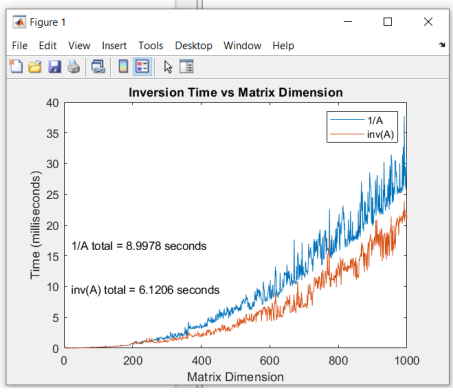
Michael Dang – 16257750

MATH434

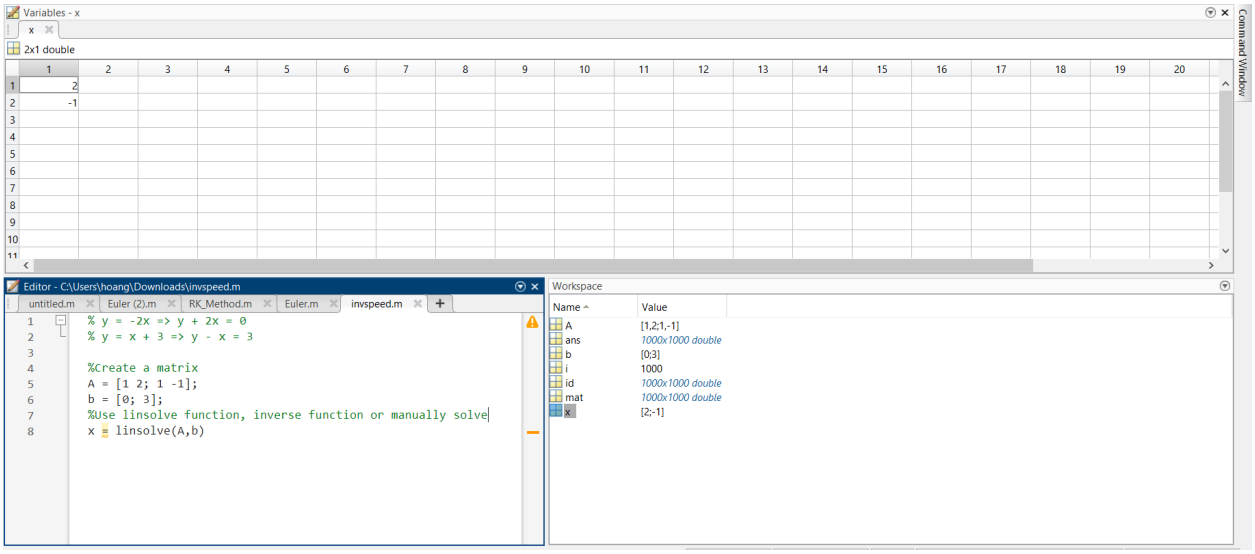
Lab2

1.

```
%invspeed.m
%invspeed.m calculates the time to perform two types of matrix inversions;
%it then plots the time versus matrix dimension.
for i=1:1000
    id=eye(i);
    mat=rand(i);
    tic;
    id/mat;
    x(i,1)=toc;
    tic;
    inv(mat);
    x(i,2)=toc;
end
plot(1000*x)
title('Inversion Time vs Matrix Dimension')
legend('1/A', 'inv(A)')
xlabel('Matrix Dimension')
ylabel('Time (milliseconds)')
text(20, max(x(:,2))*700,['1/A total = ', num2str(sum(x(:,1))), ' seconds'])
text(20, max(x(:,2))*400,['inv(A) total = ', num2str(sum(x(:,2))), ' seconds'])
```



2. a.



b.

