs t')
grid on
subplot(1,3,2),plot(t,v,'r')
xlabel('t (seconds)')
ylabel('v (meters/second)')
title('Graph of v vs t')
grid on
subplot(1,3,3),plot(t,a,'k')
xlabel('t (seconds)')
ylabel('a (meters/second2)')
title('Graph of a vs t')
grid on
```

➤ Numbers

Program Output:

t	x	v	a
0	0	5.0470	-0.1800
0.1000	0.4975	4.9660	-1.8200
0.2000	0.9806	4.7400	-3.0400
0.3000	1.4367	4.4090	-3.9700
0.4000	1.8570	4.0045	-4.3800
0.5000	2.2361	3.5720	-4.4600
0.6000	2.5725	3.1410	-4.2300
0.7000	2.8673	2.7365	-3.9200
0.8000	3.1235	2.3660	-3.5600
0.9000	3.3448	2.0385	-3.1200
1.0000	3.5355	1.7540	-2.6300

Graphics:



## 1.4. Practice Questions

1. Prove the first differential formulations DP5, DD3 and DB3 and the second differential formulations DP3, DD4 and DB4 as shown in Table 1 by Taylor series expansion.
2. The motion of a parachutist from the moment he exits the airplane until just before the umbrella expands is called free fall. The equation of motion can be expressed by the equation :

$$d = \frac{1}{c} \left( \frac{mgm}{c} \left( 1 - e^{-\frac{ct}{m}} \right) \right)$$

with  $d$  distance traveled,  $m$  mass of the parachutist,  $c$  coefficient of air friction and  $t$  plunge time. If  $m = 68.1$  kg,  $g = 9.8$  m/sec<sup>2</sup>, and  $c = 12.5$  kg/sec, then:

- a. Create a program to determine the speed and acceleration of the parachutist at 1, 2, 3 ..., 10.
  - b. Graph the relationship between  $d$ ,  $v$  and  $a$  with  $t$  on one window.
3. The voltage  $E = E(t)$  in an electrical circuit has the equation :

$$E(t) = L \frac{dI}{dt} + R I(t)$$

where  $R$  is resistance and  $L$  is inductance. Value of  $R = 2 \Omega$ ,  $L = 0.05 \text{ H}$  and value of  $I(t)$  as in the following table:

$t$ (seconds)	$I(t)$ in Amperes
1,0	8.2277
1,1	7.2428
1,2	5.9908
1,3	4.5266
1,4	2.9122
1,5	1.2146
1,6	-0.4974
1,7	-2.1559
1,8	-3.6962
1,9	-5.0598
2,0	-6.1962

- a. Create a program to determine the values of  $I' = dI/dt$  and  $E$  when  $t = 1.0; 1.1; 1.2; 1.3; \dots 2.0$  seconds with the first differential formulation.
  - b. Graph  $I$ ,  $E$ , and  $dI/dt$  as a function of  $t$ .
4. The data on the measurement of the distance  $d = d(t)$  of an object moving with respect to time is labeled as follows:

$t$ (seconds)	$d(t)$ in meters
1	0.3386
2	1.3111
3	2.8573
4	4.9224
5	7.4571
6	10.4168
7	13.7610
8	17.4530
9	21.4599
10	25.7516

Find its velocity and acceleration at time  $t = 1, 2, 3, \dots 10$  seconds.

### 2.1. Destination

After completing this module, students are expected to have the ability to solve physical problems using numerical integration methods through the steps of preparing numerical formulations, preparing algorithms, preparing flow charts and creating computer programs.

### 2.2. Numerical Integration Formula

Integration means combining parts that are small into a larger whole. Mathematically, function integration is expressed by:

$$I = \int_a^b f(x) dx = F(b) - F(a) \quad (2.1)$$

where  $F(x)$  is the anti-difference of the function  $f(x)$  that satisfies  $dF/dx = f(x)$ . The integrand function in equation (2.1) can be :

- > Simple continuous functions, for example: polynomial, exponential and trigonometric.
- > Complex continuous functions that are difficult or impossible to solve by analytical methods
- > Functions whose values have been tabulated

When analytical methods fail to solve the integration problem of a function, a possible alternative method is the numerical method. The numerical method is a way of calculating the area under the function by dividing the integration interval (between  $a$  and  $b$ ) into smaller bars and then finding the approximate area of each bar formed and summing it up as a whole. Numerical integration formulas are broadly grouped into 2, namely:

- > Classic Formula, which is a formula based on taking an interval (segment) with equal spacing. An example of this formulation is the Newton-Cotes Method (Trapezoidal and Simpson methods)
- > Modern Formula, which is a formula based on interval retrieval (segments) with spaces not necessarily equal. Examples of this formulation are the Romberg Integration Method and Gauss Quadrature.

### 2.2.1. Classic Formulation

The Newton-Cotes method is based on converting the integrand function  $f(x)$  into an  $n$ th-order polynomial function of the form shown in equation (2.2).

$$f_n(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_{n-1}x^{n-1} \quad (2.2)$$

If  $n = 1$  it is called the trapezium method, if  $n = 2$  it is called the Simpson 1/3 method and if  $n = 3$  it is called the Simpson 3/8 method.

#### 2.2.1.1. Single-Segment Trapezoidal Method (TST)

- > The integration interval ( $a$  to  $b$ ) is divided by one segment  $h$  with  $h = b - a$
