Numerical Linear Algebra

Rao
Politecnico di Milano

Originally written in: **2024-09-16**

Last updated at: **2024-11-09**

Contents

1 Matrix Decompositions and Factorization

1.1 QR Factorization

Let $A \in \mathbb{R}(m \times n)$ be a rectangular matrix, then

$$A = QR \tag{1.1}$$

where $Q \in \mathbb{R}(m \times m)$ is an orthogonal matrix and $R \in \mathbb{R}(m \times n)$ is an upper trapezoidal matrix.

One version of the QR factorization is *reduced QR factorization*. Let A be an $m \times n$ matrix. The reduced QR factorization of A is a factorization of the form:

$$A = \hat{Q}\hat{R} \tag{1.2}$$

where $Q \in \mathbb{R}(m \times n)$ is an rectangular matrix and $R \in \mathbb{R}(n \times n)$ is an upper triangular matrix.

1.2 Cholesky Factorization

Let $A \in \mathbb{R}^{n \times n}$ be a *symmetric and positive definite* (SPD) matrix. Then, there exists a unique upper triangular matrix $R \in \mathbb{R}^{n \times n}$ with positive diagonal entries such that:

$$A = R^T R \tag{1.3}$$

This factorization is called *Cholesky factorization*.

CHOLESKY FACTORIZATION

Let $r_{11} = \sqrt{a_{11}}$.

For $k = 2, ...$

$$r_{ij} = \frac{1}{r_{ii}}\left(a_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj}\right)$$
$$r_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} r_{ki}^2}$$

The computational cost of the Cholesky factorization is $O(n^3/3)$.

1.3 Schur Decomposition

If $A \in \mathbb{C}^{n \times n}$ then there is a unitary matrix $U \in \mathbb{C}^{n \times n}$ such that:

$$U^H A U = T \tag{1.4}$$

where T is an upper triangular matrix. The diagonal elements of T are the eigenvalues of A. $U = [\boldsymbol{u}_1, ..., \boldsymbol{u}_n]$ are called Schur vectors. They are generally not the eigenvectors of A.

2 Norms

The essential notions of **size and distance** in a vector space are captured by norms. These are the *yardsticks* with which we measure approximations and convergence throughout numerical linear algebra.

2.1 Vector Norms

A norm is a function $\|\cdot\| : \mathbb{C}^m \to \mathbb{R}$ that assigns a real-valued length to each vector. In order to conform to a reasonable notion of length, a norm must satisfy the following three conditions. For all vectors x, y and for all scalars $\alpha \in \mathbb{C}$:

1. *Nonnegativity*: $\|\boldsymbol{x}\| \geq 0$ and $\|\boldsymbol{x}\| = 0$ if and only if $x = 0$.
2. *Triangle Inequality*: $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$.
3. *Homogeneity*: $\|\boldsymbol{\alpha x}\| = |\alpha| \, \|\boldsymbol{x}\|$.

The above conditions allow for different notions of lenght, and at times it is useful to have the flexibility.

$$\begin{aligned} \|\boldsymbol{x}\|_1 &= \sum_{i=1}^{m} |x_i| \\ \|\boldsymbol{x}\|_2 &= \sqrt{\sum_{i=1}^{m} |x_i|^2} \\ \|\boldsymbol{x}\|_\infty &= \max_{\{1<i<m\}} |x_i| \\ \|\boldsymbol{x}\|_p &= \left(\sum_{i=1}^{m} |x_i|^p \right)^{\frac{1}{p}} (p \leq 1 < \infty) \end{aligned} \tag{2.1}$$

Aside from the $p-$norms, the most useful norms are the *weighted p norms*, where each of the coordinates of a vector space is given its own weight. In general, given any norm $\|\cdot\|$, the *weighted p norm* is defined as:

$$\|\boldsymbol{x}\|_w = \|\boldsymbol{W x}\| \tag{2.2}$$

Here W is the diagonal matrix with the ith diagonal entry is the weight $w_i \neq 0$. For example. a weighted 2-norm is specified as follows:

$$\|\boldsymbol{x}\|_W = \left(\sum_{i=1}^{m} |w_i x_i|^2 \right)^{\frac{1}{2}} \tag{2.3}$$

Thm Cauchy-Schwarz Inequality **theorem 2.1.1**

For any two vectors $x, y \in \mathbb{R}^n$, the following inequality holds:

$$|(x, y)| = |x^T y| \leq \|\boldsymbol{x}\|_2 \|\boldsymbol{y}\|_2 \tag{2.4}$$

Where strict equality holds if and only if x and y are linearly dependent.

We recall that the scalr product in $\mathbb{R}^n$ can be realyed to the p-norms by the H"older inequality:

$$|(x, y)| \leq \|\boldsymbol{x}\|_p \|\boldsymbol{y}\|_q \qquad \text{Where} \qquad \frac{1}{p} + \frac{1}{q} = 1 \tag{2.5}$$

Thm Norm continuity **theorem 2.1.2**

Any vector norm $\|\cdot\|$ defined on V is a continous function of its argument, namely, $\forall > 0, \exists C > 0$ such that if $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\| \leq \varepsilon$ then $\|\boldsymbol{x}\| - \|\hat{\boldsymbol{x}}\| \leq C\varepsilon$, for any $x, \hat{x} \in V$.

Thm **theorem 2.1.3**

let $\|\cdot\|$ be a norm of $\mathbb{R}^n$ and $A \in \mathbb{R}^{n\times n}$ be a matrix with n linearly independent columns. Then, the function $\|\cdot\|_{A^2}$ acting from $\mathbb{R}^n$ in to $\mathbb{R}$ defined as:

$$\|\boldsymbol{x}\|_{A^2} = \|\boldsymbol{A}\boldsymbol{x}\| \tag{2.6}$$

is a norm on $\mathbb{R}^n$.

Thm Energy Norm **theorem 2.1.4**

Let $A \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix. Then, the *energy norm* is defined as:

$$\|\boldsymbol{x}\|_A = \sqrt{\boldsymbol{x}^T A \boldsymbol{x}} \tag{2.7}$$

Thm Convergence **theorem 2.1.5**

Let $\|\cdot\|$ be a norm in a finite dimensional space V. Then:

$$\lim_{k\to\infty} x^{(k)} = x \iff \lim_{k\to\infty} \|\boldsymbol{x}^{(\boldsymbol{k})} - \boldsymbol{x}\| = 0 \tag{2.8}$$

where $x \in V$ and $x^{(k)}$ is a sequence of vectors in V.

2.2 Matrix Norms

In dealing with a space of matrices, certain special norms are more useful thant the vector norms. These are the *induced matrix norms*, defined in terms of the behavior of a matrix as an operator between its normed domain and range spaces.

Def Matrix Norm **definition 2.2.1**

A *matrix norm* is a mapping $\|\cdot\| : \mathbb{R}^{m\times n} \to \mathbb{R}$ such that:

1. $\|\boldsymbol{A}\| \geq 0 \forall A \in \mathbb{R}^{m\times n}$ and $\|\boldsymbol{A}\| = 0$ if and only if $A = 0$.
2. $\|\boldsymbol{\alpha}\boldsymbol{A}\| = |\alpha|\ \|\boldsymbol{A}\|\ \forall A \in \mathbb{R}^{m\times n}$ and $\alpha \in \mathbb{C}$.
3. $\|\boldsymbol{A} + \boldsymbol{B}\| \leq \|\boldsymbol{A}\| + \|\boldsymbol{B}\|\ \forall A, B \in \mathbb{R}^{m\times n}$(triangular inequality)

Def **definition 2.2.2**

We say that a matrix norm $\|\cdot\|$ is *compatible* or *consistent* with a vector norm $\|\cdot\|$ if:

$$\|\boldsymbol{Ax}\| \leq \|\boldsymbol{A}\| \, \|\boldsymbol{x}\| \qquad \forall x \in \mathbb{R}^n \tag{2.9}$$

More generally, given three norms, all denoted by $\|\cdot\|$, albeit defined on $\mathbb{R}^m, \mathbb{R}^n . \mathbb{R}^{m\times n}$, respectively, we say that they are consistent if if $\forall x \in \mathbb{R}^n, Ax = y \in \mathbb{R}^m$, we have that $\|\boldsymbol{y}\| \leq \|\boldsymbol{A}\| \, \|\boldsymbol{x}\|$.

Def **Sub multiplicative** **definition 2.2.3**

We say that a matrix norm $\|\cdot\|$ is *sub-multiplicative* if $\forall A \in \mathbb{R}^{n\times m}, \forall B \in \mathbb{R}^{m\times q}$ we have that

$$\|\boldsymbol{AB}\| \leq \|\boldsymbol{A}\| \, \|\boldsymbol{B}\| \tag{2.10}$$

Def **Fronebius Norm** **definition 2.2.4**

The norm

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{\mathrm{tr}(AA^H)} \tag{2.11}$$

is a matrix norm called the *Frobenius norm*. And it is compatible with the Euclidean vector norm $\|\cdot\|_2$. Indeed.

$$\|\boldsymbol{Ax}\|_2^2 = \sum_{i=1}^{m}|\sum_{j=1}^{n}a_{ij}x_j|^2 \leq \sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2\sum_{j=1}^{n}|x_j|^2 = \|\boldsymbol{A}\|_F^2\|\boldsymbol{x}\|_2^2 \tag{2.12}$$

Thm Induced Matrix Norm **theorem 2.2.1**

Let $\|\cdot\|$ be a vector norm. The function:

$$\|\boldsymbol{A}\| = \sup_{x\neq 0} \frac{\|\boldsymbol{Ax}\|}{\|\boldsymbol{x}\|} \tag{2.13}$$

is a matrix norm called *induced matrix norm or natural matrix norm.*

Proof: Check definition 2.2.1.

