Homework 4

1. **Inverse of a block matrix.** Let $B$ and $D$ be invertible matrices of sizes $m \times m$ and $n \times n$, respectively, and let $C$ be any $m \times n$ matrix. Find the inverse of the block matrix

\[
A = \begin{bmatrix} B & C \\ 0 & D \end{bmatrix}
\]

in terms of $B^{-1}$, $C$, and $D^{-1}$.

*Hints.*

- Your goal is to find matrices $W$, $X$, $Y$, and $Z$ (in terms of $B^{-1}$, $C$, and $D^{-1}$) that satisfy

\[
A \begin{bmatrix} W & X \\ Y & Z \end{bmatrix} = I.
\]

Use block matrix multiplication to express this as a set of four matrix equations that you can then solve.

- You can get an idea of what the solution should look like by considering the case when $B$, $C$, and $D$ are scalars.

2. **Affine combinations of left inverses.** Let $Z$ be a tall $m \times n$ matrix with linearly independent columns, and let $X$ and $Y$ be left inverses of $Z$. Show that for any scalars $\alpha$ and $\beta$ satisfying $\alpha + \beta = 1$, $\alpha X + \beta Y$ is also a left inverse of $Z$.

3. **Rows and columns of a matrix and its inverse.** Suppose the $n \times n$ matrix $A$ is invertible, with inverse $B = A^{-1}$. We let the $n$-vectors $a_1, \ldots, a_n$ denote the columns of $A$, and $b_i^T, \ldots, b_n^T$ the rows of $B$. For each of the following statements, choose the correct response and justify your answer. **True** means the statement always holds, with no further assumptions. **False** means the statement does not always hold, without further assumptions.

   (a) For any $n$-vector $x$, we have $x = \sum_{i=1}^n (b_i^T x) a_i$. True False

   (b) For any $n$-vector $x$, we have $x = \sum_{i=1}^n (a_i^T x) b_i$. True False

   (c) For $i \neq j$, $a_i \perp b_j$. True False

   (d) For any $i$, $\|b_i\| \geq 1/\|a_i\|$. True False

   (e) For any $i$ and $j$, $b_i + b_j \neq 0$. True False

   (f) For any $i$, $a_i + b_i \neq 0$. True False
4. **Reverse-time linear dynamical system.** A linear dynamical system has the form

\[ x_{t+1} = Ax_t, \]

where \( x_t \) in the \((n\text{-vector})\) state in period \( t \), and \( A \) is the \( n \times n \) dynamics matrix. This formula gives the state in the next period as a function of the current state. We want to derive a recursion of the form

\[ x_{t-1} = A^{\text{rev}} x_t, \]

which gives the previous state as a function of the current state. We call this the **reverse time linear dynamical system**.

(a) When is this possible? When it is possible, what is \( A^{\text{rev}} \)?

(b) For the specific linear dynamical system with dynamics matrix

\[ A = \begin{bmatrix} 3 & 2 \\ -1 & 4 \end{bmatrix}, \]

find \( A^{\text{rev}} \), or explain why the reverse time linear dynamical system doesn’t exist.

5. **Solving linear equations in Julia.** Generate a random \( 20 \times 20 \) matrix \( A \) and a random \( 20\)-vector \( b \) using the following code:

\[
A = \text{rand}(20, 20) \\
b = \text{rand}(20)
\]

(1t’s very likely that the matrix you generate will be invertible.)

We solve the linear equation \( Ax = b \), *i.e.*, compute the solution \( x = A^{-1}b \) in Julia using several methods. In each case, you should check that the \( x \) you compute satisfies the equations by evaluating and reporting the norm of the residual, \( \|Ax - b\| \). (This should be very small.)

(a) Using the backslash operator:

\[ x = A \backslash b \]

(b) Computing the inverse of \( A \) explicitly:

\[ x = \text{inv}(A) \times b \]

(c) Using QR factorization, from the formula \( x = R^{-1}Q^Tb \):

\[
Q, R = \text{qr}(A) \\
x = R \backslash (Q' \times b)
\]

(You should check that the matrix \( Q \) obtained is very nearly orthogonal, \( R \) is an upper triangular matrix, and that \( A \) is very near \( QR \).)

