

MAT 343 Laboratory 4

Plotting and computer animation in MATLAB

In this laboratory session we will learn how to

1. Plot in MATLAB
2. The geometric properties of special types of matrices (rotations, dilations, reflections).
3. Perform simple computer animation in MATLAB

Plotting in MATLAB

If \mathbf{x} and \mathbf{y} are vectors of the same length, the command `plot(x,y)` will produce a plot of all the (x_i, y_i) pairs, and each point will be connected to the next by a line segment. If the x -coordinates are taken close enough together, the graph should resemble a smooth curve. The command `plot(x,y,'o')` will plot the ordered pairs with `o`'s, but will not connect the points.

For example, to plot the function $f(x) = \frac{e^{x/10} \sin x}{x+1}$ on the interval $[0, 10]$, set

```
x = 0:0.2:10;  
y = exp(x/10).*sin(x)./(x+1);
```

(note the `.` before the `*` and `/`, as these operations must be done component-wise).

The command `plot(x,y)` will generate the graph of the function. To compare the graph with that of $\sin x$, we could set `z = sin(x)`; and use the command

```
plot(x,y,x,z)
```

to plot both curves at the same time.

It is also possible to do more sophisticated types of plots in MATLAB, including polar coordinates, three-dimensional surfaces, and contour plots.

Given the points (x_i, y_i) , rather than creating the vectors \mathbf{x} and \mathbf{y} , an equivalent approach, which often is more convenient, is to create a $2 \times n$ matrix of data:

$$A = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ y_1 & y_2 & \cdots & y_n \end{bmatrix}$$

then typing

```
plot(A(1,:),A(2,:))
```

In other words, the first row of A gives the x 's and the second row the y 's.

EXAMPLE 1

Consider the triangle whose vertices are $(-0.5, -1)$, $(0, 1)$, $(0.5, -1)$.

- (a) Enter the x and y coordinates of these points as MATLAB vectors by setting

```
x = [-0.5, 0, 0.5, -0.5]  
y = [-1, 1, -1, -1]
```

The coordinates of the first points are repeated in the fourth column of \mathbf{x} and \mathbf{y} because we want to connect the third point back to the first. The command `plot(x,y)` draws this triangle in the graph window. The command `axis([-2,2,-2,2])` will rescale the axes so that the triangle does not take up the entire figure.

- (b) Now let's store the entire figure in a matrix T by storing the x -coordinate in the first row and the y -coordinate in the second row:

```
T = [-0.5, 0, 0.5, -0.5; -1, 1, -1, -1]
```

(if x and y have already been entered we can create T simply by typing $T = [x; y]$).
The triangle can then be plotted by typing

```
plot(T(1,:),T(2,:))
```

Again, set `axis([-2,2, -2, 2])`.

Rotation Matrices in \mathbb{R}^2

Any 2-dimensional matrix of the form

$$Q = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

is called a rotation matrix. If \mathbf{v} is a vector in \mathbb{R}^2 , the product $Q\mathbf{v}$ is the vector \mathbf{v} rotated θ radians in the counterclockwise direction.

To rotate the vector in the clockwise direction we simply replace θ with $-\theta$, thus

$$Q^{-1} = \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix}$$

Since $\cos(-\theta) = \cos \theta$ and $\sin(-\theta) = -\sin(\theta)$, we have

$$Q^{-1} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} = Q^T$$

EXAMPLE 2

We can rotate the triangle T from Example 1 by an angle of $\pi/4$ in the counterclockwise direction by multiplying the matrix T by

$$Q = \begin{bmatrix} \cos(\pi/4) & -\sin(\pi/4) \\ \sin(\pi/4) & \cos(\pi/4) \end{bmatrix}$$

```
clf % clear all settings for the plot
T = [-0.5, 0, 0.5, -0.5; -1, 1, -1, -1];
plot(T(1, :), T(2, :), 'linewidth', 2) % plot the triangle
hold on
Q = [cos(pi/4), -sin(pi/4); sin(pi/4), cos(pi/4)];
QT = Q*T; % rotate the triangle
plot(QT(1, :), QT(2, :), '-r', 'linewidth', 2) % plot the rotated triangle
title('Example of Rotation') % add a title
legend('original triangle', 'rotated triangle') % add a legend
grid on % add a grid
axis equal % set the axis equal
hold off
```