1. If $\|\boldsymbol{Ax}\| \geq 0$, then it follows that $\|\boldsymbol{A}\| = \sup_{\|\boldsymbol{x}\|=1}\|\boldsymbol{Ax}\| \geq 0$. Moreover,

$$\|\boldsymbol{A}\| = \sup_{x\neq 0} \frac{\|\boldsymbol{Ax}\|}{\|\boldsymbol{x}\|} = 0 \iff \|\boldsymbol{Ax}\| = 0 \forall x \neq 0 \tag{2.14}$$

and $Ax = 0 \forall x \neq 0$ if and only if $A = 0$; therefore, $\|\boldsymbol{A}\| = 0$ if and only if $A = 0$.

2. Given a scalar α, we have that:

$$\|\boldsymbol{\alpha A}\| = \sup_{x\neq 0} \frac{\|\boldsymbol{\alpha Ax}\|}{\|\boldsymbol{x}\|} = \sup_{x\neq 0} |\alpha| \frac{\|\boldsymbol{Ax}\|}{\|\boldsymbol{x}\|} = |\alpha| \sup_{x\neq 0} \frac{\|\boldsymbol{Ax}\|}{\|\boldsymbol{x}\|} = |\alpha| \ \|\boldsymbol{A}\| \tag{2.15}$$

3. Finally, triangular inequality holds. Indeed, by definition of supremum, if $x \neq 0$ then:

$$\frac{\|\boldsymbol{Ax}\|}{\|\boldsymbol{x}\|} \leq \|\boldsymbol{A}\| \Rightarrow \|\boldsymbol{Ax}\| \leq \|\boldsymbol{A}\| \ \|\boldsymbol{x}\| \tag{2.16}$$

So that, taking x with unit norm, one gets:

$$\|(\boldsymbol{A}+\boldsymbol{B})\boldsymbol{x}\| \leq \|\boldsymbol{Ax}\| + \|\boldsymbol{Bx}\| \leq \|\boldsymbol{A}\| + \|\boldsymbol{B}\| \tag{2.17}$$

from which it follows that $\|\boldsymbol{A}+\boldsymbol{B}\| = \sup_{\|\boldsymbol{x}\|=1}\|(\boldsymbol{A}+\boldsymbol{B})\boldsymbol{x}\| \leq \|\boldsymbol{A}\| + \|\boldsymbol{B}\|$.

Relevant instances of induced matrix norms are the so-called *p-norms*:

$$\|\boldsymbol{A}\|_p = \sup_{\boldsymbol{x}\neq 0} \frac{\|\boldsymbol{Ax}\|_p}{\|\boldsymbol{x}\|_p} \tag{2.18}$$

The 1-norm(column sum norn):

$$\|\boldsymbol{A}\|_1 = \max_{j=1,\dots,n}^{n} \sum_{i=1}^{m} |a_{ij}| \tag{2.19}$$

The infinity-norm(row sum norm):

$$\|\boldsymbol{A}\|_\infty = \max_{i=1,\dots,m}^{m} \sum_{j=1}^{n} |a_{ij}| \tag{2.20}$$

Moreover, we have $\|\boldsymbol{A}\|_1 = \|\boldsymbol{A^T}\|_\infty$ and, if A is self-adjoint or real sysmetric, then $\|\boldsymbol{A}\|_1 = \|\boldsymbol{A}\|_\infty$.

A special discussion is deserved by the *2-norm* or *spectral norm* for which the following theorem holds.

Thm Spectral Norm **theorem 2.2.2**

Let $\sigma_1(A)$ be the largest singular value of A. Then, the 2-norm of A is given by:

$$\|\boldsymbol{A}\|_2 = \sqrt{\rho(A^H A)} = \sqrt{\rho(A^H A)} == \sigma_1(A) \tag{2.21}$$

In particular, if A is hermitian (or real and symmetric), then $\|\boldsymbol{A}\|_2 = \rho(A)$.

Proof: Since $A^T A$ is hermitian, there exists a unitary matrix U such that

$$U^H A^H A U = \text{diag}(\mu_1, ..., \mu_n) \tag{2.22}$$

where μ_i are the positive eigenvalues of $A^H A$. Let $y = U^H x$, then:

$$\begin{aligned} \|\boldsymbol{A_2}\| &= \sup_{x \neq 0} \frac{\sqrt{(A^H A x, x)}}{\sqrt{(x, x)}} = \sup_{y \neq 0} \frac{\sqrt{(U^H A^H A U y, y)}}{\sqrt{(y, y)}} \\ &= \sup_{y \neq 0} \frac{\sqrt{\sum_{i=1}^n \mu_i \ |y_i|^2}}{\sqrt{\sum_{i=1}^n |y_i|^2}} = \sqrt{\max_{i=1,...,n} |\mu_i|} \end{aligned} \tag{2.23}$$

If A is hermitian, the same considerations as above apply directly to A. Finally, if A is unitary, we have

$$\|\boldsymbol{A}\|_2 = \sup_{x \neq 0} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} = \sup_{x \neq 0} \frac{\|\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} = 1 \tag{2.24}$$

2.3 Sequences and Series of Matrices

A sequence of matrices A^k is said to *converge* to a matrix $A \in \mathbb{R}^{n \times n}$ if

$$\lim_{k \to \infty} \|\boldsymbol{A^k} - \boldsymbol{A}\| = 0 \tag{2.25}$$

The choice of the norm does not influence the result since in $\mathbb{R}^{n \times n}$ all norms are equivalent.

Thm Convergence of Sequences of Matrices **theorem 2.3.1**

Let A be a square matrix; then

$$\lim_{k \to \infty} A^k = 0 \Leftrightarrow \rho(A) < 1 \tag{2.26}$$

3 Principles of Numerical Mathematics

3.1 Well-posedness and Condition Number

Consider the following problem: find x such that:

$$F(x, d) = 0 \tag{3.1}$$

where F is a function of x and d. And three types of problems can be considered:

1. *direct problem*: given F and d, find x;
2. *inverse problem*: given F and x, find d;
3. *identification problem*: given x and d, find F.

Problems Eq. (3.1) are ***well-posed*** if it admits a *unique* solution, and the solution depends continuously on the data.

A problem which does not enjoy the property above is called ill posed or unstable and before undertaking its numerical solution it has to be regularized, that is, it must be suitably transformed into a well-posed problem.

Let D be the set of admissible data, i.e. the set of the values of d in correspondance of which problem Eq. (3.1) admits a unique solution. Continuous dependence on the data means that small perturbations on the data d of D yield "small" changes in the solution x.

Precisely, let $d \in D$ and denoted by δd a perturbation admissible in the sense that $d + \delta d \in D$ and by δx the corresponding change in the solution, in such a way that:

$$F(x + \delta x, d + \delta d) = 0 \tag{3.2}$$

Then, we require that:

$$\begin{gathered} \exists \eta_0 = \eta_0(d) > 0, \exists K_0 = K_0(d) \text{ such that} \\ \text{if } \|\boldsymbol{\delta d}\| \leqslant \eta_0 \text{ then } \|\boldsymbol{\delta x}\| \leqslant K_0 \, \|\boldsymbol{\delta d}\| \end{gathered} \tag{3.3}$$

The norms used for the data and for the solution may not coincide, whenever d and x represent variables of different kinds.

e.g. Wellposedness of Linear Systems **example 3.1.1**

Consider the problem of solving a linear system $Ax = b$. The problem is well-posed if it has below two properties:

1. The problem has a unique solution x, which means that the matrix A is invertible.
2. The solution depends continously on the data.

The Eq. (3.3) is however more suitable to express in the following the concept of *numerical stability*, that is, the property that small perturbations on the data yield perturbations of the same order on the solution.

3.1.1 Absolute Condition Number

Let δx denote a small perturbation of x, and wirte $\delta f = f(x + \delta x, d) - f(x, d)$. The absolute condition number is then defined as:

$$K_{\text{abs}} = \lim_{\delta \to 0} \sup_{\|\boldsymbol{\delta x}\| \leq \delta} \frac{\|\boldsymbol{\delta f}\|}{\|\boldsymbol{\delta x}\|} \tag{3.4}$$

For most problems, the limit of the supremum in this formula can be interpreted as a supremum over all infinitesimal perturbations δx, and in the interest of readability, we shall generally write the formula simply as

$$K = \sup_{\delta x} \frac{\|\boldsymbol{\delta f}\|}{\|\boldsymbol{\delta x}\|} \tag{3.5}$$

with the understanding that δx and δf are infinitesimal.

If f is differentiable, we can evaluate the absolute condition number by means of the derivative of f. Let $J(x)$ be the matrix whose i, j entry is the partial derivative $\partial f_i / \partial x_j$, evaluated at x. The definition of derivative gives us, $\delta f \approx J(x)\delta x$, with equality in the limit $\|\boldsymbol{\delta x}\| \to 0$. The absolute condition number is then:

$$K = \|\boldsymbol{J}(\boldsymbol{x})\| \tag{3.6}$$

3.1.2 Relative Condition Number

When we are concerned with relative changes, we need the notion of relative condition. The *relative condition number* is defined as:

$$K = \lim_{\delta \to 0} \sup_{\|\boldsymbol{\delta x}\| \leq \delta} \left(\frac{\|\boldsymbol{\delta f}\|}{\|\boldsymbol{f}(\boldsymbol{x})\|} / \frac{\|\boldsymbol{\delta x}\|}{\|\boldsymbol{x}\|} \right) \tag{3.7}$$

or, assuming δx and δf are infinitesimal,

$$K = \sup_{\delta x} \frac{\frac{\|\boldsymbol{\delta f}\|}{\|\boldsymbol{f}(\boldsymbol{x})\|}}{\frac{\|\boldsymbol{\delta x}\|}{\|\boldsymbol{x}\|}} \tag{3.8}$$

If f is differentiable, we can express this equality in terms of the Jacobian matrix $J(x)$, as follows:

$$K = \frac{\|\boldsymbol{J}(\boldsymbol{x})\|}{\|\boldsymbol{f}\boldsymbol{x}\| \ / \ \|\boldsymbol{x}\|} \tag{3.9}$$

Problem Eq. (3.1) is called *ill-conditioned* if $K(d)$ is "big" for any admissible datum d (the precise meaning of "small" and "big" is going to change depending on the considered problem).