- > The integrand function is a first-order polynomial
- > The integration value is approximated by the area of the trapezium ( $I \approx \text{area of the trapezium}$ ).

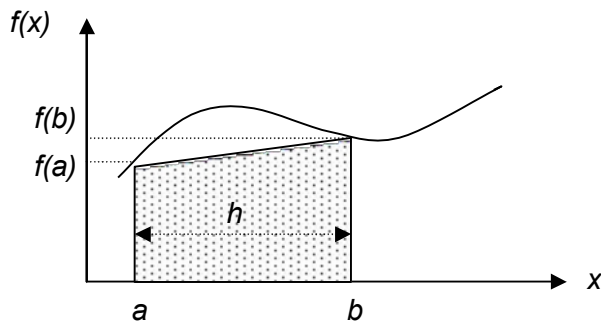
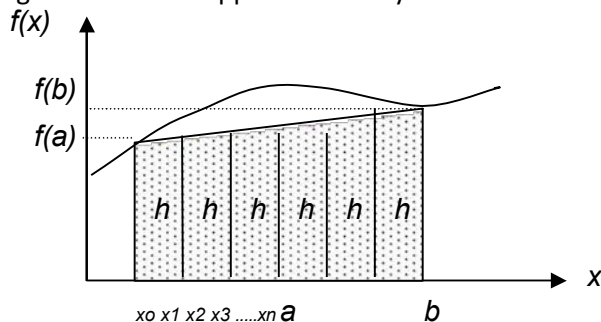


Figure 5. Single-segment Trapezoidal method

$$I = \int_a^b f(x) dx \approx \int_a^b f_1(x) dx = \frac{h}{2} [f(a) + f(b)] \quad (2.3)$$

#### 2.2.1.2. Double-Segmented Trapezoidal Method

- > The integration interval ( $a$  to  $b$ ) is divided into  $n$  segments, with the width of each segment  $h = (b-a)/n$
- > The integrand function is a first-order polynomial
- > The integration value is approximated by the sum of the trapezium areas of each segment





$$\begin{aligned}
I &= \int_a^b f(x) dx \approx \int_{x_0}^{x_1} f_1(x) dx + \int_{x_1}^{x_2} f_1(x) dx + \int_{x_2}^{x_3} f_1(x) dx + \dots + \int_{x_{n-1}}^{x_n} f(x) dx \\
I &= \frac{h}{2} [f(x_0) + f(x_1)] + \frac{h}{2} [f(x_1) + f(x_2)] + \frac{h}{2} [f(x_2) + f(x_3)] + \dots + \frac{h}{2} [f(x_{n-1}) + f(x_n)] \\
I &= \frac{h}{2} \left[ f(x_0) + 2 \sum_{i=1}^{n-1} f(x_i) + f(x_n) \right] \tag{2.4}
\end{aligned}$$

### 2.2.1.3. Simpson 1/3 Single Method (S3T)

- > The integration interval ( $a$  to  $b$ ) is divided by two segments  $h$  with  $h = (b-a)/2$
- > Using 3 abscissa points:  $x_0$ ,  $x_1$  and  $x_2$

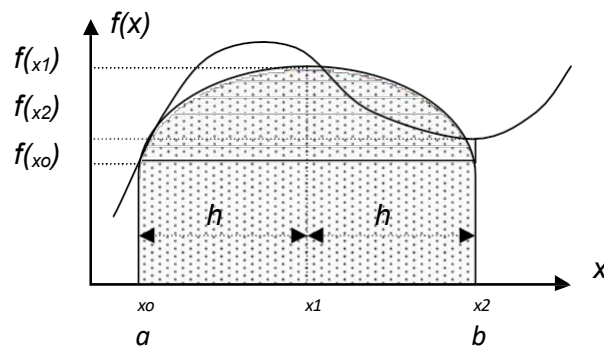


Figure 7. Single 1/3 Simpson method

- > Integrant function is a second-order Lagrange polynomial  $\rightarrow$  quadratic

$$\begin{aligned}
f(x) &= f_2(x) \\
&= \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f(x_0) + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f(x_1) \\
&\quad + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f(x_2) \tag{2.5}
\end{aligned}$$

- > Integration value:

$$I = \int_{x_0}^{x_2} f(x) dx = \int_{x_0}^{x_2} f_2(x) dx = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)] \tag{2.6}$$

### 2.2.1.4. Simpson 1/3 Multiple Segment Method (S3SG)

- > Integration interval ( $a$  to  $b$ ) divided by  $n$  segments with each segment  $h = (b-a)/n$

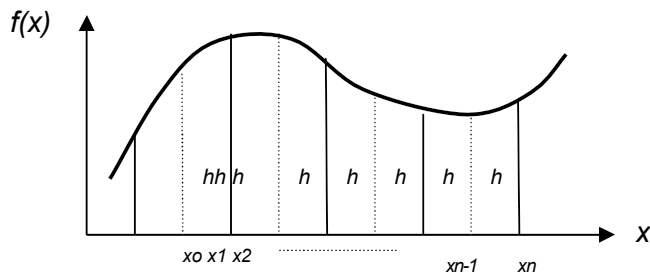


Figure 8. Simpson's method 1/3 multiple segments

- > The integration value is the sum of all segments:

$$\begin{aligned}
 I &= \int_a^b f(x) dx = \int_{x_0}^{x_1} f_2(x) dx + \int_{x_1}^{x_2} f_2(x) dx + \int_{x_2}^{x_3} f_2(x) dx + \dots + \int_{x_{n-2}}^{x_{n-1}} f_2(x) dx + \int_{x_{n-1}}^{x_n} f_2(x) dx \\
 I &= \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)] + \frac{h}{3} [f(x_2) + 4f(x_3) + f(x_4)] \\
 &\quad + \frac{h}{3} [f(x_4) + 4f(x_5) + f(x_6)] + \dots + \frac{h}{3} [f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)] \\
 I &= \frac{h}{3} \left[ f(x_0) + 4 \sum_{\substack{i=1 \\ i=\text{odd}}}^{n-1} f(x_i) + 2 \sum_{\substack{i=2 \\ i=\text{even}}}^{n-2} f(x_i) + f(x_n) \right] \quad (2.7)
 \end{aligned}$$

### 2.2.1.4. Simpson 3/8 Single Method (S8T)

- > Integration interval ( $a$  to  $b$ ) divided by  $n$  segments with each segment  $h = (b-a)/3$
- > Using 4 abscissa points:  $x_0, x_1, x_2$  and  $x_3$

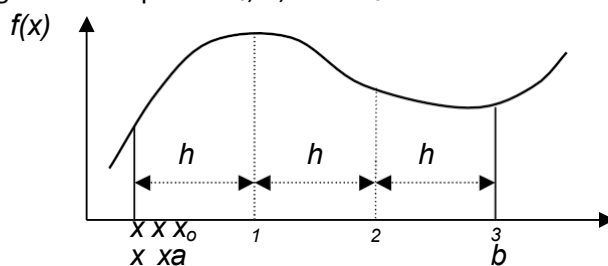


Figure 9. Single 3/8 Simpson method

- > The integrand function is a third-order Lagrange polynomial
- > Integration value:

$$I = \int_{x_0}^{x_3} f(x) dx = \int_{x_0}^{x_3} f_3(x) dx$$

$$I = \frac{3h}{8} [f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)] \quad (2.8)$$

### 2.2.1.5. Simpson Method 3/8 Multiple Segments

- > Integration interval (a to b) divided by n segments with each segment  $h = (b-a)/n$

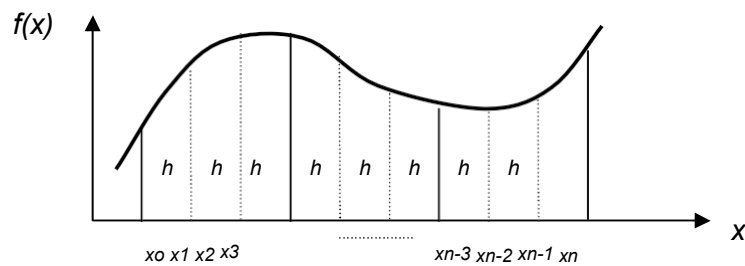


Figure 8. Simpson's method 1/3 multiple segments

- > The integration value is the sum of all segments:

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

$$= \int_{x_0}^{x_3} f_3(x) dx + \int_{x_3}^{x_6} f_3(x) dx + \int_{x_6}^{x_9} f_3(x) dx + \dots + \int_{x_{n-3}}^{x_n} f_3(x) dx$$

$$I = \frac{3h}{8} [f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)] + \frac{3h}{8} [f(x_3) + 3f(x_4) + 3f(x_5) + f(x_6)]$$

$$+ \frac{3h}{8} [f(x_6) + 3f(x_7) + 3f(x_8) + f(x_9)] + \dots$$

$$+ \frac{3h}{8} [f(x_{n-3}) + 3f(x_{n-2}) + 3f(x_{n-1}) + f(x_n)]$$

$$I = \frac{3h}{8} \left[ f(x_0) + 3 \sum_{\substack{i=1 \\ i \neq keli}}^{n-1} f(x_i) + 2 \sum_{\substack{i=3 \\ i=keli}}^{n-1} f(x_i) + f(x_n) \right] \quad (2.9)$$

## Romberg Integration

Romberg integration is based on the successive use of the trapezium rule. The method of obtaining the estimated value of the integral is done by using two other estimated integrals to get a more accurate result. This method is called *Richardson extrapolation*.