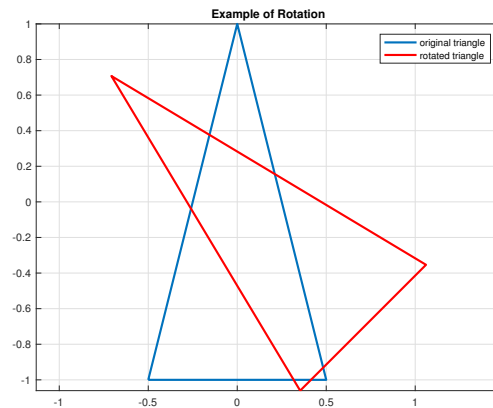


Figure L4a: Original triangle and rotated triangle

Remarks:

- The `hold on` command concatenates all subsequent plot commands on the same graph. Type `hold off` to start a new plot.
- The `'r-'` plots the figure in red.
- `grid on` adds a grid to the picture
- `axis equal` sets the same scale on the x and y axis.
- `'linewidth', 2` sets the width of the line to 2 points (the default is 0.5).

Dilation and Contraction

Consider the matrix

$$D = \begin{bmatrix} \alpha & 0 \\ 0 & \alpha \end{bmatrix}.$$

If we multiply this matrix by any vector \mathbf{v} we obtain the vector $\alpha\mathbf{v}$ which is a dilation or contraction of the original vector, depending on whether α is greater or smaller than one.

EXAMPLE 3

Consider the case where the dilation factor α is 1.25. We can plot the original triangle T together with the dilated version by executing the following commands:

```
clf
T = [-0.5, 0, 0.5, -0.5; -1, 1, -1, -1];
plot(T(1,:),T(2,:), 'linewidth',2)
hold on
D = [1.25, 0; 0, 1.25];
DT = D*T;
plot(DT(1,:),DT(2,:), '-r', 'linewidth',2)
title('Example of Dilation')
legend('original triangle', 'dilated triangle')
grid on
axis equal
hold off
```

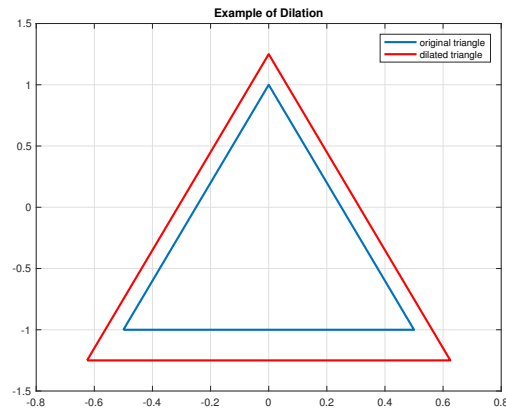


Figure L4b: Original triangle and rotated triangle

Reflection Matrices

The reflection matrix about the line L with direction the unit vector \mathbf{u} is given by $R = 2\mathbf{u}\mathbf{u}^T - I$, where I is the identity matrix.

For instance, to obtain the reflection matrix about the line at 45 degrees we can take $\mathbf{u} = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right)^T$.

Then $\mathbf{u}\mathbf{u}^T = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}$ and $R = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. (Note that this answer makes perfect sense, since the reflection of the vector $\mathbf{x} = (x_1, x_2)^T$ about the line at 45 degrees is the vector $(x_2, x_1)^T$).

Combinations of Transformations

One can perform a combination of two (or more) transformations by multiplying by the corresponding matrices in the appropriate order.

EXAMPLE 4

We can see the effect of first reflecting the triangle about the line at 45 degrees and then rotating it $\pi/4$ radians counterclockwise, by plotting the product $\mathbf{Q}\mathbf{R}\mathbf{T}$. Note the order of the matrices; here we first apply the reflection (R is the matrix that multiplies T first) and then we apply the rotation.

```
clf
T = [-0.5, 0, 0.5, -0.5; -1, 1, -1, -1];
plot(T(1,:),T(2,:), 'linewidth',2)
hold on
Q = [cos(pi/4), -sin(pi/4); sin(pi/4), cos(pi/4)];
R = [0, 1; 1, 0];
QRT = Q*R*T;
plot(QRT(1,:),QRT(2,:), '-r', 'linewidth',2)
title('Example of reflection and rotation')
legend('original triangle', 'modified triangle')
grid on
axis equal
hold off
```

