WBMA063-05 – Numerical Linear Algebra
Mock Exam
2025-2026

Instructions:

1. **Write your name and student number of the top of each sheet of writing paper!**
2. Use the writing (lined) and scratch (blank) paper provided, raise your hand if you need more paper.
3. Start each question on a new page.

This exam consists of 4 questions for a total of 90 points. 10 points are free.

1. Let $A \in \mathbb{R}^{n\times n}$ be a non-singular matrix.
   (a) (4 points) Define the LU decomposition of A. Also define the pivoted LU decomposition of A.

   Solution: The LU decomposition of A is $A = LU$ where $L \in \mathbb{R}^{n\times n}$ is unit lower triangular (lower triangular with 1s on the diagonal) and $U \in \mathbb{R}^{n\times n}$ is upper triangular. The pivoted LU decomposition is of the form $PA = LU$, where L and U are as before and $P \in \mathbb{R}^{n\times n}$ is a permutation matrix. That means P has columns of the identity but in any order.

   (b) (8 points) Assume the LU decomposition of $A = LU$ exists. Show that the dominant computational cost of computing L and U is $\mathcal{O}(\frac{2}{3}n^3)$ flops.

   Solution: Suppose we have completed $k-1$ steps of the LU decomposition algorithm, and define

   $$A \leftarrow A - L_1U_1 - \cdots - L_{k-1}U_{k-1}.$$

In step k of the LU algorithm, we take $p = A(k,k)$ to be the pivot and define

$$\boldsymbol{L}_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ A(k+1,k)/p \\ \vdots \\ A(n,k)/p \end{bmatrix}, \quad \boldsymbol{U}_k = \begin{bmatrix} 0 & \dots & 0 & A(k,k) & A(k,k+1) & \dots & A(k,n) \end{bmatrix},$$

which costs $n-k$ flops. Then we compute the left-lower $(n-k)\times(n-k)$ block of $\boldsymbol{L}_k\boldsymbol{U}_k$, which costs $(n-k)^2$ flops. Finally, we need to compute the new A, which comes down to $(n-k)^2$ minusses in the left-lower block of $A - \boldsymbol{L}_k\boldsymbol{U}_k$. So in total in step k, we have

$$n-k+(n-k)^2+(n-k)^2 \sim 2(n-k)^2 \text{ flops.}$$

The total dominant cost is then

$$\begin{aligned} \sum_{k=1}^{n-1} 2(n-k)^2 &= 2n^2\sum_{k=1}^{n-1} 1 + 2\sum_{k=1}^{n-1} k^2 - 4n\sum_{k=1}^{n-1} k \\ &\sim 2n^3 + 2\frac{n^3}{3} - 4n\frac{n^2}{2} \\ &= \frac{2}{3}n^3. \end{aligned}$$

(c) (5 points) Suppose you have computed a pivoted LU decomposition of A, $PA = LU$. Describe an efficient algorithm to explicitly compute the inverse A^{-1}. What is the computational cost of your algorithm?
Hint: consider building A^{-1} one column at a time.

Solution: Let $A^{-1} = \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 & \dots & \boldsymbol{b}_n \end{bmatrix}$. Since $AA^{-1} = I$, we have $A\boldsymbol{b}_i = \boldsymbol{e}_i$ for all $1 \le i \le n$, where e_i is the ith basis vector $[0,\dots,1,\dots 0]^T$. So given a pivoted LU decomposition of A, an algorithm to build the inverse would be:

For $i = 1,\dots,n$
- $\hat{\boldsymbol{e}}_i = P\boldsymbol{e}_i$
- $L\boldsymbol{y}_i = \hat{\boldsymbol{e}}_i$ Solve with forward substitution
- $U\boldsymbol{b}_i = \boldsymbol{y}_i$ Solve with backward substitution

The cost of this algorithm is n times $\mathcal{O}(2n^2)$ flops, which is $\mathcal{O}(2n^3)$.

