Math 22AL Lab #3

0.1 Notes

1. **Plotting Graphs** `to see plots, you have to use one of the terminals in the Mathematics department computer lab. Connecting from home via ssh will not allow you to see plots.`

2. `Green typewriter text` represents comments you must type. Each comment is worth one point.

3. `Blue typewriter text` represents commands you must type. Each command is worth one point. MATLAB is case-sensitive, so commands are case-sensitive as well. Beware of typos! Also, do not copy and paste commands. Special characters are sometimes used (e.g. the left quote character) which MATLAB does not like.

1 Objectives

In this lab, you will explore the following topics using MATLAB:

- Using the plotting commands of MATLAB to create graphs.
- Creating and using m-files
- Exploring linear transformations in $\mathbb{R}^2$

2 Header

Start MATLAB and Enter your information:

```
TYPE:    textmatlab
TYPE:    diary LAB3.text
COMMENT: % [your last name]
COMMENT: % [your first name]
COMMENT: % [the date, in any format]
```

3 Plotting Graphs

Note: to see plots, you have to use one of the terminals in the Mathematics department computer lab. Connecting from home via `ssh` will not allow you to see plots.

3.1 Generating 2-D Graphs

To graph a line segment passing through the points $(-1, 2)$ and $(4, -3)$,

TYPE: `vt1=datestr(now)`

TYPE: `x = [-1 2]; y = [4 -3];`
TYPE: `plot(x, y)`

Use the `axis` command to rescale the axes so that the line segment does not take up the entire window.

TYPE: `axis([-3 6 -4 5])`

You may chose a color for your lines by specifying it with a string.

TYPE: `plot(x, y, 'r')`

The line is now red! You can also make it dashed or dotted by specifying it with a string.

TYPE: `plot(x, y, 'g--')`

The line is now dashed and green! See `http://www.mathworks.com/help/matlab/ref/plot.html` for information on line styles. You can also add a title and label to the graph.

TYPE: `title('Plotting x and y')`
TYPE: `xlabel('x axis')`
TYPE: `ylabel('y axis')`

To add a new graph to the same plot without removing the original graph, use the `hold` command.

TYPE: `hold on`
TYPE: `y = -y`
TYPE: `plot(x, y, 'r')`
TYPE: `x = -x; plot(x, y, 'g')`
TYPE: `axis([-5 5 -5 5])`

To stop adding graphs,

TYPE: `hold off`

Of course, graphing three line segments is not very interesting. Let us graph some more complicated functions.

TYPE: `t = linspace(0, 4, 100);`
TYPE: `y = exp(t); plot(t, y, 'r'), hold on`
TYPE: `y = -2*exp(t); plot(t, y, 'b')`
TYPE: `y = 3* exp(t); plot(t, y, 'm'), hold off`

The comma allows you to put multiple commands on the same line; unlike the semicolon If MATLAB printed a very long vector, then you forgot a semicolon somewhere.

3.2 3-D Mesh Surface Plots

If you have a function $Z(x, y)$, you can plot its surface in 3-D using the commands `mesh`, `meshgrid`, and `surf`. For more information, use the `help` command or see the documentation online. Here are some examples.

TYPE: `A = eye(9); mesh(A)`

It's the identity matrix in 3-D!

TYPE: `[t,s] = meshgrid(-1:.2:4, -3:.2:3);`
TYPE: `z =sin(s.*t); mesh(z)`

4 M-Files

MATLAB can execute a sequence of statements stored in an ordinary text file with a `.m` extension. Because of this extension, such files are called "m-files".

One type of m-file is the function m-file. The first line of a function m-file must define the m-file as a function, specify its name, and specify its input and output variable names. A function m-file's function name and file name must be identical. For example, a function named "changerows" must be stored in a file named "changerows.m". When a function has more than one output variable, the output variables are enclosed by brackets: for example, `[m, n] = size(A)`. If your m-file is not in the working directory (this will default to your home directory), you must specify the path to it.

Using a text editor (e.g., `pico`), create the following file and save it as `rowchange.m`

```
function rowchange(A, c, d)
display(A)
r = A(c,:);
A(c,:)  = A(d,:);
A(d,:)  = r;
display(A)
```

(The `display` function prints a variable to the screen.)

After saving the file, reopen MATLAB if it was not already open (do not forget to use the `diary` command).

TYPE: `rng shuffle`
TYPE: `A = round(10 * rand(5))`
TYPE: `rowchange(A, 1, 4)`

Try it for other rows.

