Table of Contents

MATH2019 (2019-2020) Coursework 2

NAME: JAKE DENTON STUDENT ID: 14322189

```
clear all;
close all;
clc
```

Question 1

See the file forwardElim.m

Question 2

Following code uses forward elimination (forwardElim.m) to obtain augmented matrices A2, A3, A4, and subsequently obtains the solution x.

```
A=[1,-1,2,-1,-8;1,1,1,0,-2;2,0,2,-2,-14;1,-3,6,3,2]; %define augmented
 matrix A
format shortg
for m=1:4 %this for loop finds the matrix obtained by m rounds of
 forward-elimination and displays it
    [Am]=forwardElim(A,m); %implements forward elimination step
    disp(['A',num2str(m),' is the following augmented matrix:']);
    disp(Am);%displays the matrix after m rounds of row operations
end
LinA=[1,-1,2,-1;1,1,1,0;2,0,2,-2;1,-3,6,3]; %this is the 4x4 matrix
 representing the linear system
A4=forwardElim(A,m);%defines A4 as final echelon form of A
x=backwardSub(A4);%implements backward substitution to obtain solution
 vector x
disp('The solution vector x is as follows:');%rest of the code
 displays x and confirms x is the solution vector
disp(x);
b=LinA*x;%if x is correct then b will be the constant vector which is
 the (n+1)st column of A
disp('The solution vector x reproduces As (n+1)st column when left
 multiplied by 4x4 matrix of system:');
```

```
disp(b)
```

```
A1 is the following augmented matrix:
     1    -1     2    -1    -8
     1     1     1     0    -2
     2     0     2    -2   -14
     1    -3     6     3     2

A2 is the following augmented matrix:
     1    -1     2    -1    -8
     0     2    -1     1     6
     0     2    -2     0     2
     0    -2     4     4    10

A3 is the following augmented matrix:
     1    -1     2    -1    -8
     0     2    -1     1     6
     0     0    -1    -1    -4
     0     0     3     5    16

A4 is the following augmented matrix:
     1    -1     2    -1    -8
     0     2    -1     1     6
     0     0    -1    -1    -4
     0     0     0     2     4

The solution vector x is as follows:
    -7
     3
     2
     2

The solution vector x reproduces As (n+1)st column when left
 multiplied by 4x4 matrix of system:
    -8
    -2
   -14
     2
```

Question 3

Following code produces a figure with the timings of forwardElim.m and backwardSub.m for different n

```
for n=50:50:1500 %defines a range of n values we'd like to investigate
    [A,b]=testMat(n);%uses the function given to provide an augmented
 nx(n+1) matrix
    AUG=[A,b]; %sets the augmented matrix from step above to the
 matrix AUG to put into forwardElim
    tic %starts MATLAB's in-built timer so that forwardElim.m can be
 timed
    ech=forwardElim(AUG,n);%performs the forward elimination step
    toc %stops the timer
```