(d) (10 points) Let $A = A^T \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix. Suppose we computed a Cholesky decomposition of A in floating point arithmetic with a backward stable algorithm. Then, still in floating poitn arithmetic, we use the computed Cholesky factors to solve the linear system $A\boldsymbol{x} = \boldsymbol{b}$. Show the computed solution $\hat{\boldsymbol{x}}$ is backward stable. You may use the fact that a triangular solve algorithm is backward stable.

Solution: Since the computed Cholesky factor $\hat{R}$ is backward stable, we have

$$\hat{R}^T\hat{R} = A + \delta A_{\text{CH}} \quad \text{for some } \delta A_{\text{CH}} \text{ such that } \frac{\|\delta A\|}{\|A\|} = \mathcal{O}(\varepsilon),$$

where ε is machine precision. Since forward substitution is backward stable, we have that the computed solution to $\hat{R}^T\boldsymbol{y} = \boldsymbol{b}$ satisfies

$$(\hat{R}^T + \delta\hat{R}_1^T)\hat{\boldsymbol{y}} = \boldsymbol{b} \quad \text{for some } \delta\hat{R}_1 \text{ such that } \frac{\|\delta\hat{R}_1\|}{\|\hat{R}\|} = \mathcal{O}(\varepsilon).$$

Similarly, the computed solution to $\hat{R}x = \hat{y}$ satisfies

$$(\hat{R} + \delta\hat{R}_2)\hat{x} = \hat{y} \quad \text{for some } \delta\hat{R}_2 \text{ such that } \frac{\|\delta\hat{R}_2\|}{\|\hat{R}\|} = \mathcal{O}(\varepsilon).$$

We can combine this to find

$$\begin{aligned}
\boldsymbol{b} &= (\hat{R}^T + \delta\hat{R}_1^T)\hat{\boldsymbol{y}} \\
&= (\hat{R}^T + \delta\hat{R}_1^T)(\hat{R} + \delta\hat{R}_2)\hat{\boldsymbol{x}} \\
&= (\hat{R}^T\hat{R} + \hat{R}^T\delta\hat{R}_2 + \delta\hat{R}_1^T\hat{R} + \delta\hat{R}_1^T\delta\hat{R}_2)\hat{\boldsymbol{x}} \\
&= (A + \delta A_{\text{CH}} + \hat{R}^T\delta\hat{R}_2 + \delta\hat{R}_1^T\hat{R} + \delta\hat{R}_1^T\delta\hat{R}_2)\hat{\boldsymbol{x}}
\end{aligned}$$

Define

$$\delta A = \delta A_{\text{CH}} + \hat{R}^T\delta\hat{R}_2 + \delta\hat{R}_1^T\hat{R} + \delta\hat{R}_1^T\delta\hat{R}_2.$$

Then

$$\begin{aligned}
\|\delta A\| &\le \|\delta A_{\text{CH}}\| + \|\hat{R}^T\|\|\delta\hat{R}_2\| + \|\delta\hat{R}_1^T\|\|\hat{R}\| + \|\delta\hat{R}_1^T\|\|\delta\hat{R}_2\| \\
&= \mathcal{O}(\varepsilon\|A\|) + \mathcal{O}(\varepsilon\|\hat{R}\|^2).
\end{aligned}$$

Now all we need to show is that $\|\hat{R}\|^2 = \mathcal{O}(\|A\|)$, and then we have backward stability. There are many ways to do this. We note that

$$\|\hat{R}\|_2^2 = \|\hat{R}^T\hat{R}\|_2,$$

which can be noted with e.g. the SVD. Then, immediately

$$\|\hat{R}\|_2^2 = \|\hat{R}^T\hat{R}\|_2 \le \|A\|_2 + \mathcal{O}(\varepsilon\|A\|_2) = \mathcal{O}(\|A\|_2).$$

(e) (4 points) An alternative approach to solve symmetric positive definite linear systems is to use Conjugate Gradients (CG). Reflect on the different contexts in which you might prefer a Cholesky-based approach to CG, and on the different contexts in which you might prefer CG to Cholesky. Explain your answer.

