

## MATH 3940 Problem Set 1 Solutions - Matlab

**Question 2:** (a) `>> A=[-1 2 2 5 1; 0 3 1 2 1; 1 -4 1 0 -2; 0 5 3 1 1; 3 -6 0 4 3];`

`>> det(A)`

`ans = 282.0000`

`>> inv(A)`

```
ans =  -0.4929  1.2234  0.3475  -0.1950  0.0532
       -0.1418  0.5319  0.0496  -0.0993  -0.0638
       0.1738  -0.7766  0.0142  0.4716  0.0532
       0.0638  0.5106  0.1277  -0.2553  -0.0213
       0.1241  -0.8404  -0.4184  0.3369  0.1809
```

(b) `>> B=[ 7 5 9 2 -1]';`

`>> x=A\B`

`x = 5.3511`

`1.9787`

`-1.6489`

`3.6596`

`-6.6064`

**Question 3:** (b) Using the Matlab command `lu`, we find that

`>> A=[0 1 2 -1; 1 1 -1 0; -1 -1 1 3; 1 2 0 1];`

`>> [L U P]=lu(A)`

`L = 1 0 0 0`

`0 1 0 0`

`1 1 1 0`

`-1 0 0 1`

`U = 1 1 -1 0`

`0 1 2 -1`

`0 0 -1 2`

`0 0 0 3`

`P = 0 1 0 0`

`1 0 0 0`

`0 0 0 1`

`0 0 1 0`

```
>> B=[-1 5 1 9]';
```

We write this forward substitution program in M file

```
function X=forsub(A,B)
```

```
% A is an n x n lower triangular nonsingular matrix and B is an n x 1 matrix. 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-1)*X(1:k-1))/A(k,k);
```

```
End
```

We recall in Matlab as follows:

```
>> Y=forsub(L,P*B)
```

```
Y =    5
```

```
   -1
```

```
    5
```

```
    6
```

Then we use the backward substitution program in M file

```
function X=backsub(A,B)
```

```
% A is an n x n upper triangular nonsingular matrix and B is n x 1 matrix. Find the dimension of B and initialize X
```

```
n=length(B);
```

```
X=zeros(n,1);
```

```
X(n)=B(n)/A(n,n);
```

```
for k=n-1:-1:1
```

```
    X(k)=(B(k)-A(k,k+1:n)*X(k+1:n))/A(k,k);
```

```
end
```

```
>> X=backsub(U,Y)
```

```
X =    1
```

```
     3
```

```
    -1
```

```
     2
```

#### **Question 4:**

(b) >> A=[1 2 -1; 2 8 -4; -1 -4 3];

```
>> U=chol(A)
```

```
U =    1    2   -1
```

```
     0    2   -1
```

```
     0    0    1
```

```
>> B=[ -1 -10 7]';
```

Using forward substitutions and backward substitution program from previous question,

```
>> Y=forsub(U',B)
```

```
Y =  -1
```

```
     -4
```

```
      2
```

```
>> X=backsub(U,Y)
```

```
X =   3
```

```
     -1
```

```
      2
```

**Question 5:** (b) M file for Jacobi method is

function [X,k]=jacobi2(A,B,P,tol,maxite)

%A is an N X N nonsingular matrix, B is an N X 1 nonsingular matrix,

% P is an N X 1 nonsingular matrix initial guess, tol is tolerance for P

N=length(B);

for k=1:maxite

for j=1:N

X(j)=(B(j)-A(j,[1:j-1,j+1:N])\*P([1:j-1,j+1:N]))/A(j,j);

end

error=abs(norm(X'-P));

relerr=error/norm(X);

P=X';

if (error<tol)|(relerr<tol)

break

end

end

X=X';

```
>> A=[1 2 -1; 2 8 -4; -1 -4 3];
```

```
>> B=[0 6 -2]';
```

```
>>[X,k]=jacobi2(A,B,[0 0 0]',10^(-6),35)
```

```
X =  1.0e+04 *
```

```
    6.5540
```

```
    2.5961
```

```
   -4.0179
```

```
k =   35          The iterations diverge.
```

(d) M file for Gauss-Siedel method is

```
function [X,k]=gauseid(A,B,P,tol,maxite)
% A is an N X N nonsingular matrix, B is an N X 1 nonsingular matrix,
% P is an N X 1 nonsingular matrix initial guess, tol is tolerance for P
Digits=8;
N=length(B);
for k=1:maxite
    for j=1:N
        if j==1
            X(1)=(B(1)-A(1,2:N)*P(2:N))/A(1,1);
        elseif j==N
            X(N)=(B(N)-A(N,1:N-1)*(X(1:N-1)))/A(N,N);
        else
            % X contains the kth approximations and P the (k-1)st
            X(j)=(B(j)-A(j,1:j-1)*(X(1:j-1))-A(j,j+1:N)*P(j+1:N))/A(j,j);
        end
    end
    error=norm(X'-P);
    relerr=error/norm(X);
    P=X';
    if (error<tol) || (relerr<tol)
        break
    end
end
X=X';
>> A=[1 2 -1; 2 8 -4; -1 -4 3];
>> B=[0 6 -2]';
>> [X, k]=gauseid(A,B,[0 0 0]',10^(-6),35)
X =  -3.0000
      2.0000
      1.0000
k =  33
```

The iterations converge in 33 iterations.