MATLAB PROJECT 1

Please include this page in your Group file, as a front page. Type in the group number and the names of all members WHO PARTICIPATED in this project.

GROUP # 4

FIRST & LAST NAMES (UFID numbers are NOT required):

_1._Edwin Salcedo

_2._Anisha Paul

_3._Brandon Tran

_4._John Dinh

_5._Ananya Kakaveti

_6._Yaaseen Mohammad

By including your names above, each of you had confirmed that you did the work and agree with the work submitted.

Exercise 1

```
type usenorank.m
```

```
function [R,x] = usenorank(A,b)
format
x = [];
[m,n] = size(A);
fprintf('A is %i by %i matrix\n', m,n)
[R, pivot] = rref([A b]);
disp('the reduced echelon form of [A b] is')
disp(R)
disp('the vector of indexes of the pivot columns of [A b] is')
disp(pivot)
N = numel(pivot);

test1 = 1;
test2 = 1;

numr = size(R, 1);
numc = size(R, 2);

%condition 1
if ismember(numc, pivot)
    test1 = 0;
end

%condition 2
for i = 1:numr
    if any(R(i, 1:numc-1))
        test2 = 1;
    else
        if ~isequal(R(i, numc), 0)
            test2 = 0;
            break
        end
    end
end



test1
test2

if isequal(test1, test2, 1)
    disp('the system is consistent')
elseif isequal(test1, test2, 0)
    disp('the system is inconsistent')
    return
else
    disp('test1 and test2 disagree - something is not quite right!')
    return
end


test3 = 0;
test4 = 0;

%condition 3
for i = 1:numc-1
    if ismember(i, pivot)
        test3 = 1;
    else
```

```
        test3 = 0;
        break
    end
end




%condition 4
comp = R(:, 1:numc-1);
if isequal(comp, eye(numr, numc-1))
    test4 = 1;
end

if isequal(comp, eye(numc, numc))
    test4 = 1;
end




test3
test4

if isequal(test3, test4, 1)
    disp('the solution is unique')
elseif isequal(test3, test4, 0)
    disp('there are infinitely many solutions')
else
    disp('test3 and test4 disagree - something is definitely wrong')
    return
end

%display
y = [];
for i = 1:numr
    if any(R(i, :))
        y = [y ; R(i, :)];
    else
        break
    end
end

R = y;
R

if isequal(test3, test4, 1)
    x = R(:, size(R, 2));
end
if isequal(test3, test4, 0)
    x = zeros(n, 1);
    x(pivot, 1)=R(:, size(R, 2));
end

if closetozeroroundoff(A*x-b, 7)==0
    disp('a solution of the system Ax=b is')
    x
else
    disp('check the code!')
    x = []
end
end
```

```
%(a)
A=magic(5)
```

```
A = 5×5
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
```

```
b=rand(5,1)
```

```
b = 5×1
    0.3763
    0.1909
    0.4283
    0.4820
    0.1206
```

```
[R,x]=usenorank(A,b);
```

```
A is 5 by 5 matrix
the reduced echelon form of [A b] is
    1.0000         0         0         0         0   -0.0063
         0    1.0000         0         0         0    0.0134
         0         0    1.0000         0         0   -0.0036
         0         0         0    1.0000         0    0.0216
         0         0         0         0    1.0000   -0.0006

the vector of indexes of the pivot columns of [A b] is
     1     2     3     4     5
test1 = 1
test2 = 1
the system is consistent
test3 = 1
test4 = 1
the solution is unique
R = 5×6
    1.0000         0         0         0         0   -0.0063
         0    1.0000         0         0         0    0.0134
         0         0    1.0000         0         0   -0.0036
         0         0         0    1.0000         0    0.0216
         0         0         0         0    1.0000   -0.0006
a solution of the system Ax=b is
x = 5×1
   -0.0063
    0.0134
   -0.0036
    0.0216
   -0.0006
```

```
%(b)
A=magic(5)
```

```
A = 5×5
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
```

```
b=zeros(5,1)
```

```
b = 5×1
     0
     0
     0
     0
     0
```

```
[R,x]=usenorank(A,b);
```

```
A is 5 by 5 matrix
the reduced echelon form of [A b] is
     1     0     0     0     0     0
     0     1     0     0     0     0
     0     0     1     0     0     0
     0     0     0     1     0     0
     0     0     0     0     1     0

the vector of indexes of the pivot columns of [A b] is
     1     2     3     4     5
test1 = 1
test2 = 1
the system is consistent
test3 = 1
test4 = 1
the solution is unique
R = 5×6
     1     0     0     0     0     0
     0     1     0     0     0     0
     0     0     1     0     0     0
     0     0     0     1     0     0
     0     0     0     0     1     0
a solution of the system Ax=b is
x = 5×1
     0
     0
     0
     0
     0
```

```
%(c)
A=magic(4)
```

```
A = 4×4
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

```
b=rand(4,1)
```

```
b = 4×1
    0.5895
    0.2262
    0.3846
    0.5830
```

```
[R,x]=usenorank(A,b);
```

```
A is 4 by 4 matrix
the reduced echelon form of [A b] is
    1.0000         0         0    1.0000         0
         0    1.0000         0    3.0000         0
```

```
         0         0    1.0000   -3.0000         0
         0         0         0         0    1.0000

the vector of indexes of the pivot columns of [A b] is
     1     2     3     5
test1 = 0
test2 = 0
the system is inconsistent
```

```
%(d)
A=magic(4)
```

```
A = 4×4
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

```
b=ones(4,1)
```

```
b = 4×1
     1
     1
     1
     1
```

```
[R,x]=usenorank(A,b);
```

```
A is 4 by 4 matrix
the reduced echelon form of [A b] is
    1.0000         0         0    1.0000    0.0588
         0    1.0000         0    3.0000    0.1176
         0         0    1.0000   -3.0000   -0.0588
         0         0         0         0         0

the vector of indexes of the pivot columns of [A b] is
     1     2     3
test1 = 1
test2 = 1
the system is consistent
test3 = 0
test4 = 0
there are infinitely many solutions
R = 3×5
    1.0000         0         0    1.0000    0.0588
         0    1.0000         0    3.0000    0.1176
         0         0    1.0000   -3.0000   -0.0588
a solution of the system Ax=b is
x = 4×1
    0.0588
    0.1176
   -0.0588
         0
```

```
%(e)
A=[1 2 3 4;-1 -2 -3 -4;2 4 -3 -1;1 2 -1 5;3 6 1 2]
```

```
A = 5×4
     1     2     3     4
    -1    -2    -3    -4
     2     4    -3    -1
     1     2    -1     5
```

```
     3     6     1     2
```

```
b=sum(A,2)
```

```
b = 5×1
    10
   -10
     2
     7
    12
```

```
[R,x]=usenorank(A,b);
```

```
A is 5 by 4 matrix
the reduced echelon form of [A b] is
     1     2     0     0     3
     0     0     1     0     1
     0     0     0     1     1
     0     0     0     0     0
     0     0     0     0     0

the vector of indexes of the pivot columns of [A b] is
     1     3     4
test1 = 1
test2 = 1
the system is consistent
test3 = 0
test4 = 0
there are infinitely many solutions
R = 3×5
     1     2     0     0     3
     0     0     1     0     1
     0     0     0     1     1
a solution of the system Ax=b is
x = 4×1
     3
     0
     1
     1
```

```
%(f)
A=[magic(5),randi(10,5,2)]
```

```
A = 5×7
    17    24     1     8    15     3    10
    23     5     7    14    16     3     8
     4     6    13    20    22     7     4
    10    12    19    21     3     3     6
    11    18    25     2     9     9     2
```

```
b=rand(5,1)
```

```
b = 5×1
    0.9063
    0.8797
    0.8178
    0.2607
    0.5944
```

```
[R,x]=usenorank(A,b);
```

```
A is 5 by 7 matrix
the reduced echelon form of [A b] is
    1.0000         0         0         0         0   -0.0750    0.2321    0.0139
         0    1.0000         0         0         0    0.0365    0.1936    0.0067
         0         0    1.0000         0         0    0.2769   -0.1744    0.0002
         0         0         0    1.0000         0   -0.1327    0.2244   -0.0034
         0         0         0         0    1.0000    0.2788   -0.0141    0.0357

the vector of indexes of the pivot columns of [A b] is
     1     2     3     4     5
test1 = 1
test2 = 1
the system is consistent
test3 = 0
test4 = 0
there are infinitely many solutions
R = 5×8
    1.0000         0         0         0         0   -0.0750    0.2321    0.0139
         0    1.0000         0         0         0    0.0365    0.1936    0.0067
         0         0    1.0000         0         0    0.2769   -0.1744    0.0002
         0         0         0    1.0000         0   -0.1327    0.2244   -0.0034
         0         0         0         0    1.0000    0.2788   -0.0141    0.0357
a solution of the system Ax=b is
x = 7×1
    0.0139
    0.0067
    0.0002
   -0.0034
    0.0357
         0
         0
```

```
%(g)
A=[magic(5);zeros(2,5)]
```

```
A = 7×5
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
     0     0     0     0     0
     0     0     0     0     0
```

```
b=[rand(5,1);zeros(2,1)]
```

```
b = 7×1
    0.0225
    0.4253
    0.3127
    0.1615
    0.1788
         0
         0
```

```
[R,x]=usenorank(A,b);
```

```
A is 7 by 5 matrix
the reduced echelon form of [A b] is
    1.0000         0         0         0         0    0.0114
         0    1.0000         0         0         0   -0.0140
         0         0    1.0000         0         0    0.0086
         0         0         0    1.0000         0    0.0011
```

```
         0         0         0         0    1.0000    0.0099
         0         0         0         0         0         0
         0         0         0         0         0         0

the vector of indexes of the pivot columns of [A b] is
     1     2     3     4     5
test1 = 1
test2 = 1
the system is consistent
test3 = 1
test4 = 1
the solution is unique
R = 5×6
    1.0000         0         0         0         0    0.0114
         0    1.0000         0         0         0   -0.0140
         0         0    1.0000         0         0    0.0086
         0         0         0    1.0000         0    0.0011
         0         0         0         0    1.0000    0.0099
a solution of the system Ax=b is
x = 5×1
    0.0114
   -0.0140
    0.0086
    0.0011
    0.0099
```

```
%(h)
A=[magic(5);randi(10,2,5)]
```

```
A = 7×5
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
     5     6     7     7     1
     1     5     7     1     4
```

```
b=rand(7,1)
```

```
b = 7×1
    0.5309
    0.6544
    0.4076
    0.8200
    0.7184
    0.9686
    0.5313
```

```
[R,x]=usenorank(A,b);
```

```
A is 7 by 5 matrix
the reduced echelon form of [A b] is
     1     0     0     0     0     0
     0     1     0     0     0     0
     0     0     1     0     0     0
     0     0     0     1     0     0
     0     0     0     0     1     0
     0     0     0     0     0     1
     0     0     0     0     0     0

the vector of indexes of the pivot columns of [A b] is
     1     2     3     4     5     6
```

```
test1 = 0
test2 = 0
the system is inconsistent
```

```
%(i)
A=randi([-5 5],5,3)
```

```
A = 5×3
    -2    -5     0
    -4    -3     0
     1    -4     4
     3    -2     0
    -1    -1     5
```

```
b=sum(A,2)
```

```
b = 5×1
    -7
    -7
     1
     1
     3
```

```
[R,x]=usenorank(A,b);
```

```
A is 5 by 3 matrix
the reduced echelon form of [A b] is
     1     0     0     1
     0     1     0     1
     0     0     1     1
     0     0     0     0
     0     0     0     0

the vector of indexes of the pivot columns of [A b] is
     1     2     3
test1 = 1
test2 = 1
the system is consistent
test3 = 1
test4 = 1
the solution is unique
R = 3×4
     1     0     0     1
     0     1     0     1
     0     0     1     1
a solution of the system Ax=b is
x = 3×1
     1
     1
     1
```

```
%(j)
A=ones(5)
```

```
A = 5×5
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
```

```
b=sum(A,2)
```

```
b = 5×1
     5
     5
     5
     5
     5
```

```
[R,x]=usenorank(A,b);
```

```
A is 5 by 5 matrix
the reduced echelon form of [A b] is
     1     1     1     1     1     5
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0

the vector of indexes of the pivot columns of [A b] is
     1
test1 = 1
test2 = 1
the system is consistent
test3 = 0
test4 = 0
there are infinitely many solutions
R = 1×6
     1     1     1     1     1     5
a solution of the system Ax=b is
x = 5×1
     5
     0
     0
     0
     0
```