The estimates and errors in the multiple segment trapezium method can be expressed generally as follows:

$$I = I(h) + E(h) \quad (2.10)$$

where  $I$  is the exact value of the integral,  $I(h)$  is the approximation value for using  $n$  segments of the trapezium method with  $h = (b-a)/n$ , and  $E(h)$  is the truncation error. If we use two step sizes  $h_1$  and  $h_2$  then the integration value for step size  $h_2$  is :

$$I = I(h_2) + E(h_2) \quad (2.11)$$

with

$$E(h) = \frac{I(h_1) - I(h_2)}{1 - (h_1/h_2)^2} \quad (2.12)$$

which is obtained by combining the use of step size  $h_1$ . Substitution

equation (2.12) to equation (2.11) obtained the estimated value of integration:

$$I \approx I(h_2) + \frac{I(h_1) - I(h_2)}{1 - (h_1/h_2)^2} \quad (2.13)$$

For the case  $h_2 = h_1/2$ , equation (2.13) becomes :

$$I \approx I(h_2) + \frac{1}{2^2 - 1} [I(h_2) - I(h_1)]$$

$$I \approx \frac{4}{3} I(h_2) - \frac{1}{3} I(h_1) \quad (2.14)$$

The general form of Romberg integration is:

$$I = I_{j,k} \approx \frac{4^{k-1} I_{j+1,k-1} - I_{j,k-1}}{4^{k-1} - 1} \quad (2.15)$$

with :

$I_{j,k}$  = corrected integral

$I_{j+1,k-1}$  = better integral

$I_{j,k-1}$  = less accurate integral

$j$  = related to the number of segments;  $j=1$  for 1 segment,  $j=2$  for 2 segments,  $j=3$  for 4 segments, and so on; number of segments =  $2^{j-1}$

$k$  = level of integration;  $k=1$  corresponds to the error for a single segment

Based on equation (2.15), it can be seen that the improved integration value requires two less accurate integration values. The relationship is often called the recursion relationship. The recursive relations of Romberg integration are shown in Table 1 below:

Table 1. Recursive relations of Romberg integration

$j$	$I_{j,1}$	$I_{j,2}$	$I_{j,3}$	$I_{j,4}$	$I_{j,5}$
1	$I_{1,1}$	$I_{1,2}$	$I_{1,3}$	$I_{1,4}$	$I_{1,5}$
2	$I_{2,1}$	$I_{2,2}$	$I_{2,3}$	$I_{2,4}$	
3	$I_{3,1}$	$I_{3,2}$	$I_{3,3}$		
4	$I_{4,1}$	$I_{4,2}$			
5	$I_{5,1}$				

Based on the table, if  $j = 1, k = 2$  which means we determine the value of  $I_{1,2}$  then we need  $I_{1,1}$  and  $I_{2,1}$  so that the integral value is :

$$I_{1,2} \approx \frac{4 I_{2,1} - I_{1,1}}{4 - 1} = \frac{4 I_{2,1} - I_{1,1}}{3} \quad (2.16)$$

The program algorithm for the Romber integration method is as follows:

```

FUNCTION Romberg (a, b, maxit, es)
  LOCAL I(10, 10)
  n = 1
  I1,1 = TrapEq(n, a, b)
  iter = 0
  DO
    iter = iter + 1
    n = 2iter
    Iiter+1,1 = TrapEq(n, a, b)
    DOFOR k = 2, iter + 1
      j = 2 + iter - k
      Ij,k = (4k-1 * Ij+1,k-1 - Ij,k-1) / (4k-1 - 1)
    END DO
    ea = ABS((I1,iter+1 - I2,iter) / I1,iter+1) * 100
    IF (iter ≥ maxit OR ea ≤ es) EXIT
  END DO
  Romberg = I1,iter+1
END Romberg

```

## Gauss quadrature

Gauss quadrature is an integral estimation technique by taking basis points freely so that the estimated value of the integral becomes better. This technique is clearly different from the Newton-Cotes formulation which takes certain and fixed basis points. An illustration of the difference between the two techniques is shown in Figure 9.

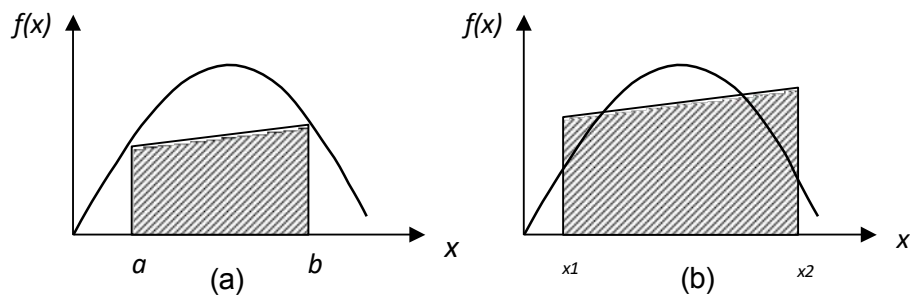


Figure 9. (a). Trapezium rule using the Newton-Cotes method.  
(b). Trapezium rule using the Gauss Quadrature method

In this section, the type of Gauss Quadrature that will be discussed is the *Gauss-Legendre* formula derived using the indeterminate coefficient method.

### a. Gauss-Legendre Two-Point Formula

In the single-segment trapezium rule in the Newton-Cotes formula, the integration value is expressed by :

$$I \approx \frac{(b-a)}{2} f(a) + \frac{(b-a)}{2} f(b) \quad (2.17)$$

In Gauss Quadrature, the value of integration is expressed by an equation of the form:

$$I \approx c_1 f(x_1) + c_2 f(x_2) \quad (2.18)$$

where  $c_1$  and  $c_2$  are unknown coefficients. The points  $x_1$  and  $x_2$  are non-fixed points so the function value still has to be found. Therefore, in equation (2.18) there are four unknown values that must be found. To determine these values, four boundary conditions are required.

The steps to determine the value of the unknown coefficient are as follows:

- © Let equation (2.18) be suitable for the integral of a constant and a linear function so that we can obtain two conditions.

© Suppose equation (2.18) is also suitable for the integral of a parabolic function ( $f(x) = x^2$ ) and a cubic function ( $f(x) = x^3$ ) so that we can obtain two conditions others.

With this assumption, we can determine the four unknown values. The four equations to determine these values are:

$$c_1 f(x_1) + c_2 f(x_2) = \int_{-1}^1 1 dx = 2 \quad (2.19.a)$$

$$c_1 f(x_1) + c_2 f(x_2) = \int_{-1}^1 x dx = 0 \quad (2.19.b)$$

$$c_1 f(x_1) + c_2 f(x_2) = \int_{-1}^1 x^2 dx = 2 \quad (2.19.c)$$

$$c_1 f(x_1) + c_2 f(x_2) = \int_{-1}^1 x^3 dx = 0 \quad (2.19.d)$$

The solution to equations (2.19.a) to (2.22.d) for  $c_1 = c_2 = 1$  is :

$$x_1 = \frac{1}{\sqrt{3}} = -0,577350269 ..$$

$$x_2 = \frac{1}{\sqrt{3}} = 0,577350269 ..$$

If the four values are substituted into equation (2.17), the two-point Gauss-Legendre formulation will be obtained as follows:

$$I \approx f\left(\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right) \quad (2.23)$$

The use of integration limits from -1 to 1 in equations (2.19.a) to (2.19.d) is to ease the arithmetic and make the formulation as general as possible. What if the limits of integration are from  $a$  to  $b$ ? We use a new variable  $x_k$  that has the following relationship with the variable  $x$ :

$$x = a_0 + a_1 x_k \quad (2.24)$$

© If the lower bound  $x = a$  corresponds to  $x_k = -1$  then equation (2.24) becomes

$$a = a_0 - a_1 \quad (2.25)$$

© If the upper bound  $x = b$  corresponds to  $x_k = 1$  then equation (2.24) becomes :

$$b = a_0 + a_1 \quad (2.26)$$

The simultaneous solution of equations (2.25) and (2.26) is :

$$a_0 = \frac{b+a}{2} \quad (2.27)$$

$$a_1 = \frac{b-a}{2} \quad (2.28)$$

If equations (2.27) and (2.28) are substituted into equation (2.24) then we get :

$$x = \frac{b+a}{2} + \frac{b-a}{2} x_k \quad (2.29)$$

The derivative of equation (2.29) is :

$$dx = \frac{b-a}{2} dx_k \quad (2.30)$$

If equations (2.29) and (2.30) are inserted into the function integration equation at the boundary  $a$  to  $b$  then obtained :

$$\begin{aligned} \int_a^b f(x) dx &= \int_{-1}^1 f\left(\frac{b+a}{2} + \frac{b-a}{2} x_k\right) \left(\frac{b-a}{2}\right) dx_k \\ &= \left(\frac{b-a}{2}\right) \int_{-1}^1 f\left(\frac{b+a}{2} + \frac{b-a}{2} x_k\right) dx_k \\ &= \frac{b-a}{2} \left[ f\left(\frac{b+a}{2} + \frac{b-a}{2} x_1\right) + f\left(\frac{b+a}{2} + \frac{b-a}{2} x_2\right) + \dots + f\left(\frac{b+a}{2} + \frac{b-a}{2} x_n\right) \right] \end{aligned} \quad (2.31)$$

where  $n$  is the number of points used. For  $n = 2$ ,  $c_1 = c_2 = 1$  so the equation becomes:

$$\int_a^b f(x) dx = \frac{b-a}{2} \left[ f\left(\frac{b+a}{2} + \frac{b-a}{2} x_1\right) + f\left(\frac{b+a}{2} + \frac{b-a}{2} x_2\right) \right] \quad (2.32)$$

### PRACTICE QUESTIONS

1. Use the multiple segment trapezium rule to determine the estimated value of the following integration:

$$\int_0^2 \sqrt{4x + x^2} dx$$

Use the rule with the number of segments  $n = 4$ .