(a) Determine how long it takes for your computer to solve a system of \( n = 2000 \) linear equations in \( n = 2000 \) variables (with invertible coefficient matrix) using Julia’s \( \backslash \) operator. You may use the following code.

\[
A = 1 + \text{rand}(2000, 2000) \\
b = \text{ones}(2000) \\
@time A\backslash b;
\]

(b) Julia is rather clever about how it solves systems of equations with \( \backslash \). Determine how long it takes for your computer to solve the following system of \( n = 2000 \) linear equations in \( n = 2000 \) variables.

\[
L = 1 + \text{rand}(2000, 2000) \\
\text{for } i = 1:2000 \\
  \text{for } j = i+1:2000 \\
    L[i, j] = 0 \\
  \text{end} \\
\text{end} \\
b = \text{ones}(2000) \\
@time L\backslash b;
\]

(c) Can you explain why the times differ by so much between the two systems, i.e., what is special about the matrix \( L \) as opposed to \( A \)? Make a hypothesis about what you think Julia is doing behind the scenes.

7. Sensitivity of solution of linear equations. Let \( A \) be an invertible \( n \times n \) matrix, and \( b \) and \( x \) be \( n \)-vectors satisfying \( Ax = b \). Suppose we now perturb the \( j \)th entry \( b \) by \( \epsilon \neq 0 \) (which is a traditional symbol for a small quantity), which means that \( b \) becomes \( \bar{b} = b + \epsilon e_j \). Let \( \bar{x} \) be the \( n \)-vector that satisfies \( A\bar{x} = \bar{b} \), i.e., the solution of the linear equations using the perturbed right-hand side. We are interested in \( \|x - \bar{x}\| \), which is how much the solution changes due to the change in the right-hand side.

(a) Show that \( \|x - \bar{x}\| \) does not depend on \( b \); it only depends on the matrix \( A \), \( \epsilon \), and \( j \).

(b) How would you find the index \( j \) that maximizes the value of \( \|x - \bar{x}\| \)? By part (a), your answer should be in terms of \( A \) (or quantities derived from \( A \)) and \( \epsilon \) only.

(c) Try this out in Julia with the following values:

\[
A = \begin{bmatrix} 1/2 & 1/3 & 1/4 \\ 1/3 & 1/4 & 1/5 \\ 1/4 & 1/5 & 1/6 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad \epsilon = 0.1.
\]

To prevent numerical imprecision errors, use the following code to build the data.
\[
A = \begin{bmatrix}
1/2 & 1/3 & 1/4 \\
1/3 & 1/4 & 1/5 \\
1/4 & 1/5 & 1/6
\end{bmatrix}
\]
\[
b = [1.0, 1.0, 1.0]
\]
\[
\text{epsilon} = 0.1
\]

Which \(j\) do you pick to maximize \(\|x - \tilde{x}\|\), and what value do you get for \(\|x - \tilde{x}\|\)? Check your answer from part (b) by direct calculation (i.e., simply finding \(\tilde{x}\) after perturbing entry \(j = 1, 2, 3\) of \(b\)).

8. Interpolation of rational functions. In this problem, you will find a rational function (i.e., ratio of polynomials)

\[
f(t) = \frac{c_0 + c_1 t + c_2 t^2}{1 + d_1 t + d_2 t^2}
\]

that satisfies the following (interpolation) conditions:

\[
f(1) = 2, \quad f(2) = 5, \quad f(3) = 9, \quad f(4) = -1, \quad f(5) = -4.
\]

Your job is to find numbers \(c_0, c_1, c_2, d_1\) and \(d_2\) for which these conditions hold.

(a) Let \(x = (c_0, c_1, c_2, d_1, d_2)\). Explain how to formulate this problem as finding a vector \(x\) that satisfies a system of linear equations \(Ax = b\). Be sure to specify what \(A\) and \(b\) are.

(b) Solve the system of linear equations in Julia to find the coefficients. What do you get for your vector \(x\)? Please include your code as part of your answer.

(c) Using MMAPlot, plot your rational function in Julia as a line plot and the 5 given points as a scatter plot. You can use the following code to produce the data needed for plotting, given that you have a vector \(x\) from part (b) ready to go.

\[
t = -2:0.01:8
f = (x[1] + x[2]*t + x[3]*t.^2) ./ (1 + x[4]*t + x[5]*t.^2)
points_t = [1, 2, 3, 4, 5]
points_f = [2, 5, 9, -1, -4]
\]

Please include your code and plots as part of your answer.