Bonus: N provides the indices of the pivot columns in the matrix. If the rank of the matrix A is equal to the rank of the augmented matrix [A b], then the system is consistent. This means that the size of N must be equal to the rank. If the rank of the coefficient matrix A is less than the rank of the augmented matrix [A b], then the system is inconsistent because the row-reduced echelon form of the matrix would contain a pivot in the final column, or a row of zeroes followed by a non-zero value in the final column; thus, no solution can exist. This means that the size of N would be larger than the rank of A.

Exercise 2

Part I

```
type givens.m
```

```
function G=givens(m,i,j,theta)
format
G=[];
% Givens works under the conditions (1 <= i < j<= m) and (m >= 2)
if (i >= 1)& (i < j) & (m >=2) & (j <= m)
    G = eye(m)
    c = cos(theta);
    s = sin(theta);
    G(i,i)=c; G(i,j)=-s; G(j,i)=s; G(j,j)=c;
    disp('the Givens rotation matrix G is')
    disp(G)
else
    disp('Givens rotation matrix cannot be constructed')
    return %terminates function
end
```

```
%(a)
m=1;i=1;j=2;theta=pi
```

```
theta = 3.1416
```

```
G=givens(m,i,j,theta);
```

```
Givens rotation matrix cannot be constructed
```

```
%(b)
m=4;i=3;j=2;theta=pi/2
```

```
theta = 1.5708
```

```
G=givens(m,i,j,theta);
```

```
Givens rotation matrix cannot be constructed
```

```
%(c)
m=5;i=2;j=4;theta=pi/4
```

```
theta = 0.7854
```

```
G=givens(m,i,j,theta);
```

```
G = 5×5
     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1
the Givens rotation matrix G is
    1.0000         0         0         0         0
         0    0.7071         0   -0.7071         0
         0         0    1.0000         0         0
         0    0.7071         0    0.7071         0
         0         0         0         0    1.0000
```

```
%(d)
```

```
m=2;i=1;j=2;theta=-pi/2
```

```
theta = -1.5708
```

```
G=givens(m,i,j,theta);
```

```
G = 2×2
     1     0
     0     1
the Givens rotation matrix G is
    0.0000    1.0000
   -1.0000    0.0000
```

```
%(e)
m=3;i=1;j=2;theta=pi
```

```
theta = 3.1416
```

```
G=givens(m,i,j,theta);
```

```
G = 3×3
     1     0     0
     0     1     0
     0     0     1
the Givens rotation matrix G is
   -1.0000   -0.0000         0
    0.0000   -1.0000         0
         0         0    1.0000
```

```
a1 = [1;1;0]; a2 = [-1;1;0]; a3 = [1;0;1];
A = [a1 a2 a3]
```

```
A = 3×3
     1    -1     1
     1     1     0
     0     0     1
```

```
GA = [-1 1 -1; -1 -1 0; 0 0 1]
```

```
GA = 3×3
    -1     1    -1
    -1    -1     0
     0     0     1
```

```
GpA = G * A
```

```
GpA = 3×3
   -1.0000    1.0000   -1.0000
   -1.0000   -1.0000    0.0000
         0         0    1.0000
```

```
if closetozeroroundoff(GA,7) == round(closetozeroroundoff(GpA,7))
    disp('Predicted matrix matches observed matrix.')
else
    disp('Matrices do not match.')
end
```

```
Predicted matrix matches observed matrix.
```

```
type closetozeroroundoff.m
```

```
function B=closetozeroroundoff(A,p)
A(abs(A)<10^-p)=0;
B=A;
end
```

Part II

```
type givensrot.m
```

```
function G=givensrot(m,i,j,a,b)
G=eye(m);
r=hypot(a,b);
c=a/r;
s=b/r;
G(i,i)=c; G(i,j)=-s; G(j,i)=s; G(j,j)=c;
```

```
type uppertrian.m
```

```
function R = uppertrian(A)
format
[m,n]=size(A);
R=A;
k=min(m,n);
for i=1:k       % iterate through columns
    j=m;        % j starts at bottom
    while j > i
        if R(j,i) ~= 0      % checks if entry is NOT 0
            b = R(j,i);     % row entry
            a = R(i,i);     % main diagonal entry
            G = givensrot(m,i,j,a,b);
            R = G' * R;
        end
        j = j - 1;          % j goes to row above
    end
end

R=closetozeroroundoff(R,12);
disp('the output matrix R is')
disp(R)

test1 = 1;
test2 = 1;
if ~istriu(R)
    test1 = 0;
end

for i=1:n
    if (closetozeroroundoff(norm(A(:,i))-norm(R(:,i)),7) ~= 0)
        test2=0;
        break
    end
end

if test1 & test2
disp('A has been reduced correctly to an uppertriangular matrix R')
else
disp('the output matrix R is not what was expected?!')
end
```

```
%(a)
A=ones(2)
```

```
A = 2×2
```

```
     1     1
     1     1
```

```
R = uppertrian(A);
```

```
the output matrix R is
    1.4142    1.4142
         0         0

A has been reduced correctly to an uppertriangular matrix R
```

```
%(b)
A=magic(3)
```

```
A = 3×3
     8     1     6
     3     5     7
     4     9     2
```

```
R = uppertrian(A);
```

```
the output matrix R is
    9.4340    6.2540    8.1620
         0    8.2394    0.9655
         0         0   -4.6314

A has been reduced correctly to an uppertriangular matrix R
```

```
%(c)
A=magic(4)
```

```
A = 4×4
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

```
R = uppertrian(A);
```

```
the output matrix R is
   19.4422   10.5955   10.9041   18.5164
         0   16.0541   15.7259    0.9848
         0         0    1.9486   -5.8458
         0         0         0         0

A has been reduced correctly to an uppertriangular matrix R
```

```
%(d)
A=[magic(3),ones(3,2)]
```

```
A = 3×5
     8     1     6     1     1
     3     5     7     1     1
     4     9     2     1     1
```

```
R = uppertrian(A);
```

```
the output matrix R is
    9.4340    6.2540    8.1620    1.5900    1.5900
         0    8.2394    0.9655    0.6137    0.6137
         0         0   -4.6314   -0.3088   -0.3088
```

```
A has been reduced correctly to an uppertriangular matrix R
```

```
%(e)
A=[magic(3);ones(2,3)]
```

```
A = 5×3
     8     1     6
     3     5     7
     4     9     2
     1     1     1
     1     1     1
```

```
R = uppertrian(A);
```

```
the output matrix R is
    9.5394    6.3945    8.2815
         0    8.2529    0.9747
         0         0    4.6333
         0         0         0
         0         0         0

A has been reduced correctly to an uppertriangular matrix R
```

```
%(f)
A=triu(magic(5))
```

```
A = 5×5
    17    24     1     8    15
     0     5     7    14    16
     0     0    13    20    22
     0     0     0    21     3
     0     0     0     0     9
```

```
R = uppertrian(A);
```

```
the output matrix R is
    17    24     1     8    15
     0     5     7    14    16
     0     0    13    20    22
     0     0     0    21     3
     0     0     0     0     9

A has been reduced correctly to an uppertriangular matrix R
```

```
%(g)
A=tril(magic(3))
```

```
A = 3×3
     8     0     0
     3     5     0
     4     9     2
```

```
R = uppertrian(A);
```

```
the output matrix R is
    9.4340    5.4060    0.8480
         0    8.7622    1.5311
         0         0    0.9678

A has been reduced correctly to an uppertriangular matrix R
```

```
%(h)
A=[1 1 2 0;0 0 1 3;0 0 2 4;0 0 3 5;1 0 -2 3]
```

```
A = 5×4
     1     1     2     0
     0     0     1     3
     0     0     2     4
     0     0     3     5
     1     0    -2     3
```

