AMATH 301 – Autumn 2019
Homework 3

Due: 7:00pm, October 23, 2019.

Instructions for submitting:

- Scorelator problems are submitted as a MATLAB script (`.m` file). You should **NOT** upload your `.dat` files. The `.dat` files should be created by your script. You have 5 attempts on Scorelator.
- Writeup problems are submitted to Gradescope as a single `.pdf` file that contains text and plots. Put the problems in order and label each writeup problem. When you submit, **you must indicate which problem is which on Gradescope**. **All code you used for this part of the assignment should be included either at the end of the problem or at the end of your .pdf file.**

Scorelator problems

1. There are a number of applications where one might want to rotate an object in $\mathbb{R}^3$ (3-dimensional space) about a certain axis. Two such examples are (a) in computer graphics where objects are represented on the screen as objects in $\mathbb{R}^3$ (see e.g., Ray Tracing if you are interested) and (b) in aircraft control where rotations about the different axes are called *yaw, pitch,* and *roll* (see e.g., aircraft principle axes if you are interested). To rotate a vector $\mathbf{v} = (v_x, v_y, v_z)^T$ by an angle θ about a vector $\mathbf{u} = (u_x, u_y, u_z)^T$, you can multiply $\mathbf{v}$ on the left by the matrix

$$R = \begin{pmatrix} \cos\theta + u_x^2(1-\cos\theta) & u_x u_y(1-\cos\theta) - u_z\sin\theta & u_x u_z(1-\cos\theta) + u_y\sin\theta \\ u_y u_x(1-\cos\theta) + u_z\sin\theta & \cos\theta + u_y^2(1-\cos\theta) & u_y u_z(1-\cos\theta) - u_x\sin\theta \\ u_z u_x(1-\cos\theta) - u_y\sin\theta & u_z u_y(1-\cos\theta) + u_x\sin\theta & \cos\theta + u_z^2(1-\cos\theta) \end{pmatrix}.$$

   In other words, $R\mathbf{v}$ is the vector $\mathbf{v} = (v_x, v_y, v_z)^T$ rotated by an angle θ around the vector $\mathbf{u} = (u_x, u_y, u_z)^T$.

   (a) Define the matrix A to be the matrix that rotates a vector by an angle of $\theta = \pi/7$ around the vector $\mathbf{u} = (1,1,1)/\sqrt{3}$. Use the `lu` command to compute the LU decomposition of A, **using partial pivoting (this means you also have the permutation matrix** P. This will give three matrices: L, U, and P. Calculate the product ULP and save the resulting matrix in `A1.dat`.

(b) I have a vector $\mathbf{y} = (9, -1, 27)^T$ which was rotated by the matrix A defined above. Before it was rotated, the vector was called $\mathbf{x}$. I want to know what $\mathbf{x}$ was before rotating it. Setup the equation which is used to find $\mathbf{x}$ and then solve for $\mathbf{x}$ using LU decomposition. Save the resulting vector $\mathbf{x}$ in `A2.dat`.

2. Consider the truss bridge diagrammed below. The bridge is composed of steel beams that are connected at joints or *nodes*. Each beam exerts a force, F_j, on the nodes that it touches. When the bridge is at rest, these forces balance. A positive force means that the beam is in tension: it is being pulled apart by the nodes. A negative force means that the beam is being pushed inwards and is in compression.

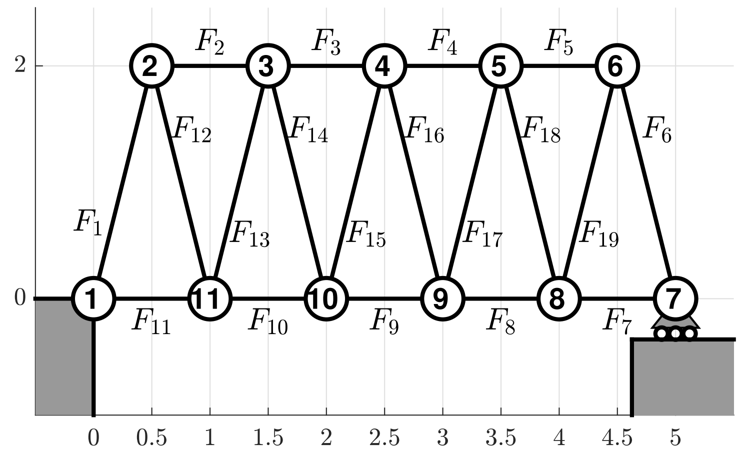

Figure 1: A truss bridge

Cars and trucks will pass over this bridge, and we would like to figure out what forces will act on the bridge as its load changes. Let W_8, W_9, W_{10}, and W_{11} be the weights, measured in Newtons, of vehicles on the bridge located at nodes 8, 9, 10, and 11 respectively.