5 Linear Transformations

Create the following m-file and save it as `lintr2.m`

```
function lintr2(obj, A)
whitebg('w')
plot(obj(1,:), obj(2,:), 'k')
hold on
y = A * obj;
display(y)
plot(y(1,:), y(2,:), 'r')
hold off
```

This m-file plots the points in the matrix `obj`, and then plots the points in the matrix `A*obj`. Now reopen MATLAB if it was not already open (do not forget to use the `diary` command). Enter the following matrix; it describes the vertices in a drawing of a house.

TYPE: `house = [0 0; 0 1; 1 2; 2 1; 2 0; 3 0; ...`
TYPE: `1.75 0; 1.75 .5; 1.25 .5; 1.25 0; -1 0]'`

Do not forget the transpose!
To see the house,
TYPE: `lintr2(house, eye(2))`

Create a matrix that reflects across the x-axis.

TYPE: `reflectX = [1 0; 0 -1]`
TYPE: `lintr2(house, reflectX)`

The house should be in black and its reflection about the x-axis in red. Do the same thing for the y-axis:

TYPE: `reflectY = [-1 0; 0 1]`
TYPE: `lintr2(house, reflectY)`

Explain what happened.
COMMENT: `% [an explanation of what happened]`

To rotate the house 30 degrees counterclockwise,
TYPE: `rot30 = [cos(pi/6) -sin(pi/6); sin(pi/6) cos(pi/6)]`
TYPE: `lintr2(house, rot30)`

This should rotate your house 30 degrees. Can you rotate your house 45 degrees?

Now we will consider shear.

TYPE: `shearX = [1 2; 0 1]`
TYPE: `lintr2(house, shearX)`

This shears the house horizontally. For vertical shear,

TYPE: `shearY = [1 0; 0 3]`
TYPE: `lintr2(house, shearY)`

You can combine these transformations by multiplying them. For example,

TYPE: `lintr2(house, reflectY * shearX)`

In which order are the transformations being applied?

If the reflection is applied first,
TYPE: `s5a1 = 'reflectY'`

If the shear is applied first,
TYPE: `s5a1 = 'shearX'`

Now try the other order.
TYPE: `lintr2(house, shearX * reflectY)`

In which order are the transformations being applied here?

If the reflection is applied first,
TYPE: `s5a2 = 'reflectY'`

If the shear is applied first,
TYPE: `s5a2 = 'shearX'`

TYPE: `vtm=datestr(now)`

6 Matrix Multiplication

Matrix multiplication can be expressed and observed in several ways; we will see a few of them.

TYPE: `A = [-2 1 0; 2 0 1; 8 5 3; 1 2 -3]`
TYPE: `B = [1 5 -2; 0 -7 5; 1 3 -2 ]`

Enter the following commands. Some will give errors; this is expected.

TYPE: `A * B`
TYPE: `B * A`
TYPE: `inv(A)`
TYPE: `inv(B)`
TYPE: `inv(A*A')`
TYPE: `inv(A'*A)`
TYPE: `size(A'*A)`
TYPE: `size(A*A')`

There should have been three errors.

COMMENT: `% [an explanation of why the first error occurred]`
COMMENT: `% [an explanation of why the second error occurred]`
COMMENT: `% [an explanation of why the third error occurred]`

Let A and B be $\ell \times m$ and $m \times n$ matrices, respectively. Then the matrix product $C = A \cdot B$ exists. There are multiple ways to view matrix multiplication:

1. The element in the ith row and jth column of C, c_{ij}, is equal to the dot product of the ith row of A with the jth row of B. That is,

$$c_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj}.$$

2. The kth column of C is A times the kth column of B.

3. The kth row of C is the kth row of A times B.

TYPE: `C = A * B`

Find the entry in position $(2, 3)$ of the matrix C in two ways.

TYPE: `C(2,3)`
TYPE: `A(2,:)  * B(:,3)`

Find the second column of C in two ways.

TYPE: `C(:,2)`
TYPE: `A * B(:,2)`

Find the second row of C in two ways.
TYPE: `C(2,:)`
TYPE: `A(2,:)  * B`

Find C in two different ways.

TYPE: `C = A * B`
TYPE: `A(:,1) * B(1,:)  + A(:,2) * B(2,:)  + A(:,3) * B(3,:)`

Do you think you can justify why
`AB == A(:,1) * B(1,:)  + A(:,2) * B(2,:)  + A(:,3) * B(3,:)`?
(You do not have to answer.)
TYPE: `vt2=datestr(now)`
TYPE: `vt3=vt2-vt1`
TYPE: `vt4=vt2-vtm`