NLA theoretical questions from previous exams

Claudio Tessa

2025-2026

1 Consider the linear system $A\mathbf{x} = \mathbf{b}$, where $A \in \mathbb{R}^{n\times n}$ is an invertible matrix:

1. Describe the gradient method for the numerical solution of the above linear system. Describe the applicability conditions, the algorithm and the interpretation of the scheme as a minimization problem. **Define the notation employed**.

The gradient method requires A to be SPD. In this case, solving the linear system $A\mathbf{x} = \mathbf{b}$ is equivalent to minimizing the quadratic function $\Phi : \mathbb{R}^n \to \mathbb{R}$

$$\Phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} - \mathbf{x}^T\mathbf{b}$$

Since A is SPD, $\Phi(\mathbf{x})$ defines a paraboloid with global minimum in $\mathbf{x}$. Since $\nabla\Phi(\mathbf{x}) = A\mathbf{x} - \mathbf{b}$ (assuming A is symmetric), we have that the minimum ($\nabla\Phi(\mathbf{x}) = \mathbf{0}$) coincides with the solution of $A\mathbf{x} = \mathbf{b}$.

The update rule is

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \nabla\Phi(\mathbf{x}^{(k)})$$

where α_k is the step size. The optimal parameter α_k is chosen by asking that

$$\frac{\mathrm{d}\Phi\left(\mathbf{x}^{(k+1)}\right)}{\mathrm{d}\alpha_k} = 0 \qquad \Longrightarrow \qquad \alpha_k = \frac{\left(\mathbf{r}^{(k)}\right)^T \mathbf{r}^{(k)}}{\left(\mathbf{r}^{(k)}\right)^T A\mathbf{r}^{(k)}}$$

Note that $\nabla\Phi(\mathbf{x}) = A\mathbf{x} - \mathbf{b} = -\mathbf{r}$. Therefore, the residual $\mathbf{r}^{(k)}$ at step k is

$$\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)} = -\nabla\Phi(\mathbf{x}^{(k)})$$

Algorithm 1 Gradient method (steepest descent)

Given $\mathbf{x}^{(0)}$, compute $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$
while stopping criteria **do**
  $\alpha_k = \dfrac{\left(\mathbf{r}^{(k)}\right)^T \mathbf{r}^{(k)}}{\left(\mathbf{r}^{(k)}\right)^T A\mathbf{r}^{(k)}}$ ▷ $\approx n$ flops if A is sparse
  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{r}^{(k)}$ ▷ $\approx n$ flops
  $\mathbf{r}^{(k+1)} = (I - \alpha_k A)\mathbf{r}^{(k)}$ ▷ $\approx n$ flops
end while

Notation:

- $\mathbf{x}^{(k)}$ approximate solution at step k;
- $\nabla\Phi(\mathbf{x})$ gradient of $\Phi(\mathbf{x})$ with respect to the vector $\mathbf{x}$
- $\mathbf{r}^{(k)}$ residual at iteration k

2. State the main theoretical result for the Gradient Method. **Define the notation employed**.

Let $\mathbf{e}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}$ the error at iteration k. We have that

$$\|\mathbf{e}^{(k+1)}\|_A \leq \frac{K(A)-1}{K(A)+1}\|\mathbf{e}^{(k)}\|_A \qquad \Longrightarrow \qquad \|\mathbf{e}^{(k)}\|_A \leq \left(\frac{K(A)-1}{K(A)+1}\right)^k \|\mathbf{e}^{(0)}\|_A$$

Therefore the method converges **linearly** at a rate determined by the condition number $K(A)$. A large $K(A)$ (poorly conditioned matrix) means slower convergence, and viceversa.

3. Describe the Conjugate Gradient (CG) method. Recall the interpretation of the scheme as a Krylov subspace method and state the main theoretical result. **Define the notation employed**.