2. Numerical integration with Simpson's rule 3/8 multiple segments is expressed with the following formulation:

$$I = \frac{3h}{8} \left[ f(x_0) + 3 \sum_{\substack{i=1 \\ i \neq k \text{ or } l \\ \text{pa tan } 3}}^{n-1} f(x_i) + 2 \sum_{\substack{i=1 \\ i \neq k \text{ or } l \\ \text{pa tan } 3}}^{n-1} f(x_i) + f(x_n) \right]$$



Make it:

- Algorithm
- Flow chart
- Use the rule to calculate the integral value of :

$$\int_0^1 \frac{1}{1+x} dx \quad \text{with}$$

number of segments  $n = 4$ .

- Find the integration value of the following function:

$$I = \int_0^1 \sqrt{1+9x} dx$$

with integration:

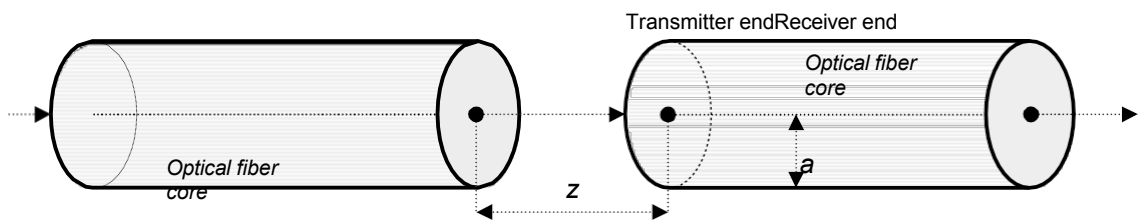
- Third-order Romberg method ( $I_{1,4}$ )
- Three-point Gauss-Legendre method

- The optical power efficiency at longitudinal splitting of multiform ripple index optical fiber is expressed by the following integration equation:

$$\eta = \frac{4}{\pi} \frac{NA^2}{\omega} \int_0^{\omega} \frac{\omega x}{(2)^2} \sqrt{\frac{1}{(2)^2} \left( \frac{\omega x}{2} \right)^2} \frac{\omega}{(1+\omega^2)^2} d\omega$$

with  $\omega = \frac{z}{\sqrt{1+NA^2}}$ ,  $x = z/a$ ,  $NA$  numerical aperture,  $z$  distance

longitudinal separation and  $a$  the radius of the optical fiber.



Create a program using the multi-segment trapezoidal integration method to calculate the value of optical power efficiency under the conditions:

- Program input:  $NA = 0.1$  ( $0 < NA < 1$ )
- Value of  $x$  from 0 to 50 with step 0.5 ( $x = 0:0.5:50$ )
- Value  $h = 0.01$
- Program output:
  - Display of  $\eta$  values for all  $x$  values
  - Relationship graph between efficiency  $\eta$  and  $x$

## PROGRAM ALGORITHM

- Define the integrand function:  
 $f = \text{inline}('(\text{acos}(0.5*x.*y)-0.5*x.*y.*\text{sqrt}(1-0.5*x.*y).^2)).*(y./(1+y.^2).^2)', 'x', 'y');$
- Input the value  $na$  ( $0 < na < 1$ ) from the keyboard
- Give the value of the lower limit of integration:  $w_0 = 0$ ;
- Calculate the upper limit of integration:  $w = na/\text{sqrt}(1-na^2)$
- Give index value = 0
- For  $j = 0$  to 50 with a step of 0.5:
  - $x = j$
  - index = index + 1
  - give step value  $h = 0.01$
  - Calculate the function value at the lower boundary:  $f_{w_0} = f(x, w_0)$ ;
  - Calculate the function value at the upper boundary:  $f_w = f(x, w)$ ;
  - Calculate the number of segments:  $n = (w - w_0)/h$ ;
  - Initialize the middle term with zero:  $st = 0$
  - For  $i$  from 0 to  $n-1$ :
    - ▶ Calculate the value of the middle term:  $st = st + f(x, w_0 + i*h)$ ;
  - Calculate the integration value with the trapezoidal method:  
 $it(\text{index}) = (4/(\pi*na^2))*0.5*h*(f_{w_0} + 2*st + f_w)$ ;
- Display the integration value as a column matrix:  $[it']$
- For  $j=0:0.5:50$ , draw the graph of it as a function of  $j=0:0.5:50$ ;  
 $\text{plox}(j, it, 'b')$   
 $\text{grid on}$

5. The wave function in Fresnel diffraction is expressed in the form of the following equation:

$$E_{\text{F}} = \frac{i\alpha}{2D} e^{-ikD} [C(x) \boxtimes iS(x)][C(y) \boxtimes iS(x)]$$

with  $C$  and  $S$  are the Fresnel integrals expressed as follows:

$$C(\omega) = \int_0^{\omega} \cos \frac{\pi u^2}{2} du$$

$$S(\omega) = \int_0^{\omega} \sin \frac{\pi u^2}{2} du$$

The variable  $\omega$  is the end coordinate of the casement and  $u$  is a dummy variable.

Make a program with Simpson 1/3 and 3/8 method for :

a. Calculate the values of  $C(\omega)$  and  $S(\omega)$  as follows:

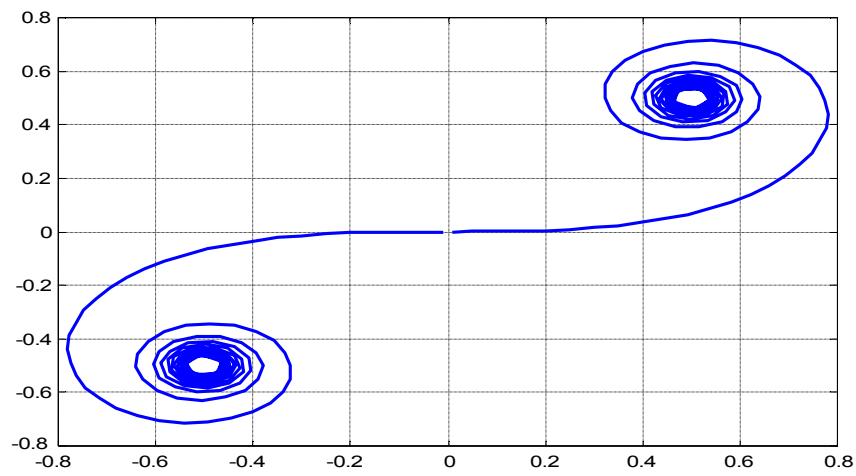
- Value of  $h = 0.01$ ;

➤  $\omega$  value from 0 to 8 with step 0.1  $\rightarrow \omega = 0:0.1:8$

- b. Create graph  $C(\omega)$  versus  $S(\omega)$  for region positive and negative regions

$\rightarrow \text{plot}(C,S,-C,-S)$

An illustration of the program results for the graph of  $C(\omega)$  versus  $S(\omega)$  is shown in the following figure:



## III DETERMINING THE ROOTS OF AN

## EQUATION

## PURPOSE

After completing this module, students are expected to be able to use the method of determining the roots of an equation to solve related physical problems through the steps of preparing numerical formulations, preparing algorithms, preparing flow charts and creating computer programs.

## THEORY

The root of an equation is basically the value of a variable that makes the equation zero. Suppose the equation is  $y = f(x)$ , if  $x$  causes  $y = 0$  then  $x$  is called the root of the equation. Numerical methods to determine the root of an equation can be grouped into two, namely closed methods and open methods. Closed methods use initial estimates that confine the root value, for example: bisection method and false position method (regula falsi). Open methods do not require an initial estimate that brackets the true root value. Examples of open methods are: simple one-point iteration method, Newton-Raphson method and Secant method.

**Bisection Method**

The divide-by-two method is a method of finding the roots of a function that begins with the placement of an interval with each interval always divided by two. If the function values at the interval boundaries differ in sign, then the function value in the middle is evaluated. The location of the root is determined to be in the middle of the subinterval. The process is repeated until an adequate estimate is obtained. The outline of the Bagidua method is shown in the following algorithm:

- *Step 1:* Choose a lower estimate  $x_b$  and an upper estimate  $x_a$  ( $x_b < x_a$ ) for the function value to change sign along the interval  $\Rightarrow f(x_b) \cdot f(x_a) < 0$ ;

➤ *Step 2:* First estimate  $x_1 = (x_0 + b)/2$ ;

- > **Step 3:** Evaluate  $f(x_b)f(x_r)$ :
  - a. If  $< 0$ , then root in the first subinterval  $\Rightarrow x_a = x_r$ ; go to step 4
  - b. If  $> 0$ , then root in the second subinterval  $\Rightarrow x_b = x_r$ ; go to step 4
  - c. If  $= 0$ , then root  $= x_r \Rightarrow$  stop computing
- > **Step 4:** Calculate the new estimate:  $x_r = (x_b + x_a)/2$ ;
- > **Step 5:** Decide whether the new estimate is accurate enough. If "yes"  $\Rightarrow$  stop the computation, if "no"  $\Rightarrow$  return to step 3.

