

Section exam 3 Numerical Analysis: Linear Algebra

Assigned Nov 21, 2024. Due Wednesday Dec 4, 2024, at noon.

Instructions:

- *This is an open book exam.*

You are allowed to consult the text book. You are also allowed to do research in other text books, on the internet, etc.

- The exam should be worked *individually*. Do not consult with other students.
- Please type your solutions if at all possible.
- Some of the questions involve writing Matlab code. If at all possible, please stick to Matlab. However, if you absolutely insist, you can use other similar languages, as long as the syntax is sufficiently similar that your code is readable to somebody who does not know the language. If you use a different language, adding comments describing what each line does would be helpful.
- The following files are provided on Canvas and the course webpage:
 - `exam3_student_version.m` provides starting points for problems 4 and 5.
 - `matrix_Y.mat` holds the matrix \mathbf{Y} that you need in problem 4 in Matlab format.
 - `matrix_Y.txt` holds the matrix \mathbf{Y} that you need in problem 4 in plain text format.

Question 1: (20p) Let \mathbf{A} be an invertible sparse matrix of size $m \times m$ with the singular value decomposition

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^*.$$

Further, let \mathbf{b} be a vector of size $m \times 1$, and suppose that you seek to solve the linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{1}$$

for \mathbf{x} . Suppose that m is large enough that you cannot use dense methods, but that you can afford to apply \mathbf{A} and \mathbf{A}^* to vectors.

(a) (7 points) Since \mathbf{A} is invertible, the system (1) is mathematically equivalent to the system

$$\mathbf{A}^*\mathbf{A}\mathbf{x} = \mathbf{A}^*\mathbf{b}. \tag{2}$$

Would you be able to use Conjugate Gradients to solve (2)? If you answer yes, then discuss the costs, and the pros and cons of this approach; identify some situations where this approach may work well, and some where it may not work well. If you answer no, then describe why CG cannot be used.

(b) (6 points) Consider the matrix

$$\mathbf{B} = \begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^* & \mathbf{0} \end{bmatrix}.$$

Let \mathbf{u}_i and \mathbf{v}_i be a pair of left and right singular vectors of \mathbf{A} , so that $\mathbf{A}\mathbf{v}_i = \sigma_i\mathbf{u}_i$. Prove that the vectors

$$\begin{bmatrix} \mathbf{u}_i \\ \mathbf{v}_i \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \mathbf{u}_i \\ -\mathbf{v}_i \end{bmatrix}$$

are both eigenvectors of \mathbf{B} , and then provide the full eigenvalue decomposition of \mathbf{B} , expressed in terms of the matrices \mathbf{U} , \mathbf{D} , and \mathbf{V} .

(c) (7 points) In order to solve (1), you could in principle instead solve the linear system

$$\begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^* & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix}. \tag{3}$$

Observe that the coefficient matrix in (3) is symmetric. Would you be able to use Conjugate Gradients to solve (3)? If you answer yes, then discuss the costs, and the pros and cons of this approach; identify some situations where this approach may work well, and some where it may not work well. If you answer no, then describe why CG cannot be used.

Question 2: (20 points)

(a) (6 points) Consider a symmetric positive definite (spd) matrix \mathbf{A} that has been partitioned as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{12}^* & \mathbf{A}_{22} \end{bmatrix}.$$