Editor - C:\Users\hoang\Downloads\invspeed.m

untitled.m | Euler (2).m | RK_Method.m | Euler.m | invspeed.m

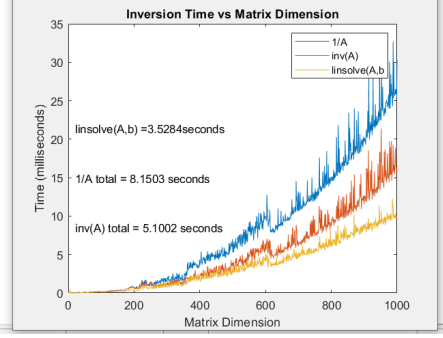
```
%invspeed.m
%invspeed.m calculates the time to perform two types of matrix inversions;
%it then plots the time versus matrix dimension.
for i=1:1000
    id=eye(i);
    mat=rand(i);
    tic;
    id/mat;
    x(i,1)=toc;
    tic;
    inv(mat);
    x(i,2)=toc;
    tic;
    b = ones(i,1);
    linsolve(mat,b);
    x(i,3)=toc;
end
plot(1000*x)
title('Inversion Time vs Matrix Dimension')
legend('1/A', 'inv(A)','linsolve(A,b')
xlabel('Matrix Dimension')
ylabel('Time (milliseconds)')
text(20, max(x(:,2))*700,['1/A total = ', num2str(sum(x(:,1))), ' seconds'])
text(20, max(x(:,2))*400,['inv(A) total = ', num2str(sum(x(:,2))), ' seconds'])
text(20, max(x(:,2))*1000,['linsolve(A,b) =', num2str(sum(x(:,3))), 'seconds'])
```

Workspace

Name	Value
A	[1,2;1,-1]
ans	1000x1 double
b	1000x1 double
i	1000
id	1000x1000 double
mat	1000x1000 double
x	1000x3 double

Figure 1

File Edit View Insert Tools Desktop Window Help



3. a.

Editor - C:\Users\hoang\Downloads\lot.m

RK_Method.m | Euler.m | lot.m

```
function lot(N, R)
%lot picks lottery numbers. N numbers from 1 to R are chosen.
if (N <= R & R > 0)
    tot = randperm(R);
%  returns  a  row  vector  containing  a  random  permutation  of  the  in
%repeating %elements.
    x = tot(1:N);
    disp(['Your lottery numbers are: ', num2str(x)])
end
```

Command Window

```
>> lot(3,600)
Your lottery numbers are: 80  248  253
>> lot(3,600)
Your lottery numbers are: 49  105  157
>> lot(3,600)
Your lottery numbers are: 315  580  310
>> lot(3000,600)
fx >>
```

b.

Editor - C:\Users\hoang\Downloads\mystats.m

RK_Method.m | Euler.m | lot.m | mystats.m

```
function [avg, med] = mystats(x)
n = length(x);
avg = mymean(x,n);
med = mymedian(x,n);
end
function a = mymean(v,n)
% MYMEAN Example of a local function.
a = sum(v)/n;
end
function m = mymedian(v,n)
% MYMEDIAN Another example of a local function.
w = sort(v);
if rem(n,2) == 1
    m = w((n + 1)/2);
else
    m = (w(n/2) + w(n/2 + 1))/2;
end
end
```

Command Window

```
>> x = rand(1,300);
>> mystats(x)

ans =

    0.4658

fx >>
```

4.

Editor - C:\Users\hoang\Downloads\l.m

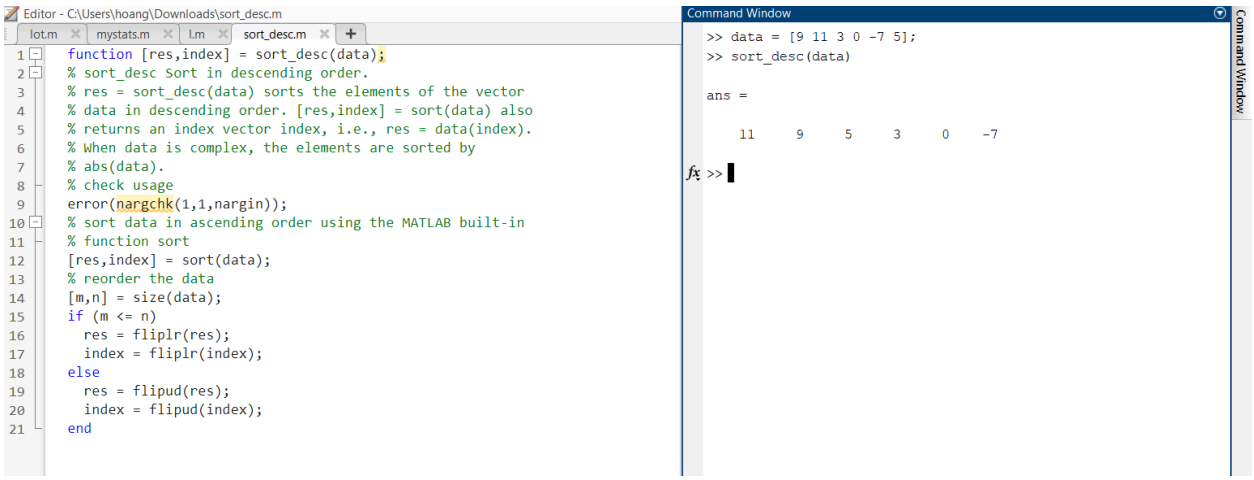
lot.m | mystats.m | l.m

```
%matrix A and vector b
function [x1, x2] = l(A,b)
x1 = A\b;
x2 = inv(A)*b;
end
```

Command Window

```
>> A

A =

     2     3     1
     4    -1     0
     1     2     3

>> b

b =

     2
     4
     1

>> l(A,b)

ans =

     1
     0
     0

fx >>
```

5.



