MA 302 – Spring 2019
Homework 4: Due on Wednesday, April 3

Instructions:
The code submitted should be your own creation. You may consult MATLAB's documentation or the notes from class. The submission of codes obtained from online sources is a violation of Loyola's honor code.

Exercise 1: Linear Transformations
Download the provided script `linear_trans.m`. Provide MATLAB commands to visualize the following transformations. In each case use the MATLAB command `saveas(gcf,'prob1_ex1.jpg')` to save the image visualizing problem 1 exercise 1, for example. Your figures should contain a title describing the transformation and axis labels. Your plot should include both the original and transformed shapes. Write a small script for each problem.

1. Rotation by an angle $\theta = \frac{\pi}{8}$
2. Reflection along the x-axis.
3. A random orthogonal transformation. Mark two reference points and show how the orthogonal transformation modifies them.
4. A translation along the vector `v =[3,0]` and a dilation by a factor of 2
5. Write a script that animates the following sequence of 2000 transformations.
   Implementation notes:

   (a) Use the commands

   ```
   v = VideoWriter('mymovie');
   open(v);
   ```

   to open the movie file `mymovie.avi`

   (b) In a loop that runs from $k = 1$ to 2000, for the shape provided in `linear_trans.m`, rotate by 1 degree in an anti-clockwise direction and scale the image by a factor of $c = 0.999$ and for values of $k <= 1000$ and for values of $k > 1000$, rotate by 1 degree in an anti-clockwise direction dilate the shape by $c = 1.001$. For each iteration, plot the transformed image using `connect_dots` with an axis size of 5 and save the movie frames for the animation using the commands

   ```
   frame = getframe(gcf);
   writeVideo(v,frame);
   ```

   (c) After exiting the loop call `close(v)` to finish writing the animation.

   (d) Your current directory should contain a file `'mymovie.avi'`, check to make sure it works.
6. Add your own non-trivial twist to the animation and save that in a file `'mymovie2.avi'`.
7. Plot each of the following affine transformations and give a brief description of their geometric effects. In each case pick 2 sample reference points to track the effects of the transformation.

   (a)

$$\boldsymbol{T}_1 = \boldsymbol{A}_1\vec{v} + \vec{b}_1 = \begin{bmatrix} 0.85 & 0.04 \\ -0.04 & 0.85 \end{bmatrix} \vec{v} + \begin{bmatrix} 0 \\ 1.6 \end{bmatrix}$$

(b)

$$T_2 = A_2\vec{v} + \vec{b}_2 = \begin{bmatrix} 0.20 & -0.26 \\ 0.23 & 0.22 \end{bmatrix}\vec{v} + \begin{bmatrix} 0 \\ 1.6 \end{bmatrix}$$

(c)

$$T_3 = A_3\vec{v} + \vec{b}_3 = \begin{bmatrix} -0.15 & 0.28 \\ 0.26 & 0.24 \end{bmatrix}\vec{v} + \begin{bmatrix} 0 \\ 0.44 \end{bmatrix}$$

(d)

$$T_4 = A_4\vec{v} = \begin{bmatrix} 0 & 0 \\ 0 & 0.16 \end{bmatrix}\vec{v}$$

Exercise 2: Fractals

1. Write a MATLAB function `fern_generator.m` that implements the following affine transformations N iterations to generate a figure of a fern fractal. Your function should return the amount of time it takes for the fractal to be plotted. On each iteration, each transformation is picked at random following the following probabilities:

   (a) 85% of the time:

   $$T_1 = A_1\vec{v} + \vec{b}_1 = \begin{bmatrix} 0.85 & 0.04 \\ -0.04 & 0.85 \end{bmatrix}\vec{v} + \begin{bmatrix} 0 \\ 1.6 \end{bmatrix}$$

   (b) 7% of the time:

   $$T_2 = A_2\vec{v} + \vec{b}_2 = \begin{bmatrix} 0.20 & -0.26 \\ 0.23 & 0.22 \end{bmatrix}\vec{v} + \begin{bmatrix} 0 \\ 1.6 \end{bmatrix}$$

   (c) 7% of the time:

   $$T_3 = A_3\vec{v} + \vec{b}_3 = \begin{bmatrix} -0.15 & 0.28 \\ 0.26 & 0.24 \end{bmatrix}\vec{v} + \begin{bmatrix} 0 \\ 0.44 \end{bmatrix}$$

   (d) 1% of the time:

   $$T_4 = A_4\vec{v} = \begin{bmatrix} 0 & 0 \\ 0 & 0.16 \end{bmatrix}\vec{v}$$

   Implementation notes

   - For each iteration, use the MATLAB function `randsample` (see class notes for usage) to pick a random integer corresponding to T_i, $\quad i = 1 \cdots 4$ with the given probabilities.
   - Construct a `switch` statement to execute the transformation corresponding to the output from `randsample`. Plot each point after the transformation.
   - Use the MATLAB functions `tic` and `toc` to time your iterations.
   - Use plot with `'MarkerSize'` of `2.5` to plot each point.
   - You do not need to have separate colors for each transformations.

   Testing
   Run your code for $N = 5000, 25000$ and 50000. Provide the CPU times and plots of the fern for each value of N.

2. Write a new function `fern_generator_optimized.m` that speeds up the computations above in the following way:

   (a) Preallocate memory for a matrix to store all the points computed from the iterations using the MATLAB function `zeros`

   (b) Plot all the points at once after completing the iteration.

   Run your code for $N = 5000, 25000, 50000$ and provide the CPU times in your summary file.

Submission of exercises

Place all your files (`m-files, image, movie, summary.txt, diary.txt`) in a folder named `lastname_hwN` and zip the folder to create a file `lastname_hwN.zip`. Email your zip file `lastname_hwN.zip` to `pchidyagwai@loyola.` with subject `MA302_hwN`.