

ME3001 - Spring 2024

Weekly Activity 6: Using Two-Dimensional Arrays

Graphics - The Patch Object + Transformations

Overview :

You are going to practicing using two dimensional arrays in MATLAB. These are often referred to as Matrices and this is where MATLAB gets its name! There are many tools built in for working with matrices. If you are curious check the help. Today we are going to practice the basics and learn a new way to make graphics in MATLAB.

Basic Form of a Matrix:

The **size** of a matrix is the **number of rows** and the **number of columns** in the matrix respectively. The matrix A shown below has m rows and n columns.

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ & \cdot & & \\ & a_{ij} & & \\ & \cdot & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

Initializing a Two Dimensional Matrix:

- Choose a valid variable name and use the *square brackets* to create a matrix with hardcoded data.
- Commas (,) separate columns of the matrix. Spaces also work.
- Semicolons (;) separate rows of the matrix. A newline will also work.

Accessing and Assigning:

- Basically the same as with 1D
- `x = A(3,5)` % access element in row 3 column 5
- `y = 25)`
- `A(6,2) = y` % assign element in row 6 column 2

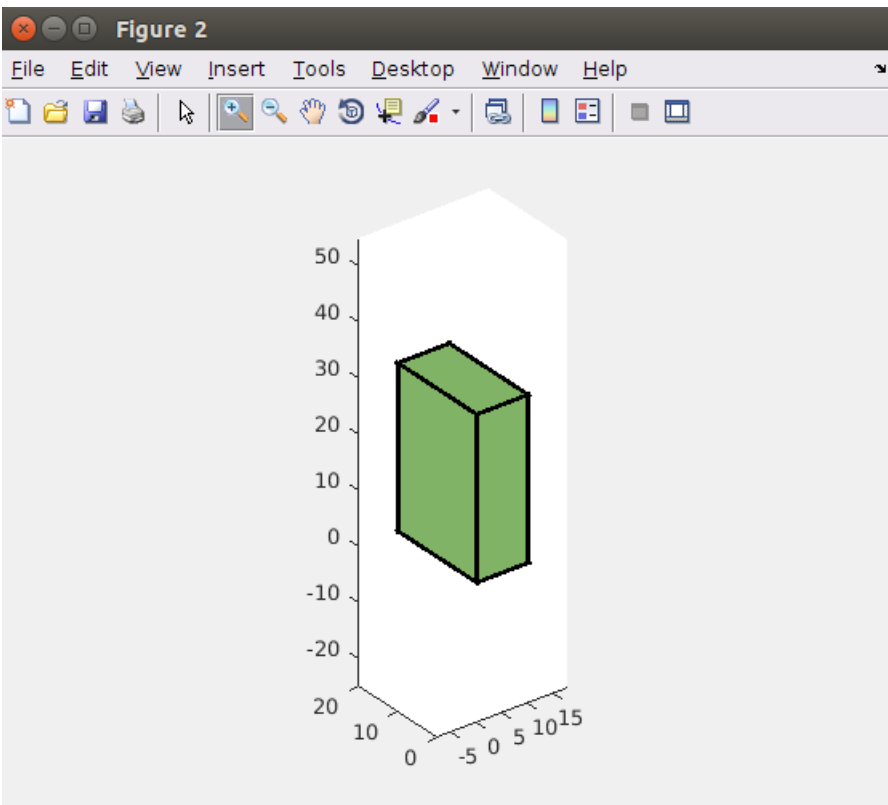
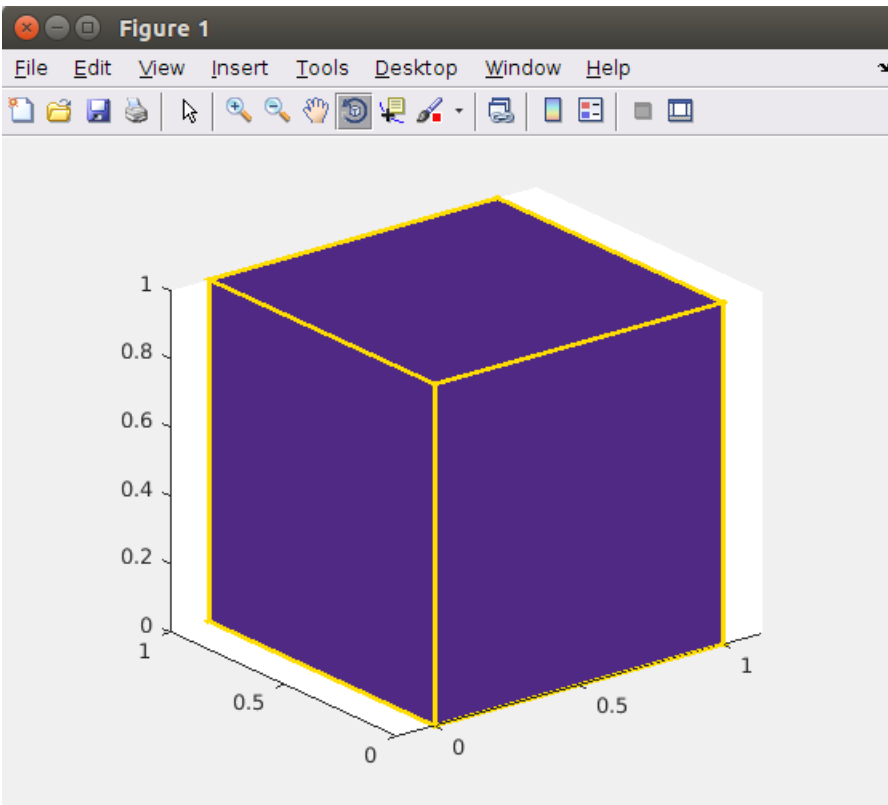
Example :

