MATLAB
تعليـــم الماتلاب خطوة بخطــــــــوة

إعداد وتقديم

المهندس:- احمد محمد الفلاح الرابطى.

التخصص:- الهندسة الكهربية والإلكترونية.

القسم:- التحكم الألى.

الجامعة:- (جامعة الجبل الغربي, جامعة الفاتح, جامعة الله آباد)

المهنة والخبرة:- (معيد في كلية الهندسة, مهندس في شركة ليبيانا, مهندس في مصنع الأدوية بالرابطة, معلم في المعهد الصناعي بالرابطة, مهندس منفذ ومتابع لصيانة منظومة الدراسة والامتحانات في شعبية الجبل الغربي)

البريد الألكترونى:- Ahmad_Engineer21@yahoo.com

المستوى التعليمي:- بكالوريوس في الهندسة الكهربية شعبة التحكم الألى من جامعة الجبل الغربي. ودبلوما في الدراسات العليا في الهندسة الكهربية شعبة التحكم الألى من جامعة الفاتح. والآن دراسة الماجستير في جامعة الله أباد في الهند وتحضير لمناقشة رسالة الماجستير.

السنة الدراسية:- 2010 – 2011 ف.

الهواية:- المطالعة والشطرنج وكرة القدم.

العنوان:- ليبيا – غريان – الرابطة.

مقـــــــدمـــــة

بسم الله الدى لا يحمد ولا يغفر ولا يسأل إلا هو وحده لا شريك له نعبده ولا نشرك به شيء و صلى اللهم وسلم على النبي المصطفى خاتم النبيين, وعلى أله الطاهرين البررة وعلى أصحابه الأكرمين الدين نشروا الدين في البلدان وحملوا القران وحفظوا السنة, وعلى زوجاته الطاهرات أمهات المؤمنين وبعد...

فمن خلال دراستي في لغة الماتلاب لاحظت أن هناك عدة كتب تشرح البرمجة بلغة الماتلاب ولكن توجد ندرة في الأمثلة العملية في هده الكتب فنلاحظ الكاتب يكتفي بكتابة مثال ام مثالين بسيطين قد لا يعطى طالب العلم مراده وأيضا من خلال ملاحظتي لتخبط الكبير لبعض الطلاب في كتابة البرامج بلغة الماتلاب في معمل الحاسوب. وعدم فهم كيف يتم تصحيح الأخطاء . لهدا كتبت لكم قدر كبير من الأمثلة العملية مع الخرج لتوصيل الفكرة بسهولة ويسر وبسرعة وبدون تعقيد. وقد تأكدت من النتائج للبرامج كلها في الحاسوب. . وكل هدا في سبيل تيسير العلم فنسأل الله أن يجزينا عن هدا العمل كامل الجزاء في يوم تزل فيه الأقدام انه نعم المولى ونعم النصير. ونسأل كل من استفاد من هدا العمل الدى أخد منى ساعات طوال لتحضيره وإخراجه لكم على مثل هده الصورة المنظمة والواضحة أن يدعوا لنا في ظهر الغيب ونسأل الله القبول وعدم الرياء والنفاق فهو نعم المولى ونعم النصير .

مفخرة للإنسان العلم

واحـذر يفوتـك فخـر ذاك المغـرس ....العلـم مغـرس كـل فخـر فافتخـر
من همته في مطعـم أو ملبـس ....واعلـم بـأن العـلـم لـيـس يـنـالـه
فـي حالتيـه عـارياً أو مكتـسـِـي..............................إلا أخـو العلـم الـذي يعـنـي بــه
واهجـر لـه طيــب الرقــاد وعـبِّــس.........................فاجعل لنفسـك منـه حظـاً وافـراً
كنت الرئيس وفخر ذلك المجلس ....فلعـل يومـاً إن حضـرت بمجلـسٍ

اللذة في طلب العلم

مــن وصــل غـانـيـةٍ وطـيــب عـنــاق ....ســهـري لتنـقـيـح الـعـلـوم ألـــذ لـــي
أحـلـــى مـــن الـدوكـــاء والـعـشـــاق ....وصـريـر أقـلامــي عـلــى صفحـاتـهـا
نقـري لألـقـي الرمـل عــن أوراقـــي..................... وألـــذ مـــن نـقــر الـفـتـاة لـدفـهـا
في الدرس أشهى من مدامة ساقي ....وتـمـا يـلـي طـربـاً لـحـل عـويـصـةٍ
نـومـاً وتـبـغـي بـعــد ذلـــك لـحـاقـي .... وأبـيــت ســهــران الـدجــا وتـبـيـتـه

1- الجمع فى الماتلاب

```
%---------------------------------
clc
clear
a=4;
b=5;
c=7;
d=a+b+c
%---------------------------------
clc
clear
a=[2 3 4 6 7 8 9 10];
sum(a)
%----------------------------------
```

2- الطرح فى الماتلاب

```
%---------------------------------
clc
clear
a=4;
b=5;
c=7;
d=a-b-c
%---------------------------------
```

3- الضرب فى الماتلاب

```
%---------------------------------
clc
clear
a=4;
b=5;
c=7;
d=a*b*c
%---------------------------------
clc
clear
a=4;
b=5;
c=7;
d=conv(a,b)      %or  d=conv(a,conv(b,c))
f=conv(d,c)
%----------------------------------
 s=[4 5 7];
 prod(s)
%-------------------------
```

4- القسمة فى الماتلاب

```
%---------------------------------
clc
clear
a=4;
b=5;
c=7;
d=a/b
f=d/c
%---------------------------------
```

```
clc
clear
a=4;
b=5;
c=7;
d=deconv(a,b)      %or  d=deconv(deconv(a,b),c)
f=deconv(d,c)
%---------------------------------
```

5-تمثيل الجدر والدالة الأسية واللوغاريتم الطبيعى والدوال المثلثية فى الماتلاب

```
  ((5*log10(x)+2*x^2*sin(x)+sqrt(x)*lin(x))
f=-----------------------------------------
      (exp(6*x^3)+3*x^4+sin(lin(x)))
```

```
%--------------------------------
clc
clear
x=1;
f=deconv((5*log10(x)+2*x^2*sin(x)+sqrt(x)*log(x)),(exp(6*x^3)+3*x^4+s
in(log(x))))
%--------------------------------
clc
clear
x=1;
f=(5*log10(x)+2*x^2*sin(x)+sqrt(x)*log(x))/(exp(6*x^3)+3*x^4+sin(log
(x)))
%---------------------------------
```

6- التفاضل فى الماتلاب

```
%--------------------------------
clc
clear
syms x
f=((x^5)+(5*x^4)+(4*x^3)-(2*x^2)-(8*x)+9)
d=diff(f,x)
%--------------------------
clc
clear
syms x
f=((x^5)+(5*x^4)+(4*x^3)-(2*x^2)-(8*x)+9)
d=diff(f,2)
%-----------------------------
%--------------------------------
clc
clear
syms x
f=(1/(1+x^2))
d=diff(f,x)
%--------------------------
```

7- التكامل فى الماتلاب

```
%--------------------------------
clc
clear
syms x
f=(1/(1+x^2))
```

```
d=int(f,x)
%---------------------------
clc
clear
syms x
f=((x^5)+(5*x^4)+(4*x^3)-(2*x^2)-(8*x)+9)
d=int(f,x)
%------------------------------
%---------------------------
clc
clear
syms x
f=((x^5)+(5*x^4)+(4*x^3)-(2*x^2)-(8*x)+9)
d=int(f,1,2)
%------------------------------
%---------------------------
clc
clear
syms x a b
f=((x^5)+(5*x^4)+(4*x^3)-(2*x^2)-(8*x)+9)
d=int(f,a,b)
%------------------------------
```

8- حل المعادلات التفاضلية

Example 1

Find the total solution of the ODE

$$\frac{d^2y}{dt^2} + 4\frac{dy}{dt} + 3y = 3e^{-2t}$$

subject to the initial conditions $y(0) = 1$ and $y'(0) = -1$

Solution:

$$y_N(t) = k_1e^{-t} + k_2e^{-3t}$$

(We must remember that the constants k_1 and k_2 must be evaluated from the total response).

To find the forced response, we assume a solution of the form

$$y_F = Ae^{-2t}$$

$$4Ae^{-2t} - 8Ae^{-2t} + 3Ae^{-2t} = 3e^{-2t}$$

from which $A = -3$ and the total solution is

$$y(t) = y_N + y_F = k_1 e^{-t} + k_2 e^{-3t} - 3e^{-2t}$$

The constants k_1 and k_2 are evaluated from the given initial conditions. For this example,

$$y(0) = 1 = k_1 e^0 + k_2 e^0 - 3e^0$$

or

$$k_1 + k_2 = 4$$

$$y'(0) = -1 = \left.\frac{dy}{dt}\right|_{t=0} = -k_1 e^{-t} - 3k_2 e^{-3t} + 6e^{-2t}\Big|_{t=0}$$

or

$$-k_1 - 3k_2 = -7$$

yields $k_1 = 2.5$ and $k_2 = 1.5$.

$$y(t) = y_N + y_F = 2.5e^{-t} + 1.5e^{-3t} - 3e^{-2t}$$

```
%---------------------------------
clc
clear
syms x t
y=dsolve('D2y+4*Dy+3*y=3*exp(-2*t)', 'y(0)=1', 'Dy(0)=-1')
ezplot(y,[0 4])
%pretty(y)
%-----------------------------------
```

9- حل معادلتين وثلاث معادلات للإخراج الثوابت

Example1

$$19k3+25k4=0$$
$$25k3-19k4=4$$

Example2

$$r1+r2=1$$
$$0.683r1+3.817r2+r3=2$$
$$0.393r1+3.817r3=-1$$

Solution

```
%-----------------------------------
clc
syms k3 k4
f1=19*k3+25*k4;
f2=25*k3-19*k4-4;
[k3 k4]=solve(f1,f2 )
%--------------------------------
clc
syms r1 r2 r3
f1=r1+r2-1;
f2=0.683*r1+3.817*r2+r3-2;
```

```
f3=0.393*r1 + 3.817*r3+1
[r1 r2 r3]=solve(f1,f2,f3)
%-------------------------------------------
```

Example3

x^2*y^2+z=0
x-(y/2)-alpha+z=0
x+z+y=0

Solution

```
%--------------------------------
clc
syms x y z alpha
x^2*y^2+z=0
x-(y/2)-alpha+z=0
x+z+y=0
[x,y,z]=solve('x^2*y^2+z','x-(y/2)-alpha+z','x+z+y')
%----------------------------------
```

10- إيجاد الجذور من معادلة متعددة الحدود والعكس

Example 1

Find the roots of the polynomial

$$p_1(x) = x^4 - 10x^3 + 35x^2 - 50x + 24$$

$$p_2(x) = x^5 - 7x^4 + 16x^2 + 25x + 52$$

Solution:

```
%------------------------------------
clc
p1=[1 -10 35 -50 24];
f1=roots(p1)
%------------------------------------
p2=[1 -7 0 16 25 52];
f2=roots(p2)
%-----------------------------------
```

Example 2

1- It is known that the roots of a polynomial are $1, 2, 3,$ and 4. Compute the coefficients of this polynomial.

2- It is known that the roots of a polynomial are $-1, -2, -3, 4+j5$ and $4-j5$. Find the coefficients of this polynomial.

Solution:

```
%------------------------------
clc
r1=[1 2 3 4];
f1=poly(r1)
r2=[-1 -2 -3 -4+5j -4-5j ];
f2=poly(r2)
%-----------------------------------
```

Example 3

Evaluate the polynomial

$$p_5(x) = x^6 - 3x^5 + 5x^3 - 4x^2 + 3x + 2$$

at $x = -3$.

Solution:

```
%--------------------------------
clc
p1=[1 -3 0 5 -4 3 2];
f1=polyval(p1,-3)
%----------------------------------
```

Example 4

Let

$$p_1 = x^5 - 3x^4 + 5x^2 + 7x + 9$$

and

$$p_2 = 2x^6 - 8x^4 + 4x^2 + 10x + 12$$

1- Compute the product $p_1 \cdot p_2$ using the **conv(a,b)** function.

2- Compute the product $p_1 \cdot p_2$ using the **[q,r]=deconv(c,d)** function.

Solution

```
%--------------------------------
clc
p1=[1 0 -3 0 5 7 9];
p2=[2 -8 0 0 4 10 12];
f1=conv(p1,p2)
[q,r]=deconv(p1,p2)
%----------------------------------
```

Example 5

Let

$$p_5 = 2x^6 - 8x^4 + 4x^2 + 10x + 12$$

1- Compute the derivative $\frac{d}{dx}p_5$ using the **polyder(p)** function.

2- Compute the integration p_5 using the **polyint(p)** function.

Solution:

```
%--------------------------------
clc
clear
p5=[2 0 -8 0 4 10 12];
f1=polyder(p5)
f2=polyint(p5)
%----------------------------------
```

Example 6

$$R(x) = \frac{p_{num}}{p_{den}} = \frac{(x^2 - 4.8372x + 6.9971)(x^2 + 0.6740x + 1.1058)(x + 1.1633)}{(x^2 - 3.3520x + 3.0512)(x^2 + 0.4216x + 1.0186)(x + 1.0000)(x + 1.9304)}$$

Find $\frac{p_{num}}{p_{den}}$ in polynomial form using the **collect(s)** function that is used to multiply two or more symbolic expressions to obtain the result in polynomial form. We must remember that the **conv(p,q)** function is used with numeric expressions only, that is, polynomial coefficients.

solution

```
%--------------------------------
clc
clear
syms x
pnum=collect((x^2-4.8372*x+6.9971)*(x^2+0.6740*x+1.1058)*(x+1.1633))
pden=collect((x^23.3520*x+3.0512)*(x^2+0.4216*x+1.0186)*(x+1.0000)*(x
+1.9304))
R=pnum/pden
pretty(R)
%----------------------------------
```

Example 7

finds the residues, poles and direct term of a partial fraction expansion of the ratio of two polynomials B(s)/A(s) .If there are no multiple roots,

```
 B(s)     R(1)     R(2)          R(n)
 ----  =  -------- + -------- + ... + -------- + K(s)
 A(s)    s - P(1)  s - P(2)      s - P(n)
```

[R,P,K] = residue (B,A)

$$\frac{b}{a} = \frac{x^4+2x^3-4x^2+5x+1}{x^5+4x^4-2x^3+6x^2+2x+1}$$

solution

```
%--------------------------------
clc
b=[1 2 -4 5 1];
a=[1 4 -2 6 2 1];
[R,P,K] = residue(b,a)
%----------------------------------
```

R = 0.2873 , -0.0973 + 0.1767i, -0.0973 - 0.1767i, 0.4536 + 0.0022i, 0.4536 - 0.0022i

P = -4.6832, 0.5276 + 1.0799i, 0.5276 - 1.0799i, -0.1860 + 0.3365i, -0.1860 - 0.3365i

K =0

11- كيفية رسم الدوال فى الماتلاب

Example 1

Write the MATLAB code that produces a simple plot for the waveform defined as

$$y = f(t) = 3e^{-4t}\cos 5t - 2e^{-3t}\sin 2t + \frac{t^2}{t+1}$$

in the $0 \le t \le 5$ seconds interval.