In the CG method, we introduce an updating direction $\mathbf{d}^{(k+1)}$ in such a way that it is A-conjugate to all the previous directions $\mathbf{d}^{(j)}$ with $j \leq k$ (i.e., orthogonal with respect to the scalar product induced by A):

$$\left(\mathbf{d}^{(k+1)}, \mathbf{d}^{(j)}\right)_A = \left(\mathbf{d}^{(k+1)}, A\mathbf{d}^{(j)}\right) = 0 \qquad \forall j \leq k \quad \begin{matrix}\text{conjugate}\\ \text{directions}\end{matrix}$$

The algorithm then becomes:

Algorithm 2 Conjugate Gradient (CG) method

Given $\mathbf{x}^{(0)}$, compute $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$
Set $\mathbf{d}^{(0)} = \mathbf{r}^{(0)}$
while stopping criteria **do**
  $a_k = \dfrac{\left(\mathbf{d}^{(k)}\right)^T \mathbf{r}^{(k)}}{\left(\mathbf{d}^{(k)}\right)^T A\mathbf{r}^{(k)}}$
  $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$ ▷ Update $\mathbf{x}^{(k+1)}$ along $\mathbf{d}^{(k)}$
  $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha_k A\mathbf{d}^{(k)}$ ▷ Update $\mathbf{r}^{(k+1)}$
  $\beta_k = \dfrac{\left(A\mathbf{d}^{(k)}\right)^T \mathbf{r}^{(k+1)}}{\left(A\mathbf{d}^{(k)}\right)^T \mathbf{d}^{(k)}}$
  $\mathbf{d}^{(k+1)} = \mathbf{r}^{(k+1)} - \beta_k \mathbf{d}^{(k)}$ ▷ Update $\mathbf{d}^{(k+1)}$
end while

β_k is used as a scalar coefficient to update the search direction.

Consider the k-th Krylov subspace generated by A from $\mathbf{r}^{(0)}$:

$$\mathcal{K}_k(A, \mathbf{r}^{(0)}) = \text{span}\{\mathbf{r}^{(0)}, A\mathbf{r}^{(0)}, \ldots, A^{k-1}\mathbf{r}^{(0)}\}$$

We have that

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \mathbf{r}^{(k-1)} = \mathbf{x}^{(0)} + \mathbf{r}^{(0)} + \cdots + \mathbf{r}^{(k-1)} = \mathbf{x}^{(0)} + p_{k-1}(A)\mathbf{r}^{(0)}$$

where $p_k(z) := (1-z)^k$ is a polynomial of degree k.

Since $p_{k-1}(A)\mathbf{r}^{(0)} \in \mathcal{K}_k(A, \mathbf{r}^{(0)})$, we have that

$$\mathbf{x}^{(k)} \in \mathbf{x}^{(0)} + \mathcal{K}_k(A, \mathbf{r}^{(0)})$$

The CG is a krylov space solver, because by definition it chooses the step length α_k such that $\mathbf{x}^{k+1}$ is locally optimal on the search line.

Moreover, we are sure that the approximate solutions $\mathbf{x}^{(k)}$ given by the CG method are **optimal** as they minimize the A-norm of the error.

The CG method converges to the exact solution in at most n iterations. At each iteration k, the error $\mathbf{e}^{(k)} = \mathbf{x} - \mathbf{x}^{(k)}$ can be bounded by

$$\left\|\mathbf{e}^{(k)}\right\|_A \leq \frac{2c^k}{1+c^{2k}} \left\|\mathbf{e}^{(0)}\right\|_A \qquad \text{with} \qquad c = \frac{\sqrt{K(A)}-1}{\sqrt{K(A)}+1}$$

4. Comment on the main differences between the Gradient and the Conjugate Gradient methods.

The main differences are:

- **Geometric interpretation**:
  - The *gradient* method looks for the direction of the steepest descent "right now" (at every iteration k), even if moving in that direction "undoes" previous progress. Gives orthogonal residuals ($\mathbf{r}_{k+1} \perp \mathbf{r}_k$);
  - The *CG* method chooses a direction that is A-conjugate to all previous directions, ensuring that once we minimize the error in a specific direction, we never have to revisit it. Gives orthogonal residuals compared to all the previous residuals ($\mathbf{r}_{k+1} \perp \mathcal{K}_k$).
- **Convergence rate**:
  - The *gradient* method reduces the error by a factor proportional to $\frac{K(A)-1}{K(A)+1}$;
  - The *CG* method reduces the error by a factor proportional to $\frac{\sqrt{K(A)}-1}{\sqrt{K(A)}+1}$.