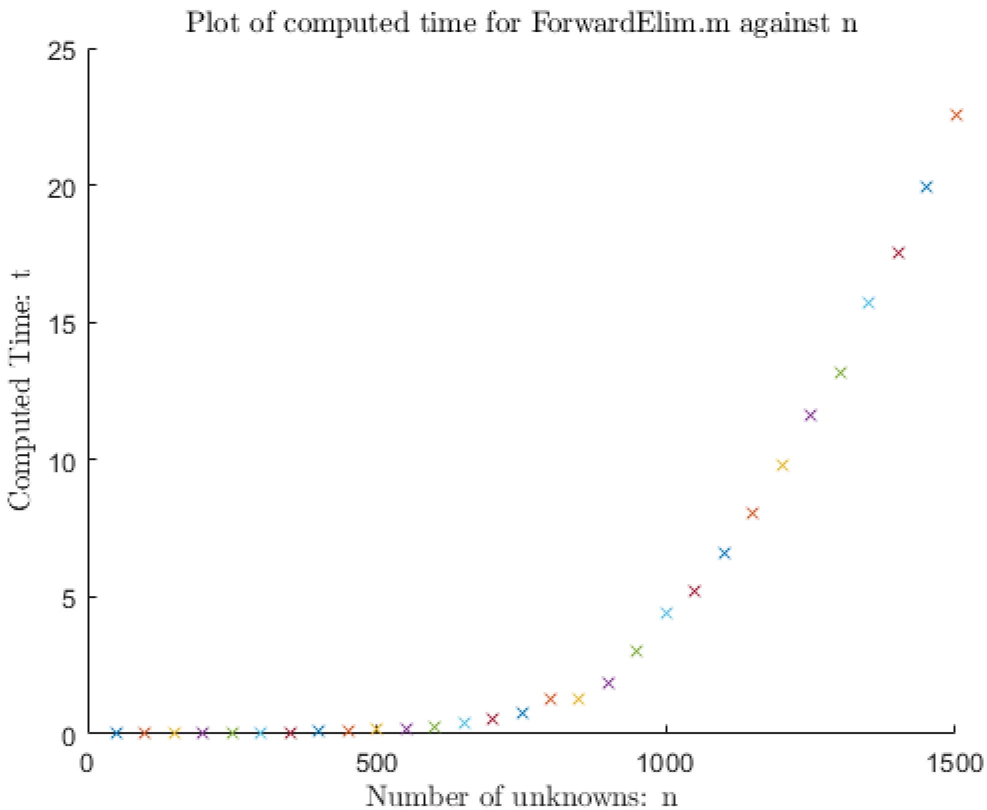
```
    t=toc; %sets t as equal to the returned time taken given by toc
    hold all %allows all points to be plotted on the same set of axes
    loglog(n,t,'x'); %plots point on logarithmic scaled axes
end
set(groot,'DefaultTextInterpreter','latex'); %rest of this code
 formats the graph
title('Plot of computed time for ForwardElim.m against n');
xlabel('Number of unknowns: n');
ylabel('Computed Time: t');


% Explanation for the observed behaviour in the plots for large n:

% From the Gaussian elimination algorithm, when we have an nxn system
 it
% requires (n^3/3)+(n^2/2)-(5n/6) multiplications/divisions and
% (n^3/3)-(n/3) additions/subtractions over the course of the
% forward-elimination step. Thus for large n (such as those that I
 have
% used for my graph), the number of operations altogether is
 approximately
% (2*n^3/3). This cubic order can be observed from the graph as it has
 a
% slow gradual slope until it starts increasing almost vertically from
% n=1000. This is due to the fact that beyond this each increase of 50
 in n
% vastly adds to the number of operations required to find the echelon
% form. To show this consider the difference in the approx. number of
% operations between n and (n+50): (2/3)*[(n+50)^3-n^3]=100*n^2+O(n).
 From
% this, if n=1000,the next term has 10^8 more operations! Due to these
% operations, the time taken to complete the task increases
 proportionally
% leading to the steepness that can be seen from the graph.
```

```
Elapsed time is 0.000891 seconds.
Elapsed time is 0.001094 seconds.
Elapsed time is 0.002934 seconds.
Elapsed time is 0.006986 seconds.
Elapsed time is 0.015575 seconds.
Elapsed time is 0.034123 seconds.
Elapsed time is 0.056892 seconds.
Elapsed time is 0.089128 seconds.
Elapsed time is 0.107402 seconds.
Elapsed time is 0.151272 seconds.
Elapsed time is 0.196329 seconds.
Elapsed time is 0.286015 seconds.
Elapsed time is 0.423425 seconds.
Elapsed time is 0.550860 seconds.
Elapsed time is 0.780787 seconds.
Elapsed time is 1.270580 seconds.
Elapsed time is 1.253027 seconds.
Elapsed time is 1.848547 seconds.
Elapsed time is 2.992399 seconds.
```

```
Elapsed time is 4.429805 seconds.
Elapsed time is 5.227550 seconds.
Elapsed time is 6.621979 seconds.
Elapsed time is 8.020206 seconds.
Elapsed time is 9.830387 seconds.
Elapsed time is 11.606277 seconds.
Elapsed time is 13.167179 seconds.
Elapsed time is 15.697947 seconds.
Elapsed time is 17.534501 seconds.
Elapsed time is 19.934994 seconds.
Elapsed time is 22.564901 seconds.
```