Solution:

```
%--------------------------------
clc
clear
t=0: 0.01: 5            % Define t-axis in 0.01 increments
y=3.* exp(-4.* t).* cos(5 .* t)-2.* exp(-3.* t).* sin(2.* t) + t.^2./
(t+1)
plot(t,y);
grid;
xlabel('t');
ylabel('y=f(t)');
title('Plot for Example A.13')
%----------------------------------
```

Example 2

Plot the functions

$$y = \sin^2 x, \quad z = \cos^2 x, \quad w = \sin^2 x \cdot \cos^2 x, \quad v = \sin^2 x / \cos^2 x$$

in the interval $0 \le x \le 2\pi$ using 100 data points.

1-Use the **plot** command to display these functions on same windows on the same graph.

2-Use the **subplot** command to display these functions on four windows on the same graph.

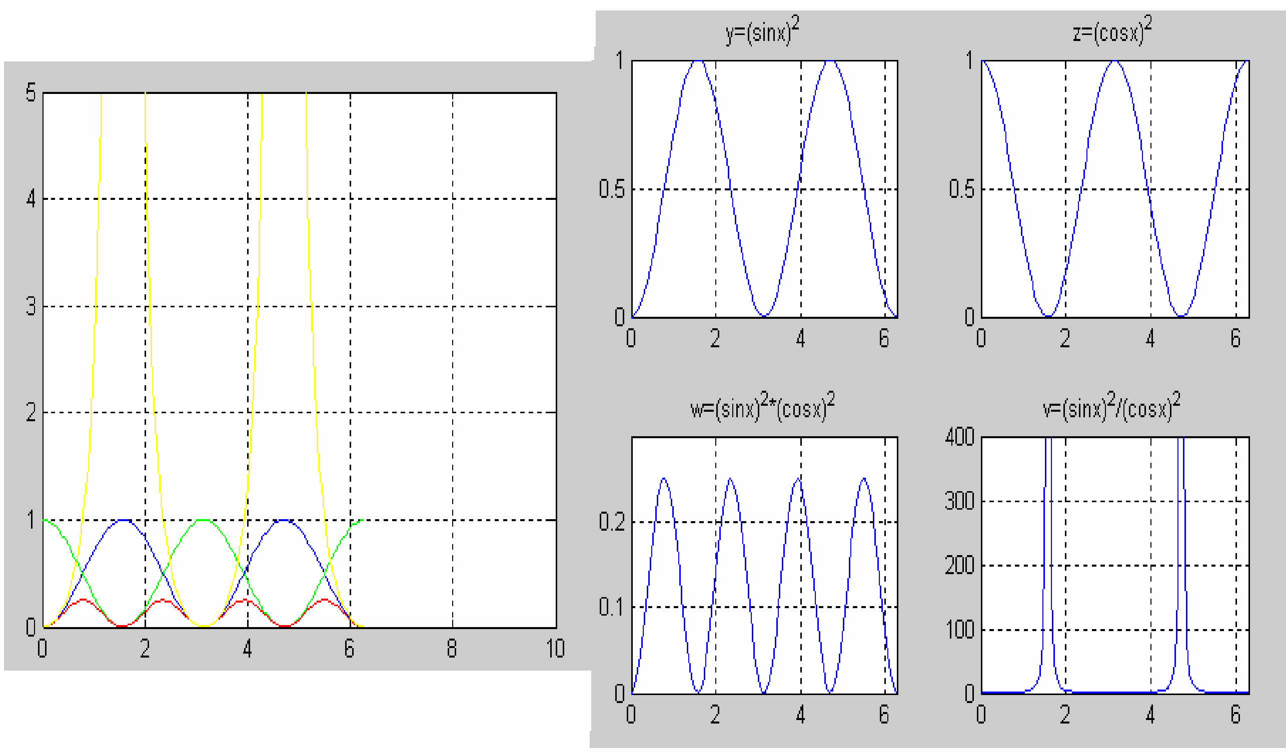
Solution:

1-Use the **plot** command to display these functions on same windows on the same graph.

```
%--------------------------------
clc
clear
x=linspace(0,2*pi,100);         % Interval with 100 data points
y=(sin(x).^ 2);
z=(cos(x).^ 2);
w=y.* z;
v=y./ (z+eps);                  % add eps to avoid division by zero
plot(x,y,'b',x,z,'g',x,w,'r',x,v,'y');
grid on
axis([0 10 0 5]);
%----------------------------------------------------------------
```

2-Use the **subplot** command to display these functions on four windows on the same graph.

```
%-----------------------------------------------------------------
clc
clear
x=linspace(0,2*pi,100);         % Interval with 100 data points
y=(sin(x).^ 2);
z=(cos(x).^ 2);
w=y.* z;
v=y./ (z+eps);
subplot(221);                   % upper left of four subplots
plot(x,y);
axis([0 2*pi 0 1]);
title('y=(sinx)^2');
grid on
subplot(222);                   % upper right of four subplots
plot(x,z);
axis([0 2*pi 0 1]);
title('z=(cosx)^2');
grid on
subplot(223);                   % lower left of four subplots
plot(x,w);
axis([0 2*pi 0 0.3]);
title('w=(sinx)^2*(cosx)^2');
grid on
subplot(224);                   % lower right of four subplots
plot(x,v);
axis([0 2*pi 0 400]);
title('v=(sinx)^2/(cosx)^2');
grid on
%---------------------------------
```



مثال :- اكتب برنامج يحسب جدول الضرب ويعرضه فى شكل منظم باستعمال الأمر for؟

الحل

```
clc
a=0;
 disp('----------------------------------------------------')
for i=1:10;
    b=0;
       for j=1:10;
           c(j) =a*b;
           b=b+1;
       end
       c
 disp('----------------------------------------------------')
a=a+1;
end
```

تمثيـــــــــــل المصفوفات فى الماتـــــــــــلاب

1- العمليات الحسابية للمصفوفات فى الماتلاب((Matrix Operations))

Example C.1

Compute $A+B$ and $A-B$ given that

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 4 \end{bmatrix} \text{ and } B = \begin{bmatrix} 2 & 3 & 0 \\ -1 & 2 & 5 \end{bmatrix}$$

Solution:

$$A+B = \begin{bmatrix} 1+2 & 2+3 & 3+0 \\ 0-1 & 1+2 & 4+5 \end{bmatrix} = \begin{bmatrix} 3 & 5 & 3 \\ -1 & 3 & 9 \end{bmatrix}$$

and

$$A-B = \begin{bmatrix} 1-2 & 2-3 & 3-0 \\ 0+1 & 1-2 & 4-5 \end{bmatrix} = \begin{bmatrix} -1 & -1 & 3 \\ 1 & -1 & -1 \end{bmatrix}$$

Check with MATLAB:

```
%-----------------------------------------------------------
clc
clear
A=[1 2 3; 0 1 4];                    % Define matrices A
B=[2 3 0; -1 2 5];                   % Define matrices B
m1=A+B                                  % Add A and B
m2=A-B                                  % Subtract B from A
%------------------------------------------------------------
```

Example C.2

Multiply the matrix

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

by

a. $k_1 = 5$

b. $k_2 = -3+j2$

Solution:

a.

$$k_1 \cdot A = 5 \times \begin{bmatrix} 1 & -2 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 5\times 1 & 5\times(-2) \\ 5\times 2 & 5\times 3 \end{bmatrix} = \begin{bmatrix} 5 & -10 \\ 10 & 15 \end{bmatrix}$$

b.

$$k_2 \cdot A = (-3+j2) \times \begin{bmatrix} 1 & -2 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} (-3+j2)\times 1 & (-3+j2)\times(-2) \\ (-3+j2)\times 2 & (-3+j2)\times 3 \end{bmatrix} = \begin{bmatrix} -3+j2 & 6-j4 \\ -6+j4 & -9+j6 \end{bmatrix}$$

Check with MATLAB:

```
%-----------------------------------------------------------
clc
clear
k1=5;                       % Define scalars k1
k2=(-3 + 2*j);              % Define scalars k2
A=[1 -2; 2 3];              % Define matrix A
m1=k1*A                     % Multiply matrix A by constant k1
m2=k2*A                     %Multiply matrix A by constant k2
%------------------------------------------------------------
```

Example C.3

Matrices C and D are defined as

$$C = \begin{bmatrix} 2 & 3 & 4 \end{bmatrix} \text{ and } D = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix}$$

Compute the products $C \cdot D$ and $D \cdot C$

Solution:

The dimensions of matrices C and D are respectively $1 \times 3 \quad 3 \times 1$; therefore the product $C \cdot D$ is feasible, and will result in a 1×1, that is,

$$C \cdot D = \begin{bmatrix} 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} = \begin{bmatrix} (2) \cdot (1) + (3) \cdot (-1) + (4) \cdot (2) \end{bmatrix} = \begin{bmatrix} 7 \end{bmatrix}$$

The dimensions for D and C are respectively $3 \times 1 \quad 1 \times 3$ and therefore, the product $D \cdot C$ is also feasible. Multiplication of these will produce a 3×3 matrix as follows:

$$D \cdot C = \begin{bmatrix} 1 \\ -1 \\ 2 \end{bmatrix} \begin{bmatrix} 2 & 3 & 4 \end{bmatrix} = \begin{bmatrix} (1) \cdot (2) & (1) \cdot (3) & (1) \cdot (4) \\ (-1) \cdot (2) & (-1) \cdot (3) & (-1) \cdot (4) \\ (2) \cdot (2) & (2) \cdot (3) & (2) \cdot (4) \end{bmatrix} = \begin{bmatrix} 2 & 3 & 4 \\ -2 & -3 & -4 \\ 4 & 6 & 8 \end{bmatrix}$$

Check with MATLAB:

```
%-----------------------------------------------------------
clc
clear
C=[2 3 4];               % Define matrices C and D
D=[1; -1; 2];            % Define matrices C and D
m1=C*D                   % Multiply C by D
m2=D*C                   % Multiply D by C
%------------------------------------------------------------
```

2- حساب المحددات للمصفوفات ((Determinants of Matrices))

Example C.4

Matrices A and B are defined as

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \text{ and } B = \begin{bmatrix} 2 & -1 \\ 2 & 0 \end{bmatrix}$$

Compute $detA$ and $detB$.

Solution:

$$detA = 1 \cdot 4 - 3 \cdot 2 = 4 - 6 = -2$$

$$detB = 2 \cdot 0 - 2 \cdot (-1) = 0 - (-2) = 2$$

Check with MATLAB:

```
%----------------------------------------------------------
clc
clear
A=[1 2; 3 4];
B=[2 -1; 2 0];          % Define matrices A and B
det(A)                  % Compute the determinant of A
det(B)                  % Compute the determinant of B
%-----------------------------------------------------------
```

Example C.5

Compute $detA$ and $detB$ if matrices A and B are defined as

$$A = \begin{bmatrix} 2 & 3 & 5 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix} \text{ and } B = \begin{bmatrix} 2 & -3 & -4 \\ 1 & 0 & -2 \\ 0 & -5 & -6 \end{bmatrix}$$

Solution:

det(A)= 2(0*0-1*1) - 3(1*0-1*2) + 5(1*1-0*2) = -2+6+5 = 9

det(B) = 2(0*(-6)-(-2)*(-5)) - (-3)(1*(-6)-0*(-2)) +4(1*(-5)-0*0)= -18

Check with MATLAB:

```
%----------------------------------------------------------
clc
clear
A=[2 3 5; 1 0 1; 2 1 0];         % Define matrix A
B=[2 -3 -4; 1 0 -2; 0 -5 -6];    % Define matrix B
det(A)                  % Compute the determinant of A
det(B)                  % Compute the determinant of B
%-----------------------------------------------------------
```

3- قاعدة ((Cramer's Rule))

Let us consider the systems of the three equations below

$$a_{11}x + a_{12}y + a_{13}z = A$$
$$a_{21}x + a_{22}y + a_{23}z = B$$
$$a_{31}x + a_{32}y + a_{33}z = C$$

and let

$$\Delta = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \quad D_1 = \begin{vmatrix} A & a_{11} & a_{13} \\ B & a_{21} & a_{23} \\ C & a_{31} & a_{33} \end{vmatrix} \quad D_2 = \begin{vmatrix} a_{11} & A & a_{13} \\ a_{21} & B & a_{23} \\ a_{31} & C & a_{33} \end{vmatrix} \quad D_3 = \begin{vmatrix} a_{11} & a_{12} & A \\ a_{21} & a_{22} & B \\ a_{31} & a_{32} & C \end{vmatrix}$$

Cramer's rule states that the unknowns x, y, and z can be found from the relations

$$x = \frac{D_1}{\Delta} \qquad y = \frac{D_2}{\Delta} \qquad z = \frac{D_3}{\Delta}$$

provided that the determinant Δ (delta) is not zero.

Example C.10

Use Cramer's rule to find v_1, v_2, and v_3 if

$$2v_1 - 5 - v_2 + 3v_3 = 0$$
$$-2v_3 - 3v_2 - 4v_1 = 8$$
$$v_2 + 3v_1 - 4 - v_3 = 0$$

and verify your answers with MATLAB.

