P45 1, 2, 3

1.Use Program 2.1 to solve the system $UX = B$, where

$$U = [u_{ij}]_{10\times10} \quad and \quad u_{ij} = \begin{cases} cos(ij) & i \le j, \\ 0 & i > j. \end{cases}$$

and $B = [b_{i1}]_{10\times1}$ and $b_{i1} = tan(i)$.

```
%The following commands are completed in the command line window
U=zeros(10,10);
for i=1:10
    for j=1:10
        if i<=j
            U(i,j)=cos(i*j);
        end
    end
end
B=zeros(10,1);
for k=1:10
    B(k)=tan(k);
end

X = backsub(U,B)
X =

 -748.9196
 -398.4748
 -207.2299
  -83.9979
   42.8938
  -75.7439
   51.4968
  -17.5075
   -0.1486
    0.7519
```

Solution:
$X = (-748.9196, -398.4748, -207.2299, -83.9979, 42.8938, -75.7439, 51.4968, -17.5075, -0.1486, 0.7519)^T$

2.$Forward - substitution\ algoritm.$ A linear system $AX = B$ is called lower triangular provided that $a_{ij} = 0$ when $i < j$. Construct a program forsub, analogous to Program 2.1, to solve the following lower-triangular system. $Remark$ This program will be used in Section 2.3.

```
%forsub.m
function X = forsub(A,B)
%Input  -A is a nxn Lower-triangular nonsingular matrix
%       -B is a nx1 matrix
%Output -X is the solution to the linear system AX=B
%Find the dimension of B and initialize X
n=length(B);
X=zeros(n,1);
X(1)=B(1)/A(1,1);
for k=2:n
    X(k)=(B(k)-A(k,1:k)*X(1:k))/A(k,k);
end
```

Test:

solve the lower-triangular system $AX = B$.

$$\begin{pmatrix} 2 & 0 & 0 & 0 \\ -1 & 4 & 0 & 0 \\ 3 & -2 & -1 & 0 \\ 1 & -2 & 6 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 6 \\ 5 \\ 4 \\ 2 \end{pmatrix}$$

```
%The following commands are completed in the command line window
A=[2,0,0,0;-1,4,0,0;3,-2,-1,0;1,-2,6,3];
B=[6,5,4,2];
X = forsub(A,B)

X =
     3
     2
     1
    -1
```

3.Use forsub to solve the system $LX = B$, where

$$L = [l_{ij}]_{20\times 20} \text{ and } l_{ij} = \begin{cases} i+j & i \geq j, \\ 0 & i < j, \end{cases} \text{ and } B = [b_{i1}]_{20\times 1} \text{ and } b_{i1} = i.$$

```
%The following commands are completed in the command line window
L=zeros(20,20);
B=zeros(20,1);
for i=1:20
    for j=1:20
        if i>=j
            L(i,j)=i+j;
        end
    end
end
for k=1:20
    B(k)=k;
end

X = forsub(L,B)
X =
    0.5000
    0.1250
    0.0625
    0.0391
    0.0273
    0.0205
    0.0161
    0.0131
    0.0109
    0.0093
    0.0080
    0.0070
    0.0062
    0.0055
    0.0050
    0.0045
    0.0041
    0.0038
    0.0035
```

```
    0.0032
```

Solution:
$X = (0.5000, 0.1250, 0.0625, 0.0391, 0.0273, 0.0205, 0.0161, 0.0131, 0.0109, 0.0093, 0.0080,$
$0.0070, 0.0062, 0.0055, 0.0050, 0.0045, 0.0041, 0.0038, 0.0035, 0.0032)^T$