3.1.3 Codution of Matrix-Vector Multiplication

Now we come to one of the condition numbers of fundamental importance in numerical linear algebra.

Fix $A \in \mathbb{C}^{m \times n}$ and consider the problem of computing Ax from input x; that is, we are going to determine a condition number corresponding to perturbations of x but not A. Working directly

from the definition of K, with $\|\cdot\|$ denoting an arbitrary vector norm and the corresponding induced matrix norm, we have:

$$K = \sup_{\delta x}\left(\frac{\|\boldsymbol{A}(\boldsymbol{x}+\boldsymbol{\delta x})-\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{A}\boldsymbol{x}\|}\right)/\left(\frac{\|\boldsymbol{\delta x}\|}{\|\boldsymbol{x}\|}\right) = \sup_{\delta x}\frac{\|\boldsymbol{A}\boldsymbol{\delta x}\|}{\|\boldsymbol{\delta x}\|}/\frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \tag{3.10}$$

that is,

$$K = \|\boldsymbol{A}\|\,\frac{\|\boldsymbol{x}\|}{\|\boldsymbol{A}\boldsymbol{x}\|} \tag{3.11}$$

This is an exact formula for K, dependent on both A and x.

Suppose A is square and nonsingular. Then we can use the fact that $\|\boldsymbol{x}\| \,/\, \|\boldsymbol{A}\boldsymbol{x}\| \le \|\boldsymbol{A}^{-1}\|$ to loosen Eq. (3.11) to a bound independent of x:

$$K \le \|\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\| \tag{3.12}$$

Or, one might write this as:

$$k = \alpha\,\|\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\| \tag{3.13}$$

with

$$\alpha = \frac{\|\boldsymbol{x}\|}{\|\boldsymbol{A}\boldsymbol{x}\|}/\,\|\boldsymbol{A}^{-1}\| \tag{3.14}$$

If $\|\cdot\| = \|\cdot\|_2$ this will occur whenever x is a multiple of a minimal right singular vector of A.

3.1.4 Condition number of a Matrix

The product $\|\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\|$ comes up so often that it has its own name: it is the *condition number* of A:

$$K(A) = \|\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\| \tag{3.15}$$

Thus, in this case the term *condition number* is attached to a matrix, not a problem. If $K(A)$ is small, A is said to be *well-conditioned*; if it is large, A is *ill-conditioned*. If A is singular, it is customary to write $K(A) = \infty$.

Note that if $\|\cdot\| = \|\cdot\|^2$, then $\|\boldsymbol{A}\| = \sigma_1$ and $A^{-1} = \frac{1}{\sigma_n}$, where σ_1 and λ_n are the maximum abnd minimum singular value of A. Thus

$$K(A) = \frac{\sigma_1}{\sigma_n} \tag{3.16}$$

In the 2-norm, and it is this formula that is generally used for computing 2-norm condition numbers of matrices.

For a rectangular matrix $A \in \mathbb{C}^{m\times n}$ of full rank, $m \ge n$, the condition numver is defined in terms of the **pseudoinverse**: $K(A) = \|\boldsymbol{A}\|\,\|\boldsymbol{A}^{+}\|$. Since A^+ is motivated by least squares problems, this definition is most useful in the case $\|\cdot\| = \|\cdot\|^2$, where we have

$$K(A) = \frac{\sigma_1}{\sigma_n} \tag{3.17}$$

3.1.5 Condition Number of a System of Equations

Specifically, let us hold b fixed and consider the behavior of the problem $A \to x = A^{-1}b$ when A is perturbed by infinitesimal δA. Then x must change by infinitesimal δx such that:

$$(A + \delta A)(x + \delta x) = 0 \tag{3.18}$$

Using the equality $Ax = b$ and dropping the doubly infinitesimal term $(\delta A)(\delta x)$, we obtain $(\sigma A)x + A(\sigma)x = 0$. that is, $\sigma x = -A^{-1}(\sigma A)x$. This equation implies $\|\boldsymbol{\sigma x}\| \leq \|\boldsymbol{A^{-1}}\| \, \|\boldsymbol{\sigma A}\| \, \|\boldsymbol{x}\|$, or equivalently:

$$\frac{\sigma x}{\|\boldsymbol{x}\|} / \frac{\sigma A}{\|\boldsymbol{A}\|} \leq \|\boldsymbol{A^{-1}}\| \, \|\boldsymbol{A}\| = K(A) \tag{3.19}$$

Thm **theorem 3.1.1**

Let b be fixed and consider the problem of computing $x = A^{-1}b$, where A is square and nonsingular. The condition number of this problem with respect to perturbations in A is

$$K(A) = \|\boldsymbol{A}\| \, \|\boldsymbol{A^{-1}}\| \tag{3.20}$$

3.2 Stability of Numerical Methods

We shall henceforth suppose the problem Eq. (3.1) to be well-posed and a numerical method for the approximate solution of Eq. (3.1) will consist, in general, of a sequence of approximate problems:

$$F(x_n, d_n) = 0 \tag{3.21}$$

depending on a certain parameter n (to be defined case by case). The undetstood expectation is that $x_n \to x$ as $n \to \infty$, that is, the sequence of approximate solutions **converges** to the exact solution.

For that, it is necessary that $d_n \to d$ and $F_n \to F$, as $n \to \infty$. Precisely, if the datum d of Eq. (3.1) is admissible for F_n, we say that Eq. (3.21) is consistent if:

$$F_{n(x,d)} = F_{n(x,d)} - F(x, d) \to 0 \text{ for } n \to \infty \tag{3.22}$$

3.2.1 Relations between Stability and Covergence

The concepts of stability and convergence are strongly connected.

Thm **theorem 3.2.1**

If problem Eq. (3.1) is well-posed, a *necessary* condition in order for the numerical problem Eq. (3.21) to be convergent is that it is stable.

4 Sparse matrices

4.1 Sparse matrices storage formats

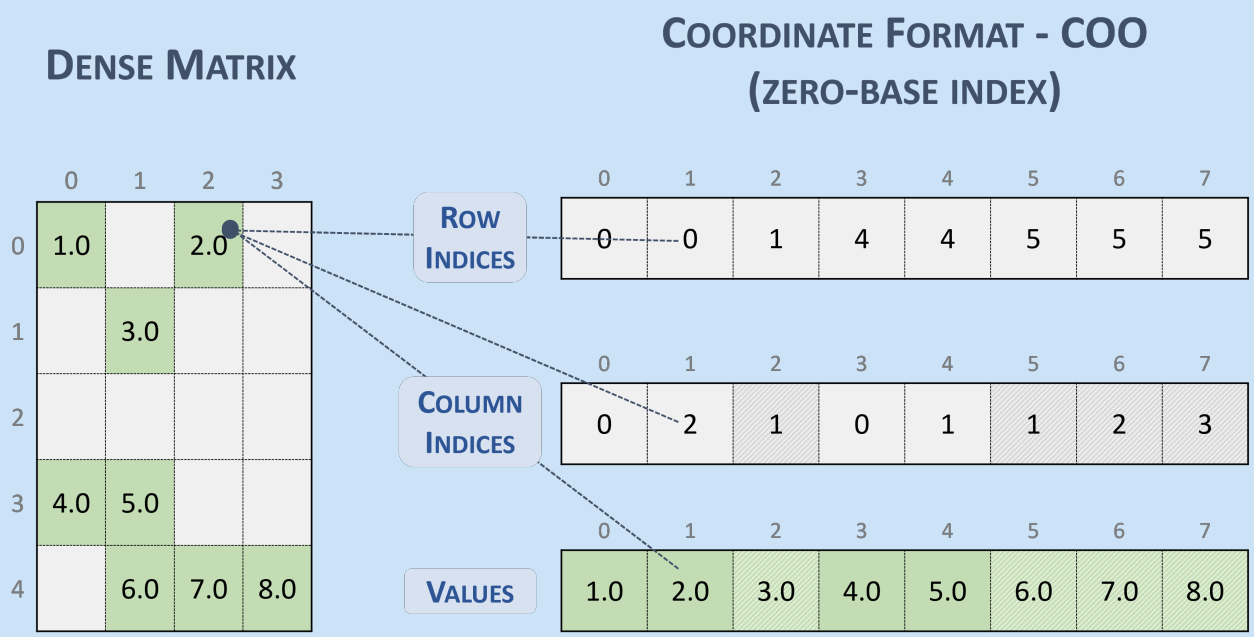
Sparse matrices are matrices that contain a large number of zero elements. The storage of these matrices can be optimized by using different formats. The most common formats are:

4.1.1 Coordinate format (COO)

The simplest storage scheme for sparse matrices is the so-called coordinate format. The data structure consists of three arrays:

1. `AA` - all the values of the nonzero elements of A in any order.
2. `JR` - the row indices of the nonzero elements of A.
3. `JC` - the column indices of the nonzero elements of A.

