Math 244: MATLAB Assignment 6

Name: Kym Derriman

RUID: kad424

Date: Nov 18, 2024

```
clear;
clc;
```

1.

```
A = [2 3 -1; -3 4 0];
B = [1 5; 0 -2; -1 3];
C = [2 -5; 1 3];
v = [1; 2; 5];
w = [2; 4];
y = [0; 3; 2];

aval = A(1,3);
fprintf('A(1,3): %d\n',aval);
Bval = B(3,2);
fprintf('B(3,2): %d\n',Bval);
yval = y(2);
fprintf('y(2): %d\n',yval);
```

2.

```
% Matrix Computations
fprintf('A*B = \n');
disp(A*B);
fprintf('B*A = \n');
disp(B*A);
fprintf('B*C = \n');
disp(B*C);
fprintf('C*A = \n');
disp(C*A);
fprintf('A*v = \n');
disp(A*v);
fprintf('B*w = \n');
disp(B*w);
fprintf('B*A*v = \n');
disp(B*(A*v));
fprintf('A(v - 3y) = \n');
disp((A*(v - 3.*y)));
```

3.

```
% Try the computation here
% A*C
```

It is not possible to compute AC because their dimensions are not compatible. The number of rows of C must equal the number of columns in A.

```
"Error using  * 

Incorrect dimensions for matrix multiplication. Check that the number of columns
in the first matrix matches the number of rows in the second matrix. To operate
on each element of the matrix individually, use TIMES (.*) for elementwise
multiplication."
```

4.

```
% Try the computation here
v2 = 2.*v;
w3 = 3.*w;
%v2 + w3
% Arrays have incompatible sizes for this operation.
```

"Error: Arrays have incompatible sizes for this operation."

The arrays do not have the same size so adding them is not defined.

5.

```
% Try the computation here
% A^2
A
A.^2
```

"Error using ^ (line 52)

Incorrect dimensions for raising a matrix to a power. Check that the matrix is square and the power is a scalar. To operate on each element of the matrix individually, use POWER (.^) for elementwise power."

The operation A.^2 squares each element of A, whereas A^2 returns an error because A does not have the correct dimensions for squaring the matrix itself (A*A).

6.

```
% Matrix C computations
C
fprintf('C^2: \n')
C^2
fprintf('C.^2: \n');
C.^2
```

C^2 multiplies the matrix C by itself (C*C), which is possible because C is a square matrix. C.^2 returns a different answer because this tells matlab to square the matrix element-wise, as in square each element individually.

7.

```
% Dot product computations
sum(v.*y)
v'*y
```

```
% Matrix transposes
B
B'
B*C
C'*B'
```

C'B' is the transpose of BC.

8.

```
% M information
M = [-1 -12 -2; 0 4 0; 3 12 4];
d = det(M)
```

```
d =
8
```

```
[V,D] = eig(M)
```

```
V = 3×3
   -0.7071    0.5547   -0.6963
         0         0    0.1741
    0.7071   -0.8321    0.6963
D = 3×3
    1.0000         0         0
         0    2.0000         0
         0         0    4.0000
```

```
v1 = [-4;1;4];
lambda1 = 4;
v2 = [-2;0;3];
labmda2 = 2;
v3 = [-1;0;1];
lambda3 = 1;
M*v1 == lambda1.*v1
```

```
ans = 3×1 logical array
   1
   1
   1
```

```
M*v2 == labmda2.*v2
```

```
ans = 3×1 logical array
   1
   1
   1
```

```
M*v3 == lambda3.*v3
```

```
ans = 3×1 logical array
   1
   1
   1
```

I gotta be honest here, this is frustrating to get this output. To get clean integer vectors, a whole annoying process is required. I used Mathematica and obtained the following:

Eigenvalues: {4,2,1}

Eigenvectors: {{-4, 1, 4}, {-2, 0, 3}, {-1, 0, 1}}

These correspond to the MATLAB output, as shown by the output of the logical arrays here, where 1 is true represents M*v = lambda*v.

9.

```
%N information
N = [-10 -12 12; 6 7 -4; -3 -4 7];
[V,D] = eig(N)
```

```
V = 3×3
   -0.8000   -0.4082    0.0000
    0.6000    0.8165    0.7071
    0.0000    0.4082    0.7071
D = 3×3
   -1.0000         0         0
         0    2.0000         0
         0         0    3.0000
```

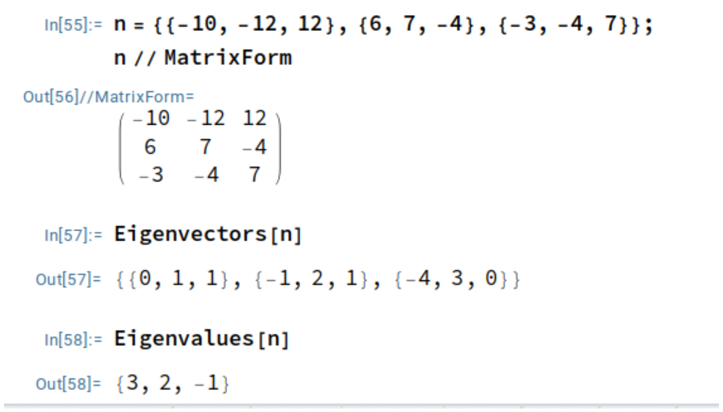
```

M = [-1 -12 -2; 0 4 0; 3 12 4];
d = det(M);
[V,D] = eig(M);
v = V((1:3),1);
v./V(1,1)
```

```
ans = 3×1
    1.0000
         0
   -1.0000
```

Okay, again, MATLAB here outputs one of the eigenvectors as [-0.4082; 0.8165; 0.4082], which is a scalar multiple of the eigenvector provided in the problem [-1;2;1], with some rounding.

Mathematica, on the other hand, outputs clean eigenvectors, with the option of numerical output similar to MATLAB if one prefers. Rutgers provides access for free, not sure why we don't use it.

Side note: I read the end of the lab and see now that it's pretty easy to make the eigenvectors clean by dividing by the smallest entry. Still... use Mathematica! :)



```
In[55]:= n = {{-10, -12, 12}, {6, 7, -4}, {-3, -4, 7}};
         n // MatrixForm
Out[56]//MatrixForm=
```

$$\begin{pmatrix} -10 & -12 & 12 \\ 6 & 7 & -4 \\ -3 & -4 & 7 \end{pmatrix}$$

```
In[57]:= Eigenvectors[n]
Out[57]= {{0, 1, 1}, {-1, 2, 1}, {-4, 3, 0}}

In[58]:= Eigenvalues[n]
Out[58]= {3, 2, -1}
```