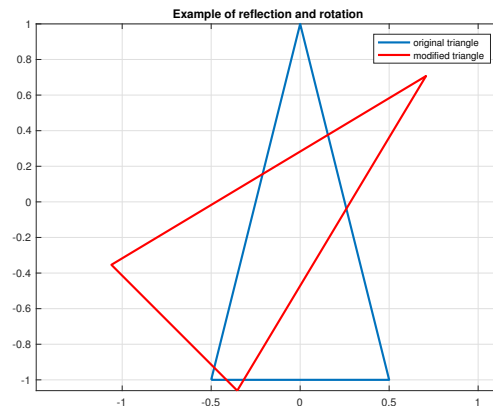


Figure L4c: Original triangle and triangle first reflected and then rotated

Animation

EXAMPLE 5

The goal of this example is to draw the triangle in its original size, then cause it to disappear, and finally to redraw it as it looks after it is dilated by a factor of 1.25. If this operation is repeated ten times in succession, the triangle will appear to be expanding or moving forward.

To do this we need to first plot the triangle and store its handle in `p`:

```
p = plot(T(1,:),T(2,:))
```

After we transform the matrix by setting $T = D \cdot T$ we can erase the original figure and draw the new figure with the command

```
set(p,'xdata',T(1,:), 'ydata',T(2,:));
```

The following are the commands that produce the effects. Enter them in MATLAB (you need not enter the comments that follow the % signs). The `pause(0.1)` command is used to adjust the speed of the plots so that we can see the graphs before they are erase. You might need to change the 0.1 to suit the speed of your machine.

```
clf                                % clear all settings for the plot
T = [-0.5,0,0.5,-0.5;-1,1,-1,-1];
D = 1.25*eye(2);
p = plot(T(1,:),T(2,:));          % plot the triangle
axis([-10,10,-10,10])             % set size of the graph
axis square                        % make the display square
figure(gcf)                        % display graphics window
% Adjust the windows on your screen so that both the command window
% and the graphics window show
hold on                            % hold the current graph
for i = 1:10
    T = D*T;                       % transform the figure
    set(p,'xdata',T(1,:), 'ydata',T(2,:)); % erase original figure and plot
                                        % the transformed figure
    pause(0.1)                     % adjust this pause rate to suit your computer.
end
hold off
```

EXERCISES

Instructions: For each of the following exercise, create an M-file to store the MATLAB commands. Copy and paste the M-file into a text document. For problems 1 and 2, include in the text document the pictures produced by MATLAB. Resize and crop the pictures so that they do not take up too much space. If the question requires written answers, include them in the text file in the appropriate location. Make sure you clearly label and separate all the Exercises. For problems 3 and 4 you do not need to include the picture.

1. Consider the original triangle T . Use the reflection matrix R introduced on page 4 to reflect the triangle about the line at 45° and plot the reflection together with the original triangle. Add a grid, a legend and title. Make sure you check that the names in the legend correspond to the correct triangles. Include the M-file as well as the figure.

2. Consider the original triangle T . First rotate the triangle 45° counterclockwise and then reflect the triangle about the line at 45° . Plot the resulting triangle and compare with the plot in Example 4. Are the results the same? Does the order of the transformations matter?

Include the M-file as well as the figure and don't forget to answer the questions.

Hint: You want to rotate first and then reflect. In Example 4 the triangle is first reflected and then rotated. To do this, $QRT = Q \cdot R \cdot T$, where Q is the rotation matrix and R is the reflection matrix. So, if you want to rotate and then reflect, in what order will you multiply the matrices?

3. Adapt the procedure developed in Example 5 to rotate the triangle counterclockwise by increments of $\pi/20$ about the origin. Stop when the triangle is in its original location and then rotate it in the clockwise direction until the triangle is in its original location again. If the pause command is set properly, it should appear that the triangle is moving around a circle.

You may want to rescale the axis to `axis([-2,2,-2,2])`

Include the M-file. Do not include the figure.

Hint: Since you are performing a computation several times, you will want to use two **for** loops: one loop for the counterclockwise rotation and another one for the clockwise rotation. Think about how many times you will need to go through the loops, keeping in mind that you are rotating the triangle counterclockwise for a full circle by increments of $\pi/20$ and then rotating the triangle clockwise back again.