An illustration of the Bagidua method can be depicted in Figure 3.

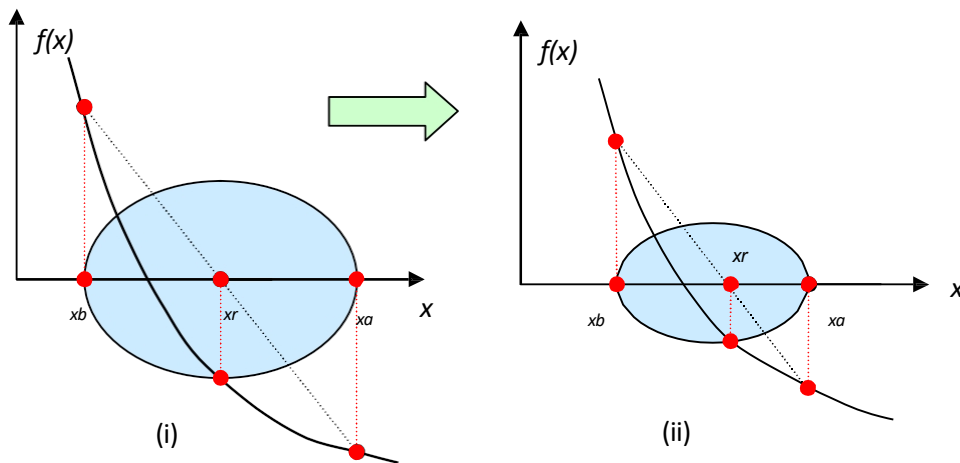


Figure 3.1. (i). Determination of the estimated root based on steps 1 and 2  
 (ii). Determination of the new estimated roots that satisfy steps 3.a. and 4

The estimated error (error) of the Bagidua method can be expressed by the following formulation:

$$\varepsilon_a = \left| \frac{x' - x}{x_r} \right| \times 100 \% \quad (3.1)$$

with  $x_r$  = old root estimate and  $x_r'$  = new root estimate.

**Example 3.1:**

Write a program to calculate the approximate value of the roots of the function  $f(x) = x^2 - 2$  by the Bagidua method using a lower estimate of 0 and an upper estimate of 2.

### False Position Method (Regula Falsi)

The false position method is done by joining points whose function values differ in sign with a straight line. The estimated root value lies at the intersection of the line with the horizontal axis ( $x$ -axis). An illustration of the false position method is shown in Figure 4.

© The formulation of the False Position (Regula Falsi) method is expressed by the equation as The following:

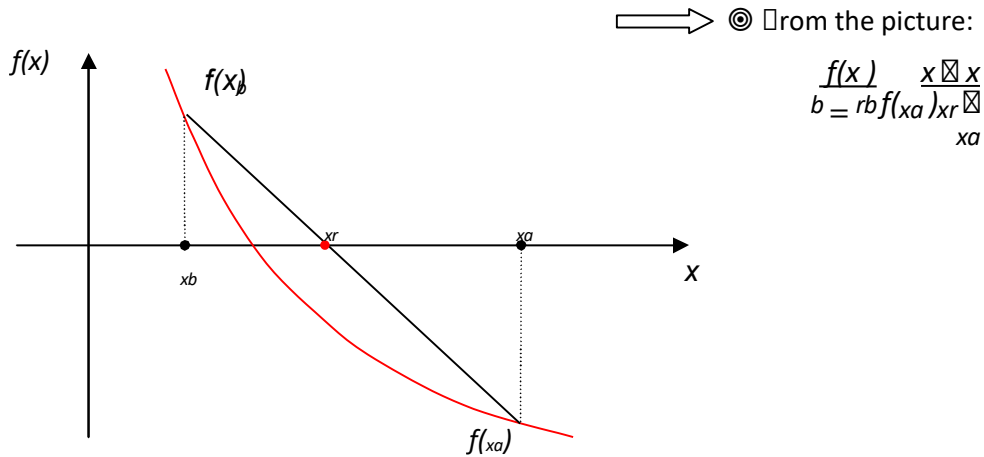


Figure 3.2. Illustration of the False Position Method

$$x_r = x_a - \frac{(x_a - x_b)f(x_a)}{f(x_a) - f(x_b)} \quad (3.2)$$

#### **Example 3. 2:**

Solve **Example 3.1** with the false position method!

### 3. Simple One-Point Iteration Method

The way to determine the roots of a function using the simple one-point iteration method is by rearranging the function  $f(x) = 0$  such that  $x$  is in the left-hand side of the equation. This value of  $x$  is then expressed as the new function  $g(x)$ . For example, the function  $f(x) = \exp(-x) - x$  can be transformed into :

$$f(x) = \exp(-x) - x = 0 \quad \longrightarrow \quad x = \exp(-x) = g(x) \quad (3.3)$$

If  $x$  is given an initial guess value  $x_i$ , then it can be used to determine a new guess value

$x_{i+1}$ .

The estimated equation can be expressed as:

$$x_{i+1} = g(x_i) \quad (3.4)$$

If the estimated root value is  $x_r$ , then the iteration can be stopped if the value of  $x_r - x_{i+1} = g(x_r) - g(x_i)$  is less than the error tolerance value. The error tolerance value can be determined as small as possible so that the estimated root value is close to the true value. The flowchart for determining the root value by the simple one-point iteration method of the function  $f(x) = \exp(-x) - x$  is shown in Figure 5.

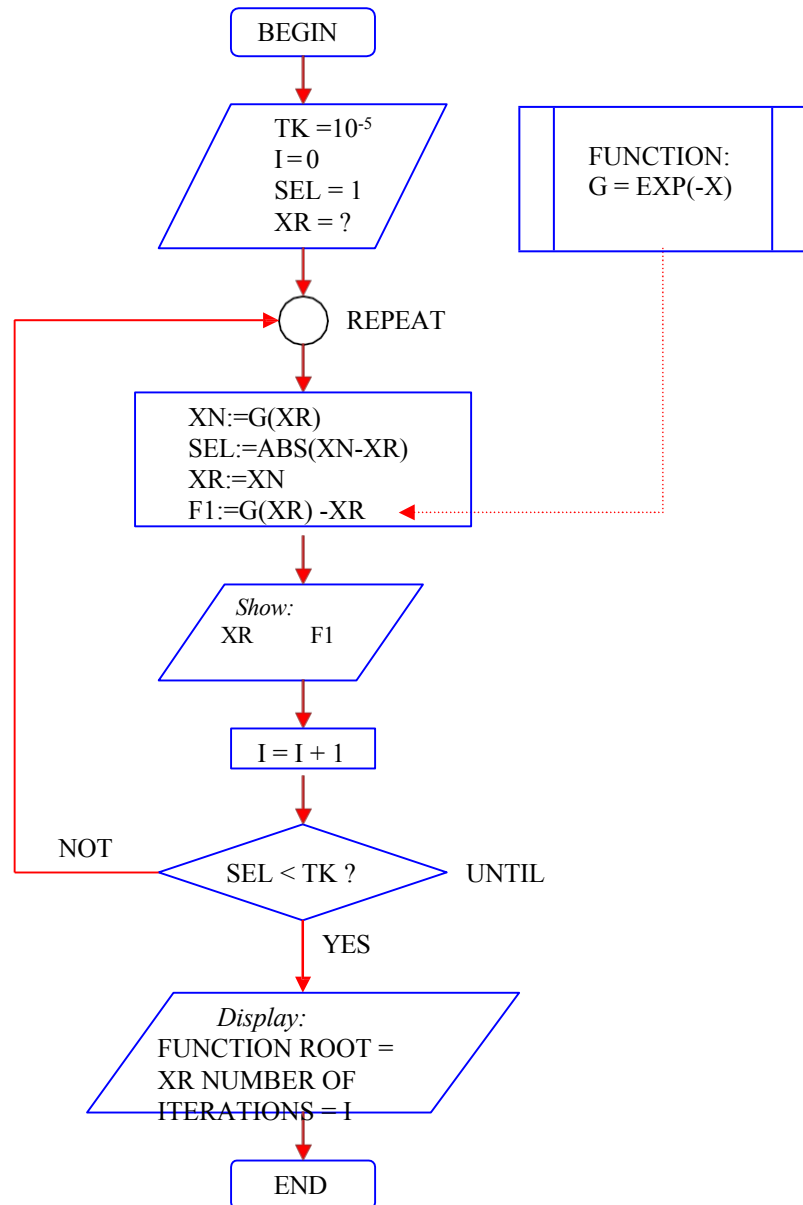


Figure 3.3. Flowchart of Simple One-point Iteration Method

**Example 3.3** roots of the function  $f(x) = e^{-x} - x$  by the Simple One-Point Iteration method



#### 4. Newton-Raphson Method

The Newton-Raphson method is a root determination technique based on the intersection of the tangent line made at the initial estimated value with the  $x$ -axis (horizontal), see Figure 3.4.