```
R = uppertrian(A);
```

```
the output matrix R is
    1.4142    0.7071         0    2.1213
         0    0.7071    2.8284   -2.1213
         0         0    3.7417    6.9488
         0         0         0    1.3093
         0         0         0         0

A has been reduced correctly to an uppertriangular matrix R
```

```
%(i)
A=hilb(4)
```

```
A = 4×4
    1.0000    0.5000    0.3333    0.2500
    0.5000    0.3333    0.2500    0.2000
    0.3333    0.2500    0.2000    0.1667
    0.2500    0.2000    0.1667    0.1429
```

```
R = uppertrian(A);
```

```
the output matrix R is
    1.1932    0.6705    0.4749    0.3698
         0    0.1185    0.1257    0.1175
         0         0    0.0062    0.0096
         0         0         0    0.0002

A has been reduced correctly to an uppertriangular matrix R
```

```
%(j)
A=[1 3 4 -1 2;2 6 6 0 -3;1 3 1 2 -1;1 3 0 3 0]
```

```
A = 4×5
     1     3     4    -1     2
     2     6     6     0    -3
     1     3     1     2    -1
     1     3     0     3     0
```

```
R = uppertrian(A);
```

```
the output matrix R is
    2.6458    7.9373    6.4254    1.5119   -1.8898
         0         0    0.8165   -0.8165    1.6330
         0         0    3.3238   -3.3238   -0.0573
         0         0         0         0   -2.7854

A has been reduced correctly to an uppertriangular matrix R
```

Solving Homogeneous Systems

Exercise 3

```
type closetozeroroundoff.m
```

```
function B=closetozeroroundoff(A,p)
A(abs(A)<10^-p)=0;
B=A;
end
```

```
type homobasis.m
```

```
function C = homobasis(A)
format
[~,n]=size(A);
R=rref(A);
rankA=rank(A);

if rankA==n
    disp('the homogeneous system has only the trivial solution')
    C=zeros(n,1);
else
    disp('the homogeneous system has non-trivial solutions')
    [~,pivot]=rref(A);
    nonpivot=setdiff(1:n,pivot);
    q=numel(nonpivot);
    j=1:q;
    fprintf('a free variable is x%i\n',nonpivot(j))
    C=zeros(n,q);
    R=R(1:rankA,nonpivot);
    C(pivot,:)=-R;
    C(nonpivot,:)=eye(q);
    if rank(C)== size(C,2) && ~any(closetozeroroundoff(A*C,5),'all')
        disp('columns of C form a basis for solution set of homogeneous system')
    else
        C=[];
    end
end
```

```
%(a)
A=[1 -1 -1 2;-2 5 4 4]
```

```
A = 2×4
     1    -1    -1     2
    -2     5     4     4
```

```
C=homobasis(A)
```

```
the homogeneous system has non-trivial solutions
a free variable is x3
a free variable is x4
columns of C form a basis for solution set of homogeneous system
C = 4×2
    0.3333   -4.6667
   -0.6667   -2.6667
    1.0000         0
         0    1.0000
```

```
%(b)
A=[1 2 -3]
```

```
A = 1×3
     1     2    -3
```

```
C=homobasis(A)
```

```
the homogeneous system has non-trivial solutions
a free variable is x2
a free variable is x3
columns of C form a basis for solution set of homogeneous system
C = 3×2
    -2     3
     1     0
     0     1
```

```
%(c)
A=magic(3)
```

```
A = 3×3
     8     1     6
     3     5     7
     4     9     2
```

```
C=homobasis(A)
```

```
the homogeneous system has only the trivial solution
C = 3×1
     0
     0
     0
```

```
%(d)
A=[magic(3), ones(3,1)]
```

```
A = 3×4
     8     1     6     1
     3     5     7     1
     4     9     2     1
```

```
C=homobasis(A)
```

```
the homogeneous system has non-trivial solutions
a free variable is x4
columns of C form a basis for solution set of homogeneous system
C = 4×1
   -0.0667
   -0.0667
   -0.0667
    1.0000
```

```
%(e)
A=magic(4)
```

```
A = 4×4
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

```
C=homobasis(A)
```

```
the homogeneous system has non-trivial solutions
a free variable is x4
columns of C form a basis for solution set of homogeneous system
C = 4×1
    -1
    -3
     3
     1
```

```
%(f)
A=[0 1 2 3;0 2 4 6]
```

```
A = 2×4
     0     1     2     3
     0     2     4     6
```

```
C=homobasis(A)
```

```
the homogeneous system has non-trivial solutions
a free variable is x1
a free variable is x3
a free variable is x4
columns of C form a basis for solution set of homogeneous system
C = 4×3
     1     0     0
     0    -2    -3
     0     1     0
     0     0     1
```

```
%(g)
A=[0 1 0 2 0 3; 0 2 0 4 0 6; 0 4 0 8 0 6]
```

```
A = 3×6
     0     1     0     2     0     3
     0     2     0     4     0     6
     0     4     0     8     0     6
```

```
C=homobasis(A)
```

```
the homogeneous system has non-trivial solutions
a free variable is x1
a free variable is x3
a free variable is x4
a free variable is x5
columns of C form a basis for solution set of homogeneous system
C = 6×4
     1     0     0     0
     0     0    -2     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
     0     0     0     0
```

```
%(h)
A=[1 0 2 0 3;2 0 5 0 6]
```

```
A = 2×5
     1     0     2     0     3
     2     0     5     0     6
```

```
C=homobasis(A)
```

```
the homogeneous system has non-trivial solutions
a free variable is x2
a free variable is x4
a free variable is x5
columns of C form a basis for solution set of homogeneous system
C = 5×3
     0     0    -3
     1     0     0
     0     0     0
     0     1     0
     0     0     1
```

```
%(i)
A=[1 0 0 2 3;2 0 0 4 6]
```

```
A = 2×5
     1     0     0     2     3
     2     0     0     4     6
```

```
C=homobasis(A)
```

```
the homogeneous system has non-trivial solutions
a free variable is x2
a free variable is x3
a free variable is x4
a free variable is x5
columns of C form a basis for solution set of homogeneous system
C = 5×4
     0     0    -2    -3
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

```
%(j)
A=hilb(4)
```

```
A = 4×4
    1.0000    0.5000    0.3333    0.2500
    0.5000    0.3333    0.2500    0.2000
    0.3333    0.2500    0.2000    0.1667
    0.2500    0.2000    0.1667    0.1429
```

```
C=homobasis(A)
```

```
the homogeneous system has only the trivial solution
C = 4×1
     0
     0
     0
     0
```

Exercise 4

```
type closetozeroroundoff.m
```

```
function B=closetozeroroundoff(A,p)
A(abs(A)<10^-p)=0;
B=A;
end
```

```
type interpolate.m
```

```
function [A,P]=interpolate(m,n,X,Y)
format
P=[];
A=ones(m,n);
i=n-1:-1:0;
A=X.^i;

disp('the matrix A of the system Ac=Y for finding coefficient vector c is')
disp(A)

if (rank([A Y]) ~= rank(A))
    disp('there is no solution')
    return
else
    disp('the system is consistent')
    if (rank(A)==size(A,2))
        disp('the solution is unique')
    else
        disp('there are infinitely many solutions')
    end
    c=A\Y;

    if (closetozeroroundoff(A*c-Y,7) == 0)
        disp('a solution is found correctly')
    else
        disp('what is wrong?!')
        return
    end

    c=closetozeroroundoff(c,12);
    disp('the vector c of the coefficients of a polynomial P is')
    disp(c)

    disp('an interpolating polynomial P is')
    P=vpa(poly2sym(c),4)

    plot(X,Y,'*'),hold on
    a=X(1);
    b=X(end);
    polyplot(a,b,c');
    hold off
end

end
```

```
type polyplot.m
```

```
function []=polyplot(a,b,p)
x=(a:(b-a)/50:b)';
y=polyval(p,x);
plot(x,y);
end
```

```
%(a)
m=4;n=4;
X=(1:m)'/m
```

```
X = 4×1
    0.2500
    0.5000
    0.7500
    1.0000
```

```
Y=randi(10,m,1)
```

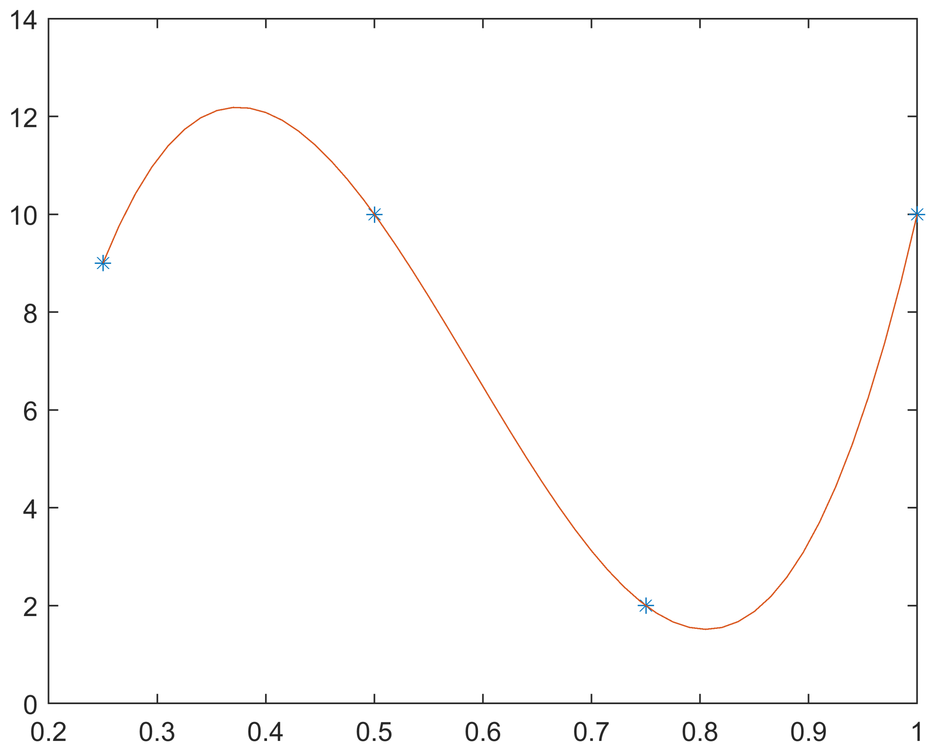
```
Y = 4×1
     9
    10
     2
    10
```