Editor - C:\Users\hoang\Downloads\sort_desc.m

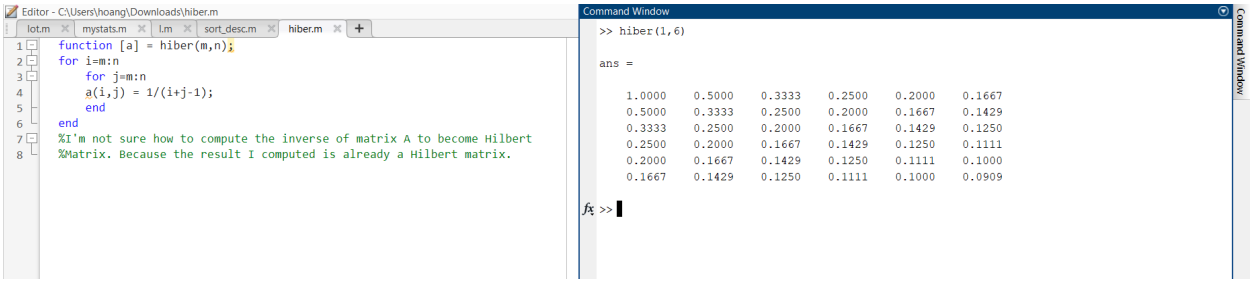
lot.m | mystats.m | l.m | sort_desc.m

```
function [res,index] = sort_desc(data);
% sort_desc Sort in descending order.
% res = sort_desc(data) sorts the elements of the vector
% data in descending order. [res,index] = sort(data) also
% returns an index vector index, i.e., res = data(index).
% When data is complex, the elements are sorted by
% abs(data).
% check usage
error(nargchk(1,1,nargin));
% sort data in ascending order using the MATLAB built-in
% function sort
[res,index] = sort(data);
% reorder the data
[m,n] = size(data);
if (m <= n)
  res = fliplr(res);
  index = fliplr(index);
else
  res = flipud(res);
  index = flipud(index);
end
```

Command Window

```
>> data = [9 11 3 0 -7 5];
>> sort_desc(data)

ans =

    11     9     5     3     0    -7

fx >>
```

6.



Editor - C:\Users\hoang\Downloads\hiber.m

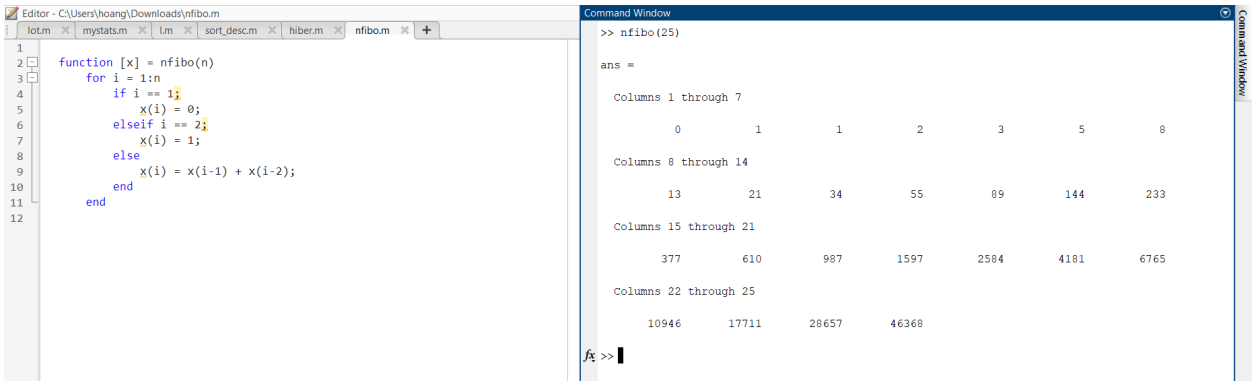
lot.m | mystats.m | l.m | sort_desc.m | hiber.m

```
function [a] = hiber(m,n);
for i=m:n
    for j=m:n
    a(i,j) = 1/(i+j-1);
    end
end
%I'm not sure how to compute the inverse of matrix A to become Hilbert
%Matrix. Because the result I computed is already a Hilbert matrix.
```