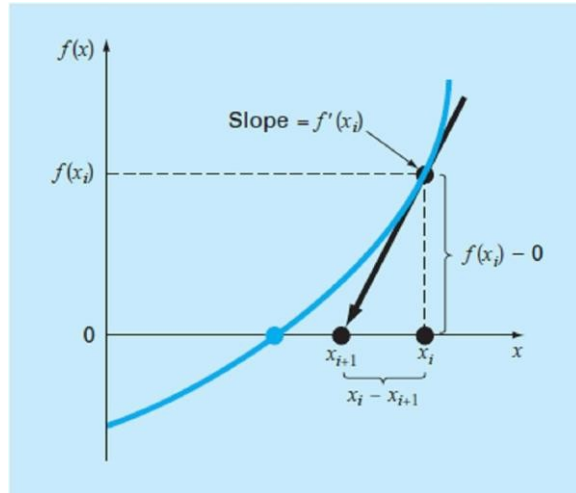


Figure 3.4. Newton Rapson Method

Determining the roots of a function with the Newton-Raphson method can be done through the first derivative of  $x$  which is equal to the *slope of the tangent line*:

$$f'(x_i) = \frac{f(x_i) - 0}{x_i - x_{i+1}} \quad (3.5)$$

which can be rearranged into iteration form as follows:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad (3.6)$$

The iteration step starts from the initial guess  $x_0$ , then  $x_1$ ,  $x_2$ ,  $x_3$  and so on are calculated. Iteration can be stopped until the value of  $x$  is close to the true value, that is, if the absolute value of  $(x_{i+1} - x_i)/x_{i+1}$  is less than or equal to a certain tolerance value.

**Example 3.4:** Find the roots of the function  $f(x) = x^2 - 2$  with initial estimate 1.

#### 5. Secant Method

The Secant method is an improvement of the Newton-Raphson method by replacing the first derivative of  $f(x)$  with a finite difference as follows:

$$f'(x) \approx \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}} \quad (3.7)$$

This replacement aims to avoid the difficulty of decreasing the function if the function is not familiar. Furthermore, if equation (3.7) is substituted to equation (3.6), the formulation of Secant method is obtained as follows:

$$x_{i+1} = x_i - \frac{(x_i - x_{i-1})f(x_i)}{f(x_i) - f(x_{i-1})} \quad (3.8)$$

### PRACTICE QUESTIONS

1. A function is expressed by the equation :

$$f(x) = \frac{(1 - 0.6x)}{x}$$

Find the roots of the function by *Bisection* and *False Position (Regula Falsi)* methods. Take the lower estimated value  $x_b = 1$  and upper  $x_a = 3$ .

2. The equation of parabolic motion of an object is expressed as :

$$y = x - 0.1x^2$$

with  $x$  horizontal position and  $y$  vertical position. Determine the farthest distance reached by the object using the *Simple One-Point Iteration* and *Newton-Raphson* methods. Take the initial estimated value of 8.

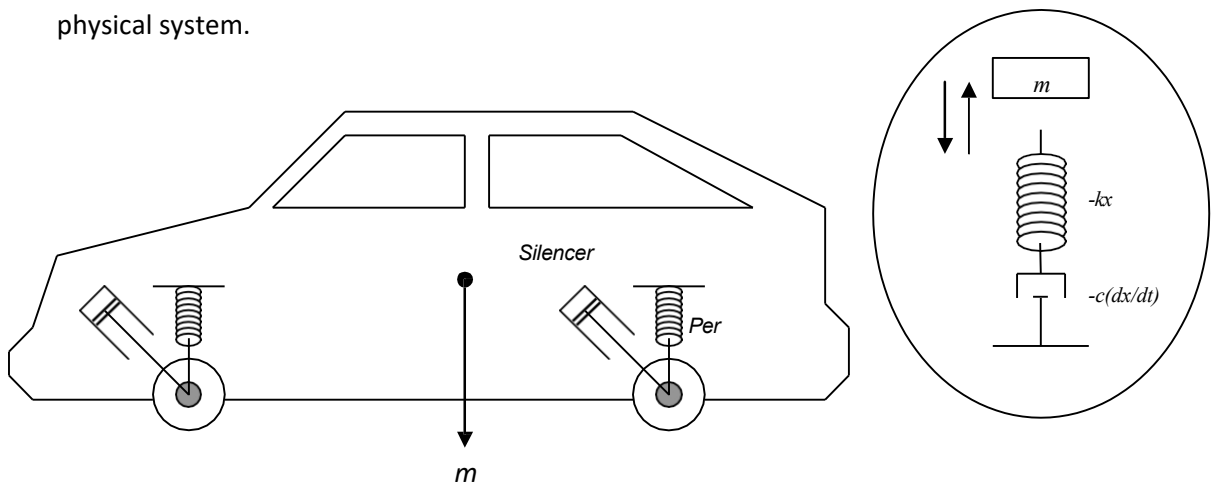
3. Find the real root of the following equation using the *Secant* method:

$$f(x) = x^3 - 6x^2 + 11x - 6$$

Take the initial estimates of 2.5 and 3.6.

4. The suspension system of a car can be approximated by a damped oscillation system.

The figure below shows the suspension system of a car and an approximation of the physical system.



The equation of motion of the system is expressed by a second-order differential equation as follows:

$$\frac{d^2x}{dt^2} + \frac{c}{m} \frac{dx}{dt} + \frac{k}{m} x = 0$$

$$+ \frac{d^2x}{dt^2} \gamma + \omega_0 x = 0$$

The ~~xxxxxx~~ equation is :  $x(t) = \left( \cos(pt) + \frac{\gamma}{p} \sin(pt) \right)$

with :  $\gamma = \frac{c}{2m}$  ;  $p = \sqrt{\omega_0^2 - \gamma^2}$  and  $\omega_0^2 > \gamma^2$

**Issues:**

Create a program using the *Secant* method to calculate the approximate value of the time when the car passes the first 3 equilibrium points. Given  $c = 14000 \text{ kg/s}$ ,  $m = 1200 \text{ kg}$ ,  $k = 1250000 \text{ kg/det}^2$  and  $x_0 = 3 \text{ cm}$ . Graph the relationship between  $x(t)$  with  $t$ .

- The relationship between pressure, volume and temperature in real gases is usually expressed by the van der Waals equation which has the form:

$$\left( p + \frac{a}{v^2} \right) (v - b) = RT$$

where  $p$  is the gas pressure,  $v = V/n$  is the molar volume of the gas,  $a$  and  $b$  are constants.

**Issues:**

Write a program using the Newton-Raphson method to calculate the molar volume of oxygen gas ( $v$ ) at gas pressures of 10, 20, 30 ... 100 atm and temperatures of 300° K, if the constants  $a = 1.360$ ;  $b = 0.03183$  and  $R = 0.082054 \text{ lt.atm}/(\text{mol.K})$  are known.

**Settlement Instructions:**

Transform the van der Waals equation into the form :

$$f(v) = \left( p + \frac{a}{v^2} \right) (v - b) - RT$$

$$f'(v) = p - \frac{2a}{v^3} - b$$

$$= \frac{1}{v^2} + \frac{1}{v^3}$$

Use the functions above to estimate the value of  $v$  using the known values of variables and constants.