e.g. Coordinate format **example 4.1.1**



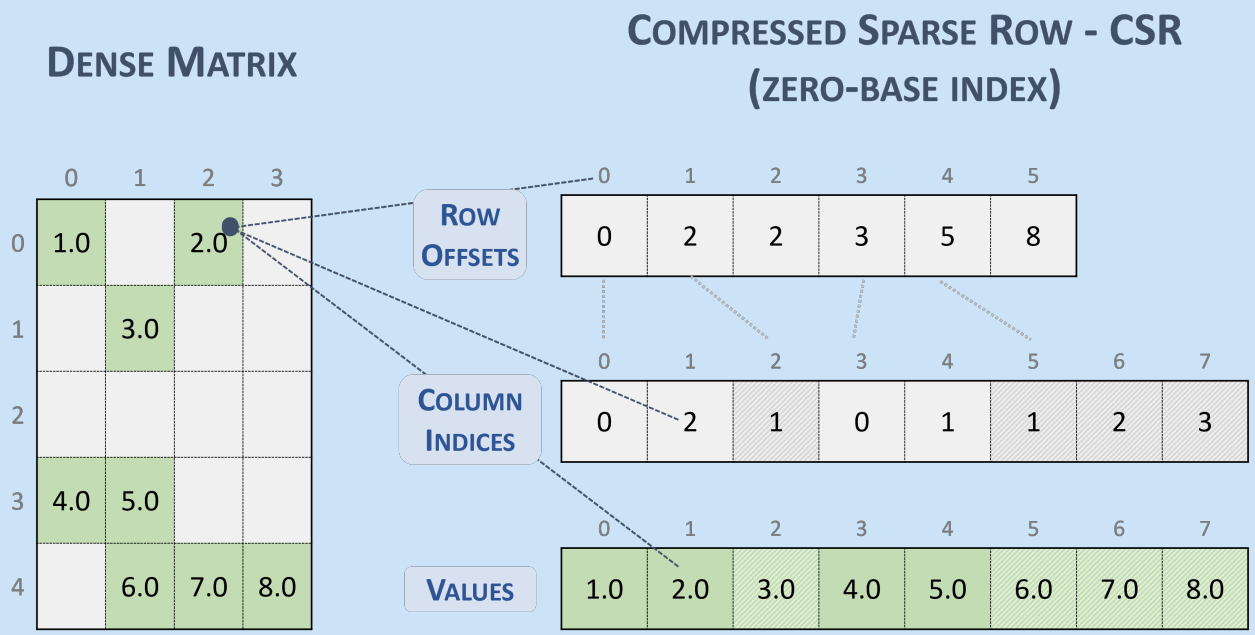
4.1.2 Compressed sparse row (CSR)

The CSR format is similar to COO, where the row indices are compressed and replaced by an array of offsets. The new data structure consists of three arrays:

1. `AA` - the real values a_{ij} sorted row by row, from row 1 to row n.
2. `JA` - the column indices of the nonzero elements of A.
3. `IA` - the row offsets. contains the pointers to the beginning of each row in the array A and JA. The content of IA is the position in the arrays AA and JA where the row i

starts. The length of IA is $n+1$, with $IA(n+1)$ containing the total number of nonzero elements in the matrix.

e.g. Compressed sparse row format **example 4.1.2**



To create a sparse matrix in the CSR format, we use the `csr_matrix` function, which is provided by the `scipy.sparse` module. Here is an example program:

```
import scipy.sparse as sp
from scipy import *

data = [1.0, 2.0, -1.0, 6.6, 1.4]
rows = [0, 1, 1, 3, 3]
cols = [1, 1, 2, 0, 4]

A = sp.csr_matrix((data, [rows, cols]), shape=(4, 5))
print(A)

>>> A.data
array([ 1. ,  2. , -1. ,  6.6,  1.4])
>>> A.indices
array([1, 1, 2, 0, 4], dtype=int32)
>>> A.indptr
array([0, 1, 3, 3, 5], dtype=int32)
```

5 Iterative methods for large linear systems

Given an $n \times n$ real matrix A and a real n-vector, the problem is: Find $\boldsymbol{x}$ belonging to $\mathbb{R}^n$ such that

$$A\boldsymbol{x} = \boldsymbol{b} \tag{5.1}$$

where $\boldsymbol{x}$ is the exact solution of the linear system $A\boldsymbol{x} = \boldsymbol{b}$. In such cases existence and uniqueness of the solution are ensured if one of the following (equivalent) hypotheses holds:

1. A is invertible
2. rank(A)=n;
3. the homogeneous system $A\boldsymbol{x} = 0$ admits only the null solution.

5.1 On the Convergence of Iterative Methods

The basic idea of iterative methods is to construct a sequence of vectors $\boldsymbol{x^k}$ that enjoy the property of *convergence*

$$\boldsymbol{x} = \lim_{k\to\infty} \boldsymbol{x^k} \tag{5.2}$$

In practice, the iterative process is stopped at the minimum value of n such that $\|\boldsymbol{x^{(n)}} - \boldsymbol{x}\| < \varepsilon$, where ε is a given tolerance and $\|\cdot\|$ is a suitable norm. However, since the exact solution is obviously not available, it is necessary to introduce suitable stopping criteria to monitor the convergence of the iteration.

To start with, we consider iterative methods of the form

$$\text{Given } \boldsymbol{x}^0, \boldsymbol{x}^{k+1} = B\boldsymbol{x^k} + \boldsymbol{f}, k \geq 0 \tag{5.3}$$

where B is an $n \times n$ square matrix called the *iteration matrix* and $\boldsymbol{f}$ is a vector that is obtained from the right-hand side $\boldsymbol{b}$.

having denoted by B an $n \times n$ square matrix called the iteration matrix and by $\boldsymbol{f}$ a vector that is obtained from the right hand side $\boldsymbol{b}$.

Def Consistent **definition 5.1.1**

An iterative method of the form Eq. (5.3) is said to be *consistent* with Eq. (5.1) if $\boldsymbol{f}$ and B are such that $\boldsymbol{x} = \boldsymbol{B}\boldsymbol{x} + \boldsymbol{f}$. Equivalently,

$$\boldsymbol{f} = (1 - B)A^{-1}\boldsymbol{b} \tag{5.4}$$

Having denoted by

$$\boldsymbol{e}^{(k)} = \boldsymbol{x}^{(k)} - \boldsymbol{x} \tag{5.5}$$

the error at the k-th step of the iteration, the condition for convergence amounts to requiring that $\lim_{k\to\infty}\|\boldsymbol{e^k}\| = 0$ for any choice of the initial datum $\boldsymbol{x}^0$.

Consistency alone does not suffice to ensure the convergence of the iterative method Eq. (5.3).

Thm Convergence of Iterative method **theorem 5.1.1**

Let Eq. (5.3) be a consistent method. Then, the sequence of vectors $\{x^k\}$ converges to the solution of Eq. (5.1) for any choice of $x^{(0)}$ iff $\rho(B) < 1$.

Proof. From Eq. (5.5) and the consistency assumption, the recursive relation $\boldsymbol{e^{k+1}} = Be^k$ is obtained:

$$\boldsymbol{e^{k+1}} = x^{k+1} - x^k = Bx^k + f - (Bx + f) = Be^k \tag{5.6}$$

Therefore,

$$e^{(k)} = B^k e^{(0)}, \forall k = 0, 1, ... \tag{5.7}$$

Thus, thanks to theorem 2.3.1, it follows that $\lim_{k\to\infty} B^k e^0 = 0$ for any $e^{(0)}$ iff $\rho(B) < 1$.

Def **definition 5.1.2**

Let B be the iteration matrix. We call:

1. $\|\boldsymbol{B^m}\|$ the *convergence factor* after m steps of the iteration.
2. $\|\boldsymbol{B}\|^{1/m}$ the *average convergence* factor after m steps;
3. $R_{m(B)} = -\frac{1}{m}\log\|\boldsymbol{B^m}\|$ the *average convergence rate* after m steps.

5.2 Stopping Criteria

The convergence of an iterative method is monitored by means of a stopping criterion. We can easily introduce the following criteria:

$$\frac{\|\boldsymbol{x} - \boldsymbol{x^{(k)}}\|}{\|\boldsymbol{x^{(k)}}\|} \leq \varepsilon \tag{5.8}$$

Unfortunately, the exact solution $\boldsymbol{x}$ is not known, we are trying to convert the problem into a residual-based stopping criterion.

Residual based stopping criteria: The iteration is stopped when the residual $\boldsymbol{r^{(k)}} = \boldsymbol{b} - A\boldsymbol{x^{(k)}}$ is small enough:

$$\frac{\|\boldsymbol{x} - \boldsymbol{x^{(k)}}\|}{\|\boldsymbol{x^{(k)}}\|} \leq K(A)\frac{\|\boldsymbol{r^{(k)}}\|}{\|\boldsymbol{b}\|} \Longrightarrow \frac{\|\boldsymbol{r^{(k)}}\|}{\|\boldsymbol{b}\|} \leq \varepsilon \tag{5.9}$$

This is a good criteria whenever the condition number $K(A)$ is not too large. If the condition number is large, the residual-based stopping criterion may be too stringent. To make the constant $K(A)$ smaller in the stopping criterion, there is a method called *preconditioning*:

$$\frac{\|\boldsymbol{x} - \boldsymbol{x^{(k)}}\|}{\|\boldsymbol{x^{(k)}}\|} \leq K(P^{-1}A)\frac{\|\boldsymbol{z^{(k)}}\|}{\|\boldsymbol{b}\|} \Longrightarrow \frac{\|\boldsymbol{z^{(k)}}\|}{\|\boldsymbol{b}\|} \leq \varepsilon \tag{5.10}$$

where $z^{(k)} = P^{-1}\boldsymbol{r}^k$.

Distance between consecutive iterations: The iteration is stopped when the distance between

consecutive iterates is small enough, define the distance $\boldsymbol{\delta}^{(k)} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}$, then the stopping criterion is:

$$\|\boldsymbol{\delta}^{(\boldsymbol{k})}\| \leq \varepsilon \tag{5.11}$$

The relation between the true error and the distance between consecutive iterates is given by:

$$\|\boldsymbol{e}^{(\boldsymbol{k})}\| \leq \frac{\|\boldsymbol{\delta}^{(\boldsymbol{k})}\|}{1-\rho(B)} \tag{5.12}$$

Therefore this is a "good" stopping criterion only if $\rho(B) \ll 1$.

5.3 Linear Iterative Methods

A general technique to devise consistent linear iterative methods is based on an additive splitting of the matrix A of the form $A = P - N$, where P and N are two suitable matrices and P is nonsingular. For reasons that will be clear in the later sections, P is called *preconditioning matrix or preconditioner.*

Precisely, given $x^{(0)}$, one can compute $x^{(k)}$ for $k \geqslant$, solving the system:

$$Px^{(k+1)} = Nx^{(k)} + b \tag{5.13}$$

The iteration matrix of method Eq. (5.13) is $B = P^{-1}N$ and the vector $f = P^{-1}b$. Alternatively, the method can be written as:

$$x^{(k+1)} = x^{(k)} + P^{-1}r^{(k)} \tag{5.14}$$

where the residual

$$r^{(k)} = b - Ax^{(k)} \tag{5.15}$$

is the vector that measures the error in the approximation $x^{(k)}$. Eq. (5.14) outlines the fact that a linear system, with coefficient matrix P, must be solved to update the solution at step $k+1$. Thus P, besides being nonsingular, ought to be easily invertible, in order to keep the overall computational cost low.(Notice that, if P were equal to A and $N = 0$, method Eq. (5.14) would converge in one iteration, but at the same cost of a direct method.