Solution: The Cholesky-based approach is guaranteed to be backward stable, but costs $\mathcal{O}(\frac{1}{3}n^3)$ flops. This makes it often the appropriate choice when A is not too large. When A is very large, then a direct method might not be possible. The choice for CG is reasonable when 1) A is very large, and/or 2) A is very sparse, since the dominant cost of a CG iteration is a matrix-vector product with A, and/or 3) we do not require a highly accurate solution, since we can stop CG early.

2. Let $A \in \mathbb{R}^{m\times n}$ with $m > n$ be a matrix with singular value decomposition $A = U\Sigma V^T$, where Σ is square.

   (a) (3 points) For each of the factors U, Σ, and V in the singular value decomposition, provide their dimension and their special structure.

   Solution: $U \in \mathbb{R}^{m\times n}$ has orthonormal columns, $\Sigma \in \mathbb{R}^{n\times n}$ is diagonal with on the diagonal non-increasing non-negative values, and $V \in \mathbb{R}^{n\times n}$ is an orthogonal matrix.

   (b) (3 points) Assume A has full column rank. Express the solution of the least squares problem

   $$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} \|A\boldsymbol{x} - \boldsymbol{b}\|_2$$

   in terms of the singular value decomposition factors. You need not prove the result.

   Solution: $\boldsymbol{x}^* = V\Sigma^{-1}U^T\boldsymbol{b}$

   (c) (8 points) Again assume A has full rank. Show that

   $$\|A\boldsymbol{x}^* - \boldsymbol{b}\|_2 = \|U_\perp^T\boldsymbol{b}\|_2,$$

   where $U_\perp \in \mathbb{R}^{m\times(m-n)}$ is the orthogonal complement of the left singular vectors U. That is, $U_\perp \in \mathbb{R}^{m\times(m-n)}$ is such that

   $$U_{\mathrm{F}} = \begin{bmatrix} U & U_\perp \end{bmatrix}$$

   is a square orthogonal matrix.

   Solution: We can write

   $$A = \begin{bmatrix} U & U_\perp \end{bmatrix} \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^T,$$

   and we know that $\boldsymbol{x}^* = V\Sigma^{-1}U^T\boldsymbol{b}$. Now

   $$\begin{aligned} \|A\boldsymbol{x}^* - \boldsymbol{b}\|_2 &= \left\|(U\Sigma V^T)(V\Sigma^{-1}U^T\boldsymbol{b}) - \boldsymbol{b}\right\|_2 \\ &= \left\|UU^T\boldsymbol{b} - \boldsymbol{b}\right\|_2 \\ &= \left\|(UU^T - I)\boldsymbol{b}\right\|_2. \end{aligned}$$

   Since

   $$U_F U_F^T = I = \begin{bmatrix} U & U_\perp \end{bmatrix} \begin{bmatrix} U^T \\ U_\perp^T \end{bmatrix} = UU^T + U_\perp U_\perp^T \implies UU^T - I = -U_\perp U_\perp^T.$$

   Now

   $$\|A\boldsymbol{x}^* - \boldsymbol{b}\|_2^2 = \|U_\perp U_\perp^T\boldsymbol{b}\|_2^2 = \boldsymbol{b}^T U_\perp U_\perp^T U_\perp U_\perp^T\boldsymbol{b} = \boldsymbol{b}^T U_\perp U_\perp^T\boldsymbol{b} = \|U_\perp^T\boldsymbol{b}\|_2^2$$

(d) (4 points) Computing the solution to a least squares problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|A\boldsymbol{x} - \boldsymbol{b}\|_2$$

via the singular value decomposition of A is backward stable. Suppose this algorithm is used in floating point arithmetic to compute the solution $\hat{\boldsymbol{x}}$ to the least squares problem. What does it mean for $\hat{\boldsymbol{x}}$ to be backward stable? Provide a mathematical expression.