```
[A,P]=interpolate(m,n,X,Y);
```

```
the matrix A of the system Ac=Y for finding coefficient vector c is
    0.0156    0.0625    0.2500    1.0000
    0.1250    0.2500    0.5000    1.0000
    0.4219    0.5625    0.7500    1.0000
    1.0000    1.0000    1.0000    1.0000

the system is consistent
the solution is unique
a solution is found correctly
the vector c of the coefficients of a polynomial P is
  266.6667
 -472.0000
  241.3333
  -26.0000

an interpolating polynomial P is
```

P = $266.7\,x^3 - 472.0\,x^2 + 241.3\,x - 26.0$

```
Warning: MATLAB has disabled some advanced graphics rendering features by switching to software OpenGL. For
more information, click here.
```




```
%(b)
m=4;n=4;
X=(1:m)'/m
```

```
X = 4×1
    0.2500
    0.5000
    0.7500
    1.0000
```

```
Y=X
```

```
Y = 4×1
    0.2500
    0.5000
    0.7500
    1.0000
```

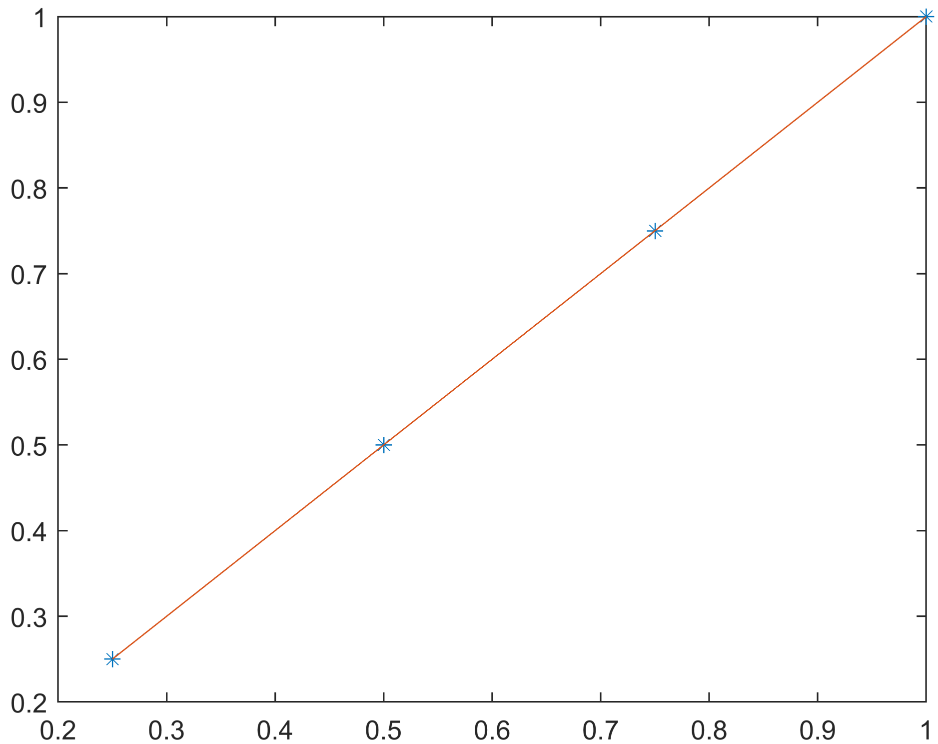
```
[A,P]=interpolate(m,n,X,Y);
```

```
the matrix A of the system Ac=Y for finding coefficient vector c is
    0.0156    0.0625    0.2500    1.0000
    0.1250    0.2500    0.5000    1.0000
    0.4219    0.5625    0.7500    1.0000
    1.0000    1.0000    1.0000    1.0000

the system is consistent
the solution is unique
a solution is found correctly
the vector c of the coefficients of a polynomial P is
     0
     0
```

```
     1
     0
```

an interpolating polynomial P is

P = x




```
%(c)
m=5;n=4;
X=(1:m)'
```

```
X = 5×1
     1
     2
     3
     4
     5
```

```
Y=X
```

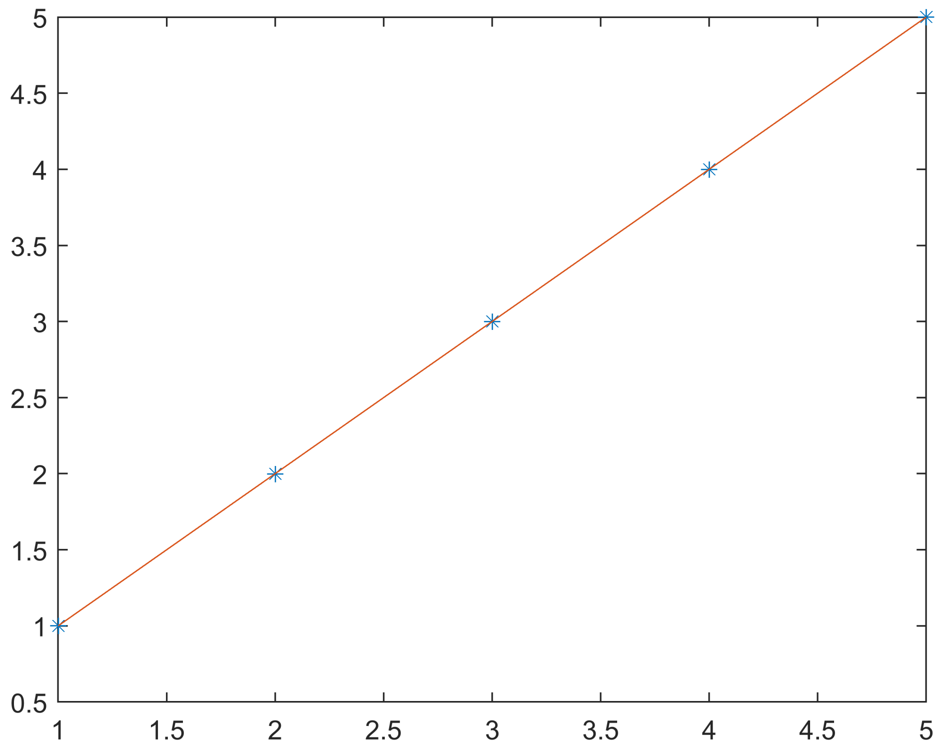
```
Y = 5×1
     1
     2
     3
     4
     5
```

```
[A,P]=interpolate(m,n,X,Y);
```

```
the matrix A of the system Ac=Y for finding coefficient vector c is
     1     1     1     1
     8     4     2     1
    27     9     3     1
```

```
    64    16     4     1
   125    25     5     1

the system is consistent
the solution is unique
a solution is found correctly
the vector c of the coefficients of a polynomial P is
         0
         0
    1.0000
         0

an interpolating polynomial P is
```

P = x




```
%(d)
m=4;n=6;
X=(1:m)'
```

```
X = 4×1
     1
     2
     3
     4
```

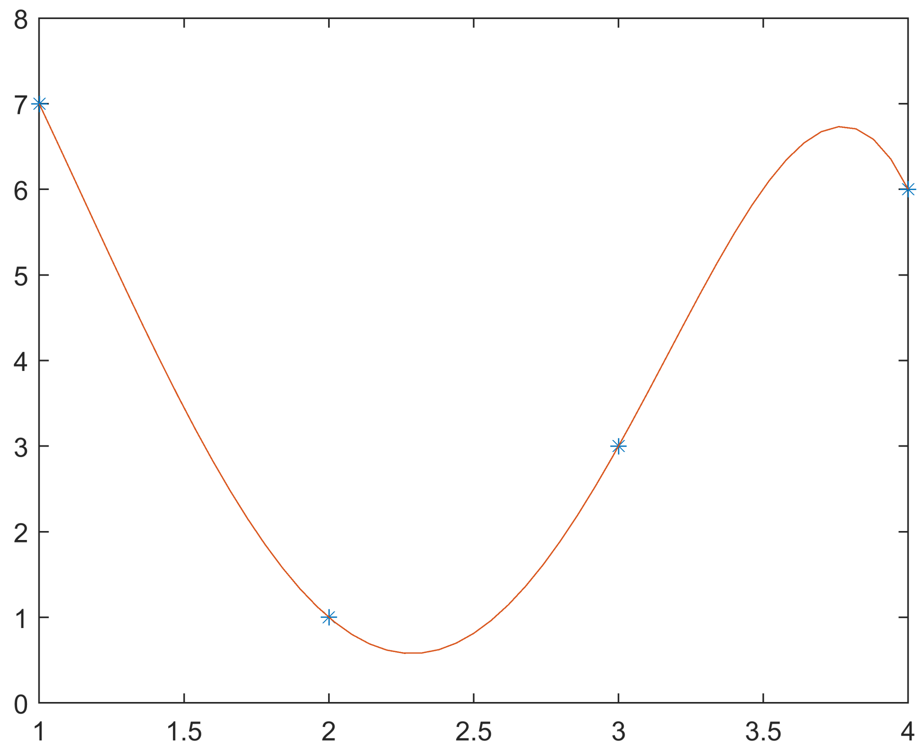
```
Y=randi(10,m,1)
```

```
Y = 4×1
     7
     1
     3
     6
```

```
[A,P]=interpolate(m,n,X,Y);
```

```
the matrix A of the system Ac=Y for finding coefficient vector c is
  Columns 1 through 5

           1           1           1           1           1
          32          16           8           4           2
         243          81          27           9           3
        1024         256          64          16           4

  Column 6

           1
           1
           1
           1

the system is consistent
there are infinitely many solutions
a solution is found correctly
the vector c of the coefficients of a polynomial P is
   -0.1656
    0.9595
         0
   -5.0866
         0
   11.2927

an interpolating polynomial P is
```

P = $-0.1656\,x^5 + 0.9595\,x^4 - 5.087\,x^2 + 11.29$