Let us mention two results that ensure convergence of the iteration Eq. (5.14), provided suitable conditions on the splitting of A are fulfilled.

5.3.1 Jacobi, Gauss-Seidel and Relaxation Methods

5.3.1.1 Jacobi Method and Over-Relaxation

If the diagonal entries of A are nonzero, we can single out in each equation the corresponding unknown, obtaining the equivalent linear system.

$$x_i = \frac{b_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}x_j}{a_{ii}}, i = 1, ..., n \tag{5.16}$$

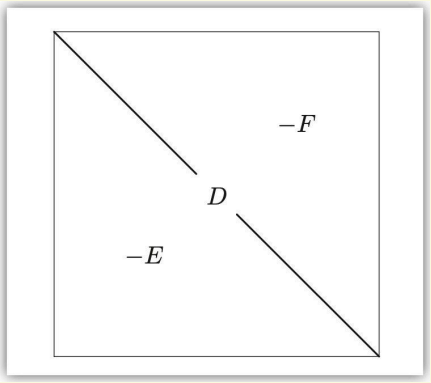
In the Jacobi method, once an arbitrarily initial guess $x^{(0)}$ is given, the solution is updated by the formula:

$$x_i^{(k+1)} = \frac{b_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij} x_j^{(k)}}{a_{ii}}, i = 1, ..., n \tag{5.17}$$

This amounts to performing the following splitting for A:

$$P = D, N = D - A = E + F$$

where D is the diagonal matrix of the diagonal entries of A, E is the lower triangular matrix, and F is the upper triangular matrix:



The iteration matrix of the Jacobi method is thus given by

$$B_j = D^{-1}(E + F) = I - D^{-1}A \tag{5.18}$$

A generalization of the Jacobi method is the over-relaxation method(or JOR), in which, having introduced a relaxation parameter ω, Eq. (5.17) is replaced by:

$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \omega \frac{b_i - \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij} x_j^{(k)}}{a_{ii}}, i = 1, ..., n \tag{5.19}$$

The corresponding iteration matrix is:

$$B_{j_w} = \omega B_j + (1 - \omega)I \tag{5.20}$$

This method is consistent if any $\omega \neq 0$ and for $\omega = 1$ it coincides with the Jacobi method.

5.3.1.2 The Gauss Seidel method

The Gauss-Seidel method differs from the Jacobi method in the fact that at the $k + 1$th step the available values of $x_i^{(k+1)}$ are being used to update the solution:

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)}}{a_{ii}}, i = 1, ..., n \tag{5.21}$$

This method amounts to performing the following splitting for A:

$$P = D - E, N = F \tag{5.22}$$

and the iteration matrix is:

$$B_{GS} = (D - E)^{-1}F \tag{5.23}$$

5.4 Stationary and Nonstationary Iterative Methods

Devoted by

$$R_p = I - P^{-1}A \tag{5.24}$$

the iteration matrix associated with Eq. (5.14). Proceeding as in the case of relaxation methods, Eq. (5.14) can be generalized introducing a relaxation (or acceleration) parameter α. This leads to the following *stationary Richardson method.*

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha P^{-1}\boldsymbol{r}^{(k)}, k \geq 0 \tag{5.25}$$

More generally, allowing α to depend on the iteration index, *the nonstationary Richardson method or semi-iterative method* is given by

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha^{(k)} P^{-1}\boldsymbol{r}^{(k)}, k \geq 0 \tag{5.26}$$

The iteration matrix at the k-th step for Eq. (5.26) is

$$B_R = I - \alpha_k P^{-1}A \tag{5.27}$$

with $\alpha_k = \alpha$ in the stationary case. If $P = I$, the family of methods Eq. (5.26) will be called *nonpreconditioned.*The Jacobi and Gauss-Seidel methods can be regarded as stationary Richardson methods with $P = D$ and $P = D - E$, respectively.

We can rewrite Eq. (5.26) in a form of greater interest for computation. Letting $\boldsymbol{z}^{(k)} = P^{-1}\boldsymbol{r}^{(k)}$(the so-called *preconditioned residual*), we have $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{z}^{(k)}$ and $\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha_k A\boldsymbol{z}^{(k)}$.

To summarize, a nonstationary Richardson method requires at each $k + 1$th step the following operations:

1. solve the linear system $P\boldsymbol{z}^{(k)} = \boldsymbol{r}^{(k)}$
2. compute the acceleration parameter α_k
3. update the solution $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{z}^{(k)}$
4. update the residual $\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha_k A\boldsymbol{z}^{(k)}$

Thm Convergence **theorem 5.4.1**

Assume that P is a nonsingular matrix and that $P^{-1}A$ has positive real eigenvalues, ordered in such a way that $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n > 0$. Then, the stationary Richardson method converges if and only if $0 \leq \alpha \leq \frac{2}{\lambda_1}$. Moreover, letting

$$\alpha_{\text{opt}} = \frac{2}{\lambda_1 + \lambda_n} \tag{5.28}$$

the spectral radius of the iteration matrix R_α is minimum if $\alpha = \alpha_{\text{opt}}$, with

$$\rho_{\text{opt}} = \min_{\{\alpha\}} \rho(R_\alpha) = \frac{\lambda_1 - \lambda_n}{\lambda_1 + \lambda_n} \tag{5.29}$$

e.g. Preconditioned Stational Richardson Method **example 5.4.1**

PRECONDITIONED STATIONAL RICHARDSON METHOD

Given arbitrary $\boldsymbol{x}^0 =$, Let $\boldsymbol{r}^0 = \boldsymbol{b} - A\boldsymbol{x}^0$

For $k = 0, 1, ...$

Compute $\alpha_{\text{opt}} = \frac{2}{\lambda_{\min}(P^{-1}A)+\lambda_{\max}(P^{-1}A)}$

Solve the linear system $P\boldsymbol{z}^{(k)} = \boldsymbol{r}^{(k)}$

Update the solution: $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_{\text{opt}}\boldsymbol{z}^{(k)}$

Update the residual: $\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha_{\text{opt}}A\boldsymbol{z}^{(k)}$

5.4.1 Preconditioning Matrices

All the methods introduced in the previous sections can be cast in the form Eq. (5.3) , so that they can be regarded as being methods for solving the system

$$(I - B)\boldsymbol{x} = \boldsymbol{f} = P^{-1}\boldsymbol{b} \tag{5.30}$$

On the other hand, since $B = P^{-1}N$, linear system $A\boldsymbol{x} = \boldsymbol{b}$ can be equivalently rewrited as

$$P^{-1}A\boldsymbol{x} = P^{-1}\boldsymbol{b} \tag{5.31}$$

The latter is the preconditioned system, being P the preconditioning matrix or left preconditioner. Right and centered preconditioners can be introduced as well,

$$AP^{-1}\boldsymbol{y} = \boldsymbol{b}, \boldsymbol{y} = P\boldsymbol{x} \tag{5.32}$$

or

$$P_L^{-1}AP_R^{-1}\boldsymbol{y} = P_L^{-1}\boldsymbol{b}, \boldsymbol{y} = P_R\boldsymbol{x} \tag{5.33}$$

There are *point preconditioners* and *block preconditioners*, depending on whether they are applied to the single entries of A or to the blocks of a partition of A. The iterative methods considered so far correspond to fixed-point iterations on a left-preconditioner system. Computing the inverse of P is not mandatory; actually, the role of P is to "precondition" the residual $\boldsymbol{r}^{(k)}$.

Since the preconditioner acts on the spectral radius of the iteration matrix, it would be useful to pick up, for a given linear system, an *optimal preconditioner*, a preconditioner which is able to make the number of iterations required for convergence independent of the size of the system.

There is not a general roadmap to devise optimal preconditioners. However, an established "rule of thumb" is that P is a good preconditioner for A if $P^{-1}A$ is near to being a normal matrix and if its eigenvalues are clustered within a sufficiently samll region of the complex field. The choice of a preconditioner must also be guided by practical considerations, noticeably, its computational cost and its memory requirements.

Preconditioners can be divided into two main categories: algebraic and functional preconditioners, the difference being that the algebraic preconditioners are independent of the problem

that originated the system to be solved, and are actually constructed via algebraic procedures, while the functional preconditioners take advantage of the knowledge of the problem and are constructed as a function of it.

5.4.2 The Gradient Method

In the special case of symmetric and positive definite matrices, however, the optimal acceleration parameter can be dynamically computed at each step k as follows.

We first notice that, for such matrices, solving system Eq. (5.14) is equivalent to minimizing the quadratic form

$$\Phi(\boldsymbol{y}) = \frac{1}{2}\boldsymbol{y}^T A\boldsymbol{y} - \boldsymbol{y}^T b \tag{5.34}$$

which is called the *energy of system*, Indeed, the gradient of Φ is given by

$$\nabla\Phi(\boldsymbol{y}) = \frac{1}{2}(A + A^T)\boldsymbol{y} - \boldsymbol{b} = A\boldsymbol{y} - \boldsymbol{b} \tag{5.35}$$

As a consequence, if $\nabla\Phi(\boldsymbol{x}) = 0$, then $\boldsymbol{x}$ is a solution of the original system. Conversely, if $\boldsymbol{x}$ is a solution, then

$$\Phi(\boldsymbol{y}) = \Phi(\boldsymbol{x} + \boldsymbol{y} - \boldsymbol{x}) = \Phi(\boldsymbol{x}) + \frac{1}{2}(\boldsymbol{y} - \boldsymbol{x})^T A(\boldsymbol{y} - \boldsymbol{x}) \tag{5.36}$$

and thus, for any $\boldsymbol{y}$, the value of Φ is minimized at $\boldsymbol{x}$. Notice that the previous relation is equivalent to

$$\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_A^2 = \Phi(\boldsymbol{y}) - \Phi(\boldsymbol{x}) \tag{5.37}$$

where $\|\cdot\|_A$ is the energy norm, defined in theorem 2.1.4.