Command Window

```
>> hiber(1,6)

ans =

    1.0000    0.5000    0.3333    0.2500    0.2000    0.1667
    0.5000    0.3333    0.2500    0.2000    0.1667    0.1429
    0.3333    0.2500    0.2000    0.1667    0.1429    0.1250
    0.2500    0.2000    0.1667    0.1429    0.1250    0.1111
    0.2000    0.1667    0.1429    0.1250    0.1111    0.1000
    0.1667    0.1429    0.1250    0.1111    0.1000    0.0909

fx >>
```

7.



Editor - C:\Users\hoang\Downloads\nfibo.m

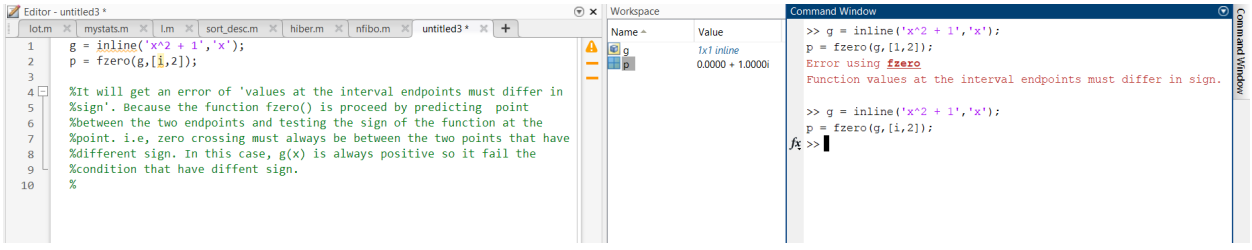
lot.m | mystats.m | l.m | sort_desc.m | hiber.m | nfibo.m

```
function [x] = nfibo(n)
    for i = 1:n
        if i == 1;
            x(i) = 0;
        elseif i == 2;
            x(i) = 1;
        else
            x(i) = x(i-1) + x(i-2);
        end
    end
```

Command Window

```
>> nfibo(25)

ans =

  Columns 1 through 7

           0           1           1           2           3           5           8

  Columns 8 through 14

          13          21          34          55          89         144         233

  Columns 15 through 21

         377         610         987        1597        2584        4181        6765

  Columns 22 through 25

       10946       17711       28657       46368

fx >>
```

8.



Editor - untitled3 *

lot.m | mystats.m | l.m | sort_desc.m | hiber.m | nfibo.m | untitled3 *

```
g = inline('x^2 + 1','x');
p = fzero(g,[i,2]);

%It will get an error of 'values at the interval endpoints must differ in
%sign'. Because the function fzero() is proceed by predicting  point
%between the two endpoints and testing the sign of the function at the
%point. i.e, zero crossing must always be between the two points that have
%different sign. In this case, g(x) is always positive so it fail the
%condition that have diffent sign.
%
```

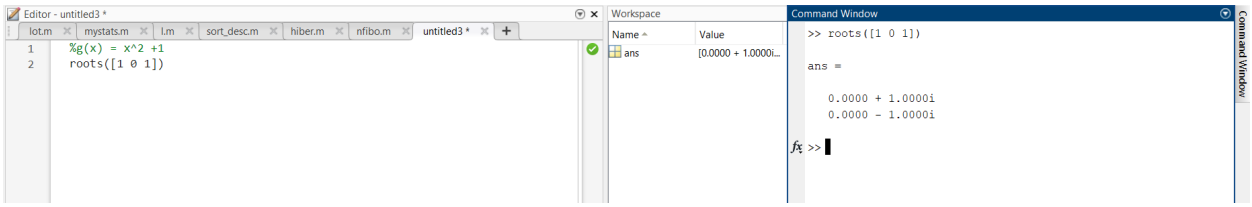
Workspace

Name	Value
g	1x1 inline
p	0.0000 + 1.0000i

Command Window

```
>> g = inline('x^2 + 1','x');
p = fzero(g,[1,2]);
Error using fzero
Function values at the interval endpoints must differ in sign.

>> g = inline('x^2 + 1','x');
p = fzero(g,[i,2]);
fx >>
```

9.



10.

This computation of eigenvalues in MATLAB can determine as the root() function for polynomial. As the professor demonstrated in class, the result shows the same. i.e, we can use root() to find the eigenvalues of the matrix as long as we can get the characteristic equation.

11.