```
%(e)
m=6;n=4;
```

```
X=(1:m)'/m
```

```
X = 6×1
    0.1667
    0.3333
    0.5000
    0.6667
    0.8333
    1.0000
```

```
Y=randi(10,m,1)
```

```
Y = 6×1
    10
    10
     2
    10
    10
     5
```

```
[A,P]=interpolate(m,n,X,Y);
```

```
the matrix A of the system Ac=Y for finding coefficient vector c is
    0.0046    0.0278    0.1667    1.0000
    0.0370    0.1111    0.3333    1.0000
    0.1250    0.2500    0.5000    1.0000
    0.2963    0.4444    0.6667    1.0000
    0.5787    0.6944    0.8333    1.0000
    1.0000    1.0000    1.0000    1.0000

there is no solution
```

```
%(f)
m=10;n=10;
X=(1:m)'
```

```
X = 10×1
     1
     2
     3
     4
     5
     6
     7
     8
     9
    10
```

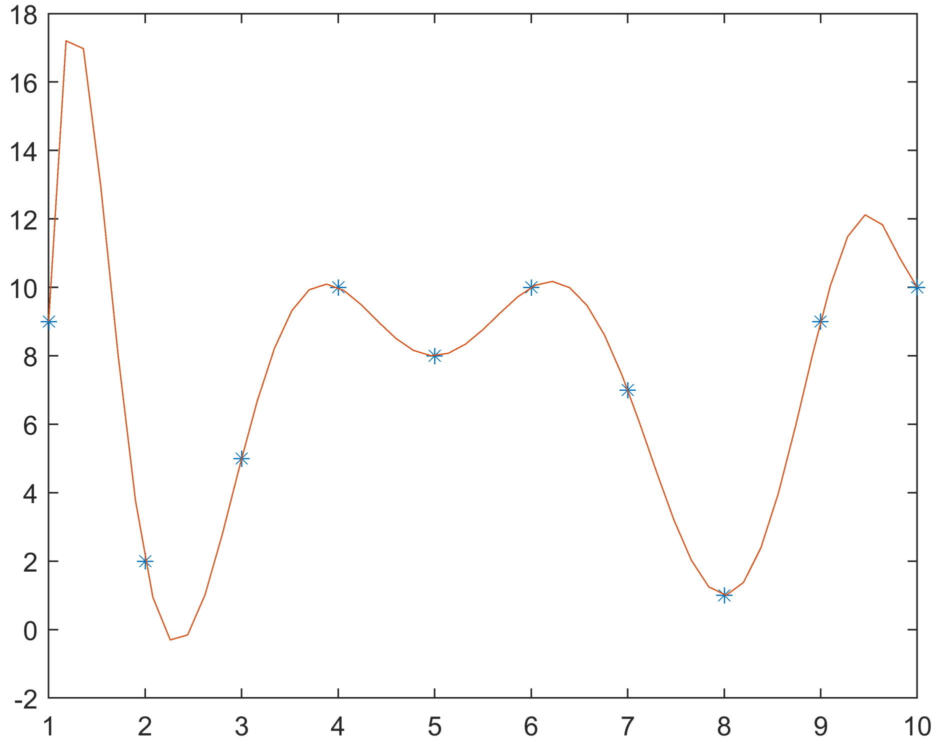
```
Y=randi(10,m,1)
```

```
Y = 10×1
     9
     2
     5
    10
     8
    10
     7
     1
     9
    10
```

```
[A,P]=interpolate(m,n,X,Y);
```

```
the matrix A of the system Ac=Y for finding coefficient vector c is
   1.0e+09 *

  Columns 1 through 6

    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0003    0.0001    0.0000    0.0000    0.0000    0.0000
    0.0020    0.0004    0.0001    0.0000    0.0000    0.0000
    0.0101    0.0017    0.0003    0.0000    0.0000    0.0000
    0.0404    0.0058    0.0008    0.0001    0.0000    0.0000
    0.1342    0.0168    0.0021    0.0003    0.0000    0.0000
    0.3874    0.0430    0.0048    0.0005    0.0001    0.0000
    1.0000    0.1000    0.0100    0.0010    0.0001    0.0000

  Columns 7 through 10

    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000

the system is consistent
the solution is unique
a solution is found correctly
the vector c of the coefficients of a polynomial P is
   1.0e+03 *

    0.0000
   -0.0000
    0.0009
   -0.0117
    0.0882
   -0.4181
    1.2316
   -2.1462
    1.9732
   -0.7090

an interpolating polynomial P is
```

P = $0.0008212\,x^9 - 0.04266\,x^8 + 0.946\,x^7 - 11.69\,x^6 + 88.22\,x^5 - 418.1\,x^4 + 1232.0\,x^3 - 2146.0\,x^2 + 1973.0\,x$




```
%(g)
m=10;n=11;
X=(1:m)'
```

```
X = 10×1
     1
     2
     3
     4
     5
     6
     7
     8
     9
    10
```

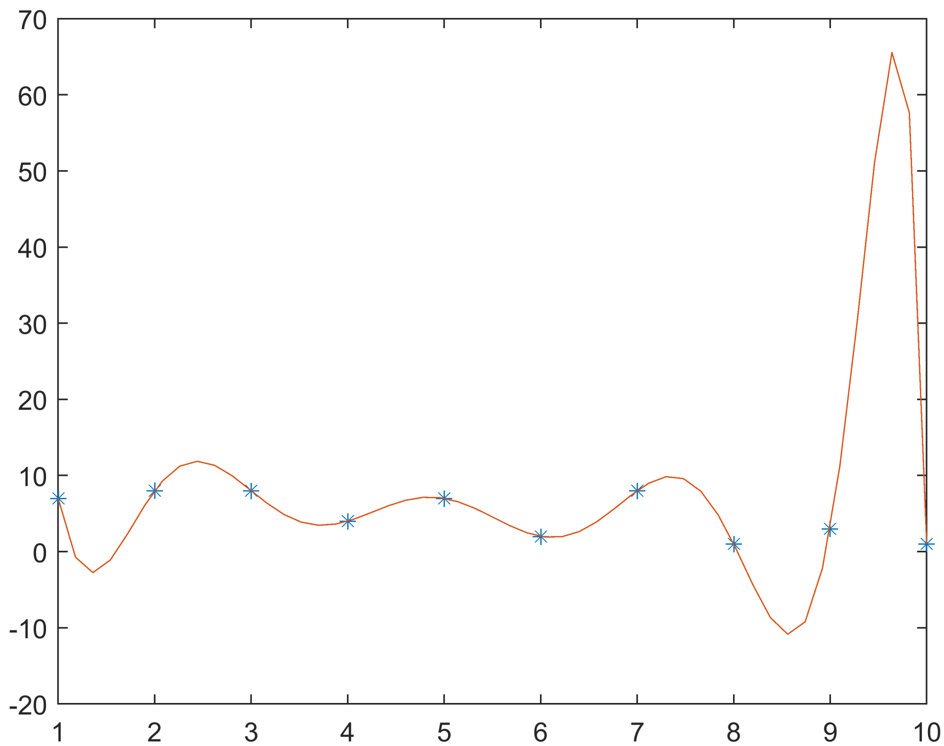
```
Y=randi(10,m,1)
```

```
Y = 10×1
     7
     8
     8
     4
     7
     2
     8
     1
     3
     1
```

```
[A,P]=interpolate(m,n,X,Y);
```

```
the matrix A of the system Ac=Y for finding coefficient vector c is
   1.0e+10 *

  Columns 1 through 6

    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0001    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0010    0.0002    0.0000    0.0000    0.0000    0.0000
    0.0060    0.0010    0.0002    0.0000    0.0000    0.0000
    0.0282    0.0040    0.0006    0.0001    0.0000    0.0000
    0.1074    0.0134    0.0017    0.0002    0.0000    0.0000
    0.3487    0.0387    0.0043    0.0005    0.0001    0.0000
    1.0000    0.1000    0.0100    0.0010    0.0001    0.0000

  Columns 7 through 11

    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000

the system is consistent
there are infinitely many solutions
a solution is found correctly
the vector c of the coefficients of a polynomial P is
   -0.0005
    0.0267
   -0.5590
    6.5247
  -46.4171
  206.3432
 -562.3525
  870.5301
 -609.6548
         0
  142.5591

an interpolating polynomial P is
```

P = $-0.0005444\,x^{10} + 0.02669\,x^9 - 0.559\,x^8 + 6.525\,x^7 - 46.42\,x^6 + 206.3\,x^5 - 562.4\,x^4 + 870.5\,x^3 - 609.7\,x^2$




```
%(h)
X=unique(sort(randi(10,10,1)))
```

```
X = 7×1
     1
     4
     5
     7
     8
     9
    10
```

```
m=numel(X);n=10;
Y=randi(10,m,1)
```

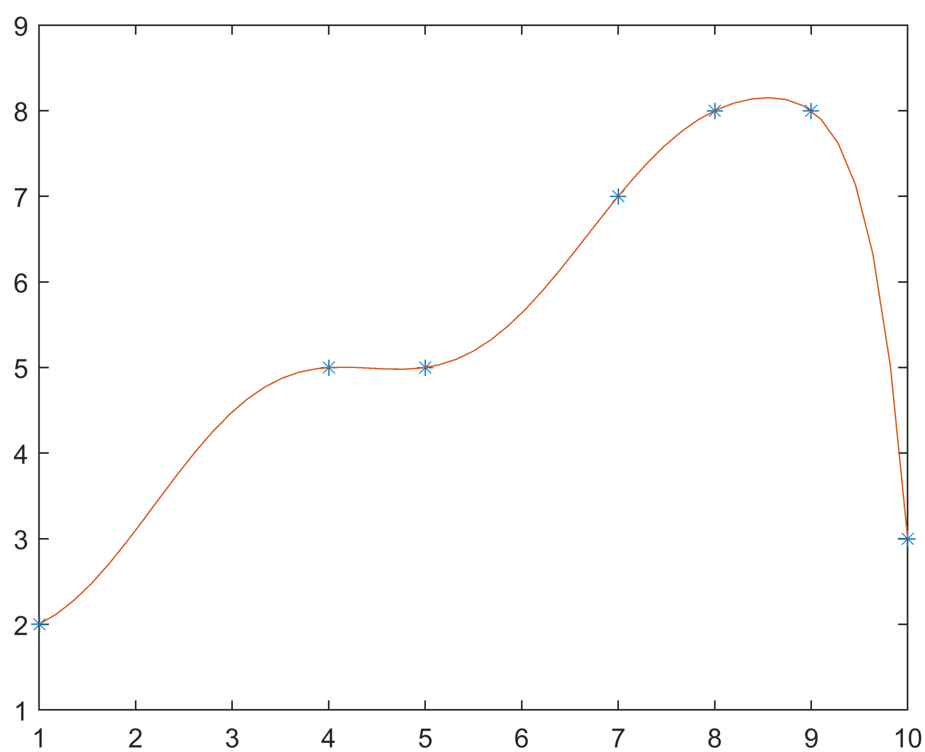
```
Y = 7×1
     2
     5
     5
     7
     8
     8
     3
```