Solution:

$$\hat{\boldsymbol{x}} = \arg\min \|(A + \delta A)\boldsymbol{x} - (b + \delta b)\|_2$$

for some δA and δb such that

$$\frac{\|\delta A\|}{\|A\|} = \mathcal{O}(\varepsilon), \quad \frac{\|\boldsymbol{\delta b}\|}{\|\boldsymbol{b}\|} = \mathcal{O}(\varepsilon),$$

where ε is machine precision.

(e) (3 points) Suppose $\hat{\boldsymbol{x}}$ is a backward stable solution to

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|A\boldsymbol{x} - \boldsymbol{b}\|_2,$$

which has exact solution $\boldsymbol{x}^*$. What can you say about the size of

$$\frac{\|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\|}{\|\boldsymbol{x}^*\|}?$$

Is it always, sometimes, or never close to machine precision? Support your reasoning.

Solution: The forward error is sometimes close, but it depends on the conditioning of A. If A is well-conditioned, say $\kappa(A) \approx 1$, then one can expect the forward error to be close to machine precision. If $\kappa(A) \gg 1$, then the forward error is big.

3. Let $A \in \mathbb{R}^{n\times n}$ be diagonalizable with eigenpairs $(\lambda_i, \boldsymbol{v}_i)$ where all $\boldsymbol{v}_i$ have unit norm. Assume the eigenvalues are all real and are ordered as

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0.$$

The power method starts by initializing a vector $\boldsymbol{x}_0 \in \mathbb{R}^n$ with unit norm, and then iterating as:

$$\begin{aligned} &\text{For } k = 0, 1, 2, \ldots \\ &\quad \boldsymbol{y}_k = A\boldsymbol{x}_k \\ &\quad \boldsymbol{x}_{k+1} = \boldsymbol{y}_k / \|\boldsymbol{y}_k\|_2. \end{aligned}$$

(a) (6 points) Expand $\boldsymbol{x}_0$ in the basis formed by the eigenvectors $\boldsymbol{v}_i$:

$$\boldsymbol{x}_0 = \sum_{i=1}^{n} \alpha_i \boldsymbol{v}_i.$$

Show

$$\boldsymbol{x}_k = \text{sign}(\lambda_1)^k \frac{\alpha_1 \boldsymbol{v}_1 + \sum_{i=2}^{n} \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^k \boldsymbol{v}_i}{\|\alpha_1 \boldsymbol{v}_1 + \sum_{i=2}^{n} \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^k \boldsymbol{v}_i\|_2}.$$

Solution: Note first we have

$$\boldsymbol{x}_k = \frac{A^k \boldsymbol{x}_0}{\|A^k \boldsymbol{x}_0\|_2}$$

and consider $A^k \boldsymbol{x}_0$:

$$A^k \boldsymbol{x}_0 = \sum_{i=1}^{n} \alpha_i A^k \boldsymbol{v}_i.$$

Since $\boldsymbol{v}_i$ is an eigenvector of A, we have

$$A\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i \implies A^2 \boldsymbol{v}_i = \lambda_i A \boldsymbol{v}_i = \lambda_i^2 \boldsymbol{v}_i \implies \cdots \implies A^k \boldsymbol{v}_i = \lambda_i^k \boldsymbol{v}_i.$$

So that

$$\begin{aligned} A^k \boldsymbol{x}_0 &= \sum_{i=1}^{n} \alpha_i A^k \boldsymbol{v}_i. \\ &= \sum_{i=1}^{n} \alpha_i \lambda_i^k \boldsymbol{v}_i \\ &= \alpha_1 \lambda_1^k \boldsymbol{v}_1 + \sum_{i=2}^{n} \alpha_i \lambda_i^k \boldsymbol{v}_i \\ &= \lambda_1^k \left(\alpha_1 \boldsymbol{v}_1 + \sum_{i=2}^{n} \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^k \boldsymbol{v}_i \right). \end{aligned}$$

The norm of this is

$$\|A^k \boldsymbol{x}_0\|_2 = |\lambda_1|^k \left\| \alpha_1 \boldsymbol{v}_1 + \sum_{i=2}^{n} \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^k \boldsymbol{v}_i \right\|_2 .$$

The result follows by seeing that

(b) (4 points) Explain what happens to $\boldsymbol{x}_k$ as k tends to infinity. Additionally, explain how the dominant eigenvalue λ_1 can be approximated from $\boldsymbol{x}_k$.