The problem is thus to determine the minimizer $\boldsymbol{x}$ of Φ by starting from a point $\boldsymbol{x}^0$, and, consequently, to select suitable directions along which moving to get as close as possible to the solution $\boldsymbol{x}$. The optimal direction, that joins the starting point $\boldsymbol{x}^{(0)}$ to the solution point $\boldsymbol{x}$, is obviously unknown a priori. Therefore, we must take a step from $\boldsymbol{x}^{(0)}$ along a given direction $\boldsymbol{p}_0$ and then fix along this latter a new point $\boldsymbol{x}^{(1)}$ from which to iterate the process until convergence.

Precisely, at the generic step k, $\boldsymbol{x}^{(k+1)}$ is computed as

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{p}^{(k)} \tag{5.38}$$

where α_k is the value which fixes the length of the step along the direction $\boldsymbol{p}^{(k)}$.

The most natural idea is to take as $\boldsymbol{p}^{(k)}$ the direction of maximum descent along the functional Φ in $\boldsymbol{x}^{(k)}$, which is given by $-\nabla\Phi\big(\boldsymbol{x}^{(k)}\big)$. This yields the gradient method, also called steepest descent method.

Due to Eq. (5.35), $\nabla\Phi\big(\boldsymbol{x}^{(k)}\big) = A\boldsymbol{x}^{(k)} - \boldsymbol{b} = -\boldsymbol{r}^{(k)}$ so that the direction of the gradient of Φ coincides with the residual $\boldsymbol{r}^{(k)}$. So it can be immediately computed using the current iterate. This shows that the gradient method, as well as the nonstationary Richardson method with $P = I$, moves at each step k along the direction of the residual $\boldsymbol{r}^{(k)}$.

To compute the parameter $\alpha^{(k)}$ let us write explicitly $\Phi\big(\boldsymbol{x}^{(k+1)}\big)$ as a function of a parameter α

$$\Phi\big(\boldsymbol{x}^{(k+1)}\big) = \frac{1}{2}\big(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{r}^{(k)}\big)^T A\big(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{r}^{(k)}\big) - \big(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{r}^{(k)}\big)^T \boldsymbol{b} \tag{5.39}$$

Differentiating with respect to α and setting it equal to zero yields the desired value of α_k

$$\alpha_k = \frac{\boldsymbol{r}^{(k)^T}\boldsymbol{r}^{(k)}}{\boldsymbol{r}^{(k)^T} A \boldsymbol{r}^{(k)}} \tag{5.40}$$

GRADIENT METHOD

Given $\boldsymbol{x}^{(0)}$, compute the residual: $\boldsymbol{r}^{(0)} = \boldsymbol{b} - A\boldsymbol{x}^{(0)}$

While(Stopping criterion is not satisfied)

> Compute parameter $\alpha_k = \frac{\boldsymbol{r}^{(k)^T}\boldsymbol{r}^{(k)}}{\boldsymbol{r}^{(k)^T} A \boldsymbol{r}^{(k)}}$
>
> Update the solution: $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{r}^{(k)}$
>
> Update the residual: $\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha_k A \boldsymbol{r}^{(k)}$

Thm Convergence of Gradient Method **theorem 5.4.2**

Let A be a symmetric and positive definite matrix. Then the gradient method is convergent for any choice of the initial datum $\boldsymbol{x}^{(0)}$. Moreover

$$\|\boldsymbol{e}^{(\boldsymbol{k+1})}\|_A \leq \frac{K(A)-1}{K(A)+1}\|\boldsymbol{e}^{(\boldsymbol{k})}\|_A \tag{5.41}$$

where $\|\cdot\|_A$ is the energy norm defined in theorem 2.1.4.

5.4.3 The Conjugate Gradient Method

The gradient method consists essentially of two phases: choosing a direction $\boldsymbol{p}^{(k)}$ and picking up a point of local minimum for Φ along that direction. The latter request can be accommodated by choosing α_k as the value of the parameter α such that $\Phi\big(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{p}^{(k)}\big)$ is minimized. Differentiating with respect to α and setting it equal to zero, we obtain the following expression for α_k:

$$\alpha_k = \frac{\boldsymbol{p}^{(k)^T}\boldsymbol{r}^{(k)}}{\boldsymbol{p}^{(k)^T} A \boldsymbol{p}^{(k)}} \tag{5.42}$$

CONJUGATE GRADIENT METHOD

Given $\boldsymbol{x}^{(0)}$, compute $\boldsymbol{r}^{(0)} = \boldsymbol{b} - A\boldsymbol{x}^{(0)}, \boldsymbol{p}^{(0)} = \boldsymbol{r}^{(0)}$

While(Stopping criterion is not satisfied)

> Compute step lenght $\alpha_k = \frac{\boldsymbol{p}^{(k)^T}\boldsymbol{r}^{(k)}}{\boldsymbol{p}^{(k)^T} A \boldsymbol{p}^{(k)}}$
>
> Update the solution: $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{p}^{(k)}$

CONJUGATE GRADIENT METHOD

Update the residual: $\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha_k A\boldsymbol{p}^{(k)}$

Compute the conjugate coefficient $\beta_k = \frac{(A\boldsymbol{p}^{(k)})^T \boldsymbol{r}^{(k+1)}}{(A\boldsymbol{p}^{(k)})^T \boldsymbol{p}^{(k)}}$

Update the direction: $\boldsymbol{p}^{(k+1)} = \boldsymbol{r}^{(k+1)} - \beta_k \boldsymbol{p}^{(k)}$

5.4.4 The Preconditioned Conjugate Gradient Method

If P is a symmetric and positive definite matrix, the preconditioned conjugate gradient method(PCG) consists of applying the CG method to the preconditioned system:

$$P^{-\frac{1}{2}} A P^{-\frac{1}{2}} \boldsymbol{y} = P^{-\frac{1}{2}} \boldsymbol{b}, \text{with } \boldsymbol{y} = P^{-\frac{1}{2}} \boldsymbol{x} \tag{5.43}$$

PRECONDITIONED CONJUGATE GRADIENT METHOD

Given $\boldsymbol{x}^{(0)}$, compute $\boldsymbol{r}^{(0)} = \boldsymbol{b} - A\boldsymbol{x}^{(0)}, \boldsymbol{z}^{(0)} = P^{-1}\boldsymbol{r}^{(0)}, \boldsymbol{p}^{(0)} = \boldsymbol{z}^{(0)}$

While(Stopping criterion is not satisfied)

Compute step length $\alpha_k = \frac{\boldsymbol{p}^{(k)^T} \boldsymbol{r}^{(k)}}{\boldsymbol{p}^{(k)^T} A\boldsymbol{p}^{(k)}}$

Update the solution: $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{p}^{(k)}$

Update the residual: $\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha_k A\boldsymbol{p}^{(k)}$

Solve the preconditioned system: $P\boldsymbol{z}^{(k+1)} = \boldsymbol{r}^{(k+1)}$

Compute the conjugate coefficient $\beta_k = \frac{\boldsymbol{z}^{(k+1)^T} \boldsymbol{r}^{(k+1)}}{\boldsymbol{z}^{(k+1)^T} \boldsymbol{z}^{(k)}}$

Update the direction: $\boldsymbol{p}^{(k+1)} = \boldsymbol{z}^{(k+1)} + \beta_k \boldsymbol{p}^{(k)}$

Thm Finite-Termination Property **theorem 5.4.3**

For an $n \times n$ SPD matrix A , the CG algorithm converges to the exact solution $x = A^{-1}b$ in at most n iterations in exact arithmetic.

Thm Error Bounds **theorem 5.4.4**

Let A be an $n \times n$ SPD matrix and let $x^{(k)}$ be the iterate of the CG method at the k-th step. Then, the error $e^{(k)} = x - x^{(k)}$ satisfies the following error bounds:

$$\|\boldsymbol{e}^{(k)}\|_A \leq \frac{2c^k}{1 + c^{2k}} \|\boldsymbol{e}^{(0)}\|_A \tag{5.44}$$

where $c = \frac{\sqrt{K(A)}-1}{\sqrt{K(A)}+1}$ and $K(A)$ is the condition number of A.

5.5 Methods Based on Krylov Subspace Iterations

Def Krylov Subspace **definition 5.5.1**

Given a nonsingular matrix $A \in \mathbb{R}(n \times n)$ and $\boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{y} \neq \boldsymbol{0}$, the mth Krylov subspace $K_m(A, \boldsymbol{y})$ generated by A and $\boldsymbol{y}$ is

$$K_m(A, \boldsymbol{y}) = \text{span}\left\{\boldsymbol{y}, A\boldsymbol{y}, A^2\boldsymbol{y}, ..., A^{m-1}\boldsymbol{y}\right\} \tag{5.45}$$

Consider the Richardson method Eq. (5.26) with $P = I$; the residual at the k-th step can be related to the initial residual as

$$\boldsymbol{r}^{(k)} = \prod_{j=0}^{k-1}(I - \alpha_j A)\boldsymbol{r}^{(0)} \tag{5.46}$$

So start at $\boldsymbol{r}^{(k)} = p_k A\boldsymbol{r}^{(0)}$, where $p_k(A)$ is a polynomial in A of degree k.