```
[A,P]=interpolate(m,n,X,Y);
```

```
the matrix A of the system Ac=Y for finding coefficient vector c is
   1.0e+09 *

  Columns 1 through 6

    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
```

```
    0.0003    0.0001    0.0000    0.0000    0.0000    0.0000
    0.0020    0.0004    0.0001    0.0000    0.0000    0.0000
    0.0404    0.0058    0.0008    0.0001    0.0000    0.0000
    0.1342    0.0168    0.0021    0.0003    0.0000    0.0000
    0.3874    0.0430    0.0048    0.0005    0.0001    0.0000
    1.0000    0.1000    0.0100    0.0010    0.0001    0.0000

  Columns 7 through 10

    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000

the system is consistent
there are infinitely many solutions
a solution is found correctly
the vector c of the coefficients of a polynomial P is
   -0.0000
    0.0005
   -0.0073
    0.0565
   -0.2187
    0.3422
         0
         0
         0
    1.8268

an interpolating polynomial P is
```

P = $-1.205\text{e-}5\, x^9 + 0.00047\, x^8 - 0.007304\, x^7 + 0.05654\, x^6 - 0.2187\, x^5 + 0.3422\, x^4 + 1.827$



Exercise 5

```
type stochastic
```

```
function [S1,S2,L,R]=stochastic(A)
L=[];
R=[];
fprintf('the vector of sums down each column is\n');
% Column sums
S1=sum(A)
fprintf('the vector of sums across each row is\n');
% Row sums
S2=sum(A,2)
S2=sum(A,2)';
notValid=0;

% Checking if A has a zero column + a zero row
% Checking if both S1 + S2 have zero entries
for i=1:size(S2,2)
    if(S2(i)==0)
        for j=1:size(S1,2)
            if(S1(j)==0)
                disp('A is neither left nor right stochastic and cannot be scaled to either of them');
                notValid=1;
            end
        end
    end
end

% If A doesn't have both a zero column + a zero row
if(notValid==0)
    % Checking if A is right stochastic (rows = 1)
    isRight=1;
    for i=1:size(S2,2)
        if(S2(i)~=1)
            isRight=0;
            break;
        end
    end

    % Checking if A is left stochastic (columns = 1)
    isLeft=1;
    for i=1:size(S1,2)
        if(S1(i)~=1)
            isLeft=0;
            break;
        end
    end

    % If A is doubly stochastic
    if(isRight==1 && isLeft==1)
        disp('A is doubly stochastic')
        L=A;
        R=A;

        % If A is only left stochastic
    else if(isRight==0 && isLeft==1)
            disp('A is only left stochastic')
            L=A; % R is empty

            % If A is only right stochastic
        else if(isRight==1 && isLeft==0)
                disp('A is only right stochastic')
                R=A; % L is empty
```

```
    % Else, checking if A can be scaled to stochastic
else
    scalableRight=1;

    % If S2 has a zero entry (there's a zero row), we're
    % scaling left + not right
    for i=1:size(S2,2)
        if(S2(i)==0)
            scalableRight=0;
        end
    end

    scalableLeft=1;

    % If S1 has a zero entry (there's a zero column), we're
    % scaling right + not left
    for k=1:size(S1,2)
        if(S1(k)==0)
            scalableLeft=0;
        end
    end

    % If A can be scaled
    disp('A is neither left nor right stochastic but can be scaled to stochastic')

    % If neither S1 nor S2 has a zero entry, we are scaling A to the left stochastic matrix L and to the
    if(scalableLeft==1 && scalableRight==1)
        for i=1:size(S1,2)
            L(:,i)=A(:,i)/S1(1,i);
        end
        for i=1:size(S2,2)
            R(i,:) = A(i,:)/S2(1,i);
        end
    end

    % Check if the matrices L and R are equal (use the function closetozeroroundoff with p=7)
    if(closetozeroroundoff(L,7)==closetozeroroundoff(R,7))
        disp('A has been scaled to a doubly stochastic matrix:')
        disp(L)
        % If S1 doesn't have a zero entry (no zero column) + S2
        % does (there's a zero row)
    else if(scalableLeft==1 && scalableRight==0)
            disp('A can be scaled to left stochastic matrix only:')
            for i=1:size(S1,2)
                L(:,i)=A(:,i)/S1(1,i);
            end
            L
        else if(scalableLeft==0 && scalableRight==1)
                disp('A can be scaled to right stochastic matrix only:')
                for i=1:size(S2,2)
                    R(i,:) = A(i,:)/S2(1,i);
                end
                R

                % else L and R are not equal, display each of them
            else
                disp('A is scaled to a left stochastic matrix:')
                L
                disp('and A is scaled to a right stochastic matrix:')
                R
            end
        end
    end
```

```
            end
        end
    end
end
```

```
type jord
```

```
function J=jord(n,r)
J=ones(n);
J=tril(triu(J,1),1)+diag(r*ones(n,1));
end
```

```
type closetozeroroundoff
```

```
function B=closetozeroroundoff(A,p)
A(abs(A)<10^-p)=0;
B=A;
end
```

```
%(a)
A=[0.5, 0, 0.5; 0, 0, 1; 0.5, 0, 0.5]
```

```
A = 3×3
    0.5000         0    0.5000
         0         0    1.0000
    0.5000         0    0.5000
```

```
[S1,S2,L,R]=stochastic(A);
```

```
the vector of sums down each column is
S1 = 1×3
     1     0     2
the vector of sums across each row is
S2 = 3×1
     1
     1
     1
A is only right stochastic
```

```
%(b)
A = transpose(A)
```

```
A = 3×3
    0.5000         0    0.5000
         0         0         0
    0.5000    1.0000    0.5000
```

```
[S1,S2,L,R]=stochastic(A);
```

```
the vector of sums down each column is
S1 = 1×3
     1     1     1
the vector of sums across each row is
S2 = 3×1
     1
     0
     2
A is only left stochastic
```

```
%(c)
```

```
A=[0.5, 0, 0.5; 0, 0, 1; 0, 0, 0.5]
```

```
A = 3×3
    0.5000         0    0.5000
         0         0    1.0000
         0         0    0.5000
```

```
[S1,S2,L,R]=stochastic(A);
```

```
the vector of sums down each column is
S1 = 1×3
    0.5000         0    2.0000
the vector of sums across each row is
S2 = 3×1
    1.0000
    1.0000
    0.5000
A is neither left nor right stochastic but can be scaled to stochastic
A can be scaled to right stochastic matrix only:
R = 3×3
    0.5000         0    0.5000
         0         0    1.0000
         0         0    1.0000
```

```
%(d)
A=transpose(A)
```

```
A = 3×3
    0.5000         0         0
         0         0         0
    0.5000    1.0000    0.5000
```

```
[S1,S2,L,R]=stochastic(A);
```

```
the vector of sums down each column is
S1 = 1×3
    1.0000    1.0000    0.5000
the vector of sums across each row is
S2 = 3×1
    0.5000
         0
    2.0000
A is neither left nor right stochastic but can be scaled to stochastic
A can be scaled to left stochastic matrix only:
L = 3×3
    0.5000         0         0
         0         0         0
    0.5000    1.0000    1.0000
```

```
%(e)
A=[0.5, 0, 0.5; 0, 0.5, 0.5; 0.5, 0.5, 0]
```

```
A = 3×3
    0.5000         0    0.5000
         0    0.5000    0.5000
    0.5000    0.5000         0
```

```
[S1,S2,L,R]=stochastic(A);
```

```
the vector of sums down each column is
S1 = 1×3
     1     1     1
```

```
the vector of sums across each row is
S2 = 3×1
     1
     1
     1
A is doubly stochastic
```

```
%(f)
A=magic(4)
```

```
A = 4×4
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

```
[S1,S2,L,R]=stochastic(A);
```

```
the vector of sums down each column is
S1 = 1×4
    34    34    34    34
the vector of sums across each row is
S2 = 4×1
    34
    34
    34
    34
A is neither left nor right stochastic but can be scaled to stochastic
A has been scaled to a doubly stochastic matrix:
    0.4706    0.0588    0.0882    0.3824
    0.1471    0.3235    0.2941    0.2353
    0.2647    0.2059    0.1765    0.3529
    0.1176    0.4118    0.4412    0.0294
```

```
%(g)
B=[1 2;3 4;5 6]; A=B*B'
```

```
A = 3×3
     5    11    17
    11    25    39
    17    39    61
```

```
[S1,S2,L,R]=stochastic(A);
```

```
the vector of sums down each column is
S1 = 1×3
    33    75   117
the vector of sums across each row is
S2 = 3×1
    33
    75
   117
A is neither left nor right stochastic but can be scaled to stochastic
A is scaled to a left stochastic matrix:
L = 3×3
    0.1515    0.1467    0.1453
    0.3333    0.3333    0.3333
    0.5152    0.5200    0.5214
and A is scaled to a right stochastic matrix:
R = 3×3
    0.1515    0.3333    0.5152
    0.1467    0.3333    0.5200
```

```
    0.1453    0.3333    0.5214
```

```
%(h)
A=jord(3,4)
```

```
A = 3×3
     4     1     0
     0     4     1
     0     0     4
```

```
[S1,S2,L,R]=stochastic(A);
```

```
the vector of sums down each column is
S1 = 1×3
     4     5     5
the vector of sums across each row is
S2 = 3×1
     5
     5
     4
A is neither left nor right stochastic but can be scaled to stochastic
A is scaled to a left stochastic matrix:
L = 3×3
    1.0000    0.2000         0
         0    0.8000    0.2000
         0         0    0.8000
and A is scaled to a right stochastic matrix:
R = 3×3
    0.8000    0.2000         0
         0    0.8000    0.2000
         0         0    1.0000
```

```
%(i)
A=randi(10,4);A(:,1)=0;A(1,:)=0
```

```
A = 4×4
     0     0     0     0
     0     1    10     5
     0     3     2     9
     0     6    10     2
```

```
[S1,S2,L,R]=stochastic(A);
```

```
the vector of sums down each column is
S1 = 1×4
     0    10    22    16
the vector of sums across each row is
S2 = 4×1
     0
    16
    14
    18
A is neither left nor right stochastic and cannot be scaled to either of them
```