  So if A is badly conditioned, CG converges faster.

- **Number of iterations**:
  - The *gradient* method theoretically iterates forever, asymptotically approaching the solution;
  - The *CG* method converges in at most n iterations in exact arithmetic (as there cannot exist more than n mutually orthogonal vectors in an n-dimensional space).

5. Describe the main changes you have to introduce if you want to use the preconditioned Gradient (PCG) and Conjugate Gradient methods and state the main conditions the preconditioner P has to satisfy. **Introduce the notation employed**
6. Describe the algebraic multigrid method as a preconditioning strategy to accelerate the convergence of the PCG method.

2 Consider the rectangular linear system $A\mathbf{x} = \mathbf{b}$, where $A \in \mathbb{R}^{m\times n}, m \geq n$:

1. Provide the definition of the solution in the least-square sense and state under which condition the problem is well posed. **Introduce the notatation employed**.

We say that $\mathbf{x}^*$ is a solution of $A\mathbf{x} = \mathbf{b}$ in the **least-squares** sense if

$$\Phi(\mathbf{x}^*) = \min_{\mathbf{y}\in\mathbb{R}^n} \Phi(\mathbf{y})$$

where $\Phi(\mathbf{y}) = \|A\mathbf{y} - \mathbf{b}\|_2^2$ (we minimize the euclidian norm of the residual).

$\mathbf{x}^*$ is found by imposing the gradient of $\Phi(\cdot)$ equal to zero at $\mathbf{x}^*$. We have that

$$\begin{aligned}\Phi(\mathbf{y}) &= (A\mathbf{y} - \mathbf{b})^T(A\mathbf{y} - \mathbf{b}) \\ &= \mathbf{y}^T A^T A\mathbf{y} - 2\mathbf{y}^T A\mathbf{b} + \mathbf{b}^T\mathbf{b}\end{aligned}$$

$$\implies \nabla\Phi(\mathbf{y}) = 2A^T A\mathbf{y} - 2A^T\mathbf{b}$$

So $\mathbf{x}^*$ must be the solution of the system

$$A^T A\mathbf{x}^* = A^T\mathbf{b}$$

2. Describe the QR factorization of the matrix A and discuss how it can be employed to solve the above linear system in the least square sense. **Introduce all the notation employed**.

The QR factorization of $A \in \mathbb{R}^{m\times n},\ m \geq n$ decomposes A into

$$A = QR$$

where

- $Q \in \mathbb{R}^{m\times m}$ is a square orthogonal matrix (its column form an orthonor-

mal basis of $\mathbb{R}^n$;

- $R \in \mathbb{R}^{m \times n}$ is an upper triangular matrix.

Let us consider instead the **reduced QR factorization** of A:

$$A = \hat{Q}\hat{R}$$

where

- $\hat{Q} \in \mathbb{R}^{m \times n}$ contains the first n columns of Q;
- $\hat{R} \in \mathbb{R}^{n \times n}$ is the top square part of R (still upper triangular);

If $A = [\mathbf{a}_1, \ldots, \mathbf{a}_n]$, to find such factorization, we need to find orthonormal vectors $[\mathbf{q}_1, \ldots, \mathbf{q}_n]$ such that

$$\text{span}(\mathbf{a}_1, \ldots, \mathbf{a}_j) = \text{span}(\mathbf{q}_1, \ldots, \mathbf{q}_j) \qquad \forall j = 1, \ldots, n$$

$$\implies \underbrace{[\mathbf{a}_1, \ldots, \mathbf{a}_n]}_{A} = \underbrace{[\mathbf{q}_1, \ldots, \mathbf{q}_n}_{\hat{Q}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & \ldots & r_{1n} \\ 0 & r_{22} & \ldots & r_{2n} \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \ldots & r_{nn} \end{bmatrix}}_{\hat{R}}$$

Then we can use the following **theorem**:

If A has full rank, the unique solution in the least-square sense of $A\mathbf{x}^* = \mathbf{b}$ is given by

$$\mathbf{x}^* = \hat{R}^{-1}\hat{Q}^T\mathbf{b}$$

where $\hat{R} \in \mathbb{R}^{n \times n}$ and $\hat{Q} \in \mathbb{R}^{m \times n}$ are the matrices of the reduced QR factorization of A.