Solution:

Rearranging the unknowns v, and transferring known values to the right side, we get

$$2v_1 - v_2 + 3v_3 = 5$$
$$-4v_1 - 3v_2 - 2v_3 = 8$$

Now, by Cramer's rule,

$$\Delta = \begin{vmatrix} 2 & -1 & 3 \\ -4 & -3 & -2 \\ 3 & 1 & -1 \end{vmatrix} \begin{matrix} 2 & -1 \\ -4 & -3 \\ 3 & 1 \end{matrix} = 6 + 6 - 12 + 27 + 4 + 4 = 35, \quad D_1 = \begin{vmatrix} 5 & -1 & 3 \\ 8 & -3 & -2 \\ 4 & 1 & -1 \end{vmatrix} \begin{matrix} 5 & -1 \\ 8 & -3 \\ 4 & 1 \end{matrix} = 15 + 8 + 24 + 36 + 10 - 8 = 85$$

$$D_2 = \begin{vmatrix} 2 & 5 & 3 \\ -4 & 8 & -2 \\ 3 & 4 & -1 \end{vmatrix} \begin{matrix} 2 & 5 \\ -4 & 8 \\ 3 & 4 \end{matrix} = -16 - 30 - 48 - 72 + 16 - 20 = -170$$

$$D_3 = \begin{vmatrix} 2 & -1 & 5 \\ -4 & -3 & 8 \\ 3 & 1 & 4 \end{vmatrix} \begin{matrix} 2 & -1 \\ -4 & -3 \\ 3 & 1 \end{matrix} = -24 - 24 - 20 + 45 - 16 - 16 = -55$$

Then,

$$v_1 = \frac{D_1}{\Delta} = \frac{85}{35} = \frac{17}{7} \qquad v_2 = \frac{D_2}{\Delta} = -\frac{170}{35} = -\frac{34}{7} \qquad v_3 = \frac{D_3}{\Delta} = -\frac{55}{35} = -\frac{11}{7}$$

We will verify with MATLAB as follows.

```
%------------------------------------------------------------
clc
clear
% The following code will compute and display the values of v1, v2
and v3.

B=[2 -1 3;-4 -3 -2; 3 1 -1];      % The elements of the determinant
D of matrix B
delta=det(B);                      % Compute the determinant D of
matrix B
d1=[5 -1 3; 8 -3 -2; 4 1 -1];     % The elements of D1
detd1=det(d1);                     % Compute the determinant of D1
d2=[2 5 3; -4 8 -2; 3 4 -1];      % The elements of D2
detd2=det(d2);                     % Compute the determinant of D2
d3=[2 -1 5; -4 -3 8; 3 1 4];      % The elements of D3
detd3=det(d3);                     % Compute he determinant of D3
v1=detd1/delta;                    % Compute the value of v1
v2=detd2/delta;                    % Compute the value of v2
v3=detd3/delta;                    % Compute the value of v3
%-------------------------------------------------------------
disp('v1=');disp(v1);             % Display the value of v1
disp('v2=');disp(v2);             % Display the value of v2
disp('v3=');disp(v3);             % Display the value of v3
%-------------------------------------------------------------
```

4-حساب معكوس المصفوفة ((The Inverse of a Matrix))

Example C.14

Matrix A is defined as

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 4 \\ 1 & 4 & 3 \end{bmatrix}$$

Compute its inverse, that is, find A^{-1}

Solution:

Here, $detA = 9+8+12-9-16-6 = -2$, and since this is a non-zero value, it is possible to compute the inverse of A using $\boxed{A^{-1} = \frac{1}{detA}adjA}$

$$adjA = \begin{bmatrix} \begin{vmatrix} 3 & 4 \\ 4 & 3 \end{vmatrix} & -\begin{vmatrix} 2 & 3 \\ 4 & 3 \end{vmatrix} & \begin{vmatrix} 2 & 3 \\ 3 & 4 \end{vmatrix} \\ -\begin{vmatrix} 1 & 4 \\ 1 & 3 \end{vmatrix} & \begin{vmatrix} 1 & 3 \\ 1 & 3 \end{vmatrix} & -\begin{vmatrix} 2 & 3 \\ 3 & 4 \end{vmatrix} \\ \begin{vmatrix} 1 & 3 \\ 1 & 4 \end{vmatrix} & -\begin{vmatrix} 1 & 2 \\ 1 & 4 \end{vmatrix} & \begin{vmatrix} 1 & 2 \\ 1 & 3 \end{vmatrix} \end{bmatrix} = \begin{bmatrix} -7 & 6 & -1 \\ 1 & 0 & -1 \\ 1 & -2 & 1 \end{bmatrix}$$

Then,

$$A^{-1} = \frac{1}{detA}adjA = \frac{1}{-2}\begin{bmatrix} -7 & 6 & -1 \\ 1 & 0 & -1 \\ 1 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 3.5 & -3 & 0.5 \\ -0.5 & 0 & 0.5 \\ -0.5 & 1 & -0.5 \end{bmatrix}$$

Check with MATLAB:

```
%-----------------------------------------------------------
clc
clear
A=[1 2 3; 1 3 4; 1 4 3];
invA=inv(A)
%format long;invA
%format short;invA
%------------------------------------------------------------
```

5- حلول المعادلات الأنية باستخدام المصفوفات ((Solution of Simultaneous Equations with Matrices))

Example C.16

For the system of the equations

$$\begin{Bmatrix} 2x_1 + 3x_2 + x_3 = 9 \\ x_1 + 2x_2 + 3x_3 = 6 \\ 3x_1 + x_2 + 2x_3 = 8 \end{Bmatrix}$$

compute the unknowns $x_1, x_2,$ and x_3 using the inverse matrix method.

Solution:

In matrix form, the given set of equations is $AX = B$ where

$$A = \begin{bmatrix} 2 & 3 & 1 \\ 1 & 2 & 3 \\ 3 & 1 & 2 \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad B = \begin{bmatrix} 9 \\ 6 \\ 8 \end{bmatrix}$$

Then,

$$X = A^{-1}B$$

or

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 \\ 1 & 2 & 3 \\ 3 & 1 & 2 \end{bmatrix}^{-1} \begin{bmatrix} 9 \\ 6 \\ 8 \end{bmatrix}$$

Next, we find the determinant $detA$, and the adjoint $adjA$

$$detA = 18 \quad and \quad adjA = \begin{bmatrix} 1 & -5 & 7 \\ 7 & 1 & -5 \\ -5 & 7 & 1 \end{bmatrix}$$

Therefore,

$$A^{-1} = \frac{1}{detA} adjA = \frac{1}{18} \begin{bmatrix} 1 & -5 & 7 \\ 7 & 1 & -5 \\ -5 & 7 & 1 \end{bmatrix}$$

we obtain the solution as follows.

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \frac{1}{18} \begin{bmatrix} 1 & -5 & 7 \\ 7 & 1 & -5 \\ -5 & 7 & 1 \end{bmatrix} \begin{bmatrix} 9 \\ 6 \\ 8 \end{bmatrix} = \frac{1}{18} \begin{bmatrix} 35 \\ 29 \\ 5 \end{bmatrix} = \begin{bmatrix} 35/18 \\ 29/18 \\ 5/18 \end{bmatrix} = \begin{bmatrix} 1.94 \\ 1.61 \\ 0.28 \end{bmatrix}$$

Check with MATLAB:

```
%----------------------------------------------------------
clc
clear
A=[2 3 1; 1 2 3; 3 1 2];
B=[9 6 8]';
X=A\B
M=inv(A)*B
%-----------------------------------------------------------
```

Example C.17

For the electric circuit of Figure C.1,

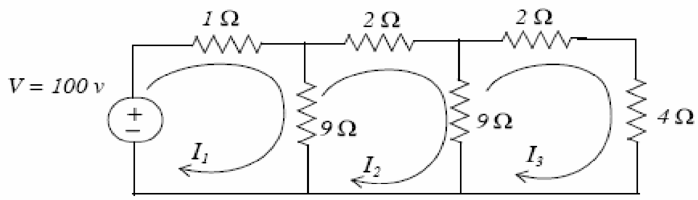


Figure C.1. Circuit for Example C.17

the loop equations are

$$\begin{aligned} 10I_1 - 9I_2 \quad\quad &= 100 \\ -9I_1 + 20I_2 - 9I_3 &= 0 \\ -9I_2 + 15I_3 &= 0 \end{aligned}$$

Use the inverse matrix method to compute the values of the currents I_1, I_2, and I_3

Solution

For this example, the matrix equation is $RI = V$ or $I = R^{-1}V$, where

$$R = \begin{bmatrix} 10 & -9 & 0 \\ -9 & 20 & -9 \\ 0 & -9 & 15 \end{bmatrix}, \quad V = \begin{bmatrix} 100 \\ 0 \\ 0 \end{bmatrix} \quad and \quad I = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix}$$

The next step is to find R^{-1}. This is found from the relation

$$R^{-1} = \frac{1}{detR} adjR$$

Therefore, we find the determinant and the adjoint of R. For this example, we find that

$$detR = 975, \qquad adjR = \begin{bmatrix} 219 & 135 & 81 \\ 135 & 150 & 90 \\ 81 & 90 & 119 \end{bmatrix}$$

Then,

$$R^{-1} = \frac{1}{detR} adjR = \frac{1}{975} \begin{bmatrix} 219 & 135 & 81 \\ 135 & 150 & 90 \\ 81 & 90 & 119 \end{bmatrix}$$

and

$$I = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \frac{1}{975} \begin{bmatrix} 219 & 135 & 81 \\ 135 & 150 & 90 \\ 81 & 90 & 119 \end{bmatrix} \begin{bmatrix} 100 \\ 0 \\ 0 \end{bmatrix} = \frac{100}{975} \begin{bmatrix} 219 \\ 135 \\ 81 \end{bmatrix} = \begin{bmatrix} 22.46 \\ 13.85 \\ 8.31 \end{bmatrix}$$

Check with MATLAB:

```
%----------------------------------------------------------
clc
clear
R=[10 -9 0; -9 20 -9; 0 -9 15];
V=[100 0 0]';
I=R\V;
disp('I1=');
disp(I(1))
disp('I2=');
disp(I(2))
disp('I3=');
disp(I(3))
M=inv(R)*V
%-----------------------------------------------------------
```

Example C.18

For the phasor circuit of Figure C.18

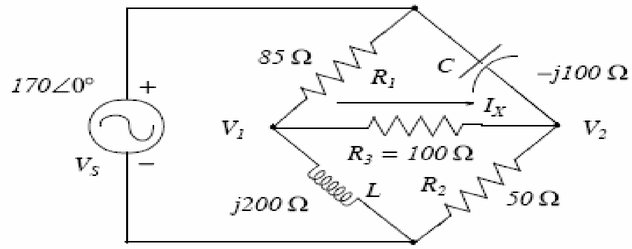


Figure C.3. Circuit for Example C.18

the current I_X can be found from the relation

$$I_X = \frac{V_1 - V_2}{R_3}$$

and the voltages V_1 and V_2 can be computed from the nodal equations

$$\frac{V_1 - 170\angle 0°}{85} + \frac{V_1 - V_2}{100} + \frac{V_1 - 0}{j200} = 0$$

and

$$\frac{V_2 - 170\angle 0°}{-j100} + \frac{V_2 - V_1}{100} + \frac{V_2 - 0}{50} = 0$$

Compute, and express the current I_x in both rectangular and polar forms by first simplifying like terms, collecting, and then writing the above relations in matrix form as $YV = I$, where $Y = Admittance$, $V = Voltage$, and $I = Current$

Solution:

The Y matrix elements are the coefficients of V_1 and V_2. Simplifying and rearranging the nodal equations , we get

$$(0.0218 - j0.005)V_1 - 0.01V_2 = 2$$
$$-0.01V_1 + (0.03 + j0.01)V_2 = j1.7$$

Next, we write in matrix form as

$$\underbrace{\begin{bmatrix} 0.0218 - j0.005 & -0.01 \\ -0.01 & 0.03 + j0.01 \end{bmatrix}}_{Y} \underbrace{\begin{bmatrix} V_1 \\ V_2 \end{bmatrix}}_{V} = \underbrace{\begin{bmatrix} 2 \\ j1.7 \end{bmatrix}}_{I}$$

where the matrices Y, V, and I are as indicated.

```
V1 = 1.0490e+002 + 4.9448e+001i
V2 = 53.4162 + 55.3439i
```

Therefore, in polar form

$$I_X = 0.518\angle -6.53°$$

Check with MATLAB:

```
%---------------------------------------------------------
clc
clear
Y=[0.0218-0.005j -0.01;-0.01 0.03+0.01j];   % Define Y,
I=[2; 1.7j];                                % Define  I,
V=Y\I;                                      % Find V
M=inv(Y)*I;
fprintf('\n');                              % Insert a line
disp('V1 = ');
disp(V(1));                                 % Display values of V1
disp('V2 = ');
disp(V(2));                                 % Display values of  V2
R3=100;
IX=(V(1)-V(2))/R3                           % Compute the value of IX
magIX=abs(IX)                               % Compute the magnitude
of IX
thetaIX=angle(IX)*180/pi                    % Compute angle theta in
degrees
%----------------------------------------------------------
```

Example 1.

Simplify the complex number z and express it both in rectangular and polar form.

$$z = \frac{(3+j4)(5+j2)(2\angle 60^0)}{(3+j6)(1+j2)}$$

Solution:

```
%-------------------------------------------------------------------
                          % Evaluation of Z
                 % the complex numbers are entered
%-------------------------------------------------------------------
clc
Z1 = 3+4*j;
Z2 = 5+2*j;
theta = (60/180)*pi;          % angle in radians
Z3 = 2*exp(j*theta);
Z4 = 3+6*j;
Z5 = 1+2*j;
%-------------------------------------------------------------------
            % Z_rect is complex number Z in rectangular form
disp('Z in rectangular form is');    % displays text inside brackets
Z_rect = Z1*Z2*Z3/(Z4+Z5)
Z_mag = abs (Z_rect);                       % magnitude of Z
Z_angle = angle(Z_rect)*(180/pi);           % Angle in degrees
disp('complex number Z in polar form, mag, phase'); % displays text
                       %inside brackets
Z_polar = [Z_mag, Z_angle]
diary
%-------------------------------------------------------------------
```

Example 1.

Write a function file to solve the equivalent resistance of series connected resistors, R1, R2, R3, …, Rn.

Solution:

```
%-------------------------------------------------------------------
function req = equiv_sr(r)
            % equiv_sr is a function program for obtaining
            % the equivalent resistance of series connected resistors
            % usage: req = equiv_sr(r)
            % r is an input vector of length n
            % req is an output, the equivalent resistance(scalar)
 n = length(r);    % number of resistors
 req = sum (r);    % sum up all resistors
end
%-------------------------------------------------------------------
```

The above MATLAB script can be found in the function file equiv_sr.m, which is available on the disk that accompanies this book.

Suppose we want to find the equivalent resistance of the series connected resistors 10, 20, 15, 16 and 5 ohms. The following statements can be typed in the MATLAB command window to reference the function equiv_sr

```
%--------------------------------------------------------------
clc
a = [10 20 15 16 5];
Rseries = equiv_sr(a)
diary
%---------------------------------------------------------------
```

The result obtained from MATLAB is

Rseries =
    66

Example 1.