Solution: We have $\boldsymbol{x}_k \to \boldsymbol{v}_1$ since everything inside the sum tends to zero. Using the Rayleigh quotient

$$\lambda_1 = \frac{\boldsymbol{v}_1^* A \boldsymbol{v}_1}{\boldsymbol{v}_1^* \boldsymbol{v}_1} \approx \frac{\boldsymbol{x}_k^* A \boldsymbol{x}_k}{\boldsymbol{x}_k^* \boldsymbol{x}_k} = \boldsymbol{x}_k^* A \boldsymbol{x}_k.$$

(c) (6 points) Assume A is symmetric. Show

$$|\boldsymbol{x}_k^* A \boldsymbol{x}_k - \lambda_1| \approx \left(\left| \frac{\lambda_2}{\lambda_1} \right|^{2k} \right),$$

using the simplification that, as $k \to \infty$,

$$\boldsymbol{x}_k \approx \operatorname{sign}(\lambda_1)^k \frac{\alpha_1 \boldsymbol{v}_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k \boldsymbol{v}_2}{\|\alpha_1 \boldsymbol{v}_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k \boldsymbol{v}_2\|_2}.$$

Solution: Since A is symmetric, its eigenvectors form an orthogonal basis, so we have $v_1^* v_2 = 0$. Let $\beta_k = \|\alpha_1 \boldsymbol{v}_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k \boldsymbol{v}_2\|_2$. Then

$$\begin{aligned}
\beta_k^2 \, \boldsymbol{x}_k^* A \boldsymbol{x}_k &\approx \left(\alpha_1 \boldsymbol{v}_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k \boldsymbol{v}_2 \right)^T A \left(\alpha_1 \boldsymbol{v}_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k \boldsymbol{v}_2 \right) \\
&= \left(\alpha_1 \boldsymbol{v}_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k \boldsymbol{v}_2 \right)^T \left(\alpha_1 \lambda_1 \boldsymbol{v}_1 + \alpha_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k \lambda_2 \boldsymbol{v}_2 \right) \\
&= \alpha_1 \lambda_1 + 2\alpha_1 \alpha_2 \lambda_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k \boldsymbol{v}_1^T \boldsymbol{v}_2 + \alpha_2^2 \left(\frac{\lambda_2}{\lambda_1}\right)^{2k} \\
&= \alpha_1 \lambda_1 + \alpha_2^2 \lambda_2 \left(\frac{\lambda_2}{\lambda_1}\right)^{2k}.
\end{aligned}$$

Where we used $\boldsymbol{v}_2^T \boldsymbol{v}_1 = 0$ and that $\|\boldsymbol{v}_1\|_2 = \|\boldsymbol{v}_2\|_2 = 1$. Similarly,

$$\beta_k^2 = \alpha_1^2 + \alpha_2^2 \left(\frac{\lambda_2}{\lambda_1}\right)^{2k}$$

So that

$$\boldsymbol{x}_k^* A \boldsymbol{x}_k \approx \frac{\alpha_1 \lambda_1 + \alpha_2^2 \lambda_2 \left(\frac{\lambda_2}{\lambda_1}\right)^{2k}}{\sqrt{\alpha_1^2 + \alpha_2^2 \left(\frac{\lambda_2}{\lambda_1}\right)^{2k}}} \to \lambda_1$$

at rate

$$\left(\frac{\lambda_2}{\lambda_1}\right)^{2k}.$$

(d) (3 points) How would you adapt the algorithm to instead find the eigenvector corresponding to the eigenvalue of A closest to -2? Explain your reasoning.