P56 2, 3

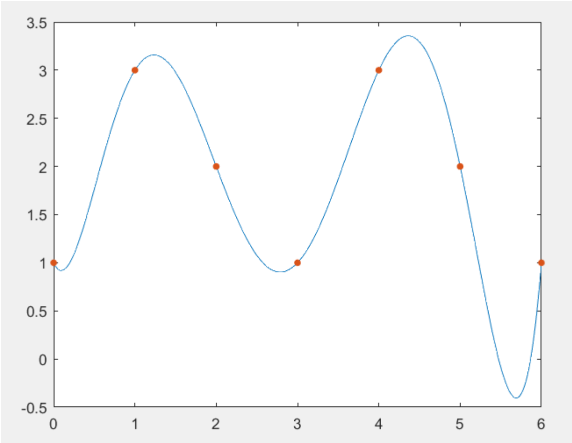
2.Use Program 2.2 to find the sixth-degree polynomial
$y = a_1 + a_2x + a_3x^2 + a_4x^3 + a_5x^4 + a_6x^5 + a_7x^6$ that passes through
$(0, 1), (1, 3), (2, 2), (3, 1), (4, 3), (5, 2)$ and $(6, 1)$. Use the **plot** command to plot the polynomial and the given points on the same graph. Explain any discrepancies in your graph.
Solution:
It is equivalent to finding such a system of equations

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2^2 & 2^3 & 2^4 & 2^5 & 2^6 \\ 1 & 3 & 3^2 & 3^3 & 3^4 & 3^5 & 3^6 \\ 1 & 4 & 4^2 & 4^3 & 4^4 & 4^5 & 4^6 \\ 1 & 5 & 5^2 & 5^3 & 5^4 & 5^5 & 5^6 \\ 1 & 6 & 6^2 & 6^3 & 6^4 & 6^5 & 6^6 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 2 \\ 1 \\ 3 \\ 2 \\ 1 \end{pmatrix}$$

```
%The following commands are completed in the command line window
A=zeros(7,7);
A(1,1)=1;
a=ones(1,7);
A(2,1:7)=1;
for i=3:7
    for j=1:7
        A(i,j)=(i-1)^(j-1);
    end
end

format long
B=[1,3,2,1,3,2,1]';
X = uptrbk(A,B)
X =
   1.000000000000000
  -1.800000000000106
  11.025000000000041
 -10.562500000000005
   3.937500000000001
  -0.637500000000000
   0.037500000000000
px=[0,1,2,3,4,5,6];
py=[1,3,2,1,3,2,1];
plot(x,polyval(flip(X),x))
hold on
scatter(px,py,20,'filled')
[polyval(flip(X),px)',py']
ans =
   1.000000000000000   1.000000000000000
   2.999999999999930   3.000000000000000
   1.999999999999924   2.000000000000000
   0.999999999999966   1.000000000000000
   3.000000000000075   3.000000000000000
   2.000000000000055   2.000000000000000
   1.000000000000923   1.000000000000000
```

The sixth-degree polynomial is
$y(x) = 1 - 1.8x + 11.025x^2 - 10.5625x^3 + 3.9375x^4 - 0.6375x^5 + 0.0375x^6$.



The difference between the value of interpolation curve at the interpolation point and the true value.

x	y(x)	the real value of y
0	1.000000000000000	1.000000000000000
1	2.999999999999930	3.000000000000000
2	1.999999999999924	2.000000000000000
3	0.999999999999966	1.000000000000000
4	3.000000000000075	3.000000000000000
5	2.000000000000055	2.000000000000000
6	1.000000000000923	1.000000000000000