Write a MATLAB function to obtain the roots of the quadratic equation

$$ax^2 + bx + c = 0$$

Solution:

```
%-----------------------------------------------------------------
function rt = rt_quad(coef)
         %
         % rt_quad is a function for obtaining the roots of
         % of a quadratic equation
         % usage: rt = rt_quad(coef)
         % coef is the coefficients a,b,c of the quadratic
         % equation ax*x + bx + c =0
         % rt are the roots, vector of length 2
         % coefficient a, b, c are obtained from vector coef
     a = coef(1); b = coef(2); c = coef(3);
     int = b^2 - 4*a*c;
if int > 0
         srint = sqrt(int);
         x1= (-b + srint)/(2*a);
         x2= (-b - srint)/(2*a);
elseif int == 0
         x1= -b/(2*a);
         x2= x1;
elseif int < 0
         srint = sqrt(-int);
         p1 = -b/(2*a);
         p2 = srint/(2*a);
      x1 = p1+p2*j;
       x2 = p1-p2*j;
end
   rt =[x1;x2];
end
%-----------------------------------------------------------------
```

We can use m-file function, rt_quad, to find the roots of the following quadratic equations:

(a) $x^2 + 3x + 2 = 0$ (b) $x^2 + 2x + 1 = 0$ (c) $x^2 - 2x + 3 = 0$

```
%---------------------------------------------------
clc
%diary ex1.dat
ca = [1 3 2];
ra = rt_quad(ca)
cb = [1 2 1];
rb = rt_quad(cb)
cc = [1 -2 3];
rc = rt_quad(cc)
diary
%---------------------------------------------------
```

```
%----aX^2+bX+c=0----------------------------------------------------
clear
clc
close all
a = input( ' a = ');
b = input( ' b = ');
c = input( ' c = ');
x1 = ( - b + sqrt( b^2-4*a*c))/(2*a)
x2 = ( - b + sqrt( b^2-4*a*c))/(2*a)
if imag(x1)==0 & imag(x2)==0
       if x1==x2
           str='ident'
       else
          str= 'real'
       end
elseif real(x1)==0 & real(x2)==0
    str='imag'
else
    str='comp'
end
bigstr=['(x1=',num2str(x1),')--','(x2=',num2str(x2),')--' ,str];
msgbox(bigstr)
%---------------------------------------------------------------------
```

مثال

برنامج لقياس الوقت الي تستغرقه للوصول لبلد علي بعد 800 كيلومتر يعني اننا سندخل طريقة المواصلات هل هى عربة أم حافلة أم طائرة, العربة تسير بسرعة 120 كيلومتر/ساعة و الحافلة بسرعة 80 كيلومتر/ساعة و الطائرة بسرعة 200 كيلومتر/ساعة

الحل

```
%---------------------------------------------------------------------

clear
clc
close all
a=input('enter your transportation method :','s');
switch a
case 'car'
    t=800/120
    msgbox(['your trip will take ',num2str(t),' hours']);
case 'bus'
    t=800/80
    msgbox(['your trip will take ',num2str(t),' hours']);
case 'plane'
    t=800/200
    msgbox(['your trip will take ',num2str(t),' hours']);
otherwise
    msgbox('inter valed tm')
end

%---------------------------------------------------------------------
```

1- تمثيل ((the **for** loops)) فى الماتلاب

Repeating with for loops

Syntax of the **for** loop is shown below

```
for k = array
    commands
end
```

The commands between the **for** and **end** statements are executed for all values stored in the **array**.

Example 1

Suppose that one-need values of the sine function at eleven evenly spaced points $\pi n/10$, for $n = 0, 1, \ldots, 10$. To generate the numbers in question one can use the **for** loop

Solution

```
%-----------------------------------------
clc
for n=0:10
    x(n+1) = sin(pi*n/10);
end
x
%----------------------------------------
%-----------------------------------------
clc
H = zeros(5);
  for k=1:5
     for l=1:5
         H(k,l) = 1/(k+l-1);
     end
  end
H
%----------------------------------------
%-----------------------------------------
clc
A = zeros(10);
for k=1:10
       for l=1:10
           A(k,l) = sin(k)*cos(l);
       end
end
%-----------------------------------------
k = 1:10;
A = sin(k)'*cos(k);
%-----------------------------------------
```

2- تمثيل ((the **while** loops)) فى الماتلاب

Repeating with while loops

Syntax of the **while** loop is

```
while expression
  statements
end
```

This loop is used when the programmer does not know the number of repetitions a priori.

Example 1

This process is continued till the current quotient is less than or equal to 0.01. What is the largest quotient that is greater than 0.01?

Solution

```
%-------------------------------------------
clc
q = pi;
  while q > 0.01
       q = q/2;
  end
q
%-------------------------------------------
```

3- تمثيل ((the **if-else-end** constructions)) فى الماتلاب

The if-else-end constructions

Syntax of the simplest form of the construction under discussion is

```
if expression
  commands
end
```

This construction is used if there is one alternative only. Two alternatives require the construction

```
if expression
  commands (evaluated if expression is true)
else
  commands (evaluated if expression is false)
end
```

If there are several alternatives one should use the following construction

```
if expression1
  commands (evaluated if expression 1 is true)
elseif expression 2
  commands (evaluated if expression 2 is true)
elseif …
.
.
.
else
  commands (executed if all previous expressions evaluate to false)
end
```

Example 1

Chebyshev polynomials $T_n(x)$, $n = 0, 1, \ldots$ of the first kind are of great importance in numerical analysis. They are defined recursively as follows

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \quad n = 2, 3, \ldots, \quad T_0(x) = 1, T_1(x) = x.$$

Implementation of this definition is easy

Solution

```
%-------------------------------------------------------
function  T = chebt(n)
        % Coefficients T of the nth Chebyshev polynomial of the first
kind.
        % They are stored in the descending order of powers.
  t0 = 1;
  t1 = [1 0];
    if n == 0
         T = t0;
       elseif n == 1;
         T = t1;
    else
  for k=2:n
    T = [2*t1 0] - [0 0 t0];
    t0 = t1;
    t1 = T;
  end
end
%-------------------------------------------------------
```

إستدعاء

```
%-------------------------------------------------------
clc
n=3
coff = chebt(n)
diary
%-------------------------------------------------------
```

Thus $T_3(x) = 4x^3 - 3x$.

4- تمثيل ((the **switch-case** constructions)) فى الماتلاب

The switch-case construction

Syntax of the switch-case construction is

```
switch expression (scalar or string)
  case value1 (executes if expression evaluates to value1)
    commands
  case value2 (executes if expression evaluates to value2)
    commands
.
.
.
  otherwise
    statements
end
```

Switch compares the input expression to each case value. Once the match is found it executes the associated commands.

Example 1

In the following example a random integer number x from the set {1, 2, … , 10} is generated. If x = 1 or x = 2, then the message Probability = 20% is displayed to the screen. If x = 3 or 4 or 5, then the message Probability = 30% is displayed, otherwise the message Probability = 50% is generated. The script file fswitch utilizes a switch as a tool for handling all cases mentioned above

Solution

```
%----------------------------------------------------------------------
clc
                      % Script file fswitch.
x = ceil(10*rand);    % Generate a random integer in {1, 2, ... , 10}
  switch x
       case {1,2}
            disp('Probability = 20%');
       case {3,4,5}
            disp('Probability = 30%');
      otherwise
            disp('Probability = 50%');
  end
%----------------------------------------------------------------------
```

Note use of the curly braces{ }after the word **case**. This creates the so-called cell array rather than the one-dimensional array, which requires use of the square brackets[].

5- دوال التقريب ((Rounding to integers. Function ceil, floor, fix and round))

We have already used two MATLAB functions **round** and **ceil** to round real numbers to integers. They are briefly described in the previous sections of this tutorial. A full list of functions designed for rounding numbers is provided below

Function	Description
floor	Round towards minus infinity
ceil	Round towards plus infinity
fix	Round towards zero
round	Round towards nearest integer

Example 1

To illustrate differences between these functions let us create first a two-dimensional array of random numbers that are normally distributed (mean = 0, variance = 1) using another MATLAB function **randn**

Solution

```
%---------------------------------------------------------------------
  clc
randn('seed', 0)   % This sets the seed of the random numbers
                   % generator to zero
  T = randn(5)
  A = floor(T)
  B = ceil(T)
  C = fix(T)
  D = round(T)
%---------------------------------------------------------------------
```

Example 1

In the following m-file functions **floor** and **ceil** are used to obtain a certain representation of a nonnegative real number

Solution

```
%--------------------------------------------------------
function [m, r] = rep4(x)
  % Given a nonnegative number x, function rep4 computes an integer m
  % and a real number r, where 0.25 <= r < 1, such that x = (4^m)*r.
if x == 0
     m = 0;
     r = 0;
  return
end
    u = log10(x)/log10(4);
if u < 0
   m = floor(u)
else
   m = ceil(u);
end
r = x/4^m;
%-------------------------------------------------------------------
```

استدعاء

```
%--------------------------------------------------------------------
  clc
  [m, r] = rep4(pi)
%--------------------------------------------------------------------
```

11-الرسم فى الماتلاب ((MATLAB graphics))

Example 1

In this example the graph of the rational function $f(x) = \dfrac{x}{1+x^2}$, -2 β x β 2, will be plotted using a variable number of points on the graph of f(x)

Solution

```
%-----------------------------------------------------------------------
clc
% Script file graph1.
% Graph of the rational function y = x/(1+x^2).
for n=1:2:5
    n10 = 10*n;
    x = linspace(-2,2,n10);
    y = x./(1+x.^2);
    plot(x,y,'r')
title(sprintf('Graph %g. Plot based upon n = %g points.',(n+1)/2, n10))
    axis([-2,2,-.8,.8])
    xlabel('x')
    ylabel('y')
    grid
    pause(3)
end
%-----------------------------------------------------------------------
clc
    % Script file graph2.
    % Several plots of the rational function y = x/(1+x^2)
    % in the same window.
k = 0;
for n=1:3:10
      n10 = 10*n;
      x = linspace(-2,2,n10);
      y = x./(1+x.^2);
      k = k+1;
 subplot(2,2,k)
 plot(x,y,'r')
 title(sprintf('Graph %g. Plot based upon n = %g points.', k, n10))
 xlabel('x')
 ylabel('y')
 axis([-2,2,-.8,.8])
 grid
 pause(3);
end
%-----------------------------------------------------------------------
```

Example 2

Using command **plot** you can display several curves in the same **Figure Window**.

We will plot two ellipses

$$\frac{(x-3)^2}{36}+\frac{(y+2)^2}{81}=1 \quad \text{and} \quad \frac{(x-7)^2}{4}+\frac{(y-8)^2}{36}=1$$

using command **plot**

Solution

```
x(t) = 3 + 6cos(t), y(t) = -2 + 9sin(t)

x(t) = 7 + 2cos(t), y(t) = 8 + 6sin(t).
```

```
%----------------------------------------------------------------------
clc
% Script file graph3.
% Graphs of two ellipses
% x(t) = 3 + 6cos(t), y(t) = -2 + 9sin(t)
               % and
% x(t) = 7 + 2cos(t), y(t) = 8 + 6sin(t).
t = 0:pi/100:2*pi;
x1 = 3 + 6*cos(t);
y1 = -2 + 9*sin(t);
x2 = 7 + 2*cos(t);
y2 = 8 + 6*sin(t);
plot(x1,y1,'r',x2,y2,'b');
axis([-10 15 -14 20])
xlabel('x')
ylabel('y')
title('Graphs of (x-3)^2/36+(y+2)^2/81 = 1 and (x-7)^2/4+(y-8)^2/36 =1.')
grid
%----------------------------------------------------------------------
```

y	yellow
m	magenta
c	cyan
r	red
g	green
b	blue
w	white
k	black

Example 3

If function **axis** is not used, then the circular curves are not necessarily circular. To justify this let us plot a graph of the unit circle of radius 1 with center at the origin

Solution

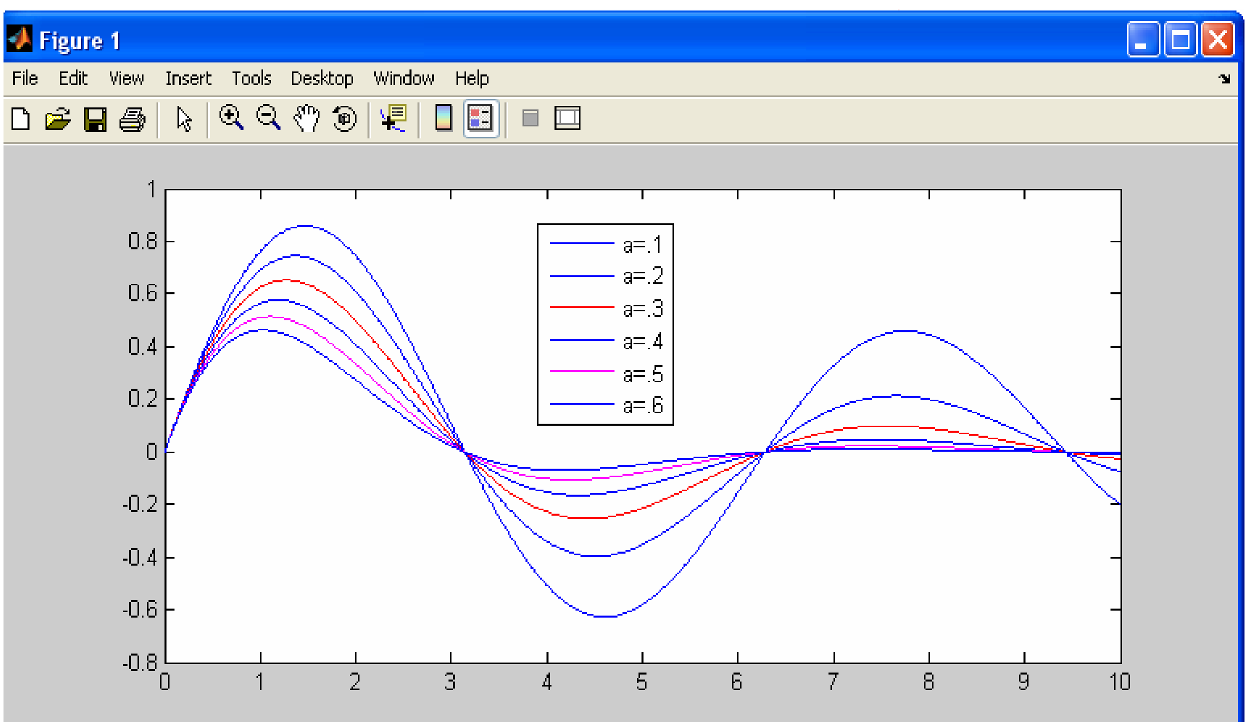
```
%----------------------------------------------------------------------
clc
t = 0:pi/100:2*pi;
x = cos(t);
y = sin(t);
plot(x,y)
%----------------------------------------------------------------------
% Script file graph4.
% Curve r(t) = < t*cos(t), t*sin(t), t >.
t = -10*pi:pi/100:10*pi;
x = t.*cos(t);
y = t.*sin(t);
plot3(x,y,t);
title('Curve u(t) = < t*cos(t), t*sin(t), t >')
xlabel('x')
ylabel('y')
zlabel('z')
grid
%----------------------------------------------------------------------
```

مثال

نقوم برسم وتحميل عدة رسمات فى نفس الشكل ونوضح كل خط بلون معين وقيمته على الرسمة

الحل

```
%------------------------------------------------------------------
clear
  clc
  close all
  x=linspace(0,10,1000);
  a=.1:.1:.6;
  c='b r m c x y';
  for i=1:6
      y=sin(x).*exp(-a(i)*x);
      plot(x,y,c(i))
      hold on
  end
  legend('a=.1','a=.2','a=.3','a=.4','a=.5','a=.6')
  %------------------------------------------------------------------
```



Example

Find first and second derivatives for F(x)=x^2+2x+2

Solution

```
%------To find first and second derivatives of Pn(x)-----
--
clc
a=[1 2 3];
syms x
p=a(1);
for i=1;
    p=a(i+1)+x*p;
end
disp('First derivative')
  p2=p+x*diff(p)
disp('Second derivative')
  p22=diff(p2)
%-------------------------------------------------
--
First derivative


  p2 =
       2+2*x


Second derivative


  p22 =
        2
```

Example

$P4(x)=3x^4-10x^3-48x^2-2x+12$ at $r=6$ deflate the polynomial with Horners algorithm Find $P3(x)$.