3. What can be done if A does not have full rank?

We can use the SVD factorization (see next point).

4. Define the SVD factorization of A. **Introduce the notation employed**.

For any real matrix $A \in \mathbb{R}^{m \times n}$, the SVD factorization of A is

$$A = U\Sigma V^T$$

where:

- $U \in \mathbb{R}^{m \times m}$ is an orthogonal matrix;
- $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{m \times n}$ is a diagonal matrix, with $p = \min(m, n)$ and $\sigma_1 \geq \cdots \geq \sigma_p \geq 0$;
- $V \in \mathbb{R}^{n \times n}$ is an orthogonal matrix.

5. Discuss under which condition and how the SVD factorization can be employed to solve the above linear system in the least square sense. **Introduce all the notation employed**.

Suppose

- $A \in \mathbb{R}^{m \times n}, m \geq n$ does not have full rank;
- The SVD of A is $A = U\Sigma V^T$.

then the unique solution in the least-squares sense of $A\mathbf{x}^* = \mathbf{b}$ is given by:

$$\mathbf{x}^* = A^\dagger \mathbf{b}$$

where $A^\dagger$ is the pseudo-inverse of A:

$$A^\dagger = V\Sigma^\dagger U^T \qquad \text{with} \qquad \Sigma^\dagger = \text{diag}\left(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_p}, 0, \ldots, 0\right) \in \mathbb{R}^{n \times m}$$

3 Consider the following eigenvalue problem: $A\mathbf{x} = \lambda\mathbf{x}$, where $A \in \mathbb{R}^{n \times n}$ is given.

1. Describe the power method for the numerical approximation of the largest in modulus eigenvalue of A. Introduce the notation, the algorithm, and the applicability conditions.

Assume A as n eigenvalues $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$, with $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ the corresponding linearly independent eigenvectors. The power method is used to identify the largest in modulus eigenvalue λ_1.

Starting from a given nonzero vector $\mathbf{x}^{(0)}$ such that $\|\mathbf{x}^{(0)}\| = 1$, the iteration scheme for $k \geq 0$ is:

$$\begin{aligned} \mathbf{y}^{(k+1)} &\leftarrow A\mathbf{x}^{(k)} \\ \mathbf{x}^{(k+1)} &\leftarrow \frac{\mathbf{y}^{(k+1)}}{\|\mathbf{y}^{(k+1)}\|} \\ \lambda^{(k+1)} &\leftarrow \left[\mathbf{x}^{(k+1)}\right]^T A\mathbf{x}^{(k+1)} \end{aligned}$$

where:

- $\mathbf{x}^{(k)}$ is the approximation of $\mathbf{x}_1$ at step k (converges to a multiple of $\mathbf{x}_1$).
- $\lambda^{(k)}$ is the approximation of λ_1 at step k (converges to λ_1).

Note: $|\lambda_1|$ must be unique;

2. State the main theoretical result.

The convergvence rate depends on the ratio $\dfrac{|\lambda_2|}{|\lambda_1|}$, where λ_2 is the second

largest in modulus eigenvalue. Therefore, the convergence rate is slow if $|\lambda_2| \approx |\lambda_1|$, and fast if $|\lambda_2| \ll |\lambda_1|$.

3. Discuss how the power method can be suitably modified in order to approximate the smallest in modulus eigenvalue of A and comment on computation cost.