We can find equations for the unknown forces with Newton's second and third laws of motion.

- x- and y-forces sum to zero on node (2).

$$\frac{-1}{\sqrt{17}}F_1 + F_2 + \frac{1}{\sqrt{17}}F_{12} = 0 \tag{1}$$

$$\frac{-4}{\sqrt{17}}F_1 - \frac{4}{\sqrt{17}}F_{12} = 0 \tag{2}$$

- x and y-forces sum to zero on node (3)

$$-F_2 + F_3 - \frac{1}{\sqrt{17}}F_{13} + \frac{1}{\sqrt{17}}F_{14} = 0 \tag{3}$$

$$\frac{-4}{\sqrt{17}}F_{13} - \frac{4}{\sqrt{17}}F_{14} = 0 \tag{4}$$

- x and y-forces sum to zero on node (4)

$$-F_3 + F_4 - \frac{1}{\sqrt{17}}F_{15} + \frac{1}{\sqrt{17}}F_{16} = 0 \tag{5}$$

$$\frac{-4}{\sqrt{17}}F_{15} - \frac{4}{\sqrt{17}}F_{16} = 0 \tag{6}$$

- x-forces sum to zero on node (5)

$$-F_4 + F_5 - \frac{1}{\sqrt{17}}F_{17} + \frac{1}{\sqrt{17}}F_{18} = 0 \tag{7}$$

$$\frac{-4}{\sqrt{17}}F_{17} - \frac{4}{\sqrt{17}}F_{18} = 0 \tag{8}$$

- x and y-forces sum to zero on node (6)

$$-F_5 + \frac{1}{\sqrt{17}}F_6 - \frac{1}{\sqrt{17}}F_{19} = 0 \tag{9}$$

$$-\frac{4}{\sqrt{17}}F_6 - \frac{4}{\sqrt{17}}F_{19} = 0 \tag{10}$$

- x-forces sum to zero on node (7)

$$-\frac{1}{\sqrt{17}}F_6 - F_7 = 0 \tag{11}$$

- x and y-forces sum to zero on node (8)

$$F_7 - F_8 - \frac{1}{\sqrt{17}}F_{18} + \frac{1}{\sqrt{17}}F_{19} = 0 \tag{12}$$

$$\frac{4}{\sqrt{17}}F_{18} + \frac{4}{\sqrt{17}}F_{19} - W_8 = 0 \tag{13}$$

- x and y-forces sum to zero on node (9)

$$F_8 - F_9 - \frac{1}{\sqrt{17}}F_{16} + \frac{1}{\sqrt{17}}F_{17} = 0 \tag{14}$$

$$\frac{4}{\sqrt{17}}F_{16} + \frac{4}{\sqrt{17}}F_{17} - W_9 = 0 \tag{15}$$

- x and y-forces sum to zero on node (10)

$$F_9 - F_{10} - \frac{1}{\sqrt{17}}F_{14} + \frac{1}{\sqrt{17}}F_{15} = 0 \tag{16}$$

$$\frac{4}{\sqrt{17}}F_{14} + \frac{4}{\sqrt{17}}F_{15} - W_{10} = 0 \tag{17}$$

- x and y-forces sum to zero on node (11)

$$F_{10} - F_{11} - \frac{1}{\sqrt{17}}F_{12} + \frac{1}{\sqrt{17}}F_{13} = 0 \tag{18}$$

$$\frac{4}{\sqrt{17}}F_{12} + \frac{4}{\sqrt{17}}F_{13} - W_{11} = 0 \tag{19}$$

Write this linear system as a matrix equation $\boldsymbol{Ax} = \mathbf{b}$ where $\boldsymbol{x}$ is a vector of the forces:

$$\boldsymbol{x} = \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_{19} \end{pmatrix}.$$

(a) Use the backslash command to solve for $\boldsymbol{x}$, the vector of forces on the bridge, when the weights of the vehicles are $W_8 = 15000$, $W_9 = 9890$, $W_{10} = 13000$, and $W_{11} = 22000$. Save $\boldsymbol{x}$ in `A3.dat`.