3.Use Program 2.2 to solve the linear system $AX = B$, where $A = [a_{ij}]_{N\times N}$ and $a_{ij} = i^{j-1}$, and $B = [b_{ij}]_{N\times 1}$, where $b_{11} = N$ and $b_{i1} = (i^N - 1)/(i-1)$ for $i \geq 2$. Use $N = 3, 7,$ and 11. The exact solution is $X = [1 \quad 1 \quad \dots \quad 1 \quad 1]'$. Explain any deviations from the exact solution.

```
%A_B.m
function [A,B] = A_B(N)
%Input -N: the dimension of B
A=zeros(N,N);
B=zeros(N,1);
for i=1:N
    for j=1:N
        A(i,j)=i^(j-1);
    end
end
B(1)=N;
for k=2:N
    B(k)=(k^N-1)/(k-1);
end
end
```

```
>> [A,B] = A_B(3);
>> X = uptrbk(A,B)

X =
```

```

     1
     1
     1

>> [A,B] = A_B(7);
>> X = uptrbk(A,B)

X =

     1
     1
     1
     1
     1
     1
     1

>> [A,B] = A_B(11);
>> X = uptrbk(A,B)

X =
    1.0001
    0.9997
    1.0004
    0.9998
    1.0001
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
    1.0000
```

When the dimension of B is 3 and 7, the error between numerical solution and real solution is small, but when the dimension of B is 11, the error between numerical solution and real solution is large.

P62 1

1.Use Program 2.3 to solve the system $AX = B$, where

$$A = \begin{pmatrix} 1 & 3 & 5 & 7 \\ 2 & -1 & 3 & 5 \\ 0 & 0 & 2 & 5 \\ -2 & -6 & -3 & 1 \end{pmatrix} \text{ and } B = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}.$$

Use the [L,U,P]=lu(A) command in MATLAB to check your answer.

$PA = LU, B = P^{-1}LUX$

```
>> A=[1,3,5,7;2,-1,3,5;0,0,2,5;-2,-6,-3,1]
>> B=[1,2,3,4]';
>> [P,X,L,U] = lufact(A,B)
P =
     0     1     0     0
     0     0     0     1
     1     0     0     0
     0     0     1     0
X =
    1.3429
    0.6857
   -3.0000
```

```
    1.8000
L =
    1.0000         0         0         0
   -1.0000    1.0000         0         0
    0.5000   -0.5000    1.0000         0
         0         0    0.5714    1.0000
U =
    2.0000   -1.0000    3.0000    5.0000
         0   -7.0000         0    6.0000
         0         0    3.5000    7.5000
         0         0         0    0.7143
>> inv(P)*L*U*X
ans =
    1.0000
    2.0000
    3.0000
    4.0000
%Use the [L,U,P]=lu(A) command in MATLAB to check your answer
>> [L,U,P]=lu(A)
L =
    1.0000         0         0         0
   -1.0000    1.0000         0         0
    0.5000   -0.5000    1.0000         0
         0         0    0.5714    1.0000

U =
    2.0000   -1.0000    3.0000    5.0000
         0   -7.0000         0    6.0000
         0         0    3.5000    7.5000
         0         0         0    0.7143

P =
     0     1     0     0
     0     0     0     1
     1     0     0     0
     0     0     1     0
>> inv(P)*L*U*X
ans =
    1.0000
    2.0000
    3.0000
    4.0000
%The result is correct
```

```
function [P,X,L,U] = lufact(A,B)
%Input  - A is an N×N matrix
%       - B is an N×1 matrix
%Output - X is an N×1 matrix containing the solution to AX=B.

[N,N]=size(A);
X=zeros(N,1);
Y=zeros(N,1);
C=zeros(1,N);
R=1:N;
for p=1:N-1
    [max1,j]=max(abs(A(p:N,p)));
    C=A(p,:);
    A(p,:)=A(j+p-1,:);
```

```
    A(j+p-1,:)=C;
    d=R(p);
    R(p)=R(j+p-1);
    R(j+p-1)=d;
    if A(p,p)==0
        'A is singular.No unique solution'
        break
    end
    for k=p+1:N
        mult=A(k,p)/A(p,p);
        A(k,p)=mult;
        A(k,p+1:N)=A(k,p+1:N)-mult*A(p,p+1:N);
    end
end
L=tril(ones(N));
for k=2:N
    L(k,1:k-1)=A(k,1:k-1);
end
U=triu(ones(N));
for k=1:N
    U(k,k:N)=A(k,k:N);
end
Y(1)=B(R(1));
for k=2:N
    Y(k)=B(R(k))-A(k,1:k-1)*Y(1:k-1);
end
X(N)=Y(N)/A(N,N);
for k=N-1:-1:1
    X(k)=(Y(k)-A(k,k+1:N)*X(k+1:N))/A(k,k);
end
P=zeros(N,N);
for i=1:N
    P(R(i),i)=1;
end
P=inv(P);
X;
L;
U;
end
```