To find the smallest in modulus eigenvalues, we can use the fact that the eigenvalues of A^{-1} are the reciprocals of those of A. Thereforen the smallest eigenvalue of A is the largest eigenvalue of A^{-1}. Instead of explicitly computing A^{-1}, we can modify the power method as follows (inverse power method):

Starting from a given nonzero vector $\mathbf{x}^{(0)}$ such that $\|\mathbf{x}^{(0)}\| = 1$, the iteration scheme for $k \geq 0$ is:

$$\text{Solve the linear system } A\mathbf{y}^{(k+1)} = \mathbf{x}^{(k)}$$

$$\text{Like with the power method } \begin{cases} \mathbf{x}^{(k+1)} \leftarrow \dfrac{\mathbf{y}^{(k+1)}}{\|\mathbf{y}^{(k+1)}\|} \\ \lambda^{(k+1} \leftarrow \left[\mathbf{x}^{(k+1)}\right]^T A\mathbf{x}^{(k+1)} \end{cases}$$

The inverse power method requires solving a linear system for each iteration, which is very expensive: $O(n^3)$. We can instead use e.g. LU factorization before the iterations start, so the cost per iteration becomes $O(n^2)$, comparable to that of the standard power method.

4. Describe the deflation technique. Present the algorithm and the applicability conditions.

After computing the dominant eigenvalue, we can compute additional eigenvalues using **deflation** technique, which removes the known eigenvalue. It constructs a new matrix B with eigenvalues $\lambda_2, \ldots, \lambda_n$, and then we can obtain λ_2 (the new dominant eigenvalue) with the power method.

(a) Choose any nonsingular matrix S such that $S\mathbf{v}_1 = \alpha\mathbf{e}_1$, a scalar multiple of $\mathbf{e}_1$ (the first column of the identity matrix I).

(b) Compute SAS^{-1}, which is a matrix of the form

$$SAS^{-1} = \begin{bmatrix} \lambda_1 & \mathbf{b}^T \\ 0 & B \end{bmatrix}$$

where $B \in \mathbb{R}^{(n-1)\times(n-1)}$

(c) Using the power method, we find λ_2 and $\mathbf{z}_2$ eigenvector of B.

(d) To recover $\mathbf{v}_2$ eigenvector of A, we can use the following:

$$\mathbf{v}_2 = S^{-1}\begin{pmatrix} \alpha \\ \mathbf{z}_2 \end{pmatrix} \qquad \alpha = \frac{\mathbf{b}^T\mathbf{z}_2}{\lambda_1 - \lambda_2}$$

Note that this requires $\lambda_2 \neq \lambda_1 (<)$.

4 Consider the following problem: find $\mathbf{x} \in \mathbb{R}^n, A\mathbf{x} = \mathbf{b}$, where $A \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^n$ are given.

1. State under which conditions the mathematical problem is well posed.

   The problem is well posed if and only if A is invertible: $\det A \neq 0$. In this case, the solution $\mathbf{x}$ exists and is unique.

2. Describe the general form of a linear iterative method for the approximate solution of $A\mathbf{x} = \mathbf{b}$ and describe the stopping criteria.

   The general form of a linear iterative method is

   $$\mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{f} \qquad k \geq 0$$

   where $B \in \mathbb{R}^{n \times n}$ (iteration matrix) and $\mathbf{f} \in \mathbb{R}^n$ uniquely identify the method.

   For an iterative method to make sense it must satisfy the convergence property

   $$\lim_{k \to +\infty} \mathbf{x}^{(k)} = \mathbf{x}$$

   therefore we must have that if $\mathbf{x}^{(k)}$ happens to be the exact solution $\mathbf{x}$, then $\mathbf{x}^{(k+1)}$ must be again equal to $\mathbf{x}$

   $$\implies \mathbf{x} = B\mathbf{x} + \mathbf{f} \implies \mathbf{f} = (I - B)\mathbf{x}$$

   This property is called **consistency**.