(b) Find the largest force (in absolute value) of the vector $\boldsymbol{x}$. Save its absolute value in `A4.dat`. You may find some of the commands `max`, `abs`, or `norm` useful.

(c) Now suppose that we add weight to the truck at position 10 in increments of 10 Newtons until the bridge collapses. Each bridge member can withstand 40600 Newtons of compression or tension (i.e., positive or negative forces.) Therefore, the bridge will collapse when the absolute value of the largest force is larger than 40600. Find the smallest weight of the truck at position 10 for which the bridge collapses. Save your answer as `A5.dat`. Find which force is the one that exceeded 40600 Newtons. If the force is F_j, save the value of j in `A6.dat`. **Note: Two forces exceed 40600 Newtons at the same time. Record the smaller of the two value of j in `A6.dat`.**

3. Create the following 19×19 matrix and 19×1 vector in MATLAB.

$$A = \begin{pmatrix} 1 & 2 & 3 & \cdots & 18 & 19 \\ 2 & 1 & 2 & \ddots & \cdots & 18 \\ 3 & 2 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 2 & 3 \\ 18 & \cdots & \ddots & 2 & 1 & 2 \\ 19 & 18 & \cdots & 3 & 2 & 1 \end{pmatrix}, \qquad \mathbf{v} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ \vdots \\ 19 \end{pmatrix}. \tag{20}$$

The matrix A can be created in MATLAB using the command `toeplitz(1:19)`. Create both the matrix A and $\mathbf{v}$.

(a) Calculate $\mathbf{b} = A\mathbf{v}$. Notice that $\mathbf{v}$ is the exact solution of $A\mathbf{x} = \mathbf{b}$.

(b) Solve $A\mathbf{x} = \mathbf{b}$ using the backslash command. Then calculate the error vector $\mathbf{e} = \mathbf{v} - \mathbf{x}$ and the residual vector $\mathbf{r} = A\mathbf{x} - \mathbf{b}$. Notice that if the backslash command gave the exact solution, both $\mathbf{e}$ and $\mathbf{r}$ would be vectors of all zeros. Calculate $||\mathbf{e}||$ and $||\mathbf{r}||$ using the `norm` command. Make a 1×2 row vector with $||\mathbf{e}||$ as the first component and $||\mathbf{r}||$ as the second component and save it in `A7.dat`.

(c) Solve $A\mathbf{x} = \mathbf{b}$ using LU decomposition. As in part (b), calculate the error and residual vectors, create a row vector with their norms, and save it in `A8.dat`.

(d) Solve $A\mathbf{x} = \mathbf{b}$ by using $\mathbf{x} = A^{-1}\mathbf{b}$ (A^{-1} is calculated using the `inv(A)` command). Again create a row vector with the norms of $\mathbf{e}$ and $\mathbf{r}$ and save the result to `A9.dat`.

4. Consider an algorithm that solves $A\mathbf{x} = \mathbf{b}$ in $\mathcal{O}(n^2 \log n)$ operations. Which of the following is true?

   A Solving $A\mathbf{x} = \mathbf{b}$ for a 100×100 matrix will take 200 times as many operations as solving $A\mathbf{x} = \mathbf{b}$ for a 10×10 matrix.

   B Solving $A\mathbf{x} = \mathbf{b}$ for a 10×10 matrix will take exactly 230.2585 operations.

   C Solving $A\mathbf{x} = \mathbf{b}$ for a 10×10 matrix will take exactly 230 operations.

   D Solving $A\mathbf{x} = \mathbf{b}$ for a 10×10 matrix will take approximately 230 operations.

   E Solving $A\mathbf{x} = \mathbf{b}$ for a 100×100 matrix will take 10 times as many operations as solving $A\mathbf{x} = \mathbf{b}$ for a 10×10 matrix.

   F Solving $A\mathbf{x} = \mathbf{b}$ for a 10×10 matrix will take exactly 100 operations.

   G Solving $A\mathbf{x} = \mathbf{b}$ for a 10×10 matrix will take approximately 100 operations.

   H Solving $A\mathbf{x} = \mathbf{b}$ for a 100×100 matrix will take 100 times as many operations as solving $A\mathbf{x} = \mathbf{b}$ for a 10×10 matrix.

   I Solving $A\mathbf{x} = \mathbf{b}$ for a 100×100 matrix will take 230 times as many operations as solving $A\mathbf{x} = \mathbf{b}$ for a 10×10 matrix.

   Save your answer to the file `A10.dat`.