P73 4, 5

4.Use Gauss-Seidel iteration to solve the following band system.

$$
\begin{array}{rrrrrrrl}
12x_1 & -\ 2x_2 & +\ \ x_3 & & & & & = 5 \\
-2x_1 & +\ 12x_2 & -\ 2x_3 & +\ \ x_4 & & & & = 5 \\
x_1 & -\ 2x_2 & +\ 12x_3 & -\ 2x_4 & +\ \ x_5 & & & = 5 \\
 & x_2 & -\ 2x_3 & +\ 12x_4 & -\ 2x_5 & +\ \ x_6 & & = 5 \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\
 & & x_{46} & -\ 2x_{47} & +\ 12x_{48} & -\ 2x_{49} & +\ \ x_{50} & = 5 \\
 & & & x_{47} & -\ 2x_{48} & +\ 12x_{49} & -\ 2x_{50} & = 5 \\
 & & & & x_{48} & -\ 2x_{49} & +\ 12x_{50} & = 5
\end{array}
$$

5.In Programs 2.4 and 2.5 the relative error between consecutive iterates is used as a stopping criterion. The problems with using this criterion exclusively were discussed in Section 1.3. The linear system $AX = B$ can be rewritten as $AX - B = 0$. If X_k is the kth iterate from a Jacobi or Gauss-Seidel iteration procedure, then the norm of the $residual$ $AX_k - B$ is, in general, a more appropriate stopping criterion.

Modify Programs 2.4 and 2.5 to use the residual as a stopping criterion. Use the modified programs to solve the band system in Problem 4.

