MSCS 6040 – TAKE HOME EXAM

assigned: Thursday, May 2
due: Friday, May 10, **by 3pm local time**

Instructions:

1. You must explain all reasoning. It is not sufficient to just write the correct answer. Correct answers with no justification will receive no credit.
2. You must submit your completed exam on D2L by **3:00pm on Friday, May 10**.
3. **In order to speed up grading, you must submit files in the following way.**

   Submit, for each question, precisely 1 ".mlx" file (not a ".m" file) and precisely 1 ".pdf" file which is the exported version of the submitted ".mlx" fie. Name these files "[lastname]-[question number].[file extension]". If you must submit an additional file for a question, name it "[lastname]-[question number]-[descriptive information about what the file is].[whatever]".

   So, for example, you submitted files might look like:

   Pantone-1.mlx
   Pantone-1.pdf
   Pantone-2.mlx
   Pantone-2.pdf
   . . .

   Note that you can write text and explanation directly inside of the ".mlx" file.
4. For any questions that ask you to write code and produce output, you **must** submit a ".mlx" file so I can run your code and verify the output.
5. **RULES FOR OUTSIDE SOURCES:**
   It is permissible to use:

   (a) Either of our textbooks, or any other published books or articles.
   (b) *Your* course notes.
   (c) The internet for base-level questions: Matlab synatax, basic definitions of terms, etc.
   (d) Me, via email or office hours.

   It is not permissible to use:

   (a) Any classmates, other students, other professors, etc.
   (b) Anybody else's course notes (it is okay if you had previously borrowed someone else's notes because you missed a day, etc).
   (c) The internet for anything remotely approaching the actual question asked.

   For example, it is okay to search the internet for "How to compute the SVD in Matlab", but it is not okay to search the internet for "Examples of numerical instability in Matlab".

   If you are in doubt about what is okay, email me!

   Any instances of plagiarism, working with classmates, or other cheating, will result in a score of 0 for the entire final exam.

The Marquette University honor code obliges students:

- To fully observe the rules governing exams and assignments regarding resource material, electronic aids, copying, collaborating with others, or engaging in any other behavior that subverts the purpose of the exam or assignment and the directions of the instructor.
- To turn in work done specifically for the paper or assignment, and not to borrow work either from other students, or from assignments for other courses.
- To complete individual assignments individually, and neither to accept nor give unauthorized help.
- To report any observed breaches of this honor code and academic honesty.

This exam has six questions, each worth 25 points. Only the top 4 scores will be recorded. You may choose to do only 4 problems, or you may choose to do more than 4.

1. The goals of this question are to introduce you to an important application of linear algebra that we did not have time to cover in this course, as well as to test your ability to use the theoretical concepts we learned to understand and apply relevant techniques to your own research.

   Principal Component Analysis is a linear algebra technique used in many applications, including machine learning, community detection, and regression. It is, at the most basic level, a way to take high-dimensional data and reduce its dimensionality in a manner that captures the most possible information.

   For example, you may conduct a medical study that gathers 100 data points from each of 50 patients. Suppose 25 of the patients come from one neighborhood, while the other 25 come from another, and you want to know whether there is any appreciable difference in the 100 parameters between the two neighborhoods. In this case, PCA could be used to reduce the dimensionality to 2, and a simple x/y-plot would visually tell you whether these two communities have significantly different measurements.

   In this exercise, you will read a brief page about how PCA works, and then use Matlab's built in routines to perform the analysis.

   First, read the following two webpages. The first is a nice interactive overview without any detail, and the second explains how it works. Both should be easy reads given the material we've learned in class.

   `http://setosa.io/ev/principal-component-analysis/`
   `https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c`

   Now, you will us PCA to analyze the "fisheriris" dataset that comes preloaded into Matlab. The British biologist Ronald Fisher took four measurements from 150 iris flowers, 50 each from three different species. If you type "load fisheriris", you will now have access to two variables: "meas", which is a 150×4 matrix of measurements, and "species", which is a 150×1 matrix saying which species each corresponding measurement.

   Fisher's question was this: "Do the three species of iris have significantly different measurements?".

   (a) Use Matlab's PCA code to reduce the dimensionality from 4 to 1, and plot the results on a 1-dimensional line. Plot each species in a different color, so you should have 50 points each of 3 different colors. By looking at the plot, answer Fisher's question.

   (b) Use Matlab's PCA code to reduce the dimensionality from 4 to 2, and plot the results on a 2-dimensional plane. Plot each species in a different color, so you should have 50 points each of 3 different colors. By looking at the plot, answer Fisher's question.