Question 4*

See the file forwardElimCP.m

Question 5*

Following code applies forward elimination with complete pivoting (forwardElimCP.m) to B1 to obtain augmented matrices B2, B3, B4, and, subsequently obtains the solution x.

```
B1 = [...
     0    -1     2    -1     0
     1    -1     1     0    -9
     2    -2     3    -3   -21
```

```
     1    -1     4     3     6
    ]; %define augmented matrix B
format shortg
for m=1:4 %this for loop finds the matrix obtained by m rounds of
 forward-elimination with complete pivoting and displays it
    [Am]=forwardElimCP(B1,m); %implements forward elimination step
 with complete pivoting
    disp(['B',num2str(m),' is the following augmented matrix:']);
    disp(Am);%displays the matrix after m rounds of row operations
end
BLin=B1(1:4,1:4); %this is the 4x4 matrix representing the linear
 system
B4=forwardElimCP(B1,m);%defines B4 as final echelon form of B1
x=backwardSub(B4);%implements backward substitution to obtain solution
 vector x
disp('The solution vector x is as follows:');%rest of the code
 displays x
disp(x);


% Following code applies forward elimination
% with complete pivoting (forwardElimCP.m) to A

A=[1,-1,2,-1,-8;1,1,1,0,-2;2,0,2,-2,-14;1,-3,6,3,2]; %define augmented
 matrix A
format shortg
for m=1:4 %this for loop finds the matrix obtained by m rounds of
 forward-elimination with complete pivoting and displays it
    [Am]=forwardElimCP(A,m); %implements forward elimination step w/
 complete pivoting
    disp(['A',num2str(m),' is the following augmented matrix:']);
    disp(Am);%displays the matrix after m rounds of row operations
end
LinA=[1,-1,2,-1;1,1,1,0;2,0,2,-2;1,-3,6,3]; %this is the 4x4 matrix
 representing the linear system
A4=forwardElimCP(A,m);%defines A4 as final echelon form of A
x=backwardSub(A4);%implements backward substitution to obtain solution
 vector x
disp('The solution vector x is as follows:');%rest of the code
 displays x
disp(x);
% Explanation: Applying backwardSub.m does not give the same column
 vector
% x as a solution. This is due to the fact that during complete
 pivoting,
% columns may be transposed according to the location of the maximum
% element. This is the equivalent of renaming the variables in the
 linear
% system (this does not change the actual values within x but does
 change
% their order within the column vector). Since these transpositions do
% occur when forwardElimCP.m is applied to A, the solution vector x
 will be
% different than that obtained from forwardElim.m in question 2.
```

```
B1 is the following augmented matrix:
     0    -1     2    -1     0
     1    -1     1     0    -9
     2    -2     3    -3   -21
     1    -1     4     3     6

B2 is the following augmented matrix:
            4          -1             1             3             6
            0       -0.75          0.75         -0.75         -10.5
            0       -1.25          1.25         -5.25         -25.5
            0        -0.5          -0.5          -2.5            -3

B3 is the following augmented matrix:
            4           3             1            -1             6
            0       -5.25          1.25         -1.25         -25.5
            0           0       0.57143      -0.57143       -6.8571
            0           0       -1.0952      0.095238        9.1429

B4 is the following augmented matrix:
            4           3             1            -1             6
            0       -5.25          1.25         -1.25         -25.5
            0           0       -1.0952      0.095238        9.1429
            0           0             0      -0.52174        -2.087

The solution vector x is as follows:
            3
            2
           -8
            4

A1 is the following augmented matrix:
     1    -1     2    -1    -8
     1     1     1     0    -2
     2     0     2    -2   -14
     1    -3     6     3     2

A2 is the following augmented matrix:
            6          -3             1             3             2
            0         1.5       0.83333          -0.5       -2.3333
            0           1        1.6667            -3       -14.667
            0           0       0.66667            -2       -8.6667

A3 is the following augmented matrix:
            6           3             1            -3             2
            0          -3        1.6667             1       -14.667
            0           0       0.55556        1.3333       0.11111
            0           0      -0.44444      -0.66667        1.1111

A4 is the following augmented matrix:
            6           3            -3             1             2
            0          -3             1        1.6667       -14.667
            0           0        1.3333       0.55556       0.11111
            0           0             0      -0.16667        1.1667
```

```
The solution vector x is as follows:
           2
           2
           3
          -7
```

Question 6

See the file forwardElimLU.m

Question 7

Following code applies forwardElimLU to obtain the LU factorisation of A, and checks whether L*U = A

```
A=[1,-1,2,-1;1,1,1,0;2,0,2,-2;1,-3,6,3]; %defines the matrix A
[L,U]=forwardElimLU(A);%implements the forwardElimLU function to
 factorise A
disp(L);%displays the lower triangular matrix L
disp(U);%displays the upper triangular matrix U
LU=L*U;%calculates the product of the output matrices L and U, if
 factorisation is correct, should be equal to A
disp(LU);%displays the value of the product above (=A)
```

```
     1     0     0     0
     1     1     0     0
     2     1     1     0
     1    -1    -3     1

     1    -1     2    -1
     0     2    -1     1
     0     0    -1    -1
     0     0     0     2

     1    -1     2    -1
     1     1     1     0
     2     0     2    -2
     1    -3     6     3
```

Question 8*

Following code solves for x

```
A=[1,-1,2,-1;1,1,1,0;2,0,2,-2;1,-3,6,3];%defines the matrix A from Q7
[L,U]=forwardElimLU(A); %factorises A into L and U
b=[-8;-2;-14;2];%defines the column vector b in the problem Ax=b
P=[0,0,0,1;0,0,1,0;0,1,0,0;1,0,0,0];%defines a permutation matrix P
RowTA=P*L;%P acts to swap the 1st/4th row and 2nd/3rd row of L
RowTb=P*b;%P swaps rows as above so that the system remains equivalent
ColRowTA=RowTA*P;%P next swaps the 1st/4th column and 2nd/3rd row,
 transforming L into an upper triangular matrix
```

```
augLy=[ColRowTA,RowTb];%forms the augmented matrix which we will use
 to solve Ly=b with the transformed L and appropriately row swapped b
y=backwardSub(augLy);%uses backward substitution to solve system Ly=b
 for y
Rowy=P*y;%applies row swaps to y so that system remains equivalent
augUx=[U,Rowy];%forms augmented matrix for use in solving Ux=y
x=backwardSub(augUx);%solves Ux=y for x, giving the solution vector to
 Ax=b
disp('The solution vector x to the system Ax=b is:');
disp(x);
```

```
The solution vector x to the system Ax=b is:
    -7
     3
     2
     2
```

Published with MATLAB® R2019a