Suppose that \mathbf{A}_{11} has the Cholesky factorization

$$\mathbf{A}_{11} = \mathbf{R}_{11}^* \mathbf{R}_{11}.$$

Then \mathbf{A} admits the partial factorization

$$\mathbf{A} = \mathbf{R}_1^* \mathbf{A}_1 \mathbf{R}_1$$

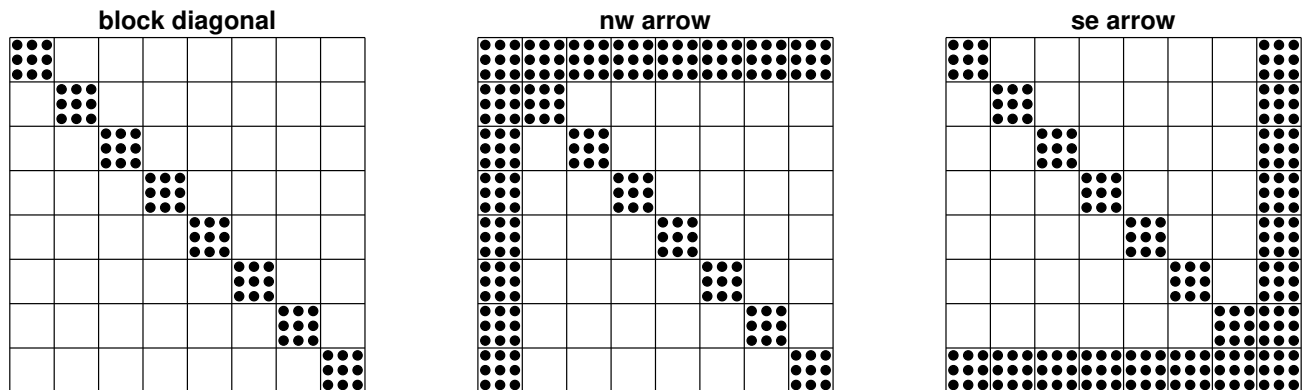
where

$$\mathbf{R}_1 = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad \text{and} \quad \mathbf{A}_1 = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}'_{22} \end{bmatrix}.$$

Give formulas for \mathbf{R}_{12} and \mathbf{A}'_{22} .

(b) (7 points) Suppose that \mathbf{A} is an spd matrix \mathbf{A} of size $N \times N$, where $N = nb$ for two positive integers n and b . Write a Matlab code that computes the Cholesky factorization of \mathbf{A} that is *blocked* with a block size b . (In other words, it will consist of a loop with $n - 1$ steps, where in each step a rank b update is applied to the bottom right block of the matrix, and a final step where the last remaining block is factorized.) Submit a printout of the code as your answer to this question. You are allowed to invoke the built-in command `chol`, but only to matrices of size $b \times b$.

(c) (7 points) Consider the Cholesky factorization of matrices with the following block patterns:



In the examples, $b = 3$ and $n = 8$.

Suppose that you write a code that exploits the zero structure of the matrices, as far as is possible in each case. What would the asymptotic flop count be for each of the patterns, expressed as a function of n and b ? To illustrate, for a general dense matrix, your answer would be $O(n^3 b^3)$

Note: For (c), you do **not** have to actually write the code – your answer consists of just the asymptotic flop counts.

Question 3: (20 points) Let \mathbf{A} be an $m \times m$ matrix that is symmetric and positive definite. Let k be a positive integer that is smaller than m , and suppose that \mathbf{G} is a matrix of size $m \times k$. If the matrix $\mathbf{G}^* \mathbf{A} \mathbf{G}$ is invertible, then we define the *Nyström* approximation to \mathbf{A} (with respect to \mathbf{G}) via

$$\mathbf{B} := (\mathbf{A} \mathbf{G}) (\mathbf{G}^* \mathbf{A} \mathbf{G})^{-1} (\mathbf{A} \mathbf{G})^*.$$

Let $\{\lambda_j\}_{j=1}^m$ denote the eigenvalues of \mathbf{A} , ordered by modulus so that

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_m|.$$

(a) (5 points) Suppose that \mathbf{A} has exact rank k , and that it has the (economy size) eigenvalue decomposition

$$\begin{array}{ccccc} \mathbf{A} & = & \mathbf{U} & \mathbf{D} & \mathbf{U}^* \\ m \times m & & m \times k & k \times k & k \times m \end{array}$$