   (c) Use Matlab's PCA code to reduce the dimensionality from 4 to 3, and plot the results in 3-dimensional space. Plot each species in a different color, so you should have 50 points each of 3 different colors. By looking at the plot, answer Fisher's question.

   You may use the internet as needed to learn about PCA and how to perform it in Matlab. You may not use the internet to search for anything specific about the iris dataset or Fisher's results.

2. Implement Arnoldi Iteration (Algorithm 33.1) in Matlab. Use your algorithm to find an orthonormal

basis for the Krylov subspace K^4 for the matrix

$$A = \begin{bmatrix} -2i & -7 & 5 & 6 & 5 & 4 & 4 & -6 \\ 10 & 3 & 1 & 3 & -4 & 7 & 6 & 1 \\ -2 & 6 & -4 & 7 & 10 & -2 & 0 & 7 \\ -10 & 5 & -1 & 7 & 9 & 7 & -8 & -8 \\ -10 & 5 & -10 & -5 & 4 & -7 & 1 & -1 \\ -5 & -6 & -2 & -1 & 8 & 2 & 9 & -7 \\ -6 & 6 & -10 & -5 & 10 & 0 & 9 & -10 \\ 7 & 9 & 0 & -1 & 9 & -8 & -7 & -3 \end{bmatrix}$$

(note that the top-left entry is imaginary) matlab code:

```
[-2*i -7 5 6 5 4 4 -6; 10 3 1 3 -4 7 6 1; -2 6 -4 7 10 -2 0 7; -10 5 -1 7 9 7 -8 -8; -10 5
  -10 -5 4 -7 1 -1; -5 -6 -2 -1 8 2 9 -7; -6 6 -10 -5 10 0 9 -10; 7 9 0 -1 9 -8 -7 -3]
```

and the vector

$$b = \begin{bmatrix} 3 \\ -7 \\ -3 \\ 3 \\ -5 \\ -6 \\ -7 \\ 7 \end{bmatrix}$$

matlab code:

```
[3; -7; -3; 3; -5; -6; -7; 7].
```

3. Use the "eig" command in Matlab to find the eigenvalues of A (from the previous question) exactly. Then, implement Power Iteration (Algorithm 27.1), and use it to find the eigenvalue / eigenvector pair corresponding to largest eigenvalue. Print the result for each iteration. How many iterations are needed to increase the number of correct decimal places?

4. Implement Rayleigh Quotient Iteration (Algorithm 27.3). Using initial vector $v = (0, 0, 0, 1, 0, 0, 0, 0)^*$ and the matrix A from Question 2, to which eigenvalue / eigenvector pair does your algorithm converge? Print the result for each iteration. How fast does it converge? Compare your answer to the true eigenvalues (using "eig") and your answer to the precious question.

5. Use the following commands in Matlab to generate a 100×100 symmetric matrix with random entries between -10 and 10:

   ```
   rng(1000)
   M = randi(21,100,100) - 10;
   M = M - tril(M,-1) + triu(M,1)'
   ```

   The "rng" command sets the random see to a particular number, so everybody should end up with the same matrix (which makes it easier for me to verify your answers). Implement the "Phase 1 / Phase 2" process for computing eigenvalues, consisting of the direct Algorithm 26.1 and the iterative Algorithm 28.2, and use your implementations to compute all of the eigenvalues and eigenvectors of M.

6. The heart of Google's methodology of indexing the web is the PageRank algorithm. It relies on principles of linear algebra to classify and rank websites by order of importance so that a Google search can yield the most relevant results.

   First, read this short article about how the PageRank algorithm works:

http://home.ie.cuhk.edu.hk/~wkshum/papers/pagerank.pdf

Another useful resource may be

http://pi.math.cornell.edu/~mec/Winter2009/RalucaRemus/Lecture3/lecture3.html.

(a) After forming the weighted matrix from all of the internet links (as described in the links above), it is necessary to compute the largest eigenvalue and eigenvector for that eigenvalue. Explain informally (without rigorous proof, but using conceptual ideas) why finding this eigenvalue / eigenvector pair tells us the solution to the question "What is the limit of the sequence $b, Ab, A^2b, A^3b, \ldots$?" as well as why that's the right question to ask?

(b) Explain how you would calculate this eigenvalue / eigenvector pair, given that the matrix involved has dimensions well over $10^9 \times 10^9$. Which algorithms from class would you use? In what order? Does the matrix have any special properties that could be used to develop an algorithm faster than the most general one?