BIL 108E Intr. to Sci. & Eng.Computing

Res.Asst.Çiğdem Toparlı

EXERCISES -3

Example-1

Evaluate the following MATLAB expressions yourself .

a. 2 / 2 * 3

b. 2 / 3 ˆ 2

c. (2 / 3) ˆ 2

d. 2 + 3 * 4 - 4

e. 2 ˆ 2 * 3 / 4 + 3

f. 2 ˆ (2 * 3) / (4 + 3)

g. 2 * 3 + 4

h. 2 ˆ 3 ˆ 2

h. i. -4 ˆ 2

Example-2

Let X=[2 5 1 6]

a)Add 16 to each element

b) Add 3 to just the odd-index element.

c) Compute the square root of each element.

d) Compute the square of each element.

Solution-2

```
>> clear all
>> X = [ 2 5 1 6 ]

X =

    2     5     1     6

>> X + 16

ans =

   18    21    17    22

>> X(1)

ans =

    2

>> X(3)

ans =

    1

>> X + [ 3 0 3 0 ]

ans =

    5     5     4     6

>> sqrt(X)

ans =

   1.4142    2.2361    1.0000    2.4495

>> X .^ 2

ans =

    4    25     1    36
```

Example-3

Given a vector t, of length n, write down the Matlab expressions that will correctly compute the following.

a) $\ln\left(2+t+t^2\right)$ b) $e^t\left(1+\cos(3t)\right)$ c) $\cos^2 t+\sin^2 t$ d) cot(t) e) $\sec^2(t)+\cot(t)-1$

Solution-3

```
>> n = 8;
>> t = rand(n, 1)

t =

    0.9575
    0.9649
    0.1576
    0.9706
    0.9572
    0.4854
    0.8003
    0.1419

>> log(2 + t + t .^ 2)

ans =

    1.3544
    1.3599
    0.7805
    1.3642
    1.3541
    1.0010
    1.2357
    0.7710

>> exp(t) .* (1 + cos(3 * t))

ans =

    0.0937
    0.0796
    2.2130
    0.0694
    0.0944
    1.8107
    0.5833
    2.2021

>> cos(t).^2 + sin(t).^2

ans =

    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000

>> cot(t)

ans =

    0.7038
    0.6928
    6.2920
    0.6844
    0.7043
    1.8959
    0.9707
    7.0005

>> sec(t) .^ 2 + cot(t) - 1

ans =

    2.7225
    2.7760
    6.3173
    2.8191
    2.7201
    2.1741
    2.0320
    7.0209
```

Example-4

Engineers often have to convert from one unit of measurement to another; this can be tricky sometimes. You need to think through the process carefully. For example, convert 5 acres to hectares, given that an acre is 4840 square yards, a yard is 36 inches, an inch is 2.54 cm, and a hectare is 10000 m^2.

Solution-4

1 yard = 36 inch = 36 * 2.54cm = 91.44 cm
1 yard kare (square yard) = 91.44^2 cm^2 = 8361.2736 cm^2 = 0.83612736 m^2
1 acre = 4840 yard kare = 4840 * 0.83612736m^2 = 4046.8564224m^2
1 hektar = 10000m^2 => 1 acre = 4046.8564224m^2 = 0.40468564224 hektar

1 acre = 0.40468564224 hektar
5 acre = 2.0234282112 hektar

Example-5

Write Matlab code that will evaluate and plot the following functions:

(c) $y = \dfrac{\sin 7x - \sin 5x}{\cos 7x + \cos 5x}$ using 200 equally spaced points on the interval $-1.57 \leq x \leq 1.57$. Use the axis command to scale the plot so that $-2 \leq x \leq 2$ and $-10 \leq y \leq 10$.

Solution-5

```
>> x = linspace(-pi / 2, pi / 2, 200);
>> y = (sin(7 * x) - sin(5 * x)) ./ (cos(7 * x) + cos(5 * x));
>> plot(x, y)
>> axis(-2, 2, -10, 10)
??? Error using ==> axis>LocSetLimits at 234
Vector must have 4, 6, or 8 elements.

Error in ==> axis at 94
            LocSetLimits(ax(j),cur_arg);

>> axis([-2, 2, -10, 10])
>> plot(x, y)
>> axis([-2 2 -10 10])
```

Example-6

It's useful to be able to work out how the period of a bond repayment changes if you increase or decrease your monthly payment P. The formula for the number of years N to repay the loan isgiven by

$$N = \frac{\ln\left(\frac{P}{P - rL/12}\right)}{12\ln(1 + r/12)}$$

How long will it take to pay off the loan of \$50 000 at \$800 a month if the interest remains at 15 percent?