   The stopping criteria is a test used to determine when to stop the iteration. Given a tolerance ε, we can typically use criteria based on:

   - **Residual**:
     $$\frac{\|\mathbf{b} - A\mathbf{x}^{(k)}\|}{\|\mathbf{b}\|} < \varepsilon$$
     where $\mathbf{b} - A\mathbf{x}^{(k)}$ is the residual at step k.
   - **Distance betweeen consecutive iterations**:
     $$\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} < \varepsilon$$

3. State the necessary and sufficient condition for convergence.

   A **consistent** iterative method with iteration matrix B converges if and only if

   $$\rho(B) < 1$$

where $\rho(B)$ is the spectral radius of B: $\rho(B) = \max\{|\lambda| : \lambda \in \lambda(A)\}$ is the maximum in modulus of all the eigenvalues of B.

4. State and prove the sufficient condition for convergence.

An iterative method converges for any initial choice of $\mathbf{x}^{(0)}$ if:

$$\|B\| < 1$$

where $\|B\|$ is the euclidian norm.

Proof. let us consider the the error at step $(k+1)$: $\mathbf{e}^{(k+1)} = \mathbf{x} - \mathbf{x}^{(k+1)}$, and a suitable vector norm $|\cdot\|$ (will use the euclidian norm as before). We have that

$$\begin{aligned}
\|e^{(k+1)}\| &= \|\mathbf{x} - \mathbf{x}^{(k+1)}\| \\
&= \|\mathbf{x} - B\mathbf{x}^{(k)} - \mathbf{f}\| \\
\text{consistency } \implies &= \|\mathbf{x} - B\mathbf{x}^{(k)} - (I - B)\mathbf{x}\| \\
&= \|B\mathbf{e}^{(k)}\| \\
&\leq \|B\|\|\mathbf{e}^{(k)}\| \\
\text{by recursion: } &\leq \|B\|\|B\|\|\mathbf{e}^{(k-1)}\| \leq \cdots \leq \\
&\leq \|B\|^{k+1}\|\mathbf{e}^{(0)}\|
\end{aligned}$$

If $|B\| < 1$, then $\lim_{k\to+\infty} \|B\|^{k+1} = 0$. Therefore,

$$\lim_{k\to+\infty} \|\mathbf{e}^{(k+1)}\| = \lim_{k\to\infty} |\mathbf{x} - \mathbf{x}^{(k+1)}\| = 0$$

$$\implies \lim_{k\to+\infty} \mathbf{x}^{(k+1)} = \mathbf{x} \qquad \text{(convergence condition)}$$

□

5. Describe the Generalized Minimal Residual Method (GMRES). Recall the interpretation of the scheme as a Krylov subspace method and the main theoretical results.

Given an initial guess $\mathbf{x}^{(0)}$ and an initial residual $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$, the GMRES looks for a solution inside the affine space

$$\mathbf{x}^{(k)} \in \mathbf{x}^{(0)} + \mathcal{K}_k(A, \mathbf{r}^{(0)})$$

where $\mathcal{K}_k(A, \mathbf{r}^{(0)}) = \text{span}\{\mathbf{r}^{(0)}, A\mathbf{r}^{(0)}, A^2\mathbf{r}^{(0)}, \ldots, A^{k-1}\mathbf{r}^{(0)}\}$ is the k-th Krylov space.

GMRES minimizes the Euclidian norm of the residual:

$$\mathbf{x}^{(k)} = \arg \min_{\mathbf{x}^* \in \mathbf{x}^{(0)} + \mathcal{K}_k} \|\mathbf{b} - A\mathbf{x}^*\|_2$$

GMRES converges after n iterations (the size of A), because the Krylov subspace will span the whole $\mathbb{R}^n$.

For convergence, we have two cases:

- If $A_S = (A + A^T)/2$ is symmetric positive definite (SPD), then

$$|\mathbf{r}^{(k)}\|_2 \leq \left[1 - \frac{\lambda_{min}^2(A_s)}{\lambda_{max}(A^T A)}\right]^{\frac{k}{2}} \|\mathbf{r}^{(0)}\|_2$$

- If A is SPD, then

$$|r^{(k)}\|_2 \leq \left[\frac{[K_2(A)]^2 - 1}{[K_2(A)]^2}\right]^{\frac{k}{2}} |\mathbf{r}^{(0)}\|_2$$

where:

- $\lambda_{min}(\cdot)$ and $\lambda_{max}(\cdot)$ are respectively the minimum and maximum eigenvalue;
- $K_2(\cdot)$ is the condition number calculated using the euclidian norm.