Exercise 6

```
type economy
```

```
function [] = economy(n)
format
A = randi(10,n)
S1 = sum(A)

for i = 1:size(A, 2)
    for j = 1:size(A, 1)
        L(j, i) = A(j , i)/S1(i);
    end
end

Sector = L;
T = array2table(Sector)

B = L;
for i = 1:size(L, 1)
    B(i, i) = B(i, i) -1;
end

s = size(B, 1);
horzcat(B, zeros(s, 1))

C = homobasis(B)

syms p
fprintf('vector of equilibrium prices with parameter p = x%i is \n', n)
x = vpa(p*C, 4)



end
```

```
type homobasis
```

```
function C = homobasis(A)
  format
  [~,n]=size(A);
  R=rref(A);
  rankA=rank(A);
  if rankA==n
      disp('the homogeneous system has only the trivial solution')
      C=zeros(n,1);
  else
      disp('the homogeneous system has non-trivial solutions')
      [~,pivot]=rref(A);
      nonpivot=setdiff(1:n,pivot);
      q=numel(nonpivot);
      j=1:q;
      fprintf('a free variable is x%i\n',nonpivot(j))
      C=zeros(n,q);
      R=R(1:rankA,nonpivot);
      C(pivot,:)=-R;
      C(nonpivot,:)=eye(q);
      if rank(C)== size(C,2) && ~any(closetozeroroundoff(A*C,5),'all')
          disp('columns of C form a basis for solution set of homogeneous system')
      else
C=[]; end
end
```

```
%(a)
n = 2;
economy(n)
```

```
A = 2×2
     8     6
     3     7
S1 = 1×2
    11    13
T = 2×2 table
```

	Sector1	Sector2
1	0.7273	0.4615
2	0.2727	0.5385

```
ans = 2×3
   -0.2727    0.4615         0
    0.2727   -0.4615         0
the homogeneous system has non-trivial solutions
a free variable is x2
columns of C form a basis for solution set of homogeneous system
C = 2×1
    1.6923
    1.0000
vector of equilibrium prices with parameter p = x2 is
x =
```

$$\begin{pmatrix} 1.692\,p \\ p \end{pmatrix}$$

```
%(b)
n = 4;
economy(n)
```

```
A = 4×4
     9     2     9     2
    10     3     3     3
     6     9    10     7
     2     3     4     5
S1 = 1×4
    27    17    26    17
T = 4×4 table
```

	Sector1	Sector2	Sector3	Sector4
1	0.3333	0.1176	0.3462	0.1176
2	0.3704	0.1765	0.1154	0.1765
3	0.2222	0.5294	0.3846	0.4118
4	0.0741	0.1765	0.1538	0.2941

```
ans = 4×5
   -0.6667    0.1176    0.3462    0.1176         0
    0.3704   -0.8235    0.1154    0.1765         0
    0.2222    0.5294   -0.6154    0.4118         0
    0.0741    0.1765    0.1538   -0.7059         0
the homogeneous system has non-trivial solutions
a free variable is x4
columns of C form a basis for solution set of homogeneous system
C = 4×1
    1.6205
    1.2722
```

```
    2.3487
    1.0000
vector of equilibrium prices with parameter p = x4 is
x =
```

$$\begin{pmatrix} 1.621\,p \\ 1.272\,p \\ 2.349\,p \\ p \end{pmatrix}$$

```
%(c)
n = 7;
economy(n)
```

```
A = 7×7
     4     8    10     8     7     9     1
     9     4     2     4     8     6     8
     6     6     6     6     5    10     9
     6     1     5     2     1     1     9
    10     1     1     7     3     5     1
     3     6     4     3    10     2     4
     8     8     2     7     2    10     3
S1 = 1×7
    46    34    30    37    36    43    35
T = 7×7 table
```

	Sector1	Sector2	Sector3	Sector4	Sector5	Sector6	Sector7
1	0.0870	0.2353	0.3333	0.2162	0.1944	0.2093	0.0286
2	0.1957	0.1176	0.0667	0.1081	0.2222	0.1395	0.2286
3	0.1304	0.1765	0.2000	0.1622	0.1389	0.2326	0.2571
4	0.1304	0.0294	0.1667	0.0541	0.0278	0.0233	0.2571
5	0.2174	0.0294	0.0333	0.1892	0.0833	0.1163	0.0286
6	0.0652	0.1765	0.1333	0.0811	0.2778	0.0465	0.1143
7	0.1739	0.2353	0.0667	0.1892	0.0556	0.2326	0.0857

```
ans = 7×8
   -0.9130    0.2353    0.3333    0.2162    0.1944    0.2093    0.0286         0
    0.1957   -0.8824    0.0667    0.1081    0.2222    0.1395    0.2286         0
    0.1304    0.1765   -0.8000    0.1622    0.1389    0.2326    0.2571         0
    0.1304    0.0294    0.1667   -0.9459    0.0278    0.0233    0.2571         0
    0.2174    0.0294    0.0333    0.1892   -0.9167    0.1163    0.0286         0
    0.0652    0.1765    0.1333    0.0811    0.2778   -0.9535    0.1143         0
    0.1739    0.2353    0.0667    0.1892    0.0556    0.2326   -0.9143         0
the homogeneous system has non-trivial solutions
a free variable is x7
columns of C form a basis for solution set of homogeneous system
C = 7×1
    1.2624
    1.0239
    1.2599
    0.7397
    0.6671
    0.8291
    1.0000
vector of equilibrium prices with parameter p = x7 is
x =
```

$$\begin{pmatrix} 1.262\ p \\ 1.024\ p \\ 1.26\ p \\ 0.7397\ p \\ 0.6671\ p \\ 0.8291\ p \\ p \end{pmatrix}$$

Exercise 7

```
format
syms x
F = @(x) atan(x) + 2*x - 1
```

```
F = function_handle with value:
    @(x)atan(x)+2*x-1
```

```
F1 = eval(['@(x)' char(diff(F(x)))])
```

```
F1 = function_handle with value:
    @(x)1/(x^2+1)+2
```

```
G=@(x) x.^3-(2/3)*x-1
```

```
G = function_handle with value:
    @(x)x.^3-(2/3)*x-1
```

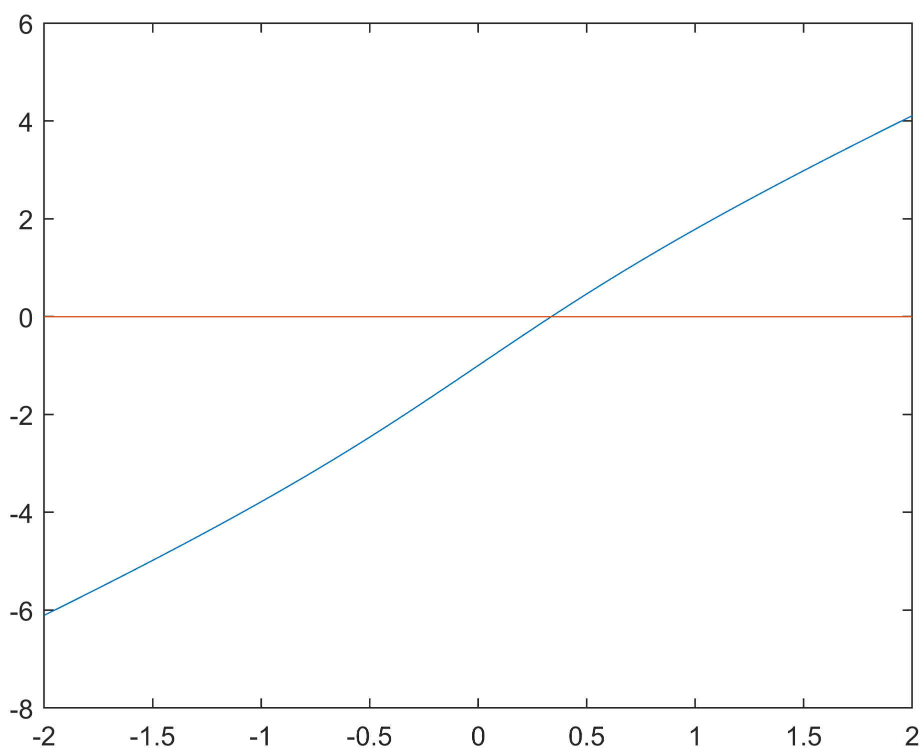
```
G1=eval(['@(x)' char(diff(G(x)))])
```

```
G1 = function_handle with value:
    @(x)3*x^2-2/3
```