Solution: The iterates

$$\boldsymbol{x}_k = \frac{(A + 2\lambda I)^{-k} \boldsymbol{x}_0}{\|(A + 2\lambda I)^{-k} \boldsymbol{x}_0\|_2},$$

and converge to the eigenvector corresponding the the smallest eigenvalue (in magnitude) of $(A + 2\lambda I)$, which corresponds to the eigenvalue of A closest to -2. So $\boldsymbol{x}_k$ converges to the corresponding eigenvector.

4. Consider a non-singular diagonalizable matrix $A \in \mathbb{R}^{n\times n}$.

(a) (4 points) The k-dimensional Krylov subspace associated to the linear system $Ax = b$ is

$$\mathcal{K}_k(A, \boldsymbol{b}) = \text{span}\left\{\boldsymbol{b}, A\boldsymbol{b}, A^2\boldsymbol{b}, \ldots, A^{k-1}\boldsymbol{b}\right\}.$$

Define a matrix $K_k \in \mathbb{R}^{n\times k}$ whose columns are the natural basis vectors for the Krylov subspace normalized; that is,

$$K_k = \begin{bmatrix} \frac{\boldsymbol{b}}{\|\boldsymbol{b}\|_2} & \frac{A\boldsymbol{b}}{\|A\boldsymbol{b}\|_2} & \frac{A^2\boldsymbol{b}}{\|A^2\boldsymbol{b}\|_2} & \cdots & \frac{A^{k-1}b}{\|A^{k-1}\boldsymbol{b}\|_2} \end{bmatrix}.$$

Qualitatively, explain why the matrix K_k becomes increasingly ill-conditioned as k increases.

Solution: The vector $A^k\boldsymbol{b}$ tends to the dominant eigenvector of A as k tends to infinity (power method). So the columns of K_k increasingly point in the same direction, which means the matrix will be increasingly ill-conditioned.

(b) (8 points) It is instead advised to use the Arnoldi algorithm to build a basis for the Krylov subspace $\mathcal{K}_k(A, \boldsymbol{b})$. This algorithm computes an orthogonal basis $\{\boldsymbol{q}_j\}$ for the Krylov subspace:

$$\mathcal{K}_k(A, \boldsymbol{b}) = \text{span}\left\{\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_k\right\}, \quad \boldsymbol{q}_i^T\boldsymbol{q}_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

The matrix

$$Q_k = \begin{bmatrix} \boldsymbol{q}_1 & \boldsymbol{q}_2 & \cdots & \boldsymbol{q}_k \end{bmatrix}$$

is then well-conditioned. The Arnoldi algorithm (in exact arithmetic) can be written as

$$\begin{aligned} &\boldsymbol{q}_1 = \boldsymbol{b}/\|\boldsymbol{b}\|_2 \\ &\text{For } j = 1, 2, \ldots, k \\ &\qquad \boldsymbol{v} = A\boldsymbol{q}_j \\ &\qquad \boldsymbol{w} = \boldsymbol{v} - \sum_{i=1}^{j} (\boldsymbol{q}_i^T\boldsymbol{v})\boldsymbol{q}_i \\ &\qquad \boldsymbol{q}_{j+1} = \boldsymbol{w}/\|\boldsymbol{w}\|_2 \end{aligned}$$

This is not how the algorithm is implemented in floating point arithmetic. Describe the algorithm as it is implemented. Then, show that after k steps, we have the identity

$$AQ_k = Q_{k+1}\tilde{H}_k,$$

for a $(k+1) \times k$ matrix $\tilde{H}_k$. Describe the structure of this matrix.

Solution: The Arnoldi algorith uses modified Gram-Schmidt, and boils down to

$$\begin{aligned}
&\boldsymbol{q}_1 = \boldsymbol{b}/\|\boldsymbol{b}\|_2 \\
&\text{For } j = 1, 2, \ldots, k \\
&\quad \boldsymbol{v} = A\boldsymbol{q}_j \\
&\quad \text{For } i = 1, 2, \ldots, j \\
&\quad\quad h_{ij} = \boldsymbol{q}_i^T \boldsymbol{v} \\
&\quad\quad \boldsymbol{v} \leftarrow \boldsymbol{v} - h_{ij}\boldsymbol{q}_i \\
&\quad h_{j+1,j} = \|\boldsymbol{v}\|_2 \\
&\quad \boldsymbol{q}_{j+1} = \boldsymbol{v}/\|\boldsymbol{v}\|_2
\end{aligned}$$