# CHAPTER IV ORDINARY DIFFERENTIAL EQUATIONS

## (PDB)

### 4.1. PURPOSE

After completing this subject module, students are expected to be able to use the Euler and Runge-Kutta methods to solve physical problems related to ordinary differential equations (PDB) through the steps of preparing numerical formulations, preparing algorithms, preparing flow charts and creating computer programs.

### 4.2. INTRODUCTION

An ordinary differential equation (PDB) is a differential equation whose function contains only one independent variable. If the function contains more than one independent variable, then the equation is called a partial differential equation (PDP). Ordinary differential equations have the form :

$$\frac{dy}{dx} = f(x, y) \quad (4.1)$$

In equation (4.1),  $x$  is the independent variable and  $y$  is the dependent variable. The general solution of the equation is :

$$y_{n+1} = y_n + \phi h \quad (4.2)$$

with  $\phi$  the slope and  $h$  the step size. The slope  $\phi$  is used to extrapolate an old price  $y_n$  to a new price  $y_{n+1}$  along a distance  $h$ . This formula can be applied step by step for further calculations. Numerically the GDP can be solved by Euler and Runge-Kutta methods.

### 4.3. EULER METHOD

The solution of the ordinary differential equation as in equation (4.2), according to Euler's method is interpreted as :

$$y_{n+1} = y_n + \frac{f(x_n, y_n)}{dy} h \quad (4.3)$$

with  $\frac{f(x_n, y_n)}{dx}$  is a differential equation evaluated at  $x_n$  and  $y_n$ .

The solution of equation (4.3) with a computer program begins by entering the initial value of  $x_0$ ,  $y_0$ , the amount of step  $h$  and the final value of  $x_m$  which will be determined by the value of  $y_m$ . Next, with the initial values  $x_0$ ,  $y_0$ , and  $h$ ,  $y_1$  and  $x_1$  are calculated. Thus, the iteration continues with the addition of a step of  $h$  at  $x$  until  $x_m$  so that the value of  $y_m$  is obtained. Repetition of the program iteration can also be done to get the value of  $y_m$  that is really close to the actual value by reducing the step value  $h$ . Iteration is stopped when the difference between the new  $y$  value and the old  $y$  value is small enough.

**Example 4.1.** : Find the distance traveled by an object at  $t = 10$  seconds if the object is moving with speed  $v = 10 + t$ , and  $S = 0$  when  $t = 0$ .

**Completion:**

**Program:**

$$y_{n+1} \rightarrow S_{n+1}; y_n \rightarrow S_n; \quad f(x_n, y_n) \rightarrow v(t_n); t_n = t_0 + i * h$$

```
clear all;
clc;
v=inline('10+t','t');
s=inline('10*t+0.5*t.^2','t');
t0=0;
tm=10;
n=100;
h=(tm-t0)/n;
sn=0;
for i=1:n
    tn=t0+i*h;
    sn1=sn+v(tn)*h;
    sn=sn1;
end
Sa=s(tm);
ea=100*abs((Sa-sn1)/Sa);
disp(['Analytical Distance Value (Sa) when t = ',num2str(tm), ' is = ',num2str(Sa)]);
disp(['Numerical Distance Value (Sn) when t = ',num2str(tm), ' is = ',num2str(sn1)]);
disp(['Error with h = ', num2str(h), ' is = ',num2str(ea), '% ']);
```

**Program Results:**

The value of the analytical distance (Sa) when t = 10 is = 150 The value of the numerical distance (Sn) when t = 10 is = 150.5 The error with h = 0.1 is = 0.33333%.

#### 4.4. RUNGE- KUTTA METHOD

There are several types of Runge-Kutta methods, but in general the iteration formula can be written in the form:

$$\begin{aligned} y_{n+1} &= y_n + f(x_n, y_n, h) \\ f(x_n, y_n, h) &= a_1 k_1 + a_2 k_2 + \dots + a_m k_m \\ h &= x_{n+1} - x_n \end{aligned} \tag{4.4}$$

with:

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + p_1 h, y_n + q_{11} k_1) \\ k_3 &= hf(x_n + p_2 h, y_n + q_{21} k_1 + q_{22} k_2) \\ &\dots \dots \dots \\ k_m &= hf(x_n + p_{m-1} h, y_n + q_{m-1,1} k_1 + q_{m-1,2} k_2 + \dots + q_{m-1,m-1} k_{m-1}) \end{aligned} \tag{4.5}$$

##### 4.4.1. First Order Runge-Kutta

The first-order Runge-Kutta method is obtained if  $m$  in equation (4.4) is 1 so that the iteration equation becomes :

$$\begin{aligned} y_{n+1} &= y_n + f(x_n, y_n, h) \\ f(x_n, y_n, h) &= a_1 k_1 \end{aligned} \tag{4.6}$$

with  $a_1 = 1$  and  $k_1$  is given by equation (4.5) so the iteration formula is :

$$y_{n+1} = y_n + f(x_n, y_n)h \tag{4.7}$$

The first-order Runge-Kutta iteration formula in equation (4.7) is the same as the Euler method iteration formula in equation (4.3).

##### 4.4.2. Second-order Runge-Kutta

The second-order Runge-Kutta method is obtained when  $m$  in equation (4.4) According to Heun, the second-order Runge-Kutta iteration is formulated as follows:

$$\begin{aligned} y_{n+1} &= y_n + \frac{1}{2}(k_1 + k_2) \\ k_1 &= hf(x_n, y_n) \\ k_2 &= hf(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1) \end{aligned} \tag{4.8}$$

#### 4.4.3. Third Order Kutta Runge

The iteration formula used in the third-order Runge-Kutta method is :

$$y_{n+1} = y_n + \frac{1}{6} \left( k_1 + 4k_2 + k_3 \right)$$

with

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right) \\ k_3 &= hf\left(x_n + h, y_n + k_1 + 2k_2\right) \end{aligned} \quad (4.9)$$

#### 4.4.4. Fourth Order Runge-Kutta

The iteration formula used in the fourth-order Runge-Kutta method is :

$$y_{n+1} = y_n + \frac{1}{6} [k_1 + 2k_2 + 2k_3 + k_4]$$

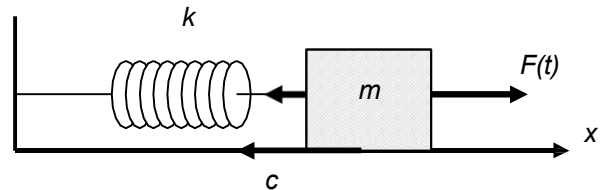
with

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right) \\ k_3 &= hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2\right) \\ k_4 &= hf(x_n + h, y_n + k_3) \end{aligned} \quad (4.10)$$

## SECOND-ORDER DIFFERENTIAL EQUATIONS

Second-order differential equations are commonly found in physics and engineering problems. For example, a forced damped oscillation system in a spring-mass circuit can be expressed by the following equation:

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + kx = F(t)$$



with  $m$  the mass of the oscillating body,  $c$  the damping constant,  $k$  the spring constant and  $F(t)$  external force (forcing force). The known initial conditions at  $t = 0$  are  $x(0)$  and  $\dot{x}(0)$ .

The solution of this case is to transform the second-order differential equation into two first-order differential equations by introducing the variable :

$$\frac{dx}{dt} = y(t) = \dot{x} = f(t)$$

so the two first-order differential equations are:

$$\frac{dx}{dt} = y(t) = x = f(t, x, y) \quad x(0) =$$

$$\frac{d^2x}{dt^2} = \frac{dy}{dt} = -\frac{c}{m}y + \frac{k}{m}x + \frac{1}{m}F(t) = g(t, x, y) \quad \text{with}$$

$$\left. \begin{array}{l} x(0) = x_0 \\ y(0) = y_0 \end{array} \right\}$$

If the Euler method is used, the form of the numerical equation is :

$$y_{n+1} = y_n + h \cdot g(y_n)$$

$$x_{n+1} = x_n + h \cdot f(t_n, x_n, y_n)$$

## PRACTICE QUESTIONS

1. The rate at which a parachutist hovers in the air after the umbrella has expanded is expressed by the formula:

$$v(t) = \frac{gm}{c} \left[ 1 - e^{-\frac{ct}{m}} \right]$$

with  $g = 9.8 \text{ m/s}^2$ ,  $m = 68 \text{ kg}$  and air resistance  $c = 12.5 \text{ kg/s}$ . Create a program to determine the distance traveled by the jumper for 10 seconds using the Euler method. The distance traveled at  $t = 0$  seconds is 0 meters. Graph the relationship between distance traveled and time for 10 seconds.