Gradescope problems

1. The `tic` and `toc` commands can be used to time how long it takes MATLAB to do something. For instance, one can do

```
tic
3 + 4
toc
```

to see how long it takes MATLAB to add 3 and 4. **Make sure to use `tic` and `toc` in `.m` files otherwise it will count the time it takes you to type in the commands in between!** In this problem we want to compare how long it takes to solve $A\mathbf{x} = \mathbf{b}$ using backslash, LU decomposition, and using the matrix inverse (`inv(A)`).

You need to do all of the following parts, but you only need to include your final plot and the discussion of the plot (part (i)) in your writeup.

(a) Create matrices `A = rand(n)` and vectors `b = rand(n,1)` for $n = 700,\ 1200,\ 2400,\ 6000$, and 9000.

(b) Using the `tic` and `toc` commands, save the time it takes to solve $A\mathbf{x} = \mathbf{b}$ using the backslash operator for each n.

(c) Using the `tic` and `toc` commands, save the time it takes to solve $A\mathbf{x} = \mathbf{b}$ using LU decomposition for each n.

(d) Using the `tic` and `toc` commands, save the time it takes to solve $A\mathbf{x} = \mathbf{b}$ using A^{-1} (`inv(A)`) for each n.

(e) On a log-log plot (`loglog` in MATLAB), plot the time it takes to solve $A\mathbf{x} = \mathbf{b}$ using the backslash operator versus n. This plot should be made with blue circles and blue lines between them (`'-bo'` in MATLAB).

(f) Using a log-log plot on the same axes as above, plot the time it takes to solve $A\mathbf{x} = \mathbf{b}$ using LU decomposition versus n. This plot should be in red diamonds with red lines in between them (`'-rd` in MATLAB) and on the same axes as the blue circles.

(g) Using a log-log plot on the same axes as above, plot the time it takes to solve $A\mathbf{x} = \mathbf{b}$ using A^{-1} (`inv(A)`) versus n. This plot should be in green squares with green lines in between and on the same axes as the other plots.

(h) Label your axes!

(i) Include the plot in your writeup and comment on how the speed of each of the three methods compare with each other. Your answer should include a comparison between each of the three methods.

2. In this problem we will compare the speed and accuracy for multiple solves of a 20×20 matrix using backslash, LU, and the matrix inverse. It is often important to consider the trade off between speed and accuracy when deciding which method to use.

The Vandermonde matrix is a matrix that comes up when one tries to create a polynomial which goes through a number of points. This task is known as *polynomial interpolation* and we will be covering it in Week 7. Vandermonde matrices are created in MATLAB using the command `vander`.

You need to do all of the following parts, but you only need to include the tables in parts (f) and (g) and the discussion in parts (h)-(j) in your writeup.

(a) Create a 20×20 matrix using the command `A = vander(1:20)`. You should put the command `warning('off','all')` in your .m file before you solve any equations with this matrix.

(b) Use a `for` loop to solve $A\mathbf{x} = \mathbf{b}$ using backslash 100 times for 100 different $\mathbf{b}$ vectors. Each time through the loop, create a new $\mathbf{b}$ vector using the `rand` command. Use `tic` and `toc` to time how long it takes MATLAB to do all 100 solves.

(c) Do the same as in part (b) using LU decomposition. For LU decomposition, you should calculate the matrices L, U, and P just once outside of your `for` loop.

(d) Do the same as in part (b) using the matrix inverse (`inv(A)`). You should calculate the inverse of your matrix, A^{-1}, just once outside of your `for` loop.

(e) Do parts (b-d) for 10000 different vectors, using the `rand` command each time.

(f) Record your results in a table, e.g.,

	Time to solve for 100 vectors	Time to solve for 10000 vectors
Backslash		
LU		
inv		

(g) Create a new random vector $\mathbf{b}$. Solve $A\mathbf{x} = \mathbf{b}$ using each of the three methods: backslash, LU, and inverse. Compute the residual vector $\mathbf{r} = A\mathbf{x} - \mathbf{b}$ for each method. Add a column to your table for the norm of the residual error in each case, e.g.,

	Time for 100 vectors	Time for 10000 vectors	$\|\|r\|\|$
Backslash time			
LU time			
inv time			

(h) From the table, which of the methods is the fastest for this task (solving $A\mathbf{x} = \mathbf{b}$ for multiple different vectors $\mathbf{b}$)?

(i) From the table, which of the methods has the most residual error?

(j) Use your answers in parts (h) and (i) to say what you think is the best method for this task. Your answer should take into account both the speed and the residual error.