It is immediate that

$$h_{j+1,j}\boldsymbol{q}_{j+1} = A\boldsymbol{q}_j - h_{1j}\boldsymbol{q}_1 - \cdots - h_{jj}\boldsymbol{q}_j$$

and thus that

$$A\boldsymbol{q}_j = h_{1j}\boldsymbol{q}_1 + \cdots + h_{jj}\boldsymbol{q}_j + h_{j+1,j}\boldsymbol{q}_{j+1}.$$

Since this holds for $j = 1, \ldots, k$, we have

$$AQ_k = Q_{k+1}\tilde{H}_k,$$

where

$$\tilde{H}_k = \begin{bmatrix} h_{11} & h_{12} & \ldots & h_{1k} \\ h_{21} & h_{22} & \ldots & h_{2k} \\ & h_{32} & \ddots & \vdots \\ & & \ddots & h_{k,k} \\ & & & h_{k_1,k} \end{bmatrix},$$

which is an upper Hessenberg matrix.

(c) (4 points) Let the eigendecomposition of A be given by $A = X\Lambda X^{-1}$. The residual at the kth GMRES iterate $\boldsymbol{x}_k$ can be bounded by

$$\frac{\|\boldsymbol{b} - A\boldsymbol{x}_k\|_2}{\|\boldsymbol{b}\|_2} \leq \kappa(X) \min_{p \in \mathcal{P}_k} \max_{\lambda \in \Lambda(A)} |p(\lambda)|,$$

where we assumed $\boldsymbol{x}_0 = \boldsymbol{0}$, $\mathcal{P}_k$ is a subset of all degree k polynomials p with $p(0) = 1$, and $\Lambda(A)$ is the set of all eigenvalues of A. Comment on the expected convergence behaviour based on the eigenvalues of A. Distinguish between the case when A is normal and when A is not normal.
Hint: A is normal when $A^T A = AA^T$. The eigenvectors of a normal matrix form an orthogonal basis.

Solution: When A is normal, X is orthogonal and thus $\kappa(X) = 1$. The new bound is

$$\frac{\|\boldsymbol{b} - A\boldsymbol{x}_k\|_2}{\|\boldsymbol{b}\|_2} \leq \min_{p \in \mathcal{P}_k} \max_{\lambda \in \Lambda(A)} |p(\lambda)|.$$

With this bound, we have that the eigenvalues of A can be used to predict the convergence behaviour of GMRES. In particular, if the eigenvalues are in a single cluster (away from zero), or in a small number of clusters (all away from zero), then convergence is fast.
When A is not normal, the eigenvalues of A are not enough to predict the convergence behaviour of GMRES. The reason is the presence of $\kappa(X)$ in the bound, which can be large.

(d) (3 points) Let A be symmetric matrix with $m < n$ distinct eigenvalues. Will GMRES converge to the exact solution in at most m iterations? Explain your reasoning.

Solution: When A is symmetric, it is normal. So now we have the bound

$$\frac{\|\boldsymbol{b} - A\boldsymbol{x}_k\|_2}{\|\boldsymbol{b}\|_2} \leq \min_{p \in \mathcal{P}_k} \max_{\lambda \in \Lambda(A)} |p(\lambda)|.$$

If A has m distinct eigenvalues $\lambda_1, \ldots, \lambda_m$, there exists a degree m polynomial

$$p(x) = \left(1 - \frac{x}{\lambda_1}\right) \ldots \left(1 - \frac{x}{\lambda_m}\right)$$

that is zero at all eigenvalues. So then

$$\frac{\|\boldsymbol{b} - A\boldsymbol{x}_m\|_2}{\|\boldsymbol{b}\|_2} \leq 0$$

and $\boldsymbol{x}_m$ must be the exact solution.