4. Adapt the procedure developed in Example 5 to show the triangle rotating in a counterclockwise direction about the origin by increments of $\pi/20$ for a total angle of $\pi/2$ (a quarter of a circle) and expanding at the same time by a factor of 1.25, then stopping and rotating in the clockwise direction as it shrinks to its original size (with a contraction factor of .8). At the end of the program, the figure should have returned to its original size and original location.

You may want to rescale the axis to `axis([-10,10,-10,10])`

Include the M-file. Do not include the figure.

Hint: Similarly to Exercise 3, you might want to use two **for** loops: one loop for the counterclockwise rotation + dilation, and another one for the clockwise rotation + contraction. Think about how many times you will need to go through the loops.

Homogeneous coordinates

Translations are not linear transformations and consequently cannot be accomplished by matrix multiplication using a 2×2 matrix. To circumvent this problem we use *homogeneous coordinates*.

The homogeneous coordinates for a point $(x_1, x_2)^T$ are $(x_1, x_2, 1)^T$.

However, when the point represented by the homogeneous coordinates $(x_1, x_2, 1)^T$ is plotted, only the x_1 and x_2 are taken into consideration. The 1 is ignored. This allows us to produce the translated coordinates for a figure by matrix multiplication. Suppose we want to translate the vector c_1 units horizontally and c_2 units vertically. In homogeneous coordinates the translated vector is $(x_1 + c_1, x_2 + c_2, 1)^T$. We need a matrix M such that

$$M(x_1, x_2, 1)^T = (x_1 + c_1, x_2 + c_2, 1)^T$$

It is easy to verify that the matrix

$$M = \begin{bmatrix} 1 & 0 & c_1 \\ 0 & 1 & c_2 \\ 0 & 0 & 1 \end{bmatrix}$$

has the desired property.

The other types of basic transformations (reflections, rotations, dilations) can all be adapted to homogeneous coordinates by augmenting the matrix by the row $(0, 0, 1)$ and the column $(0, 0, 1)^T$. For instance, in homogeneous coordinates, the dilation matrix that dilates by a factor α becomes;

$$D = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

EXAMPLE 6

Consider the matrix T representing the triangle in the previous problems. In homogeneous coordinates the matrix becomes

$$T = \begin{bmatrix} -0.5 & 0 & 0.5 & -0.5 \\ -1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

In the following M-file we translate the triangle using $c_1 = .1$ and $c_2 = .1$ for 20 times. We then translate the triangle horizontally using $c_1 = -.1$ and $c_2 = 0$ and 40 iterations.

```
clf
T=[-0.5,0,0.5,-0.5;-1,1,-1,-1;1,1,1,1]; % define the triangle in homogeneous coordinates
c1 = .1; c2 = .1;
M1 = [1,0,c1;0,1,c2;0,0,1]; % define the first translation matrix
M2 = [1,0,-c1;0,1,0;0,0,1]; % define the second translation matrix
p = plot(T(1,:),T(2,:)); % plot the original triangle
axis([-7,7,-7,7])
axis square
figure(gcf)
for i = 1:20
    T = M1*T; % compute the translated triangle
    set(p,'xdata',T(1,:), 'ydata',T(2,:)); % plot the translated triangle
    pause(0.1)
end
for i = 1:40
    T=M2*T; % compute the translated triangle
    set(p,'xdata',T(1,:), 'ydata',T(2,:)); % plot the translated triangle
    pause(0.1)
end
```

EXERCISES

5. (a) Modify the M-file in EXAMPLE 6 adding translations that bring the triangle to its original position using 20 iterations.
Include the M-file. You do not need to include the figure.
- (b) Write down a rotation matrix Q that rotates a vector in *homogeneous coordinates* $\pi/40$ radians in the counterclockwise direction (keep in mind that we are working in homogeneous coordinates, thus the rotation matrix should be a 3×3 matrix). Then modify the M-file in part (a) adding to each iteration (for all three loops) a rotation defined by the matrix Q .
Note that the triangle should NOT end up in its original location like it should for 5(a). You

might need to change the axes to see where it “lands.” Try setting `axis([-7,9,-7,7])`
Include the M-file. Do not include the figure.