In an analogous manner as for Eq. (5.46), it is seen that the iterate $\boldsymbol{x}^{(k)}$ of the Richardson method is given by

$$\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(0)} + \sum_{j=0}^{k-1}\alpha_j \boldsymbol{r}^{(j)} \tag{5.47}$$

so that $\boldsymbol{x}^{(k)}$ belongs to the following space

$$W_k = \left\{\boldsymbol{v} = \boldsymbol{x}^{(0)} + \boldsymbol{y}, \boldsymbol{y} \in K_k\left(A, \boldsymbol{r}^{(0)}\right)\right\} \tag{5.48}$$

In the nonpreconditioned Richardson method we are thus looking for an approximate solution to $\boldsymbol{x}$ in the space W_k. More generally, we can think of devising methods that search for approximate solutions of the form

$$\boldsymbol{x}^{(k)} = \boldsymbol{x}^{(0)} + q_{k-1}(A)\boldsymbol{r}^{(0)} \tag{5.49}$$

where q_{k-1} is polynomial selected in such a way that $\boldsymbol{x}^{(k)}$ be, in a sense that must be made precise, the best approximation of $\boldsymbol{x}$ in W_k. A method that looks for a solution of the form Eq. (5.46) is called a *Krylov subspace method.*

6 Solving large scale eigenvalue problems

6.1 Eigenvalues and Eigenvectors

Given a square matrix $A \in \mathbb{C}^{n\times n}$, find a scalar λ and a nonzero vector $x \in \mathbb{C}^n$ such that:

$$Ax = \lambda x \tag{6.1}$$

where:

1. The vector x is the *eigenvector*, And the scalar λ is the *eigenvalue*
2. The set of all the eigenvalues of a matrix A is called the *spectrum* of A, denoted by $\sigma(A)$.
3. The maximum modulus of all the eigenvalues is called the *spectral radius* of A and is denoted by $\rho(A) = \max_{\{\lambda\in\sigma(A)\}}|\lambda|$.

Remarks

1. The eigenvalues of a matrix are the roots of the characteristic polynomial $\det(A - \lambda I) = 0$.
2. From the Fundamental Theorem of Algebra, an $n \times n$ matrix has exactly n eigenvalues, counting multiplicities.
3. Each λ_i may be real but in general is a complex number
4. The eigenvalues $\lambda_1, \lambda_2, ..., \lambda_n$may not all have distinct values
5. Rayleigh quotient: $\lambda_i = \frac{x_i^H A x_i}{x_i^H x_i}$

6.2 The Power Method

Let $A \in \mathbb{C}^{n\times n}$ be a diagonalizable matrix and let $X \in \mathbb{C}^{n\times n}$ be the matrix of its eigenvectors $\boldsymbol{x_i}$. for $i = 1, ..., n$. Let us also suppose that the eigenvalues of A are ordered as

$$|\lambda_1| > |\lambda_2| \geq ... \geq |\lambda_n| \tag{6.2}$$

Where λ_1 has algebraic multiplicity equal to 1. Under these assumptions, λ_1, is called the *dominant eigenvalue* of A.

Given an arbitrary initial vector $\boldsymbol{q}_0 \in \mathbb{C}^n$ with unitary Euclidean norm, consider for $k = 1, 2, ...$ the following iteration based on the computation of powers of matrices, commonly known as the *power method*:

$$\begin{aligned} \boldsymbol{z}^{(k)} &= A\boldsymbol{q}^{(k-1)} \\ \boldsymbol{q}^{(k)} &= \frac{\boldsymbol{z}^{(k)}}{\|\boldsymbol{z}^{(\boldsymbol{k})}\|} \\ \nu^{(k)} &= \boldsymbol{q}^{((k))^H} A\boldsymbol{q}^{(k)} \end{aligned} \tag{6.3}$$

THE POWER METHOD

$\boldsymbol{q}_0$ =some initial vector with $\|\boldsymbol{q_0}\| = 1$
For $k = 1, 2, ...$
| Apply A: $\boldsymbol{z}^{(k)} = A\boldsymbol{q}^{(k-1)}$

THE POWER METHOD

Normalize: $\boldsymbol{q}^{(k)} = \frac{\boldsymbol{z}^{(k)}}{\|\boldsymbol{z}^{(k)}\|}$
Compute Rayleigh quotient: $\nu^{(k)} = \boldsymbol{q}^{((k))^H} A\boldsymbol{q}^{(k)}$

Let us analyze the convergence of the power method. By induction on k, we have that:

$$\boldsymbol{q}^{(k)} = A^k \frac{\boldsymbol{q}^{(0)}}{\|\boldsymbol{A^k q^{(0)}}\|}, k \geq 1 \tag{6.4}$$

This relation explains the role played by the powers of A in the method. Because A is diagonalizable, its eigenvectors $\boldsymbol{x}_i$ form a basis of $\mathbb{C}^n$ and we can write:

$$\boldsymbol{q}^{(0)} = \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i \tag{6.5}$$

Moreover, since $A\boldsymbol{x}_i = \lambda_i \boldsymbol{x}_i$, we have:

$$\begin{aligned} A^k \boldsymbol{q}^{(0)} &= \sum_{i=1}^{n} \alpha_i \lambda_i^k \boldsymbol{x}_i \\ &= \alpha_1 \lambda_1^k \left(\boldsymbol{x}_1 + \sum_{i=2}^{n} \frac{\alpha_i}{\alpha_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k \boldsymbol{x}_i \right) \end{aligned} \tag{6.6}$$

Since $\left|\frac{\lambda_i}{\lambda_1}\right| < 1$ for $i = 2, ..., n$, as k increases the term $\sum_{i=2}^{n} \frac{\alpha_i}{\alpha_1} \left(\frac{\lambda_i}{\lambda_1}\right)^k \boldsymbol{x}_i$ tends to assume an increasingly significant component in the direction of the eigenvector $\boldsymbol{x}_1$, while ts components in the other directions $\boldsymbol{x}_j$ decrease.

As $k \to \infty$, the vector $\boldsymbol{q}^{(k)}$ thus aligns itself along the direction of eigenvector $\boldsymbol{x}_1$, and the following error estimate holds at each step k.

Thm Convergence of the Power Method **theorem 6.2.1**

Let $A \in \mathbb{C}^{n \times n}$ be a diagonalizable matrix whose dominant eigenvalue is λ_1. Assuming that $\alpha_1 \neq 0$, there exists a constant $C > 0$ such that:

$$\|\tilde{\boldsymbol{q}}^{(\boldsymbol{k})} - \boldsymbol{x_1}\| \leq C \left(|\frac{\lambda_2}{\lambda_1}| \right)^k, k \geq 1 \tag{6.7}$$

where:

$$\tilde{\boldsymbol{q}}^{(k)} = \boldsymbol{x}_1 + \sum_{i=2}^{n} \frac{\alpha_i}{\alpha_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k \boldsymbol{x}_i, k = 1, 2, ... \tag{6.8}$$

Estimate Eq. (6.7) expresses the convergence of the sequence of $\tilde{\boldsymbol{q}}^{(k)}$ towards the eigenvector $\boldsymbol{x}_1$ of A. Therefore the sequence of Rayleigh quotients

$$\tilde{\boldsymbol{q}}^{((k))^H} A \tilde{\boldsymbol{q}}^{(k)} / \|\tilde{\boldsymbol{q}}^{(\boldsymbol{k})}\| = \left(\boldsymbol{q}^{(k)}\right)^H A \boldsymbol{q}^{(k)} = \nu^{(k)} \tag{6.9}$$

will converge to the dominant eigenvalue λ_1 of A. As a consequence, and the convergence will be faster when the ratio $|\frac{\lambda_2}{\lambda_1}|$ is smaller.

6.3 Deflation

6.4 The Inverse Power Method

We look for an approximation of the eigenvalue of a matrix $A \in \mathbb{C}^{n\times n}$ which is *closest* to a given number $\mu \in \mathbb{C}$, where $\mu \notin \sigma(A)$. For this, the power iteration is applied to the matrix $\left(M_\mu\right)^{-1} = (A-\mu I)^{-1}$, yielding the so-called *inverse iteration* or *inverse power method*. The number μ is called the *shift* of the method.

The eigenvalues of M_u^{-1} are $\xi = (\lambda_i - \mu)^{-1}$, let us assume that there exists an integer m such that

$$|\lambda_m - \mu| < |\lambda_i - \mu| \tag{6.10}$$

Given an arbitrary initial vector $\boldsymbol{q}_0 \in \mathbb{C}^n$ with unitary Euclidean norm, for $k = 1, 2, ...$ the following sequence is constructed:

$$\begin{aligned} (A-\mu I)\boldsymbol{z}^{(k)} &= \boldsymbol{q}^{(k-1)} \\ \boldsymbol{q}^{(k)} &= \frac{\boldsymbol{z}^{(k)}}{\|\boldsymbol{z}^{(\boldsymbol{k})}\|} \\ \nu^{(k)} &= \boldsymbol{q}^{((k))^H} A\boldsymbol{q}^{(k)} \end{aligned} \tag{6.11}$$

THE INVERSE POWER METHOD

$\boldsymbol{q}_0$ =some initial vector with $\|\boldsymbol{q_0}\| = 1$
For $k = 1, 2, ...$
  Solve linear equation $(A-\mu I)$: $\boldsymbol{z}^{(k)} = (A-\mu I)\boldsymbol{q}^{(k-1)}$
  Normalize: $\boldsymbol{q}^{(k)} = \frac{\boldsymbol{z}^{(k)}}{\|\boldsymbol{z}^{(\boldsymbol{k})}\|}$
  Compute Rayleigh quotient: $\nu^{(k)} = \boldsymbol{q}^{((k))^H} A\boldsymbol{q}^{(k)}$

Notice that the eigenvectors of M_μ are the same as those of A since $M_\mu = X(\Lambda - \mu I_n)X^{-1}$, where $\Lambda = \text{diag}(\lambda_1, ..., \lambda_n)$. For this reason, the Rayleigh quotient in Eq. (6.11) is computed directly on the matrix A. The main difference with respect to Eq. (6.3) is that at each step k a linear system with coefficient matrix $M_\mu = A - \mu I$ *must be solved*.