Solution-6

```
>> P = 800;
>> r = 0.15;
>> L = 50000;
>> N = log(P / (P - r * L / 12)) / (12 * log(1 + r / 12))

N =

   10.1954

>> N * 12

ans =

  122.3444
```

Example-7

Calculate the following summation with user input n, and plot S versus k graph.

$$s = \sum_{k=0}^{n} \frac{1}{k^2 + 1}$$

Solution-7

```
clc     % ekrani sil
clear   % butun degiskenleri temizle

% kullanicidan n degiskeninin girilmesi isteniyor
n = input('n degerini gir: ');

% toplam ve Sv degiskenleri tanimlaniyor
toplam = 0;              % toplam bir kumulatif degisken
Sv = zeros(1, n + 1);      % Sv, sigma elemanlarini tutan n+1 boyutunda
                         % sifirlardan olusan bir vektor olusturuluyor

for k = 0:n              % k ya sifirdan baslayip bir arttirarak
                         % n'ye kadar degerler ver
   toplam = toplam + 1 / (k ^ 2 + 1);
   % toplam degiskeninde k'ya sirayla verilen degerler her adimda üst üste
   % toplaniyor.
   Sv(k + 1) = toplam; % Sv de cizdirmek icin eski degerler saklaniyor
end

plot(0:n, Sv) % sifirdan n'ye kadar olan k degerleri x ekseninde

% dikkat: bu programda Sv'ler ilk n terimin toplami olarak tutuluyor. yani
% Sv(1) ilk terim, Sv(2) birinciyle ikincinin toplami, Sv(3) ilk üç terim
% toplami.
```

Example-8

Write an m-file to evaluate the following algebraic formula

$$p(t) = \begin{cases} log\left(t^2 - a\right) & if \quad t^2 > a \\ log\left(t^2\right) & if \quad t^2 \le a \end{cases}$$

where t is a number that a user enters and a = 100.

Solution-8

```
t = input('t degerini girin: ');
a = 100;

if t^2 > a
   p = log10(t ^ 2 - a);       % log10 yada log, her ikisi de dogru
else
   p = log10(t ^ 2);
end

fprintf('p nin degeri: %f olmustur.\n', p);
% yukarida %f ondalikli sayi anlaminda. basindaki 5, toplamda 5
% hane kullanilarak yazilacak noktadan sonraki 3 ise ekranda
% 3 ondaligi goster anlaminda.
```

Example-9

Write an m-file that returns the plot of $y = 5\sqrt{k^{0.5}}$ over the range $1 \leq k \leq 5$ in discrete increments of $\Delta k = 0.5$ as long as $|y| < 8$. Do not forget to label the graph.

Solution-9

```
% sorunun sadece grafigini gösteren matlab komutlari
k = 1:0.5:5            % k birden 5'e kadar 0.5 adimlarla artan bir vektör
y = 5 * sqrt(k .^ 0.5)  % y de bu vektörlere karsilik gelen degerleri tutan vektor
plot(k, y)             % bu vektörleri çizdir.

% sorunun dogru cevabi bu degildir. her adimda y'lerin degerini kontrol
% ettirmek icin for yada while döngüsünü kullanmaya gerek vardir. bu
% sadece örnek olmasi acisindan verilmistir.
```

Solution-9

```
% versiyon1
y = zeros(1, 9);    % zeros'la 9 elemanli bir dizi olustur
c = 1;              % sayac'i birden baslat

for k = 1:0.5:5
   y(c) = 5 * sqrt(k ^ 0.5);    % her adimda hesaplanan y degerini y
dizisinin bir elemanina ata
   if abs(y(c)) >= 8           % eger hesaplanan y'nin mutlak degeri
8'den büyükse
      break;                   % for döngüsünü kir
   end
   c = c + 1;                  % sayaci bir arttir
end

plot(1:0.5:5, y)             % bulunan y degerlerini grafige çiz.
```

Solution-9

```
% versiyon2
y = zeros(1, 9);    % zeros'la 9 elemanli bir dizi olustur
c = 0;              % sayac'i sifirdan baslat

for k = 1:0.5:5
   c = c + 1;      % sayaca ilk adimda bir ekleniyor, boylece dizinin sifirinci
elemanina erismesi engelleniyor
   y(c) = 5 * sqrt(k ^ 0.5);   % k'lara karsilik gelen c'ler hesaplaniyor
   if abs(y(c)) >= 8          % eger hesaplanan k'nin mutlak degeri sekizden
buyukse
      break;                % donguyu kir
   end
end

plot(1:0.5:5, y, '-o');        % hesaplanan y degerlerini grafikte noktalar
yerine
                         % o karakteri koyarak cizdir.
```

Solution-9

% hold on kullanarak çözümü
hold on

for k = 1:0.5:5
   y = 5 * sqrt(k ^ 0.5); % for döngüsündeki k'lara karsilik gelen y
degerlerini hesapla
   if (abs(y) >= 8)   % eger hesaplanan y'nin mutlak degeri
8'den buyukse
      break;   % for döngüsünden çik
   end
   plot(k, y, '-o');   % grafige her adimda k,y nokta çiftlerini
koyarak çiz
end

% sorunun bu sekilde cevabi dogrudur. ancak grafik sadece
noktalardan olusur
% noktalar arasinda cizgiler çizilmez.