5 Consider the problem: find $\mathbf{x} \in \mathbb{R}^n$ such that $A\mathbf{x} = \mathbf{b}$, where $A \in \mathbb{R}^{n\times n}$ and $\mathbf{b} \in \mathbb{R}^n$ are given.

1. Describe the LU factorization of A. Introduce the employed notation.

The LU factorization decomposes the matrix A into

$$A = LU$$

where

- $L \in \mathbb{R}^{n\times n}$ is a lower triangular matrix;
- $U \in \mathbb{R}^{n\times n}$ is an upper triangular matrix.

2. State the necessary and sufficient conditions for the existence and uniqueness of the LU factorization.

We can use the following theorem.

If A is invertible, then the LU factorization of A exists and is unique if and only if all its leading principal minors are nonzero, that is:

$$\det(A_k) \neq 0 \qquad \forall k = 1, \ldots, n-1$$

where A_k is the $k \times k$ submatrix of A consisting of the first k rows and k columns of A.

Note that if A is not invertible we need to use **pivoting** introducing a permutation matrix P. The factorization then becomes $PA = LU$.

3. Describe how the LU factorization is used to compute the approximate solution of $A\mathbf{x} = \mathbf{b}$.

Assume A is invertible. After we have the factorization $A = LU$, we can:

(a) Solve $L\mathbf{y} = \mathbf{b}$ (lower triangular system, use forward substitution);

(b) Solve $U\mathbf{x} = \mathbf{y}$ to find the final solution $\mathbf{x}$ (upper triangular system, use backwards substitution)

4. State the computational costs and prove the results.

Consider the gaussian elimination algorithm:

Algorithm 3 Gaussian elimination

for $k = 1, \dots, n-1$ **do**
  for $i = k+1, \dots, n$ **do**
    $l_{ik} = \dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$ ▷ Compute the multiplier
    for $j = k+1, \dots, n$ **do**
      $a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik} a_{kj}^{(k)}$ ▷ Update the matrix entries
    end for
    $\mathbf{b}_i^{(k+1)} = \mathbf{b}_i^{(k)} - l_{ik} b_k^{(k)}$ ▷ Update the right-hand size
  end for
end for
After $n-1$ steps, we will have: $a_{ij}^{(n)} = U, \quad l_{ij} = L, \quad \mathbf{b}^{(n)} = \mathbf{y}$

Proof. As we can see from the code, we iterate k form 1 to $n-1$. At each step k, we are

- computing the multiplier: requires $(n-k)$ divisions
- updating the matrix entries: the matrix size is $(n-k)\times(n-k)$, and each element requires 1 multiplication and 1 subtraction (2 flops). Therefore, the total cost to update is $2(n-k)^2$.

Therefore, the total computational cost C_{LU} is:

$$\implies C_{LU} = \sum_{k=1}^{n-1} [(n-k) + 2(n-k)^2] = O(n^3)$$

□

5. Describe how the incomplete LU factorization can be used as preconditioner to accelerate the convergence of a linear iterative method. Introduce the employed notation.

When A is sparse, computing the LU factorization is computationally expensive and causes "fill-in". We can instead use the incomplete LU factorization: take $P = \tilde{L}\tilde{U}$, where $\tilde{L}$ and $\tilde{U}$ are the incomplete LU factors, such that:

$$A \approx \tilde{L}\tilde{U}, \text{ with } \tilde{\ell}_{ij} = 0, \tilde{u}_{ij} = 0 \text{ if } \tilde{a}_{ij} = 0$$

$\tilde{L}, \tilde{U}$ have the same sparsity pattern as A and so memory occupation is the same.

The incomplete LU factorization improves the condition number, accelerating convergence.

6. Describe the main pivoting techniques and comment on the computational costs.