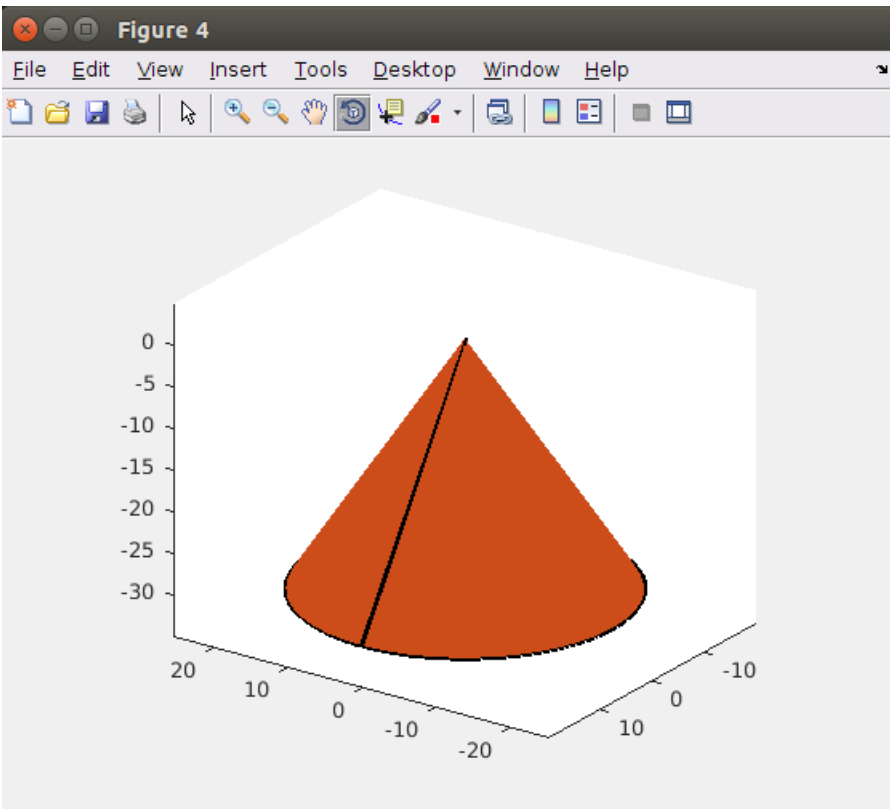
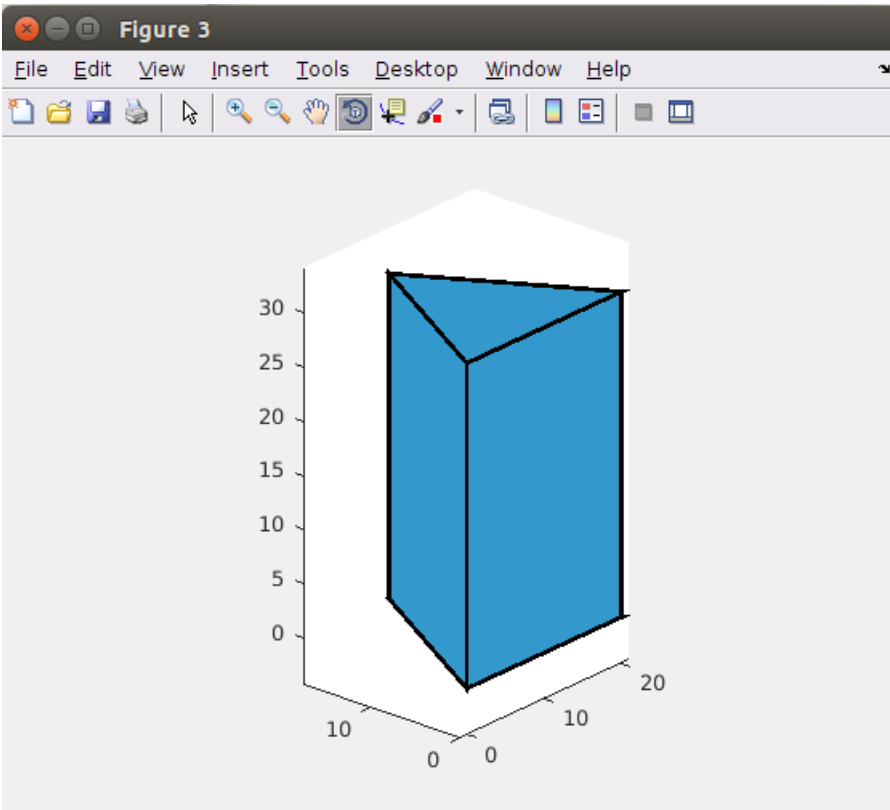
The program below generates figure1 shown on the next page using a patch object.

```
clear variables;clc;close all
s=1;
cube.vertices=[0, 0, 0
               s, 0, 0
               s, s, 0
               0, s, 0
               0, 0, s
               s, 0, s
               s, s, s
               0, s, s];

cube.faces=[1, 2, 3, 4
            5, 6, 7, 8
            1, 2, 6, 5
            2, 3, 7, 6
            3, 4, 8, 7
            4, 1, 5, 8];

figure(1)
ph=patch(cube);
ph.EdgeColor=[1.0000,0.8667,0];
ph.FaceColor=[0.3098,0.1608,0.5176];
ph.LineWidth=2;
axis equal
view(3)
```





Assignment :

1. Generate a 3D object using the patch command. Use one of the examples shown or create your own.
2. Adjust the lighting and other properties by setting the patch parameters.
3. Choose a direction and distance and translate the object using a matrix operation.
4. Choose an axis and angle and rotate the points about that axis using a rotation matrix.
5. ...

Deliverables :

- Submit working MATLAB code as a .m file which demonstrates all numbered assignment parts.
- Include at least two images of the object you have created from different viewpoint. They can be in a .pdf, .docx, or attached directly to the assignment in standard image format (.png, .jpg),