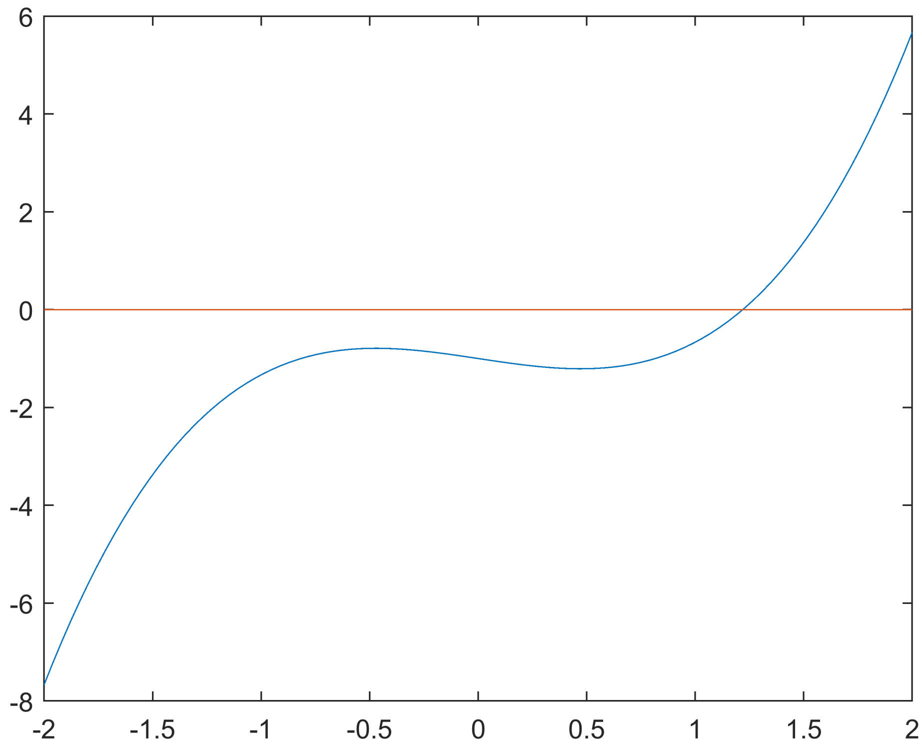
```
yzero=@(x) 0.*x
```

```
yzero = function_handle with value:
    @(x)0.*x
```

```
x=linspace(-2,2);
plot(x,F(x),x,yzero(x));
```



```
plot(x,G(x),x,yzero(x));
```




```
syms x
p=x^3-(2/3)*x-1;
r=sym2poly(p);
R=roots(r);
disp('all zeros of the polynomial are')
```

```
all zeros of the polynomial are
```

```
disp(R)
```

```
   1.2193 + 0.0000i
  -0.6097 + 0.6696i
  -0.6097 - 0.6696i
```

```
for k=1:numel(R)
    if closetozeroroundoff(R(k)-real(R(k)),12)==0
        R(k)=real(R(k));
        disp('a real zero of the polynomial is')
        disp(R(k))
    end
end
```

```
a real zero of the polynomial is
    1.2193
```

```
type newtons
```

```
function root=newtons(fun,dfun,x0)
format long
x=fzero(fun,x0);
disp('a MATLAB approximation of the real zero of the function is')
x
N=0;
while (abs(x0-x)>=(10^(-12)));
    N = N+1;
    x0=x0-(fun(x0)/dfun(x0));
end
root = x0;
disp('our approximation of the real zero of the function is')
root
disp('the number of iterations to archive the required accuracy is')
N
end
```

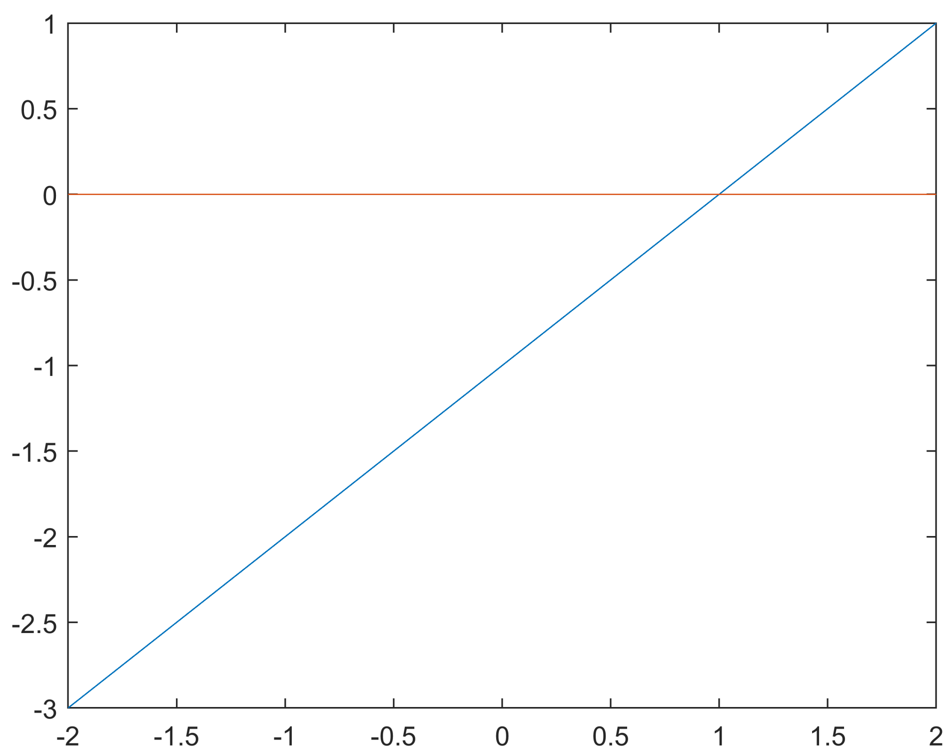
```
%Test
H=@(x) x-1
```

```
H = function_handle with value:
    @(x)x-1
```

```
H1=eval(['@(x)' char(diff(H(x)))])
```

```
H1 = function_handle with value:
    @(x)1
```

```
x=linspace(-2,2);
plot(x,H(x),x,yzero(x));
```




```
fun=H;
```

```
dfun=H1;
x0=0
```

```
x0 = 0
```

```
root=newtons(fun,dfun,x0);
```

```
a MATLAB approximation of the real zero of the function is
x =
     1
our approximation of the real zero of the function is
root =
     1
the number of iterations to archive the required accuracy is
N =
     1
```

```
x0=1
```

```
x0 =
     1
```

```
root=newtons(fun,dfun,x0);
```

```
a MATLAB approximation of the real zero of the function is
x =
     1
our approximation of the real zero of the function is
root =
     1
the number of iterations to archive the required accuracy is
N =
     0
```

(BONUS) For the first intitial interation of x0=0; N=1, this happened because the prediction was wrong and needed to go into the while loop to apply Newtons Method to find the zero intercept, in this case running it through the equation solved it. However, with the second initial interation of x0=1 N=0, this makes sense since the initial prediction is the actual zero intercept, so It wouldn't have to go through the while loop to find the zero intercept.

Part 1

```
fun=F;
dfun=F1;
x0=0.2
```

```
x0 =
   0.200000000000000
```

```
root=newtons(fun,dfun,x0);
```

```
a MATLAB approximation of the real zero of the function is
x =
   0.337328884925534
our approximation of the real zero of the function is
root =
   0.337328884925531
the number of iterations to archive the required accuracy is
N =
     3
```

```
x0=0.3
```

```
x0 =
   0.300000000000000
```

```
root=newtons(fun,dfun,x0);
```

```
a MATLAB approximation of the real zero of the function is
x =
   0.337328884925534
our approximation of the real zero of the function is
root =
   0.337328884925534
the number of iterations to archive the required accuracy is
N =
     3
```

```
x0=0.4
```

```
x0 =
   0.400000000000000
```

```
root=newtons(fun,dfun,x0);
```

```
a MATLAB approximation of the real zero of the function is
x =
   0.337328884925534
our approximation of the real zero of the function is
root =
   0.337328884925534
the number of iterations to archive the required accuracy is
N =
     3
```

Part 2

```
fun=G;
dfun=G1;
%(a)
x0=3
```

```
x0 =
     3
```

```
root=newtons(fun,dfun,x0);
```

```
a MATLAB approximation of the real zero of the function is
x =
   1.219337567364723
our approximation of the real zero of the function is
root =
   1.219337567364723
the number of iterations to archive the required accuracy is
N =
     7
```

```
%(b)
x0=2
```

```
x0 =
     2
```

```
root=newtons(fun,dfun,x0);
```

```
a MATLAB approximation of the real zero of the function is
x =
   1.219337567364723
our approximation of the real zero of the function is
root =
   1.219337567364723
the number of iterations to archive the required accuracy is
N =
     6
```

```
%(c)
x0=1.2
```

```
x0 =
   1.200000000000000
```

```
root=newtons(fun,dfun,x0);
```

```
a MATLAB approximation of the real zero of the function is
x =
   1.219337567364723
our approximation of the real zero of the function is
root =
   1.219337567364740
the number of iterations to archive the required accuracy is
N =
     3
```

```
%(d)
x0=1.1
```

```
x0 =
   1.100000000000000
```

```
root=newtons(fun,dfun,x0);
```

```
a MATLAB approximation of the real zero of the function is
x =
   1.219337567364723
our approximation of the real zero of the function is
root =
   1.219337567364727
the number of iterations to archive the required accuracy is
N =
     4
```

```
%(e)
x0=1
```

```
x0 =
     1
```

```
root=newtons(fun,dfun,x0);
```

```
a MATLAB approximation of the real zero of the function is
x =
   1.219337567364723
our approximation of the real zero of the function is
root =
```

```
   1.219337567364723
the number of iterations to archive the required accuracy is
N =
     5
```

```
%(f)
x0=sqrt(2)/3
```

```
x0 =
   0.471404520791032
```

```
root=newtons(fun,dfun,x0);
```

```
a MATLAB approximation of the real zero of the function is
x =
   1.219337567364723
our approximation of the real zero of the function is
root =
   1.219337567364734
the number of iterations to archive the required accuracy is
N =
    96
```

```
%(g)
x0=0.4714
```

```
x0 =
   0.471400000000000
```

```
root=newtons(fun,dfun,x0);
```

```
a MATLAB approximation of the real zero of the function is
x =
   1.219337567364723
our approximation of the real zero of the function is
root =
   1.219337567364724
the number of iterations to archive the required accuracy is
N =
    37
```

```
%(h)
x0=0
```

```
x0 =
     0
```

```
root=newtons(fun,dfun,x0);
```

```
a MATLAB approximation of the real zero of the function is
x =
   1.219337567364723
our approximation of the real zero of the function is
root =
   1.219337567364723
the number of iterations to archive the required accuracy is
N =
    47
```

(BONUS) When the x0 is set to be very close to the real zero, it ends up being not taking as many interations. However, once we get to complex decimal numbers that go very far it begins to take a lot more interations.

This can be seen from examples g and f although they are basically the same number f takes over double the amount of iterations g does. I believe this is because of how precise the x0 is for f, I decided to look into every iteration for x0 for g and f, and what I observed is that their first iteration basically ended up with an exponent to the same power that the decimal is in. But for the most part the closer the initial iteration, the less iteration it takes to get to the real zero.