$$A = \begin{pmatrix} 12 & -2 & 1 & 0 & 0 & 0 & \dots & 0 \\ -2 & 12 & -2 & 1 & 0 & 0 & \dots & 0 \\ 1 & -2 & 12 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 12 & -2 & 1 & \dots & 0 \\ 0 & 0 & 1 & -2 & 12 & -2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 & -2 & 12 & -2 & 1 \\ 0 & 0 & \dots & 0 & 1 & -2 & 12 & -2 \\ 0 & 0 & \dots & 0 & 0 & 1 & -2 & 12 \end{pmatrix}$$

```
%The following commands are completed in the command line window
format long
v1=12*ones(50,1);
A=diag(v1,0);
v2=-2*ones(49,1);
B1=diag(v2,1);
B2=diag(v2,-1);
v3=ones(48,1);
C1=diag(v3,2);
C2=diag(v3,-2);
A=A+B1+B2+C1+C2;
B=5*ones(50,1);
P=zeros(50,1);
[X1,k1] = jacobi(A,B,P,1e-10,100);
[X2,k2] = gseid(A,B,P,1e-10,100);
%After Jacobi iteration changed the error criterion
[X3,k3] = jacobi1(A,B,P,1e-10,100);
%After gseid iteration changed the error criterion
[X4,k4] = gseid1(A,B,P,1e-10,100);
[X1,X2,X3,X4]
[k1,k2,k3,k4]

ans =
0.463795523811805   0.463795523816549   0.463795523826606   0.463795523816536
0.537284605207901   0.537284605199964   0.537284605183169   0.537284605199954
0.509022924590280   0.509022924601329   0.509022924624674   0.509022924601317
0.498221634449666   0.498221634436173   0.498221634407749   0.498221634436159
0.498941860224472   0.498941860239761   0.498941860271864   0.498941860239747
0.499985351264499   0.499985351248129   0.499985351213909   0.499985351248116
0.500088723873387   0.500088723890134   0.500088723924979   0.500088723890121
0.500015318862541   0.500015318846051   0.500015318811947   0.500015318846038
0.499994793251290   0.499994793266974   0.499994793299207   0.499994793266961
0.499997856927928   0.499997856913466   0.499997856883973   0.499997856913453
0.500000108412254   0.500000108425198   0.500000108451385   0.500000108425185
0.500000201587955   0.500000201576686   0.500000201554102   0.500000201576673
0.500000022601398   0.500000022610944   0.500000022629889   0.500000022610931
0.499999986246451   0.499999986238571   0.499999986223111   0.499999986238558
0.499999995867643   0.499999995873978   0.499999995886263   0.499999995873964
0.500000000535260   0.500000000530292   0.500000000520791   0.500000000530279
0.500000000435245   0.500000000439038   0.500000000446196   0.500000000439025
0.500000000026763   0.500000000023937   0.500000000018691   0.500000000023924
0.499999999963968   0.499999999966016   0.499999999969761   0.499999999966003
0.499999999994002   0.499999999992554   0.499999999989960   0.499999999992541
0.500000000000754   0.500000000001743   0.500000000003489   0.500000000001730
0.500000000001569   0.500000000000916   0.499999999999788   0.500000000000903
```

```
0.499999999999591   0.499999999999994   0.500000000000683   0.499999999999981
0.500000000000140   0.499999999999919   0.499999999999556   0.499999999999906
0.499999999999924   0.499999999999991   0.500000000000107   0.499999999999973
0.499999999999924   0.499999999999907   0.500000000000107   0.499999999999990
0.500000000000140   0.500000000000945   0.499999999999556   0.499999999999936
0.499999999999591   0.499999999995043   0.500000000000683   0.500000000001056
0.500000000001569   0.500000000014202   0.499999999999788   0.499999999990590
0.500000000000754   0.499999999981357   0.500000000003489   0.500000000042652
0.499999999994002   0.500000000006022   0.499999999989960   0.499999999904018
0.499999999963968   0.499999999974528   0.499999999969761   0.500000000065118
0.500000000026763   0.500000000003067   0.500000000018691   0.500000000006478
0.500000000435245   0.500000000443867   0.500000000446196   0.500000000349304
0.500000000535260   0.500000000545105   0.500000000520791   0.500000000604841
0.499999995867643   0.499999995866622   0.499999995886263   0.499999995914885
0.499999986246451   0.499999986229971   0.499999986223111   0.499999986173183
0.500000022601398   0.500000022615558   0.500000022629889   0.500000022593249
0.500000201587955   0.500000201581408   0.500000201554102   0.500000201614978
0.500000108412254   0.500000108423546   0.500000108451385   0.500000108436582
0.499997856927928   0.499997856911217   0.499997856883973   0.499997856897408
0.499994793251290   0.499994793267151   0.499994793299207   0.499994793259522
0.500015318862541   0.500015318846831   0.500015318811947   0.500015318850123
0.500088723873387   0.500088723890258   0.500088723924979   0.500088723893320
0.499985351264499   0.499985351247956   0.499985351213909   0.499985351247756
0.498941860224472   0.498941860239695   0.498941860271864   0.498941860238928
0.498221634449666   0.498221634436195   0.498221634407749   0.498221634436069
0.509022924590280   0.509022924601346   0.509022924624674   0.509022924601452
0.537284605207901   0.537284605199966   0.537284605183169   0.537284605200004
0.463795523811805   0.463795523816549   0.463795523826606   0.463795523816546

ans =
    26    14    25    13
```

For problem 4, X2 in the program is the solution.

For problem 5, X3 in the program is solved by Jacobi iteration using the modified programs and X4 is solved by gseid iteration using the modified programs.

We can see that the number of iterations decreases usint the modified programs.