AMATH 301
Homework 1
Due: Friday, January 11, 2019

Instructions for submitting

1. Scorelator problems are submitted as a `.m` MATLAB script file. Scorelator will run your file and check the output against a solution key. Do not submit `.dat` files; your script should recreate these files when it is run by Scorelator.
2. Writeup problems are submitted to Canvas as a PDF file that contains any plots that the problems ask you to create. Label each problem so that the graders will know which problem each plot is associated with. Also upload any `.m` MATLAB script files that you used to solve the problem to Canvas.

Problem 1 (Scorelator).

Learning goals: familiarity with constructing and manipulating vectors and matrices in MATLAB.

Define the following matrices and vectors in MATLAB:

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \qquad B = \begin{pmatrix} -2 & 1 \\ 1 & -2 \end{pmatrix} \qquad C = \begin{pmatrix} 2 & -7 \\ -3 & 4 \\ 5 & -1 \end{pmatrix}$$

$$x = \begin{pmatrix} 13 \\ -4 \end{pmatrix} \qquad y = \begin{pmatrix} 5 \\ 3 \end{pmatrix} \qquad z = \begin{pmatrix} 4 \\ 7 \\ -3 \end{pmatrix}$$

The following items demonstrate operations that are common in linear algebra. Compute them and save each in a corresponding file **A1.dat, A2.dat, . . . , A10.dat**.

$A_1 = 7x$ (Scalar multiplication of a vector)

$A_2 = x + y$ (Sum of vectors)

$A_3 = 3x - 2y$ (Linear combination of vectors)

$A_4 = 0.01B$ (Scalar multiplication of a matrix)

$A_5 = A - B$ (Difference of Matrices)

$A_6 = Ax$ (Matrix-vector multiplication)

$A_7 = BA$ (Matrix-matrix multiplication)

$A_8 = C^T$ (Matrix Transpose)

$A_9 = z^T$ (Vector Transpose)

$A_{10} = CA$ (Another matrix-matrix multiplication)

Problem 2 (Scorelator).

Use the **colon operator** (e.g. `2:6`) to define the following vectors:

$$u = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{pmatrix}$$
$$v = \begin{pmatrix} 0 & 0.5 & 1.0 & 1.5 & 2.0 & 2.5 & 3 \end{pmatrix}.$$

Also define

$$D = \texttt{magic(6);}$$

Note: `magic` is a MATLAB command that produces a square matrix of the specified size that is useful for testing purposes.

Now compute:

$$\begin{aligned}
B_1 &= \text{component-wise product of } u \text{ and } v \text{ (e.g. } \texttt{.*}\text{)} \\
B_2 &= \text{component-wise division of } v \text{ divided by } u \text{ (e.g. } \texttt{./}\text{)} \\
B_3 &= \text{cube every element of } u \text{ (e.g. } \texttt{.\^{}}\text{)} \\
B_4 &= \text{the number 2 raised to the power of each element of } u \text{ (e.g. } \texttt{.\^{}}\text{)} \\
B_5 &= \text{cosine of each element of } u
\end{aligned}$$

Note: B_1, B_2, B_3 are all vectors that are the same size as u.

Use *matrix slicing and indexing* to extract the following parts of vectors and matrices:

$$\begin{aligned}
B_6 &= \text{every-other element of } v\text{, starting with the second element} \\
B_7 &= \text{the second column of } D \\
B_8 &= \text{the third row of } D \\
B_9 &= \text{the middle } 2 \times 2 \text{ submatrix of } D \\
B_{10} &= \text{the middle } 4 \times 4 \text{ submatrix of } D
\end{aligned}$$

Save each result to the corresponding file **B1.dat, B2.dat, . . . , B10.dat**.