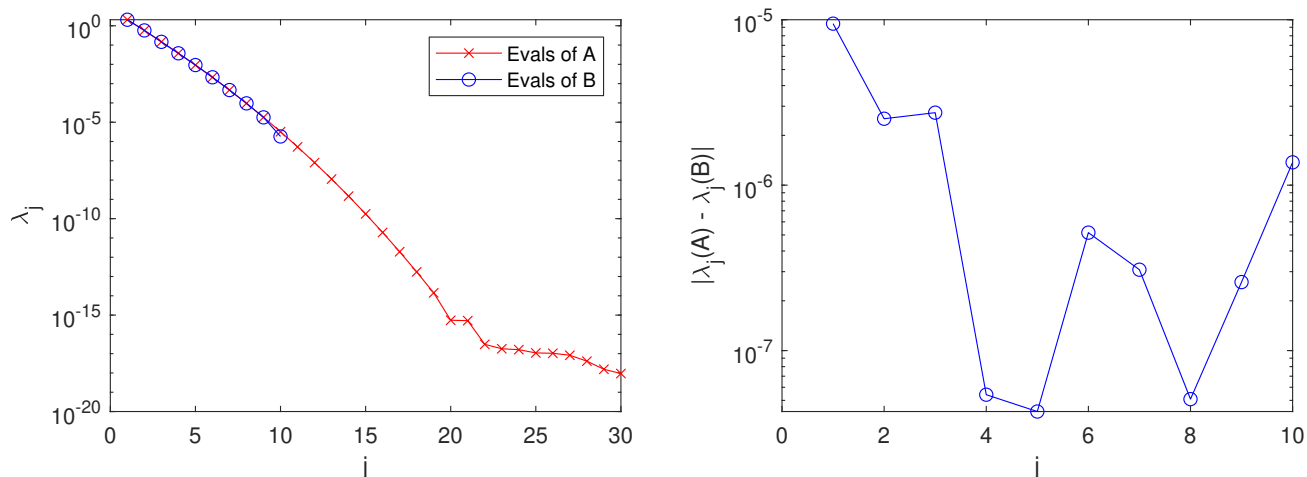
Prove that if the matrix $\mathbf{Z} := \mathbf{U}^* \mathbf{G}$ is non-singular, then the Nyström approximation is *exact*. In other words, $\mathbf{B} = \mathbf{A}$.

(b) (10 points) In parts (b) and (c), we will draw \mathbf{G} from a Gaussian random distribution. In Matlab, this is done through the command $\mathbf{G} = \text{randn}(m,k)$. It turns out that in this case, the Nyström approximation is often a very accurate approximation to \mathbf{A} if the eigenvalues $\{\lambda_j\}_{j=k+1}^m$ are small in comparison to the top ones. In particular, the eigenvalues of \mathbf{B} are very good approximations to the top k eigenvalues of \mathbf{A} .

For instance, consider the 30×30 matrix \mathbf{A} with entries

$$\mathbf{A}(i,j) = \frac{1}{\sqrt{i+j}}.$$

The left figure below plots the eigenvalues of \mathbf{A} versus the eigenvalues of \mathbf{B} for a particular instantiation of the method. The errors are shown to the right.



Code up the Nyström method, and replicate the graph! (Note that your graph may not look *exactly* like the one shown – the output depends on the draw of the random numbers!) Hand in your code and the graph.

(c) (5 points) Play with some other spd matrices whose eigenvalues decay. For your solution to (c), hand in descriptions of two matrices, and the plots analogous to those you created in (b).

Note: For problem (c), fun examples of matrices that arise in real applications are appreciated! However, you get full credit as long as you show that you tested a couple of different things.

Question 4: (20 points) In this problem, you are tasked with determining approximations to some eigenvalues and eigenvectors of a real symmetric matrix \mathbf{A} of size 100×100 . In preparing this problem, I generated a random starting vector \mathbf{q} , and then executed 49 steps of power iteration to build the 100×50 matrix

$$\mathbf{Y} = [\mathbf{q}, \mathbf{A}\mathbf{q}, \mathbf{A}^2\mathbf{q}, \mathbf{A}^3\mathbf{q}, \dots, \mathbf{A}^{49}\mathbf{q}].$$

You will find the resulting matrix \mathbf{Y} ready for download on Canvas (and on the course webpage).

In this problem, the eigenvalues of \mathbf{A} are ordered by modulus, so that

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_{100}|.$$

Using the information in \mathbf{Y} , compute approximations to the following quantities:

- (a) (8 points) The largest (in modulus) eigenvalue λ_1 .
- (b) (7 points) The first five entries of the eigenvector \mathbf{v}_1 associated with λ_1 . To resolve the normalization ambiguity, suppose that $\|\mathbf{v}_1\| = 1$ and that the first entry of \mathbf{v}_1 is positive.
- (c) (5 points) The eigenvalue λ_5 .

Please describe briefly how you estimated each quantity!

Estimate roughly how many correct digits that you have in each answer. (Guessing is fine, you do not need to rigorously motivate your guess.)