Solution

```
%------------Horner alogorithm-----------------
-
clc
a=[3 -10 -48 -2 12];
r=6;
b(1)=a(1);
p=0;
n=length(a);
for i=2:n;
    b(i)=a(i)+r.*b(i-1);
end
syms x
for i=1:n;
    p=p+b(i)*x^(4-i);
end
disp('P3(x)=')
p
%-------------------------------------------------
--
```

```
P3(x)=
        3*x^3+8*x^2-2
```

Numerical Integration

1- *Trapezoidal Rule*

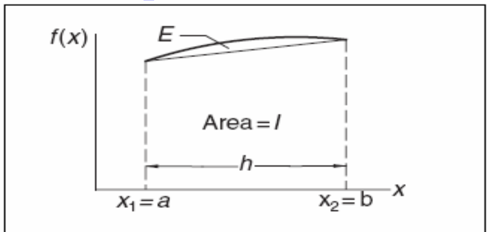


Figure . Trapezoidal rule.

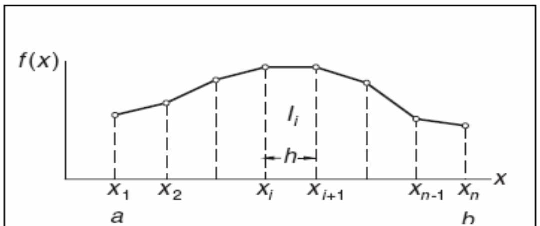


Figure . Composite trapezoidal rule.

The composite trapezoidal rule.

$$I = \sum_{i=1}^{n-1} I_i = [f(x_1) + 2f(x_2) + 2f(x_3) + \cdots + 2f(x_{n-1}) + f(x_n)]\,\frac{h}{2}$$

Example

Suppose we wished to integrate the function trabulated the table below for $f(x)=e^{x}$ over the interval from x=1.8 to x=3.4 using n=8

$$Am=\int_{a}^{b} f(x)dx=\int_{1.8}^{3.4} (e^{x})dx$$

x	1.6	1.8	2	2.2	2.4	2.6	2.8	3	3.2	3.4	3.6	3.8
f(x)	4.953	6.050	7.389	9.025	11.023	13.464	16.445	20.086	24.533	29.964	36.598	44.701

Solution

```
%---Trapezoidal Rule---------------
clc
a=1.8;
b=3.4;
h=0.2;
n=(b-a)/h
f=0;
x=2;
for i=1:n;
    %c=a+(i-1/2)*h;
    %f=f+(c^2+1);
    f=(f+exp(x))
    x=x+h;
end
Am_approx=h/2*(exp(a)+2*f+exp(b))
syms t
Am_exact=int(exp(t),1.8,3.4)
error=Am_exact-Am_approx
E_t=(error/(Am_approx+error))*100
E_a=((Am_approx-Am_exact)/Am_approx)*100
%--------------------------------
```

2- Simpson's 1/3 rule

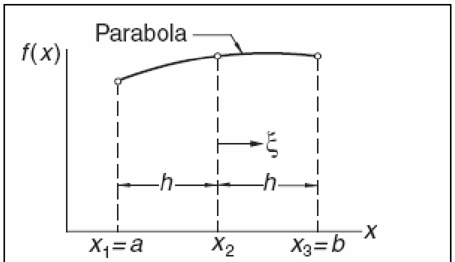


Figure . Simpson's 1/3 rule.

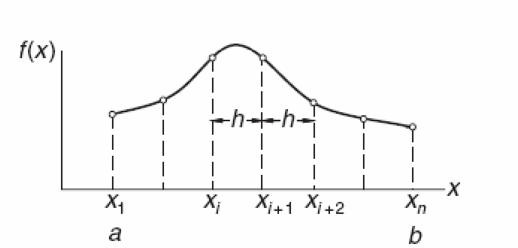


Figure. Composite Simpson's 1/3 rule.

The composite Simpson’s 1/3 rule

$$\int_a^b f(x)\,dx \approx I = [f(x_1) + 4f(x_2) + 2f(x_3) + 4f(x_4) + \cdots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)]\frac{h}{3}$$

Example

Suppose we wished to integrate the function using Simpson’s 1/3 rule and Simpson’s 3/8 rule the table below for $f(x)=e^{x}$ over the interval from x=1.8 to x=3.4 using n=8

$$Am=\int_a^b f(x)dx=\int_{1.8}^{3.4}(e^{x})dx$$

x	1.6	1.8	2	2.2	2.4	2.6	2.8	3	3.2	3.4	3.6	3.8
f(x)	4.953	6.050	7.389	9.025	11.023	13.464	16.445	20.086	24.533	29.964	36.598	44.701

Solution

```
%---Simpson’s 1/3 rule --------------------------------
clc
   a=1.8;
   b=3.4;
   h=0.2;
   n=(b-a)/h
   f=0;
   m=0;
for x=2:(h+h):3.2;
      f=(f+exp(x));
end

for x=2.2:(h+h):3;
      m=(m+exp(x));
end
  Am_approx=h/3*(exp(a)+4*f+2*m+exp(b))
syms t
  Am_exact=int(exp(t),1.8,3.4)
  error=Am_exact-Am_approx
  E_t=(error/(Am_approx+error))*100
  E_a=((Am_approx-Am_exact)/Am_approx)*100
%-----------------------------------------------------
```

3-Simpson’s 3/8 rule

The composite Simpson’s 3/8 rule

$$\int_a^b f(x)\,dx \approx I = [f(x_1) + 3f(x_2) + 3f(x_3) + 2f(x_4) + \cdots + 3f(x_{n-2}) + 3f(x_{n-1}) + f(x_n)]\frac{3h}{8}$$

```
%--------------------Simpson’s 3/8 rule --------------

clc

   a=1.8;

   b=3.4;
```

```
   h=0.2;
   n=(b-a)/h;
   f=0;
   m=0;
%----------------------------------------------------
    for x=2:h:2+h;
        f=f+exp(x)
    end
%----------------------------------------------------
    x=x+h;
    m=exp(x);
%----------------------------------------------------
     for x=2.6:h:2.6+h;
        f=f+exp(x);
     end
%----------------------------------------------------
    x=x+h;
    m=m+exp(x);
    x=x+h;
    f=f+exp(x);
%----------------------------------------------------
    Am_approx=((3*h)/8)*(exp(a)+3*f+2*m+exp(b))
%----------------------------------------------------
syms t
   Am_exact=int(exp(t),1.8,3.4)
   error=Am_exact-Am_approx
   E_t=(error/(Am_approx+error))*100
   E_a=((Am_approx-Am_exact)/Am_approx)*100
%----------------------------------------------------
```

```
%-------------------Simpson’s 3/8 rule -------------
clc
   a=1.8;b=3.4;h=0.2;n=(b-a)/h;f=0;m=0;
%---------------------------------------------------
   for x=2:h:3.2;
      switch x
        case {2,2.2}
             f=f+exp(x)
```

```
        case {2.4}
             m=exp(x);
        case {2.6,2.8}
            f=f+exp(x);
        case {3}
            m=m+exp(x);
        otherwise
           f=f+exp(x);
      end
   end
%----------------------------------------------------
    Am_approx=((3*h)/8)*(exp(a)+3*(f)+2*(m)+exp(b))
%----------------------------------------------------
syms t
   Am_exact=int(exp(t),1.8,3.4)
   pretty(Am_exact)
   error=Am_exact-Am_approx
   pretty(error)
   E_t=(error/(Am_approx+error))*100
   pretty(E_t)
   E_a=((Am_approx-Am_exact)/Am_approx)*100
   pretty(E_a)
%----------------------------------------------------
```

4-Lagrange Interpolating Polynomial Method

Lagrange's interpolation method uses the formula

$$f(x) = \frac{(x-x_1)(x-x_2)\ldots(x-x_n)}{(x_0-x_1)(x_0-x_2)\ldots(x_0-x_n)}f(x_0) + \frac{(x-x_0)(x-x_2)\ldots(x-x_n)}{(x_1-x_0)(x_1-x_2)\ldots(x_1-x_n)}f(x_1)$$
$$+ \frac{(x-x_0)(x-x_1)\ldots(x-x_{n-1})}{(x_n-x_0)(x_n-x_2)\ldots(x_n-x_{n-1})}f(x_n)$$

EXAMPLE

Given the data points

x	0	2	3
y	7	11	28

use Lagrange's method to determine y at $x = 1$.

Solution

$$\ell_1 = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} = \frac{(1-2)(1-3)}{(0-2)(0-3)} = \frac{1}{3}$$
$$\ell_2 = \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} = \frac{(1-0)(1-3)}{(2-0)(2-3)} = 1$$
$$\ell_3 = \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} = \frac{(1-0)(1-2)}{(3-0)(3-2)} = -\frac{1}{3}$$

$$y = y_1\ell_1 + y_2\ell_2 + y_3\ell_3 = \frac{7}{3} + 11 - \frac{28}{3} = 4$$

```
%-----------Lagrange's interpolation method ---------
clc
x=1;
%syms x
%--------------------------------------------------
x1=0;
x2=2;
x3=3;
%--------------------------------------------------
y0=7;
y1=11;
y2=28;
%--------------------------------------------------
l0=((x-x2)*(x-x3))/((x1-x2)*(x1-x3))
l1=((x-x1)*(x-x3))/((x2-x1)*(x2-x3))
l2=((x-x1)*(x-x2))/((x3-x1)*(x3-x2))
%--------------------------------------------------
y=y0*l0+y1*l1+y2*l2
%--------------------------------------------------
```

Example 2

Construct the polynomial interpolating the data by using Lagrange polynomials

X	1	1/2	3
F(x)	3	-10	2

Solution

```
%-----------Lagrange's interpolation method ---------
clc
syms x
%-----------------------------------------------------
x1=1;
x2=0.5;
x3=3;
%-----------------------------------------------------
y0=3;
y1=-10;
y2=2;
%-----------------------------------------------------
l0=((x-x2)*(x-x3))/((x1-x2)*(x1-x3))
l1=((x-x1)*(x-x3))/((x2-x1)*(x2-x3))
l2=((x-x1)*(x-x2))/((x3-x1)*(x3-x2))
%-----------------------------------------------------
y=y0*l0+y1*l1+y2*l2;
collect(y)
%-----------------------------------------------------

%------------Lagrange's interpolation method--------

clc
syms x
p=0;
s=[1 1/2 3];
f=[3 -10 2];
n=length(s);
for i=1:n;
    l=1;
    for j=1:n;
        if (i~=j);
            l=((x-s(j))/(s(i)-s(j)))*l;
        end
      end
  p=l.*f(i)+p;
end
p=collect(p)

%-----------------------------------------------------
```

Example 2

Construct the polynomial interpolating the data by using Lagrange polynomials

X	1	1/2	3
F(x)	3	-10	2

Solution

```
%------------Lagrange's interpolation method--------
clc
x=input(' enter value of x:')
p=0;
s=[1 1/2 3];
f=[3 -10 2];
n=length(s);
for i=1:n;
    l=1;
    for j=1:n;
        if (i~=j);
            l=((x-s(j))/(s(i)-s(j)))*l;
        end
      end
  p=l.*f(i)+p;
end
p;
fprintf('\n p(%3.3f)=%5.4f',x,p)
%---------------------------------------------------
syms x
p=0;
for i=1:n;
    l=1;
    for j=1:n;
        if (i~=j);
            l=((x-s(j))/(s(i)-s(j)))*l;
        end
      end
  p=l.*f(i)+p;
end
p=collect(p)
%---------------------------------------------------
```

p = -283/10 -53/5 *x^2 + 419/10 *x

enter value of x:5

x =5

p(5.000)=-83.8000

%---

Example 3

Find the area by lagrange polynomial using 3 nodes

X	1.8	2.6	3.4
F(x)	6.04964	13.464	29.964

Solution

```
%-----------Lagrange’s interpolation method ---------
clc
syms x
%----------------------------------------------------
x1=1.8;
x2=2.6;
x3=3.4;
%----------------------------------------------------
F0=6.04964;
F1=13.464;
F2=29.964;
%----------------------------------------------------
l0=((x-x2)*(x-x3))/((x1-x2)*(x1-x3))
A0=int(l0,1.8,3.4)
l1=((x-x1)*(x-x3))/((x2-x1)*(x2-x3))
A1=int(l1,1.8,3.4)
l2=((x-x1)*(x-x2))/((x3-x1)*(x3-x2))
A2=int(l2,1.8,3.4)
%----------------------------------------------------
F=F0*A0+F1*A1+F2*A2
collect(F)
%----------------------------------------------------
```

```
%------------Lagrange's interpolation method---
clc
syms x
format long
p=0;
s=[1.8 2.6 3.4];
f=[6.04964 13.464 29.964];
n=length(s);
for i=1:n;
    l=1;
    for j=1:n;
        if (i~=j);
            l=((x-s(j))/(s(i)-s(j)))*l;
        end
    end
  A=int(l,s(1),s(n))
  p=A*f(i)+p;
end
 p
%-----------------------------------------------
```

5-Mid Point Rule

Example

Find the mid point approximation for

$$Am=\int_{a}^{b} f(x)dx=\int_{-1}^{2}(x^2+1)dx$$

using n=6

Solution

```
%---Mid Point Rule---------------
clc
a=-1;
b=2;
n=6;
h=(b-a)/n;
f=0;
for i=1:n;
    c=a+(i-1/2)*h;
    f=f+(c^2+1);
end
Am=h*f
%--------------------------------
```

6- Taylor series

A function $f(x)$ which possesses all derivatives up to order n at a point $x = x_0$ can be expanded in a *Taylor series* as

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + \frac{f''(x_0)}{2!}(x-x_0)^2 + \ldots + \frac{f^{(n)}(x_0)}{n!}(x-x_0)^n$$

If $x_0 = 0$, reduces to

$$f(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \ldots + \frac{f^{(n)}(0)}{n!}x^n$$

Example

Compute the first three terms of the Taylor series expansion for the function

$$y = f(x) = \tan x$$

at $a = \pi/4$.