2. A differential equation has the following form:

$$\frac{dy}{dx} = -xy$$

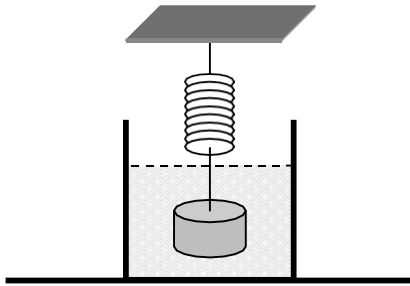
If the condition is known when  $x = 0$  the value of  $y$  ( $x=0$ ) = 1, then make a MATLAB program with the Euler method to calculate the value of  $y$  when  $x = 1, 2, 3 \dots 10$ . Make a graph of the relationship between  $x$  and  $y$ . Save the program you created with the file name **EULER\_2**.

3. Make a MATLAB program for case number 1 and number 2 using the Runge-Kutta



method of order 2, 3 and 4. Save the program you have made with the file names **RK\_2**, **RK\_3** and **RK\_4** respectively.

4. A spring-mass damped oscillatory system is described as follows:



Unknown:

load mass = 0.1 kg

damping coefficient = 0.05 kg/sec spring

constant = 0.1 N/m

Initial condition: at time  $t = 0 \rightarrow x(0) = 3$

$\rightarrow v(0) = 0$

a. Make a mathematical model of the oscillating system above and state its differential equation.

b. Use Euler's method to determine the values of  $x(t)$  and  $v(t)$  in time  $t = [0 \ 10]$  seconds. Graph  $x(t)$  versus  $t$  and  $v(t)$  versus  $t$ .

5. A non-linear pendulum swing is depicted as follows:

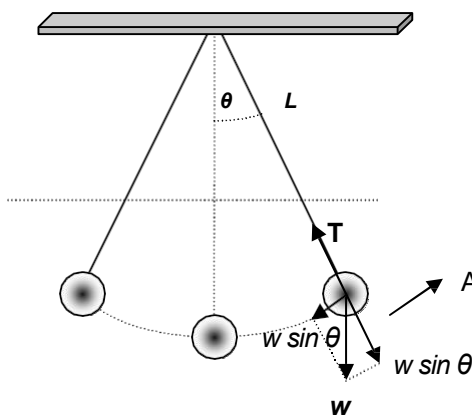


Image Description:

- ▶  $\theta$  : angle of deviation
- ▶  $L$  : length of rope
- ▶  $w$  : weight of the pendulum
- ▶  $T$  : rope tension force

The following data is known:

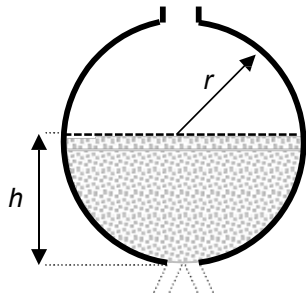
- $m=0.5$
- $L=1$
- $g=9.8$
- $\theta(t=0) = \pi/4$ ;  $\dot{\theta}(t=0) = 0$

a. Make a mathematical model of the above non-linear pendulum oscillation system and state its differential equation.

b. Use Euler's method to determine the values of  $\theta(t)$  and  $\omega(t)$  in a time range  $t = [0 \ 10]$  seconds. Graph  $\theta(t)$  versus  $t$  and  $\omega(t)$  versus  $t$ .

c. Repeat a and b for  $\theta(t=0) = \pi/18$ ;  $\dot{\theta}(t=0) = 0$  and compare the results to linear pendulum swing.

6. The picture below shows a spherical water reservoir. The reservoir is filled through the top hole and water comes out through the bottom hole.



The relationship between water level and time is expressed by the equation :

$$\frac{dh}{dt} = -\frac{c_d A \sqrt{2gh}}{\pi(2rh - h^2)}$$

with  $c_d$  an experimental constant of 0.6,  $A$  the cross-sectional area of the hole,  $r$  the radius of the reservoir,  $h$  the height of the water in the reservoir and  $g$  the acceleration of the earth's gravity.

If it is known that  $r = 2$  meters,  $g = 9.8 \text{ m/sec}^2$ ,  $A = \pi r^2$  and the initial height of water in the reservoir is 3 meters, then using the second-order Runge-Kutta method:

- create a numerical model to determine the value of  $h$  as a function of time
- Make the program algorithm of the numerical model in *a*.
- Create a program to calculate the value of  $h$  as a function of time and display the graph.

## V BOUNDARY PRICE (VALUE) ISSUES

### 5.1. PURPOSE

After completing this subject module, students are expected to be able to use the Poisson equation and the Wave equation to solve physical problems related to Boundary Prices through the steps of preparing numerical formulations, preparing algorithms, preparing flow charts and creating computer programs.

### 5.2. THEORY BASIS

Second-order differential equations related to boundary prices are expressed in the form:

$$\frac{d^2 y}{dx^2} + k^2(x)y = S(x) \quad (5.1)$$

with  $k^2$  real functions and  $S$  inhomogeneous terms. A special situation occurs if the value of  $k = 0$  or  $S = 0$ . If  $k = 0$  then equation (5.1) becomes:

$$\frac{d^2 y}{dx^2} = S(x) \quad (5.2)$$

which is called the Poisson equation. This equation can be solved with a computer program if it is converted into a form that the computer can understand. Using the Tylor series, the equation can be converted into the form :

$$y_{n+1} = 2y_n - y_{n-1} + \frac{h^2}{12}(S_{n+1} + 10S_n + S_{n-1}) \quad (5.3)$$

If  $S(x)$  is certain, then to get the value of  $y_{n+1}$ , we need the values of  $y_n$  and  $y_{n-1}$  as boundary conditions. For the second case,  $S=0$ , equation (5.1) becomes the wave equation as follows:

$$\frac{d^2 y}{dx^2} = -k^2(x)y \quad (5.4)$$

The solution of equation (5.4) using the modified Taylor series is as follows as follows:

$$y_{n+1} = \frac{\left(1 - \frac{h^2 k^2}{12}\right)^n y_n \left(1 + \frac{h^2 k^2}{12}\right)^{n-1}}{\left(1 + \frac{h^2 k^2}{12}\right)^{n+1}} \quad (5.5)$$

with  $k^2 =$  eigenvalue of a wave equation.

**Example 5.1:**

Create a program to determine the static electric potential using Poisson's equation (equation 5.2) with the following conditions:

- Charge density:  $\rho = (1/8\pi)e^{-r} \rightarrow S(r) = -4\pi\rho = -\frac{1}{2}e^{-r}$
- Boundary conditions:  $y_0 = \varphi_1 = 0$  and  $y_1 = \varphi_2 = 0.01$  with  $\varphi = \Phi$ ,  $\Phi =$  potential.
- Derivative formula (derivative):

$$\varphi_{n+1} = 2\varphi_n - \varphi_{n-1} + \frac{h^2}{12}(S_{n+1} + 10S_n + S_{n-1})$$

- Determine: the value of electric potential at  $r = 20$

**Completion:**

```
clear
clc
f=inline(' -0.5*r.*exp(-r) ','r')
N=1000;
r0=0;
rm=100;
h=(rm-r0)/N;
c=(h)^2/12;
y(1)=0;
%y(2)=1-0.5*(h+2)*exp(-h);
y(2)=0.0499;
s(1)=f(0);
s(2)=f(h);
r(1)=r0;
r(2)=h;
for i=2:N
    r(i+1)=(i)*h;
    s(i+1)=f(r(i+1));
    y(i+1)=2*y(i)-y(i-1)+c*(s(i+1)+10*s(i)+s(i-1));
end
```

```
phi=y./r;  
[r' phi']  
plot(r,phi,'b','linewidth',3)  
grid on
```