Question 5: (20 points) On Canvas (and on the course webpage) you will find a Matlab code that executes the Jacobi method that we discussed in class. This is a method that takes a given symmetric matrix \mathbf{A} and iteratively drives it to diagonal form through a series of similarity transforms. To be precise, recall from class that for any off-diagonal entry (i, j) of \mathbf{A} , it is possible to find a Givens rotation \mathbf{G} that maps the submatrix $\mathbf{A}([i, j], [i, j])$ to diagonal form:

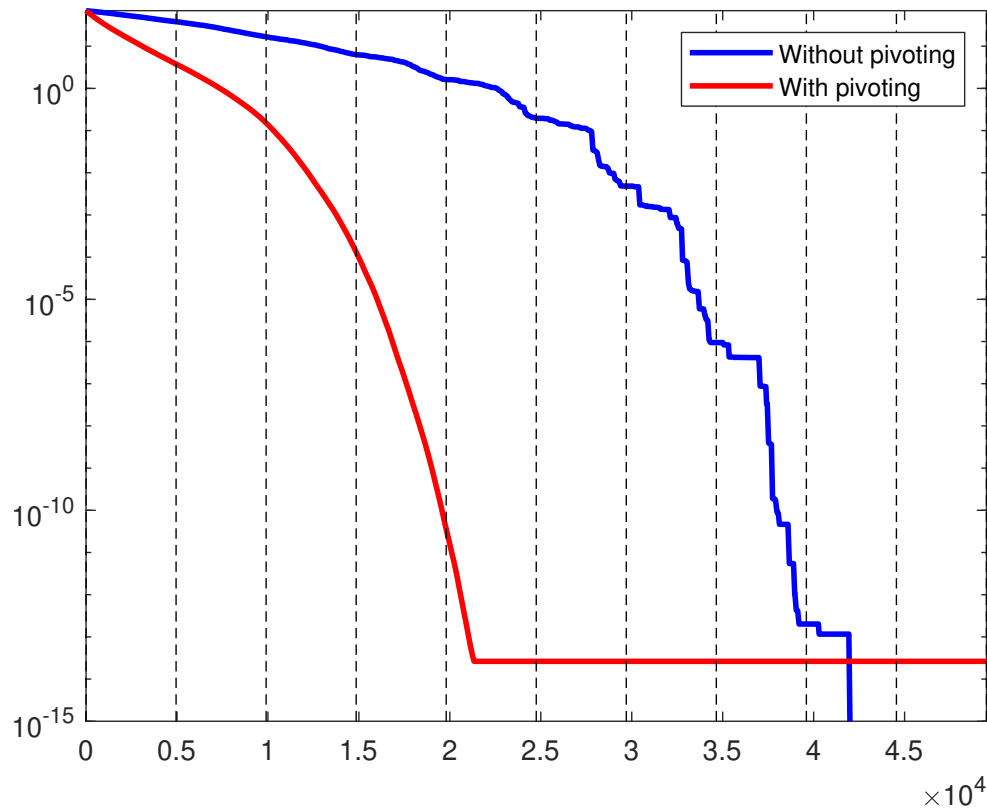
$$\mathbf{G}^* \begin{bmatrix} a_{i,i} & a_{i,j} \\ a_{i,j} & a_{j,j} \end{bmatrix} \mathbf{G} = \begin{bmatrix} a'_{i,i} & 0 \\ 0 & a'_{j,j} \end{bmatrix} \quad (4)$$

In the Jacobi iteration, you repeatedly loop over all the off-diagonal entries of \mathbf{A} . When entry (i, j) is processed, the method constructs a Givens rotation \mathbf{G} that drives the (i, j) and (j, i) entries to zero through a similarity transform where \mathbf{G}^* acts on the (i, j) rows from the left and \mathbf{G} on the (i, j) columns from the right.

In the code provided in the file `exam3_student_version.m`, the off-diagonal entries are processed in a fixed order, sweeping through one row of the matrix at a time.

To accelerate convergence, one may instead conduct a search to find the off-diagonal entry with the largest modulus, and target it instead.

The plot below shows the Frobenius norm of the off-diagonal part of the matrix after step j of the iteration. The blue line shows the error from the unpivoted method, and the red line shows the error from the pivoted method.



Write a code that implements the pivoted version of the Jacobi iteration. Submit an error plot analogous to the one shown, as well as a printout of your code.

Note: The pivoted version of Jacobi is far too expensive to use in practice. It increases the cost of one step of the method from $O(m)$ to $O(m^2)$! Efficiency would be greatly improved if you first drive \mathbf{A} to tridiagonal form, and then exploit the tridiagonal structure.