Solution:

The Taylor series expansion about point a is given by

$$f_n(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \dots$$

and since we are asked to compute the first three terms, we must find the first and second derivatives of $f(x) = \tan x$.

From math tables, $\frac{d}{dx}\tan x = \sec^2 x$, so $f'(x) = \sec^2 x$. To find $f''(x)$ we need to find the first derivative of $\sec^2 x$, so we let $z = \sec^2 x$. Then, using $\frac{d}{dx}\sec x = \sec x \cdot \tan x$, we get

$$\frac{dz}{dx} = 2\sec x \frac{d}{dx}\sec x = 2\sec x(\sec x \cdot \tan x) = 2\sec^2 x \cdot \tan x$$

Next, using the trigonometric identity

$$\sec^2 x = \tan^2 x + 1$$

and by substitution , we get,

$$\frac{dz}{dx} = f''(x) = 2(\tan^2 x + 1)\tan x$$

Now, at point $a = \pi/4$ we have:

$$f(a) = f\left(\frac{\pi}{4}\right) = \tan\left(\frac{\pi}{4}\right) = 1 \quad f'(a) = f'\left(\frac{\pi}{4}\right) = 1 + 1 = 2 \quad f''(a) = f''\left(\frac{\pi}{4}\right) = 2(1^2 + 1)1 = 4$$

and by substitution into (6.125),

$$f_n(x) = 1 + 2\left(x - \frac{\pi}{4}\right) + 2\left(x - \frac{\pi}{4}\right)^2 + \dots$$

We can also obtain a Taylor series expansion with the MATLAB **taylor(f,n,a)** function where **f** is a symbolic expression, **n** produces the first n terms in the series, and **a** defines the Taylor approximation about point a.

The following MATLAB script computes the first 8 terms of the Taylor series expansion of $y = f(x) = \tan x$ about $a = \pi/4$.

```
%------------------- Taylor series -------------
clc
   a=pi/4;
   sym x
   y=tan(x);
   z=taylor(y,8,a);
   pretty(z)
%-----------------------------------------------------
```

Example

Express the function

$$y = f(t) = e^t$$

in a Maclaurin's series.

Solution:

A Maclaurin's series has the form, that is,

$$f(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \ldots + \frac{f^{(n)}(0)}{n!}x^n$$

For this function, we have $f(t) = e^t$ and thus $f(0) = 1$. Since all derivatives are e^t, then, $f'(0) = f''(0) = f'''(0) = \ldots = 1$ and therefore,

$$f_n(t) = 1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} + \ldots$$

MATLAB displays the same result.

```
%------------------- Taylor series -------------
clc
  syms t
  fn=taylor(exp(t));
  pretty(fn)
%-----------------------------------------------------
```

Numerical Differentiation

1-Finite Difference Approximations

The derivation of the finite difference approximations for the derivatives of $f(x)$ are based on forward and backward Taylor series expansions of $f(x)$ about x, such as

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)}(x) + \cdots \quad \text{(a)}$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2!}f''(x) - \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)}(x) - \cdots \quad \text{(b)}$$

$$f(x+2h) = f(x) + 2hf'(x) + \frac{(2h)^2}{2!}f''(x) + \frac{(2h)^3}{3!}f'''(x) + \frac{(2h)^4}{4!}f^{(4)}(x) + \cdots \quad \text{(c)}$$

$$f(x-2h) = f(x) - 2hf'(x) + \frac{(2h)^2}{2!}f''(x) - \frac{(2h)^3}{3!}f'''(x) + \frac{(2h)^4}{4!}f^{(4)}(x) - \cdots \quad \text{(d)}$$

We also record the sums and differences of the series:

$$f(x+h) + f(x-h) = 2f(x) + h^2 f''(x) + \frac{h^4}{12}f^{(4)}(x) + \cdots \quad \text{(e)}$$

$$f(x+h) - f(x-h) = 2hf'(x) + \frac{h^3}{3}f'''(x) + \cdots \quad \text{(f)}$$

$$f(x+2h) + f(x-2h) = 2f(x) + 4h^2 f''(x) + \frac{4h^4}{3}f^{(4)}(x) + \cdots \quad \text{(g)}$$

$$f(x+2h) - f(x-2h) = 4hf'(x) + \frac{8h^3}{3}f'''(x) + \cdots \quad \text{(h)}$$

First Central Difference Approximations

The solution of Eq. (f) for $f'(x)$ is

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{6}f'''(x) - \cdots$$

Keeping only the first term on the right-hand side, we have

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + \mathcal{O}(h^2)$$

which is called the *first central difference approximation* for $f'(x)$. The term $\mathcal{O}(h^2)$ reminds us that the truncation error behaves as h^2.

From Eq. (e) we obtain

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + \frac{h^2}{12}f^{(4)}(x) + \cdots$$

or

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + \mathcal{O}(h^2)$$

Central difference approximations for other derivatives can be obtained from Eqs. (a)–(h) in a similar manner. For example, eliminating $f'(x)$ from Eqs. (f) and (h) and solving for $f'''(x)$ yield

$$f'''(x) = \frac{f(x+2h) - 2f(x+h) + 2f(x-h) - f(x-2h)}{2h^3} + \mathcal{O}(h^2)$$

The approximation

$$f^{(4)}(x) = \frac{f(x+2h) - 4f(x+h) + 6f(x) - 4f(x-h) + f(x-2h)}{h^4} + \mathcal{O}(h^2)$$

First Noncentral Finite Difference Approximations

These expressions are called *forward* and *backward* finite difference approximations.

Noncentral finite differences can also be obtained from Eqs. (a)–(h). Solving Eq. (a) for $f'(x)$ we get

$$\boxed{f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2}f''(x) - \frac{h^2}{6}f'''(x) - \frac{h^3}{4!}f^{(4)}(x) - \cdots}$$

Keeping only the first term on the right-hand side leads to the *first forward difference approximation*

$$\boxed{f'(x) = \frac{f(x+h) - f(x)}{h} + \mathcal{O}(h)}$$

Similarly, Eq. (b) yields the *first backward difference approximation*

$$\boxed{f'(x) = \frac{f(x) - f(x-h)}{h} + \mathcal{O}(h)}$$

Note that the truncation error is now $\mathcal{O}(h)$, which is not as good as the $\mathcal{O}(h^2)$ error in central difference approximations.

We can derive the approximations for higher derivatives in the same manner. For example, Eqs. (a) and (c) yield

$$\boxed{f''(x) = \frac{f(x+2h) - 2f(x+h) + f(x)}{h^2} + \mathcal{O}(h)}$$

Second Noncentral Finite Difference Approximations

Finite difference approximations of $\mathcal{O}(h)$ are not popular due to reasons that will be explained shortly. The common practice is to use expressions of $\mathcal{O}(h^2)$. To obtain noncentral difference formulas of this order, we have to retain more terms in the Taylor series. As an illustration, we will derive the expression for $f'(x)$. We start with Eqs. (a) and (c), which are

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{6}f'''(x) + \frac{h^4}{24}f^{(4)}(x) + \cdots$$

$$f(x+2h) = f(x) + 2hf'(x) + 2h^2 f''(x) + \frac{4h^3}{3}f'''(x) + \frac{2h^4}{3}f^{(4)}(x) + \cdots$$

We eliminate $f''(x)$ by multiplying the first equation by 4 and subtracting it from the second equation. The result is

$$f(x+2h) - 4f(x+h) = -3f(x) - 2hf'(x) + \frac{2h^2}{3}f'''(x) + \cdots$$

Therefore,

$$f'(x) = \frac{-f(x+2h) + 4f(x+h) - 3f(x)}{2h} + \frac{h^2}{3}f'''(x) + \cdots$$

or

$$f'(x) = \frac{-f(x+2h) + 4f(x+h) - 3f(x)}{2h} + \mathcal{O}(h^2)$$

This Equation is called the *second forward finite difference approximation*.

EXAMPLE

Use forward difference approximations of oh to estimate the first

% derivative of

fx = -0.1.*x.^4-0.15.*x.^3-0.5.*x.^2-0.25.*x+1.2

solution

```
%---------------------------------------------------------------
% Use forward difference approximations to estimate the first
% derivative of fx=-0.1.*x.^4-0.15.*x.^3-0.5.*x.^2-0.25.*x+1.2
clc
h=0.5;
x=0.5;
x1=x+h
fxx=[-0.1 -0.15 -0.5 -0.25 1.2]
fx=polyval(fxx,x)
fx1=polyval(fxx,x1)
tr_va=polyval(polyder(fxx),0.5)
fda=(fx1-fx)/h
et=(tr_va1-fda)/(tr_va1)*100
%----------------------------------------------------------------
```

EXAMPLE

Comparison of numerical derivative for backward difference and central difference method with true derivative and with standard deviation of 0.025

x = [0:pi/50:pi];

yn = sin(x)+0.025

True derivative=td=cos(x)

solution

```
%--------------------------------------------------------
clc
% Comparison of numerical derivative algorithms
x = [0:pi/50:pi];
n = length(x);
% Sine signal with Gaussian random error
yn = sin(x)+0.025*randn(1,n);
% Derivative of noiseless sine signal
td = cos(x);
% Backward difference estimate noisy sine signal
dynb = diff(yn)./diff(x);
subplot(2,1,1)
plot(x(2:n),td(2:n),x(2:n),dynb,'o')
xlabel('x')
ylabel('Derivative')
axis([0 pi -2 2])
legend('True derivative','Backward difference')
% Central difference
dync = (yn(3:n)-yn(1:n-2))./(x(3:n)-x(1:n-2));
subplot(2,1,2)
plot(x(2:n-1),td(2:n-1),x(2:n-1),dync,'o')
xlabel('x')
ylabel('Derivative')
axis([0 pi -2 2])
legend('True derivative','Central difference')
%--------------------------------------------------------
```

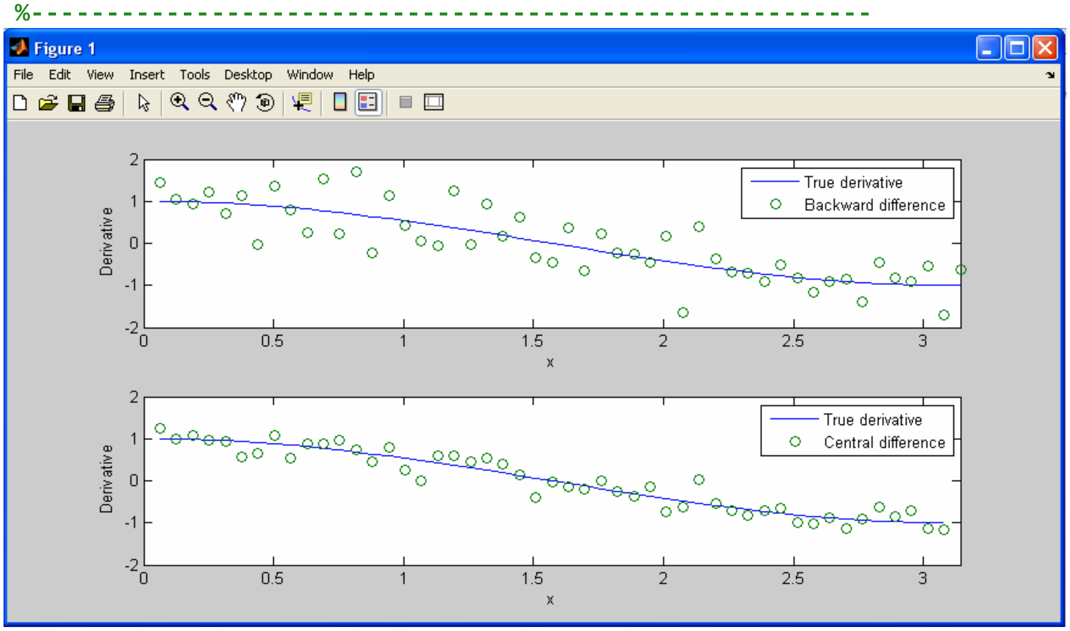


Figure. Comparison of backward difference and central difference methods

Example

Consider a `Divided Difference table` for points following

x	0	0.5	1	1.5
$f(x)$	0.0000	1.1487	2.7183	4.9811

Solution

x_k	$f[x_k]$	$f[x_k, x_{k+1}]$	$f[x_k, .., x_{k+2}]$	$f[x_k, .., x_{k+3}]$
0.0	<u>0.0000</u>			
		<u>2.2974</u>		
0.5	1.1487		<u>0.8418</u>	
		3.1392		<u>0.36306</u>
1.0	2.7183		1.3864	
		4.5256		
1.5	4.9811			

$$
\begin{aligned}
p(x) &= f(x_0) + x - x_0 f[x_0, x_1] + (x - x_0)(x - x_1) f[x_0, x_1, x_2] + (x - x_0)(x - x_1)(x - x_2) f[x_0, x_1, x_2, x_3] \\
&= 0.00 + (x - 0.0)2.2974 + (x - 0.0)(x - 0.5)0.8418 + (x - 0.0)(x - 0.5)(x - 1.0)0.36306 \\
&= 2.05803x + 0.29721x^2 + 0.36306x^3
\end{aligned}
$$

```
%----------------Divided Difference table algorithm-------------
clc
disp('******** divided difference table *********')
x=[2 4 6 8 10]
y=[4.077 11.084 30.128 81.897 222.62]
     f00=y(1);
    for i=1:4
        f1(i)=(y(i+1)-y(i))/(x(i+1)-x(i));
        f01=f1(1);
    end
 f1=[f1(1) f1(2) f1(3) f1(4)]
    for i=1:3
        f2(i)=(f1(i+1)-f1(i))/(x(i+2)-x(i));
        f02=f2(1);
    end
 f2=[f2(1) f2(2) f2(3)]
    for i=1:2
        f3(i)=(f2(i+1)-f2(i))/(x(i+3)-x(i));
        f03=f3(1);
    end
 f3=[f3(1) f3(2)]
    disp('************************************')
y=input('enter value of y:')
p4x=f00+((y-x(1))*f01)+((y-x(1))*(y-x(2))*f02+((y-x(1))*(y-
x(2))*f02))
fprintf('\np4(%3.3f)=%5.4f',y,p4x)
syms y
p4x=f00+((y-x(1))*f01)+((y-x(1))*(y-x(2))*f02+((y-x(1))*(y-
x(2))*f02))
%-------------------------------------------------------------
f1 =  3.5035    9.5220   25.8845   70.3615

f2 =       1.5046    4.0906   11.1193

f3 =              0.4310    1.1714

p4x = -293/100+7007/2000*y+12037/4000*(y-2)*(y-4)
enter value of y:8
y = 8
p4(8.000)=97.3200

% -------------------------------------------------------------
```

Example { H.W }

Find the divided differences (newten's Interpolating) for the data and compare with lagrange interpolating.