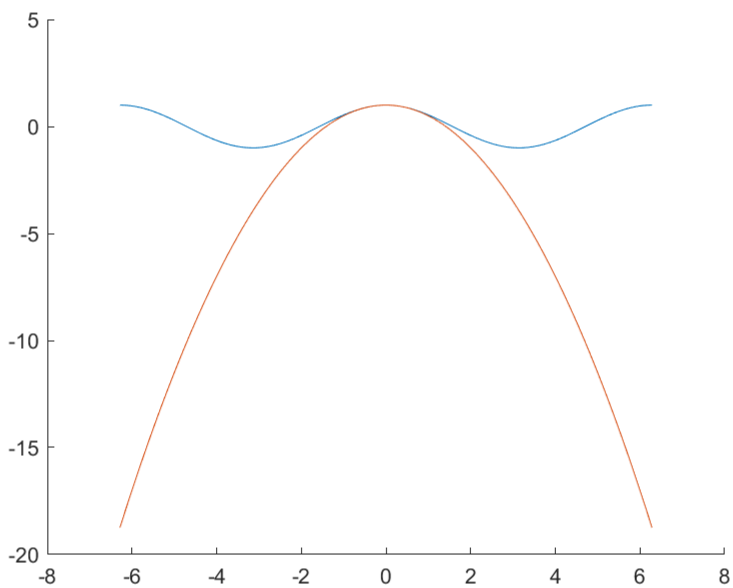
Problem 3 (Writeup).

Learning goals: practice searching online, develop familiarity with plotting and saving plots

Create a new MATLAB script and add the following:

```
xs = linspace(-2*pi,2*pi);
f = @(x) 1 - x.^2 / 2;
clf
hold on
plot(xs,cos(xs));
plot(xs,f(xs));
% commands to customize appearance go here
print(gcf,'-dpng','plot_basic.png');
```

This script generates a plot of cosine and its second-order Taylor polynomial approximation, and saves the figure as `plot_basic.png` (PNG is a good format for computer graphics).



Extend the script by adding commands to do the following:

(a) Make the plot of cosine appear in the **color black**. Set the **line width** to 3.

(b) Make the plot of the Taylor polynomial appear as a **dashed red line**. Set the line width to 2.

(c) **Label** the **x-axis** with the text "Space, x".

(d) Add a **title** with the text "Taylor Approximation of Cosine"

(e) Add a **legend** that labels Cosine and the Taylor polynomial. Set the **location** of the legend appropriately.

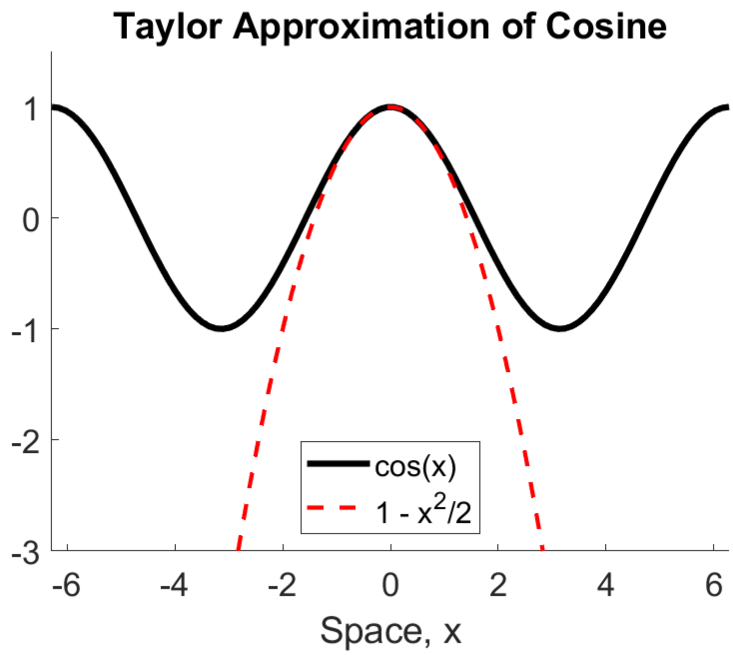
(f) Set the **axis limits** of the x-axis to $[-2\pi, 2\pi]$ and the y-axis to $[-3, 1.5]$.

(g) Set the **tickmarks** on the x-axis to be $-6, -4, -2, \ldots, 6$.

(h) Set the tickmarks of the y-axis to be $-3, -2, -1, 0, 1$.

To make these changes, you will have to search online for the appropriate MATLAB commands. The bolded keywords are hints for what to search for. Do not manually edit the figure to implement these changes: the goal is to understand how commands can be used to manipulate the figure.

After making these adjustments, the plot should look similar to the following.



NOTE: Your plot may look slightly different depending on the *size of the figure window*: a larger window will make text look relatively smaller, and a smaller figure window will make the text look relatively larger. This will not be counted against you when you are graded.