MATLAB – HW4 Upload Date: 18 Nov. 2020

Dorreen Rostami – 97243034
Zahra Hashemi – 97243072

4-1)

	Syntax	Description
polyval	y = polyval(p,x)	evaluates the polynomial p at each point in x. The argument p is a vector of length n+1 whose elements are the coefficients (in descending powers) of an nth-degree polynomial.
	[y,delta] = polyval(p,x,S)	uses the optional output structure S produced by polyfit to generate error estimates. delta is an estimate of the standard error in predicting a future observation at x by p(x).
	y = polyval(p,x,[],mu) [y,delta] = polyval(p,x,S,mu)	use the optional output mu produced by polyfit to center and scale the data. mu(1) is mean(x), and mu(2) is std(x). Using these values, polyval centers x at zero and scales it to have unit standard deviation

```
p = [5 6 -1];
x = [1 3 10 2];
y = polyval(p,x)
```

Evaluate the polynomial $5x^2+6x-1$ at the points x = 1,3,10,2

Command Window

```
>> m

y =

    10    62   559    31
```

	Syntax	Description
roots	r = roots(p)	returns the roots of the polynomial represented by p as a column vector.

```
p = [3 -2 -4];
r = roots(p)
```

solve $3x^2-2x-4 = 0$

Command Window

```
>> r

r =

    1.5352
   -0.8685
```

	Syntax	Description
poly	p = poly(r)	returns the coefficients of the polynomial whose roots are the elements of r (a vector).
	p = poly(A)	Where A is an n-by-n matrix, returns the n+1 coefficients of the characteristic polynomial of of the matrix, det(λI – A).

```
p = [1 -3 2];
r = roots(p);
poly = poly(r)
```

Command Window

```
>> m

poly =

     1    -3     2
```

4-2)

```
clear; clc;
p = input('Enter a row vector containing coefficients: ');
%vector of roots
r = roots(p);
%boolean vector that shows which roots are real or not
isReal = (imag(r) == 0);
%vector holding the elements in isReal that were equal to 1
reals = isReal(isReal == 1);
disp("Number of total roots : " + length(r));
disp("Number of real roots : " + length(reals));
disp("Number of imaginary roots : " + (length(r) - length(reals)));
```

Command Window

```
Enter a row vector containing coefficients: [1 2 3 4]
Number of total roots : 3
Number of real roots : 1
Number of imaginary roots : 2
```

4-3)

	Syntax	Description
conv	w = conv(u,v)	returns the convolution of vectors u and v.
	w = conv(u,v,shape)	Where A is an n-by-n matrix, returns the n+1 coefficients of the characteristic polynomial of of the matrix, det(λI – A).

	Syntax	Description
deconv	[q,r] = deconv(u,v)	deconvolves a vector v out of a vector u using long division, and returns the quotient q and remainder r such that u = conv(v,q)+r

```
>> p1=[1,3,7];
>> p2=[1,1];
>> pconv=conv(p1,p2)

pconv =

     1     4    10     7

>> pdeconv=deconv(pconv,p1)

pdeconv =

     1     1

>> pdeconv=deconv(pconv,p2)

pdeconv =

     1     3     7
```

4-4)

```
clc; clear;
N = [1, 5, 11, 13];
D = [1, 2, 4];
[Q,R] = deconv(N,D)
fprintf('Q(s) = %d*s + %d\n',Q(1),Q(2));
fprintf('R(s) = %d*s + %d\n',R(3),R(4));
disp("H(s) = N(s)/D(s) = Q(s) + R(s)/D(S)");
fprintf('(%d*s^3 + %d*s^2 + %d*s + %d)/(%d*s^2 + %d*s + %d) = (%d*s + %d) + (%d*s + %d)/(%d*s^2 + %d*s + %d)\n',...
    N(1),N(2),N(3),N(4),D(1),D(2),D(3),Q(1),Q(2),R(3),R(4),D(1),D(2),D(3));
```

Command Window

```
Q =

     1     3


R =

     0     0     1     1

Q(s) = 1*s + 3
R(s) = 1*s + 1
H(s) = N(s)/D(s) = Q(s) + R(s)/D(S)
(1*s^3 + 5*s^2 + 11*s + 13)/(1*s^2 + 2*s + 4) = (1*s + 3) + (1*s + 1)/(1*s^2 + 2*s + 4)
```

4-5)

	Syntax	Description
polyder	k = polyder(p)	returns the derivative of the polynomial represented by the coefficients in p, $k(x) = \frac{d}{dx}p(x)$.
	k = polyder(a,b)	returns the derivative of the product of the polynomials a and b, $k(x) = \frac{d}{dx}[a(x)b(x)]$
	[q,d] = polyder(a,b)	returns the derivative of the quotient of the polynomials a and b, $\frac{q(x)}{d(x)} = \frac{d}{dx}\left[\frac{a(x)}{b(x)}\right]$

```
>> p = polyder([3,0,2,1])

p =

     9     0     2
```

$3x^3+2x+1 \Rightarrow 9x^2+2$

	Syntax	Description
polyint	q = polyint(p,k)	returns the integral of the polynomial represented by the coefficients in p using a constant of integration k.
	q = polyint(p)	assumes a constant of integration k = 0.

```
>> q = polyint([2,1])

q =

     1     1     0
```

$2x+1 \Rightarrow x^2+x$

```
clc;clear;
N = [1 5 11 13];
D = [1 2 4];
%H(s) = N(s)/D(s)
H = deconv(N, D);
%first derivative
Nd1=polyder(N)
[q1,d1] = polyder(N,D)
Hd1 = polyder(H)
%second derivative
Nd2 = polyder(Nd1)
[q2,x2] = polyder(q1,d1)
Hd2 = polyder(Hd1)
%integral of N and H
Nintegral = polyint(N)
Hintegral = polyint(H)
```

Command Window

```
Nd1 =

     3    10    11

q1 =

     1     4    11    14    18

d1 =

     1     4    12    16    16

Hd1 =

     1

Nd2 =

     6    10

q2 =

     2    10     8   -16   -80   -64

x2 =

     1     8    40   128   304   512   640   512   256

Hd2 =

     0

Nintegral =

    0.2500    1.6667    5.5000   13.0000         0

Hintegral =

    0.5000    3.0000         0
```

4-6)

```
clear; clc;

p = conv([-1, 0, 1], [-1, 0, 1]); % p = (-x^3 + 1)^2
q = [2, 0, -3, 1]; % q = 2x^3 -3x + 1
pq = conv(p, q);  % pq = ((1 - x^3)^2)*(1 - 3x - 2x^3)

% indefinit integral (represented by the coefficents in pq)
indef_int = polyint(pq)

% definit integral over the limits of integration
a = -2;
b = 3;
def_int = diff(polyval(indef_int,[-2 3]))
```

Command Window

```
indef_int =

    0.2500         0   -1.1667    0.2000    2.0000   -0.6667   -1.5000    1.0000         0


def_int =

  959.5833
```

4-7)

	Syntax	Description
residue	[r,p,k] = residue(b,a)	finds the residues, poles, and direct term of a Partial Fraction Expansion of the ratio of two polynomials.
	[b,a] = residue(r,p,k)	converts the partial fraction expansion back to the ratio of two polynomials and returns the coefficients in b and a.

```
clear; clc;

b = [1 2]; % s + 2
a = [1 4 3 0]; % s^3 + 4s^2 + 3s
[r,p,k] = residue(b,a)
```

Command Window

```
r =

   -0.1667
   -0.5000
    0.6667


p =

    -3
    -1
     0


k =

     []
```