X	1	1/2	3
F(x)	3	-10	2

Solution

```
************** divided difference table *****************

f1 =

      26.000000000000000   4.800000000000000


f2 =

               -10.600000000000000

----------------Divided Difference table algorithm-------
----------------{ newtens Interpolating }----------------
enter value of y:5

p4(5.000)=-83.8000
px = -283/10-53/5*y^2+419/10*y


----------------------compare with ---------------------
-------------Lagranges interpolation method--------------
 enter value of x:5

 p(5.000)=-83.8000
 p = -283/10-53/5*m^2+419/10*m
```

```
%------------------------Solve H.W----------------------------
%----------------Divided Difference table algorithm-------------
%----------------{ newten's Interpolating }---------------------
clc
disp('******** divided difference table *********')
x=[1 0.5 3];
y=[3 -10 2];
     f00=y(1);
    for i=1:2;
        f1(i)=(y(i+1)-y(i))/(x(i+1)-x(i));
        f01=f1(1);
    end
 f1=[f1(1) f1(2)]
    for i=1;
        f2(i)=(f1(i+1)-f1(i))/(x(i+2)-x(i));
        f02=f2(1);
    end
 f2=f2(1)
disp('----------------Divided Difference table algorithm-----------')
disp('----------------{ newtens Interpolating }--------------------')
y=input('enter value of y:');
px=f00+((y-x(1))*f01)+((y-x(1))*(y-x(2))*f02);
fprintf('\npx(%3.3f)=%5.4f',y,px)
syms y
px=f00+((y-x(1))*f01)+((y-x(1))*(y-x(2))*f02);
px=collect(px)
%---------------------compare with --------------------------
%-------------Lagrange's interpolation method------------------
disp('---------------------compare with -------------------------')
disp('-------------Lagranges interpolation method------------------')
m=input(' enter value of x:');
p=0;
s=[1 1/2 3];
f=[3 -10 2];
n=length(s);
for i=1:n;
    l=1;
    for j=1:n;
        if (i~=j);
            l=((m-s(j))/(s(i)-s(j)))*l;
        end
      end
  p=l.*f(i)+p;
end
p;
fprintf('\n p(%3.3f)=%5.4f',m,p)
syms m
p=0;
for i=1:n;
    l=1;
    for j=1:n;
        if (i~=j);
            l=((m-s(j))/(s(i)-s(j)))*l;
        end
      end
  p=l.*f(i)+p;
end
p=collect(p)
%--------------------------------------------------------------
```

Example { H.W }

Estimate the In(3) for

Xi	2	4	6
F(x)	In(2)	In(4)	In(6)

a) Linear Interpolation.
B) Quardratic Interpolation
compare between a&b

Solution

a)Linear Interpolation.

F1(x)=f(x0)+((f(x1)-f(x0))/(x1-x0))*(x-x0)

b)Quardratic Interpolation

f2(x)=b0+b1*(x-x0)+b2*(x-x0)*(x-x1)

b0= f(x0) = 0.693147180559945;
b1= (f(x1)-f(x0))/(x1-x0) = 0.346573590279973
b2= ((f(x2)-f(x1))/(x2-x1))-b1/(x2-x0) = -0.035960259056473;

---------a) Linear Interpolation--------------------

fx1 = 0.693147180559945-0.346573590279973 (*X*-2)

inter value x:3

fx1 = 1.039720770839918

---------b) Quardratic Interpolation----------------

fx2 = 0.346573590279973*X*+(-0.035960259056473*X*+0.071920518112945)*(*X*-4)

inter value x:3

fx2 = 1.075681029896391

---------- compare between a&b---------------------
---------a) Linear Interpolation--------------------

Et1 =5.360536964281382 %

---------b) Quardratic Interpolation----------------

Et2 = 2.087293124994937 %

Quardratic Interpolation is better than Linear Interpolation

```
%----------  Solve H.W------------------------------------------
%---------a) Linear Interpolation-----------------------------
%---------b) Quardratic Interpolation------------------------
%----------  compare between a&b----------------------------
clc
x=input('inter value x:');
format long
xi=[2  4  6];
fx=[log(2)  log(4)  log(6)];
disp('---------a) Linear Interpolation-----------------------')
fx1=fx(1)+((fx(2)-fx(1))/(xi(2)-xi(1)))*(x-xi(1))
disp('---------b) Quardratic Interpolation-----------------')
b0=fx(1);
b1=(fx(2)-fx(1))/(xi(2)-xi(1));
b2=(((fx(3)-fx(2))/(xi(3)-xi(2)))-b1)/(xi(3)-xi(1));
fx2=b0+b1*(x-xi(1))+b2*(x-xi(1))*(x-xi(2));
% pretty(fx2)%expand(fx2)%collect(fx2)
disp('----------  compare between a&b----------------------')
Tv=log(3);
disp('---------a) Linear Interpolation-----------------------')
Et1=abs((Tv-fx1)/Tv)*100
disp('---------b) Quardratic Interpolation-----------------')
Et2=abs((Tv-fx2)/Tv)*100
if Et1>Et2;
    disp('Quardratic Interpolation is better than Linear Interpolation')
else
    disp('Linear Interpolation is better than Quardratic Interpolation')
end
syms x
disp('---------a) Linear Interpolation-----------------------')
fx1=fx(1)+((fx(2)-fx(1))/(xi(2)-xi(1)))*(x-xi(1))
disp('---------b) Quardratic Interpolation-----------------')
b0=fx(1);
b1=(fx(2)-fx(1))/(xi(2)-xi(1));
b2=(((fx(3)-fx(2))/(xi(3)-xi(2)))-b1)/(xi(3)-xi(1));
fx2=b0+b1*(x-xi(1))+b2*(x-xi(1))*(x-xi(2))
```

The Bisection Method for Root Approximation

we can compute the midpoint x_m of the interval $x_1 \le x \le x_2$ with

$$x_m = \frac{x_1 + x_2}{2}$$

Knowing x_m, we can find $f(x_m)$. Then, the following decisions are made:

1. If $f(x_m)$ and $f(x_1)$ have the same sign, their product will be positive, that is, $f(x_m) \cdot f(x_1) > 0$. This indicates that x_m and x_1 are on the left side of the *x–axis* crossing as shown in Figure. In this case, we replace x_1 with x_m.

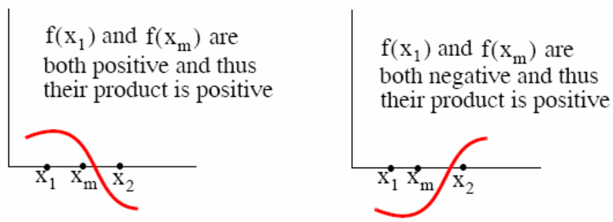


Figure . Sketches to illustrate the bisection method when $f(x_1)$ and $f(x_m)$ have same sign

2. If $f(x_m)$ and $f(x_1)$ have opposite signs, their product will be negative, that is, $f(x_m) \cdot f(x_1) < 0$. This indicates that x_m and x_2 are on the right side of the *x–axis* crossing as in Figure. In this case, we replace x_2 with x_m.

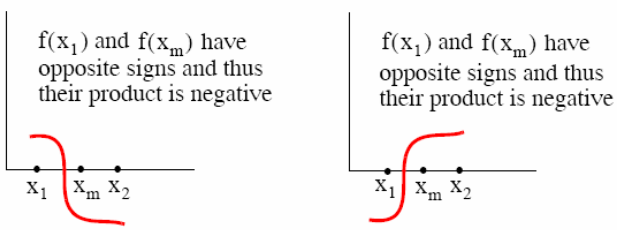


Figure . Sketches to illustrate the bisection method when $f(x_1)$ and $f(x_m)$ have opposite signs

After making the appropriate substitution, the above process is repeated until the root we are seeking has a specified tolerance. To terminate the iterations, we either:

a. specify a number of iterations

b. specify a tolerance on the error of $f(x)$

Example

Use the Bisection Method with MATLAB to approximate one of the roots of

$$y = f(x) = 3x^5 - 2x^3 + 6x - 8$$

by

a. by specifying ***16*** iterations, and using a for end loop MATLAB program

b. by specifying ***0.00001*** tolerance for f(x), and using a while end loop MATLAB program

Solution:

```
%-----------------------------------------------------------------------
function y= funcbisect01(x);
y = 3 .* x .^ 5 - 2 .* x .^ 3 + 6 .* x - 8;
% We must not forget to type the semicolon at the end of the line
above;
% otherwise our script will fill the screen with values of y
%-----------------------------------------------------------------------
```

call for function under name funcbisect01.m

```
%-----------------------------------------------------------------------
clc
x1=1;
x2=2;
disp('-------------------------')
disp('    xm              fm')  % xm is the average of x1 and x2, fm is
f(xm)
disp('-------------------------')         % insert line under xm and
fm
for k=1:16;
f1=funcbisect01(x1); f2=funcbisect01(x2);
xm=(x1+x2) / 2; fm=funcbisect01(xm);
fprintf('%9.6f %13.6f \n', xm,fm)        % Prints xm and fm on same
line;
  if (f1*fm<0)
    x2=xm;
   else
    x1=xm;
  end
end
disp('-------------------------')
x=1:0.05:2;
y = 3 .* x .^ 5 - 2 .* x .^ 3 + 6 .* x - 8;
plot(x,y)
grid
%-----------------------------------------------------------------------
```

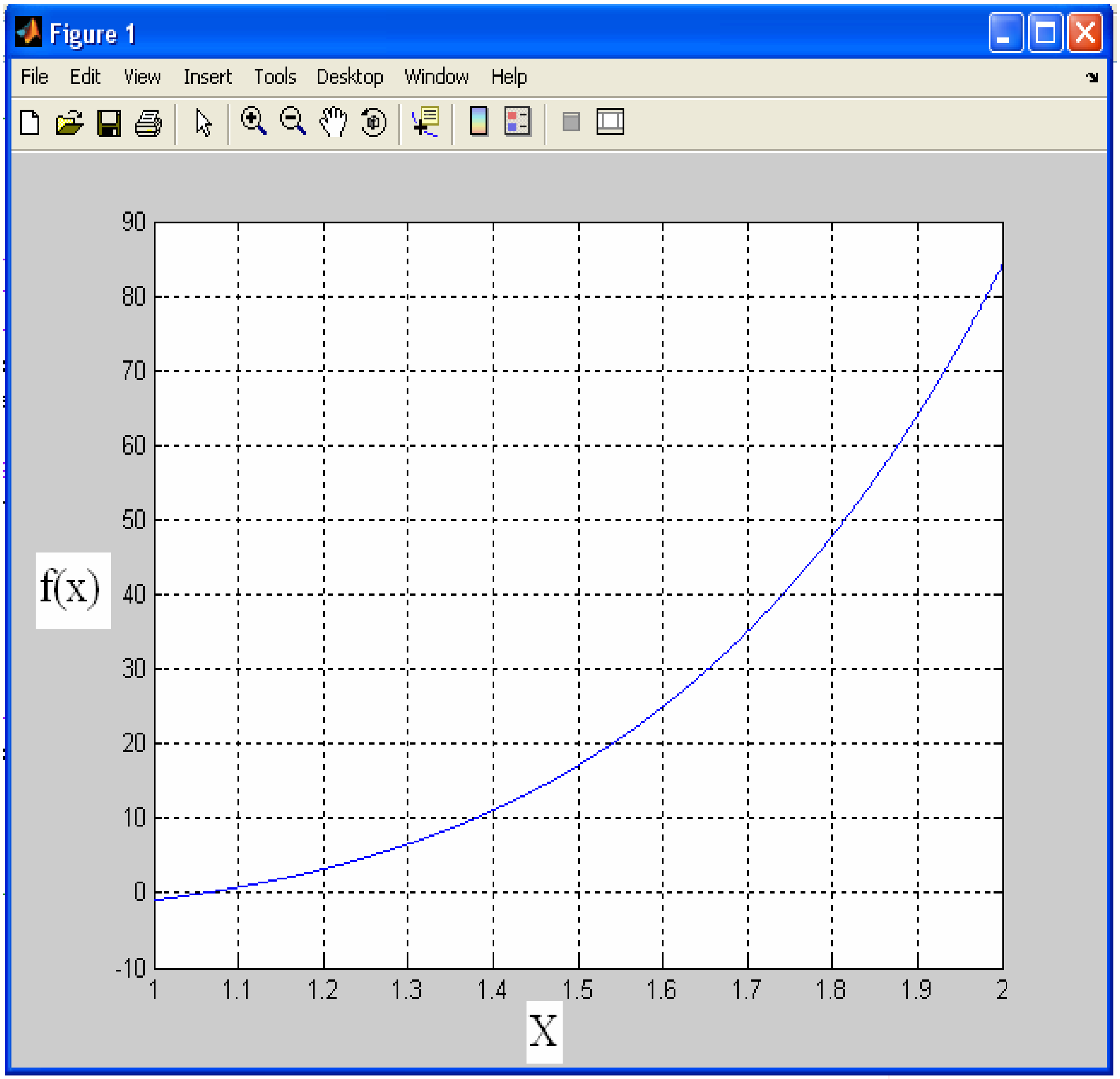


Figure 1
File Edit View Insert Tools Desktop Window Help
90
80
70
60
50
40
30
20
10
0
-10
f(x)
1 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2
X

```
%----------------------------------------------------------------------
function y= funcbisect01(x);
y = 3 .* x .^ 5 - 2 .* x .^ 3 + 6 .* x - 8;
% We must not forget to type the semicolon at the end of the line
above;
% otherwise our script will fill the screen with values of y
%----------------------------------------------------------------------
```

call for function under name funcbisect01.m

```
%----------------------------------------------------------------------
%----------------------------------------------------------------------
clc
x1=1;
x2=2;
tol=0.00001;
disp('-------------------------')
disp(' xm                 fm');
disp('-------------------------')
while (abs(x1-x2)>2*tol);
f1=funcbisect01(x1);
f2=funcbisect01(x2);
xm=(x1+x2)/2;
fm=funcbisect01(xm);
fprintf('%9.6f %13.6f \n', xm,fm);
if (f1*fm<0);
x2=xm;
else
x1=xm;
end
end
disp('-------------------------')
%----------------------------------------------------------------------
-------------------------
 xm               fm
-------------------------
 1.500000     17.031250
 1.250000      4.749023
 1.125000      1.308441
 1.062500      0.038318
 1.031250     -0.506944
 1.046875     -0.241184
 1.054688     -0.103195
 1.058594     -0.032885
 1.060547      0.002604
 1.059570     -0.015168
 1.060059     -0.006289
 1.060303     -0.001844
 1.060425      0.000380
 1.060364     -0.000732
 1.060394     -0.000176
 1.060410      0.000102
-------------------------
```

Example

Use the Bisection Method with MATLAB to approximate one of the roots of (to find the roots of)

Y=f(x)= x.^3-10.*x.^2+5;