Example-10

a) Use a `for-end` loop in a script file to calculate the sum of the first n terms of the series: $\sum_{k=1}^{n} \frac{(-1)^k k}{2^k}$. Execute the script file for $n = 4$ and $n = 20$.

b) The function sin(x) can be written as a Taylor series by:

$$\sin x = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)!}$$

Solution-10

```
n = input('Enter the number of terms ' );
S = 0;
for k = 1:n
   S = S + (-1)^k*k/2^k;
end
fprintf('The sum of the series is: %f',S)
```

Setting the sum to zero.

for-end loop.

In each pass one element of the series is calculated and is added to the sum of the elements from the previous passes.

The summation is done with a loop. In each pass one term of the series is calculated (in the first pass the first term, in the second pass the second term, and so on), and is added to the sum of the previous elements. The file is saved as Exp7-4a and then executed twice in the Command Window:

```
>> Exp7_4a
Enter the number of terms 4
The sum of the series is: -0.125000
>> Exp7_4a
Enter the number of terms 20
The sum of the series is: -0.222216
```

a

Solution-10

b) A user-defined function file that calculates sin(x) by adding n terms of Taylor's formula is shown below.

```
function y = Tsin(x,n)
% Tsin calculates the sin using Taylor formula.
% Input arguments:
% x The angle in degrees, n number of terms.
xr = x*pi/180;                      Converting the angle from degrees to radians.
y = 0;
for k = 0:n - 1
    y = y + (-1)^k*xr^(2*k + 1)/factorial(2*k + 1);      for-end loop.
end
```

The first element corresponds to $k = 0$ which means that in order to add n terms of the series, in the last loop $k = n - 1$. The function is used in the Command Window to calculate the sin(150^o) using 3 and 7 terms:

```
>> Tsin(150,3)                      Calculating sin(150°) with 3 terms of Taylor series.

ans =
    0.6523
>> Tsin(150,7)                      Calculating sin(150°) with 7 terms of Taylor series.
ans =
    0.5000                          The exact value is 0.5.
```

Example-11

Determine the 4th degree polynomial y(x) that passes through the points

$$(0,-1),\ (1,\ 1),\ (3,\ 3),\ \ (5,\ 2) \text{ and } (6,-2)$$

Solution-11

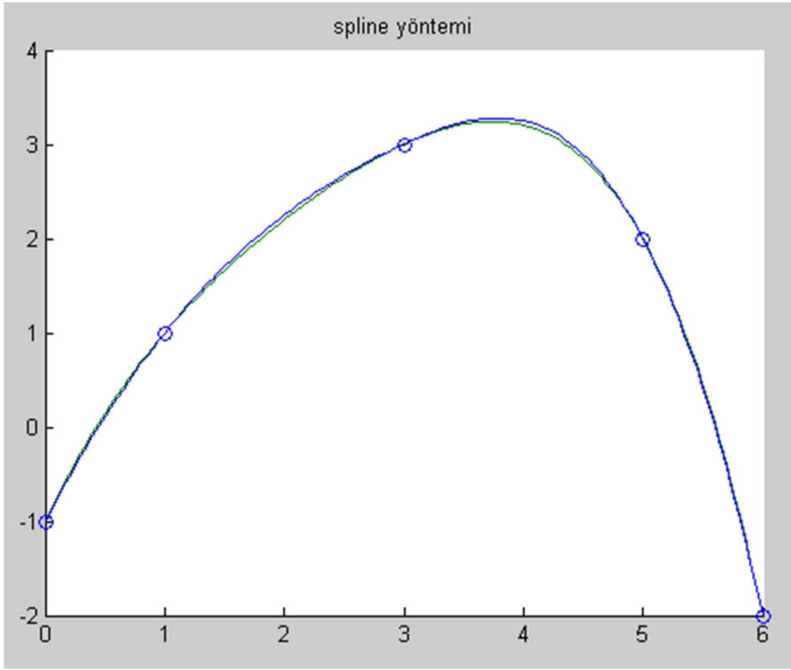
```
clc
clear all

x = [0 1 3 5 6];
y = [ -1 1 3 2 -2 ];

p = polyfit(x, y, 4);
yeni_x = 0:0.05:6;
yeni_y = polyval(p, yeni_x);

hold on
plot(x, y, 'o', yeni_x, yeni_y);

yeni_yspline = interp1(x, y, yeni_x, 'spline');
plot(yeni_x, yeni_yspline);
title('spline yöntemi');
```



Solution-11-çözüm2

```
figure ;
hold on
plot(x, y, 'o', yeni_x, yeni_y);
yeni_ynearest = interp1(x, y, yeni_x, 'nearest');
plot(yeni_x, yeni_ynearest);
title('nearest yöntemi');
figure;
hold on
plot(x, y, 'o', yeni_x, yeni_y);
yeni_ylinear = interp1(x, y, yeni_x, 'linear');
plot(yeni_x, yeni_ylinear);
title('linear yöntemi');
figure;
hold on
plot(x, y, 'o', yeni_x, yeni_y);
yeni_ycubic = interp1(x, y, yeni_x, 'cubic');
plot(yeni_x, yeni_ycubic);
title('cubic yöntemi');
```