That lies in the interval (0.6,0.8) by specifying ***0.00001*** tolerance for f(x), and using a while end loop MATLAB program

Solution:

```
%-------------------------------------------------------------------
function y= funcbisect01(x);
y = x.^3-10.*x.^2+5;
% We must not forget to type the semicolon at the end of the line
above;(% otherwise our script will fill the screen with values of y)
%-------------------------------------------------------------------
```

call for function under name funcbisect01.m

```
%-------------------------------------------------------------------
clc
x1=0.6; x2=0.8;tol=0.00001;
disp('-------------------------')
disp(' xm                  fm');
disp('-------------------------')
while (abs(x1-x2)>2*tol);
f1=funcbisect01(x1);
f2=funcbisect01(x2);
xm=(x1+x2)/2;
fm=funcbisect01(xm);
fprintf('%9.6f %13.6f \n', xm,fm);
if (f1*fm<0);
x2=xm;
else
x1=xm;
end
end
disp('-------------------------')
%-------------------------------------------------------------------
```

xm	fm
0.700000	0.443000
0.750000	-0.203125
0.725000	0.124828
0.737500	-0.037932
0.731250	0.043753
0.734375	0.002987
0.735938	-0.017453
0.735156	-0.007228
0.734766	-0.002120
0.734570	0.000434
0.734668	-0.000843
0.734619	-0.000204
0.734595	0.000115
0.734607	-0.000045

Newton–Raphson Method

The Newton–Raphson formula can be derived from the Taylor series expansion of $f(x)$ about x:

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + O(x_{i+1} - x_i)^2 \tag{a}$$

If x_{i+1} is a root of $f(x) = 0$, Eq. (a) becomes

$$0 = f(x_i) + f'(x_i)\,(x_{i+1} - x_i) + O(x_{i+1} - x_i)^2 \tag{b}$$

Assuming that x_i is a close to x_{i+1}, we can drop the last term in Eq. (b) and solve for x_{i+1}. The result is the Newton–Raphson formula

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

If x denotes the true value of the root, the error in x_i is $E_i = x - x_i$. It can be shown that if x_{i+1} is computed from Eq. (c) , the corresponding error is

$$E_{i+1} = -\frac{f''(x_i)}{2f'(x_i)}E_i^2$$

indicating that the Newton–Raphson method converges *quadratically* (the error is the square of the error in the previous step). As a consequence, the number of significant figures is roughly doubled in every iteration, provided that x_i is close to the root.

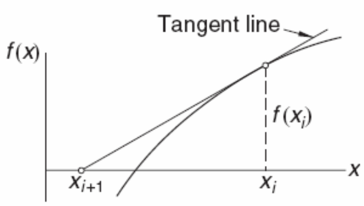


Figure (a) Graphical interpretation of the Newton–Raphson formula.

A graphical depiction of the Newton–Raphson formula is shown in Fig. (a) The formula approximates $f(x)$ by the straight line that is tangent to the curve at x_i. Thus x_{i+1} is at the intersection of the x-axis and the tangent line.

The algorithm for the Newton–Raphson method is simple: it repeatedly applies Eq. (c), starting with an initial value x_0, until the convergence criterion

$$|x_{i+1} - x_1| < \varepsilon$$

is reached, ε being the error tolerance. Only the latest value of x has to be stored. Here is the algorithm:

1. Let x be a guess for the root of $f(x) = 0$.
2. Compute $\Delta x = -f(x)/f'(x)$.
3. Let $x \leftarrow x + \Delta x$ and repeat steps 2-3 until $|\Delta x| < \varepsilon$.

EXAMPLE

A root of $f(x) = x^3 - 10x^2 + 5 = 0$ lies close to $x = 0.7$. Compute this root with the Newton–Raphson method.

Solution

The derivative of the function is $f'(x) = 3x^2 - 20x$, so that the Newton–Raphson formula in Eq. (c) is

$$x \leftarrow x - \frac{f(x)}{f'(x)} = x - \frac{x^3 - 10x^2 + 5}{3x^2 - 20x} = \frac{2x^3 - 10x^2 - 5}{x\,(3x - 20)}$$

It takes only two iterations to reach five decimal place accuracy:

$$x \leftarrow \frac{2(0.7)^3 - 10(0.7)^2 - 5}{0.7\,[3(0.7) - 20]} = 0.735\,36$$

$$x \leftarrow \frac{2(0.735\,36)^3 - 10(0.735\,36)^2 - 5}{0.735\,36\,[3(0.735\,36) - 20]} = 0.734\,60$$

Example

Use the Newton–Raphson Method to estimate the root of f(x)=e^(-x)-x, employing an initial guess of x0=0

$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$	$E_{i+1} = -\frac{f''(x_i)}{2f'(x_i)} E_i^2$

Solution

```
%------Newton-Raphson Method-------------------------
clc
x=[0];
tol=0.0000000007;
format long
for i=1:5;
     fx=exp(-x(i))-x(i);
     fxx=-exp(-x(i))-1 ;
     fxxx=exp(-x(i));
     x(i+1)=x(i)-(fx/fxx);
     T.V(i)=(abs((x(i+1)-x(i))/x(i+1)))*100;
end
for i=1:5;
     e(i)=x(6)-x(i);
     fxx=-exp(-x(6))-1 ;
     fxxx=exp(-x(6));
     e(i+1)=(-fxxx/2*fxx)*(e(i))^2;
end
if abs(x(i+1)-x(i))<tol
    disp(' enough to here')
    disp('------------')
    disp('  X(i+1) ')
    disp('-------------')
    x'
    disp('------------')
    disp('   T.V    ')
    disp('-------------')
    T.V'
    disp('------------')
    disp('   E(i+1)  ')
    disp('-------------')
    e'
    disp('-------------')
end
%----------------------------------------------------
```

```
enough to here
------------------------------
          X(i+1)
------------------------------

                   0
  0.500000000000000
  0.566311003197218
  0.567143165034862
  0.567143290409781
  0.567143290409784

-----------------------------
          T.V
-----------------------------

 1.0e+002 *

  1.000000000000000
  0.117092909766624
  0.001467287078375
  0.000000221063919
  0.000000000000005

-----------------------------
          E(i+1)
-----------------------------

  0.567143290409784
  0.067143290409784
  0.000832287212566
  0.000000125374922
  0.000000000000003
  0.000000000000000

-----------------------------
```

The secant Formula Method

A popular method of hand computation is the *secant formula* where the improved estimate of the root (x_{i+1}) is obtained by linear interpolation based two previous estimates (x_i and x_{i-1}):

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

Example

Use the The secant Formula Method to estimate the root of f(x)=e^(-x)-x, employing an initial guess of x(i-1)=0 & x(0)=0

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

Solution

```
%------The secant Formula Method ------------------------
clc
x=[0 1];
TV=0.567143290409784;
format long
for i=2:6;
    fx=exp(-x(i-1))-x(i-1);
    fxx=exp(-x(i))-x(i);
    x(i+1)=x(i)-((x(i)-x(i-1))*fxx)/(fxx-fx);
    E_T(i)=(abs((TV-x(i+1))/TV))*100;
end
    disp('------------')
    disp('  X(i+1) ')
    disp('-------------')
    x'
    disp('------------')
    disp('    E_T    ')
    disp('-------------')
    E_T'
    disp('------------')

%-----------------------------------------------------
```

```
----------------------
        X(i+1)
----------------------

                    0
   1.000000000000000
   0.612699836780282
   0.563838389161074
   0.567170358419745
   0.567143306604963
   0.567143290409705

---------------------
         E_T
---------------------

                    0
   8.032634281467328
   0.582727734700312
   0.004772693324310
   0.000002855570996
   0.000000000013997

--------------------
```

Example

Use N.R. Quadratically Method to estimate the multiple root of f(x)=x^3-5x^2+7x-3, initial guess of x(0)=0

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

Solution

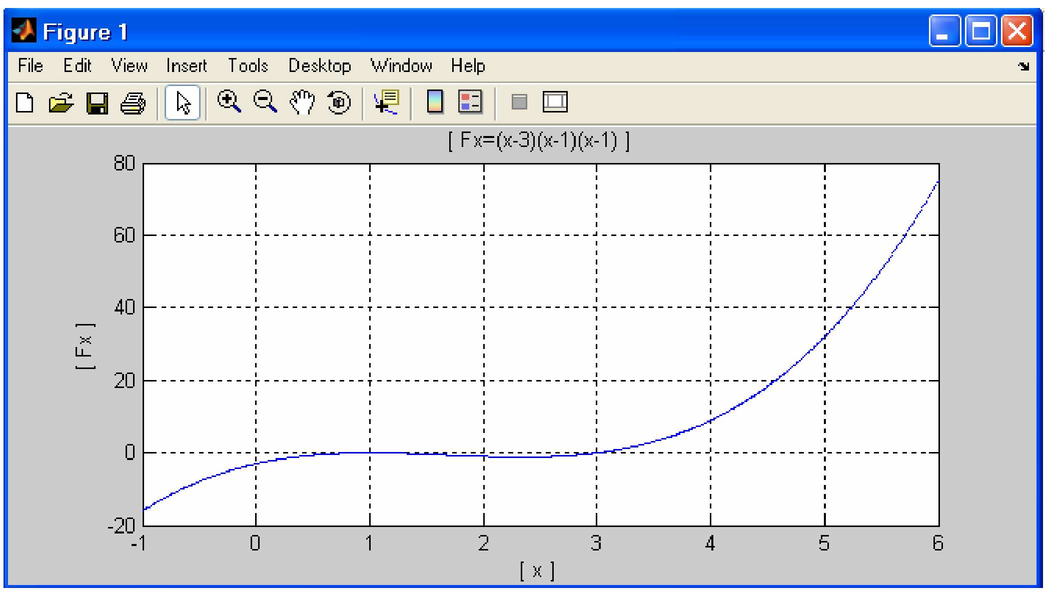
```
%------The N.R. Quadratically  Method   -----------------
clc
TV=1;
x=[0];
format long
for i=1:6;
    fx=x(i)^3-5*x(i)^2+7*x(i)-3
    fxx=3*x(i)^2-10*x(i)+7
    fxxx=6*x(i)-10
    x(i+1)=x(i)-(fx*fxx)/((fxx)^2-fx*fxxx);
    E_T(i)=(abs((TV-x(i+1))/TV))*100;
end
    disp('------------')
    disp('  X(i+1) ')
    disp('-------------')
    x'
    disp('------------')
    disp('   E_T   ')
    disp('-------------')
    E_T'
    disp('------------')

%-----------------------------------------------
%------Multiple Roots---------
%--fx=(x-3)(x-1)(x-1)---------
clc
for x=-1:0.01:6;
   fx=x.^3-5.*x.^2+7.*x-3
   plot(x,fx)
   hold on
end
grid
title('(x-3)(x-1)(x-1)')
xlabel('x')
ylabel('fx')
%-----------------------------
```

X(i+1)

0
1.105263157894737
1.003081664098603
1.000002381493816
1.000000000037312
1.000000000074625
1.000000000074625

E_T

10.526315789473696
0.308166409860333
0.000238149381548
0.000000003731215
0.000000007462475
0.000000007462475



Example

Use the Newton–Raphson Method to estimate the root of f(x)=x^3-5x^2+7x-3, initial guess of x(0)=4

$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$	$E_{i+1} = -\frac{f''(x_i)}{2f'(x_i)} E_i^2$

Solution

```
%------Newton-Raphson Method-------------------------
clc
x=[4];
tol=0.0007;
TV=3;
format long
for i=1:5;
     fx=x(i)^3-5*x(i)^2+7*x(i)-3;
     fxx=3*x(i)^2-10*x(i)+7;
     x(i+1)=x(i)-(fx/fxx);
     E_T(i)=(abs((TV-x(i+1))/TV))*100;
end
for i=1:5;
     e(i)=x(6)-x(i);
     fx=x(i)^3-5*x(i)^2+7*x(i)-3;
     fxx=3*x(i)^2-10*x(i)+7;
     fxxx=6*x(i)-10;
     e(i+1)=(-fxxx/2*fxx)*(e(i))^2;
end
if abs(TV-x(i+1))<tol
    disp(' enough to here')
    disp('-------------')
    disp('  X(i+1) ')
    disp('--------------')
    x'
    disp('-------------')
    disp('   T.V   ')
    disp('--------------')
    E_T'
    disp('-------------')
    disp('   E(i+1)  ')
    disp('--------------')
    e'
    disp('--------------')
end
%-----------------------------------------------------
```

```
    enough to here
---------------------
       X(i+1)
---------------------

   4.000000000000000
   3.400000000000000
   3.100000000000000
   3.008695652173913
   3.000074640791192
   3.000000005570623

---------------------
         T.V
---------------------

  13.333333333333330
   3.333333333333322
   0.289855072463781
   0.002488026373060
   0.000000185687436
   0.000000007462475

--------------------
       E(i+1)
--------------------

  -0.999999994429377
  -0.399999994429377
  -0.099999994429377
  -0.008695646603290
  -0.000074635220569
  -0.000000089144954

---------------------
```

Gauss Elimination Method

Example

Use the Gauss Elimination Method with MATLAB to solve the following equations

2x1+x2-x3=5---------------------------(1)

X1+2x2+4x3=10-----------------------(2)

5x1+4x2-x3=14-----------------------(3)

Solution:

```
%----- Gauss Elimination Method-----------------------------
clc
A=[2 1 -1;1 2 4;5 4 -1];
b=[5 10 14];
if size(b,2) > 1; b = b'; end % b must be column vector
n = length(b);
for k = 1:n-1 % Elimination phase
for i= k+1:n
if A(i,k) ~= 0
lambda = A(i,k)/A(k,k);
A(i,k+1:n) = A(i,k+1:n) - lambda*A(k,k+1:n);
b(i)= b(i) - lambda*b(k);
end
end
end
if nargout == 2; det = prod(diag(A)); end
for k = n:-1:1 % Back substitution phase
b(k) = (b(k) - A(k,k+1:n)*b(k+1:n))/A(k,k);
fprintf('
end
x = b
%---------------------------------------------------------
```

```
x =
     4
    -1
     2
```

ترقبوا المزيد من الشروحات للأمثلة فى التحليل العددى والرياضيات والتحكم الألى والأتصالات وإلكترونات القدرة ونظم التشغيل والدارات التماثلية والنظم الرقمية واسس الألكترونات وغيرها من المواد فى اغلب التخصصات راجين من الله سبحانه وتعالى التوفيق فلله الحمد والمنة وبه التوفيق والعصمة.

وفى الختام نسأل الله التوفيق والسعادة لى ولكم في الدنيا والأخرة . ونسأل الله الهمة في طلب العلم وبدله . واللهم صلى على النبى المصطفى وال بيته الطاهرين والصحابة والتابعين وتابع التابعين ومن تبعهم بإحسان الى يوم الدين. والسلام عليكم ورحمة الله وبركاته.