6.5 QR Iterative Method

In this section we present some iterative techniques for simultaneously approximating all the eigenvalues of a given matrix A. The basic idea consists of reducing A, by means of suitable similarity transformations, into a form for which the calculation of the eigenvalues is easier than on the starting matrix.

Let $A \in \mathbb{C}^{n\times n}$. The QR algorithm computes an upper triangular matrix T and a unitary matrix U such that $A = UTU^H$.

QR Iterative Method

Set $A^{(0)} = A$, $U^{(0)} = I$

While(Stopping criterion is not satisfied)

Compute the QR factorization: $A^{(k-1)} = Q^{(k)}R^{(k)}$

Update the matrix: $A^{(k)} = R_k Q_k$

Update the unitary matrix: $U^{(k)} = U^{(k-1)}Q^{(k)}$

Return $T = A^{(k)}, U = U^{(k)}$

Notice that

1. $A^{(k)} = R^{(k)}Q^{(k)} = \left[Q^{(k)}\right]^H A^{(k-1)}Q^{(k)}$, therefore $A^{(k)}$ is similar to $A^{(k-1)}$.
2. Moreover, from the above observaSon, we have $A^{(k)} = \left[Q^{(k)}\right]^H ... \left[Q^{(1)}\right]^H A^{(0)} \left[Q^{(1)}\right] ... Q^{(k)}$.

6.6 The Lanzos algorithm

The Lanczos algorithm can be used to efficiently find the extremal eigenvalues(maximum and minimum) of a symmetric matrix A of size $n \times n$.

It is based on computing the following decomposition of A:

$$A = Q^T T Q \tag{6.12}$$

where Q is an orthonormal basis of vectors $\boldsymbol{q}_1, ..., \boldsymbol{q}_n$ and T is a tridiagonal matrix of size $n \times n$. The decomposiSon always exists and is unique provided that $\boldsymbol{q}_1$ has been specified.

We know that $T = Q^T AQ$ which gives

$$\begin{aligned} \alpha_k &= \boldsymbol{q}_k^T A \boldsymbol{q}_k \\ \beta_k &= \boldsymbol{q}_{k+1}^T A \boldsymbol{q}_k \end{aligned} \tag{6.13}$$

The full decompostion is obtained by imposing $AQ = QT$:

$$\begin{aligned} A\boldsymbol{q}_1 &= \alpha_1 \boldsymbol{q}_1 + \beta_1 \boldsymbol{q}_2 \\ A\boldsymbol{q}_2 &= \beta_1 \boldsymbol{q}_1 + \alpha_2 \boldsymbol{q}_2 + \beta_2 \boldsymbol{q}_3 \\ &... \\ A\boldsymbol{q}_n &= \beta_{n-1} \boldsymbol{q}_{n-1} + \alpha_n \boldsymbol{q}_n \end{aligned} \tag{6.14}$$

7 Numerical methods for overdetermined linear systems of equations

Overdetermined linear systems of equations are systems of equations in which the number of equations is greater than the number of unknowns. In this case, the system is said to be *overdetermined*. When the problems are linear there is a very clean and simple way to find the optimum, if we adopt the sum-of-squares error metric.

7.1 Linear Regression

If there were no experimental uncertainty the model would fit the data exactly, but since there is noise the best we can do is minimise the error. The problem is:

$$\min_{\alpha_0,\alpha_1} \sum_{i=1}^{m} \boldsymbol{e}_i^2 = \min_{\alpha_0,\alpha_1} \sum_{i=1}^{m} (\alpha_0 + \alpha_1 T_i - L_i)^2 \tag{7.1}$$

The above problem is equivalent to the following:

$$\min_{\alpha_0,\alpha_1} \|\boldsymbol{e}\|^2 = \min_{\alpha_0,\alpha_1} \|\boldsymbol{A\alpha} - (\boldsymbol{b})\|^2 \tag{7.2}$$

with $A = [1, T_1; 1, T_2; ...; 1, T_m]$ and $\boldsymbol{\alpha} = [\alpha_0, \alpha_1]^T$, $\boldsymbol{b} = [L_1, L_2, ..., L_m]^T$.

7.2 The Least Squares Solution

The mathematical problem reads: given a matrix $A \in \mathbb{R}^{m\times n}$ and a vector $\boldsymbol{b} \in \mathbb{R}^m$, find the vector $\boldsymbol{x} \in \mathbb{R}^n$ such that:

$$A\boldsymbol{x} = \boldsymbol{b} \tag{7.3}$$

We notice that generally the above problems has no solution (in the classical sense) unless the right side b is an element of range(A). We need a new concept of solution, the basic approach is to look for an $\boldsymbol{x}$ that makes $A\boldsymbol{x}$ close to $\boldsymbol{b}$.

Def Least Squares Solution **definition 7.2.1**

Given a matrix $A \in \mathbb{R}^{m\times n}, m \geq n$, we say that $\boldsymbol{x}^* \in \mathbb{R}^n$ is a solution of the linear system $A\boldsymbol{x} = \boldsymbol{b}$ in the sense of *least squares* if

$$\Phi(\boldsymbol{x}^*) = \min_{\{\boldsymbol{y}\in\mathbb{R}^n\}} \Phi(\boldsymbol{y}) \tag{7.4}$$

where $\Phi(\boldsymbol{y}) = \|\boldsymbol{Ay} - \boldsymbol{b}\|^2$. The problem thus consists of minimising the Euclidean norm of the residual.

The solution $\boldsymbol{x}^*$ can be found by imposing the condition that the gradient of function Φ must be zero at $\boldsymbol{x}^*$.

From the definition we have

$$\nabla\Phi(\boldsymbol{x}^*) = \nabla\|\boldsymbol{Ax}^* - \boldsymbol{b}\|^2 = 2A^T(A\boldsymbol{x}^* - \boldsymbol{b}) = 0 \tag{7.5}$$

from which it follows that $A^TA\boldsymbol{x}^* = A^T\boldsymbol{b}$.

Thm Exists and Unique **theorem 7.2.1**

The system of normal equations is nonsingular if A has full rank and, in such a case, the least squares solution exists and is unique.

We notice that $B = A^T A$ is a SPD matrix. Thus, in order to solve the normal equations, one could first compute the Cholesky factorization of B, by solving two triangular systems:

$$\begin{aligned} R\boldsymbol{z} &= A^T \boldsymbol{b} \\ R\boldsymbol{x}^* &= \boldsymbol{z} \end{aligned} \tag{7.6}$$

However, $A^T A$ is very badly conditioned and, due to roundoff errors, the computation of $A^T A$ may be affected by a loss of significant digits, with a consequent loss of positive definiteness or nonsingularity of the matrix!

Thm Solution in Reduced QR Factorization **theorem 7.2.2**

Let $A \in \mathbb{R}^{m \times n}, m \geq n$, **be a full rank matrix**, and let $A = \hat{Q}\hat{R}$ be the reduced QR factorization of A. Then the unique solution in the least-square sense of the system $A\boldsymbol{x} = \boldsymbol{b}$ is given by

$$\boldsymbol{x}^* = \hat{R}^{-1}\hat{Q}^T \boldsymbol{b} \tag{7.7}$$

Moreover, the minimum of the function Φ is given by

$$\Phi(\boldsymbol{x}^*) = \sum_{i=n+1}^{m} \left[(Q^T \boldsymbol{b})_i \right]^2 \tag{7.8}$$

7.3 SVD

If A does not have full rank, the above solution techniques above fail. In this case if $\boldsymbol{x}^*$ is a solution in the least square sense, the vector $\boldsymbol{x}^* + \boldsymbol{z}$, with $\boldsymbol{z} \in N(A)$, is also a solution. We must therefore introduce a further constraint to enforce the uniqueness of the solution. Typically, one requires that $\boldsymbol{x}^*$ has minimal Euclidean norm, so that the least squares problem can be formulated as:

$$\text{Find } \boldsymbol{x}^* \text{ such that } \|A\boldsymbol{x}^* - \boldsymbol{b}\|^2 \leq \min_{\boldsymbol{x} \in \mathbb{R}^n} \|A\boldsymbol{x} - \boldsymbol{b}\|^2 \tag{7.9}$$

This problem is consistent with our formulation. If A has full rank, since in this case the solution in the least-square sense exists and is unique it necessarily must have minimal Euclidean norm. The tool for solving Eq. (7.9) is the singular value decomposition(SVD).

Def pseudo-inverse **definition 7.3.1**

Suppose that $A \in \mathbb{R}^{m \times n}$ has rank equal to r and let $A = U\Sigma V^T$ be the SVD of A. The *pseudo-inverse* of A is the matrix

$$A^\dagger = V\Sigma^\dagger U^T \tag{7.10}$$

where $\Sigma^\dagger$ is the $n \times m$ matrix obtained by taking the reciprocal of the nonzero singular values of Σ and transposing the result.

The matrix $A^\dagger$ is also called the *generalized inverse* of A. And if $n = m = \text{rank}(A)$, then $A^\dagger = A^{-1}$.

8 Direct Methods for Linear Systems

8.1 Solution of Triangular Systems

Consider the nonsingular 3×3 **lower triangular** system:

$$\begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{8.1}$$

Since the matrix is nonsingular, its diagonal entries l_{ii} are nonzero, hence we can solve sequentially for the unknown values x_i, as follows:

$$\begin{aligned} x_1 &= b_1/l_{11} \\ x_2 &= (b_2 - l_{21}x_1)/l_{22} \\ x_3 &= (b_3 - l_{31}x_1 - l_{32}x_2)/l_{33} \end{aligned} \tag{8.2}$$

This algorithm can be extended to systems $n \times n$ and is called *forward substitution*. In the case of system $L\boldsymbol{x} = \boldsymbol{b}$, with L being a nonsingular lower triangular matrix of order $n(n \geq 2)$, the method is as follows:

$$\begin{aligned} x_1 &= b_1/l_{11} \\ x_n &= \frac{1}{l_{ii}} \left(b_i - \sum_{j=1}^{i-1} l_{ij}x_j \right), i = 2, ..., n \end{aligned} \tag{8.3}$$

The number of multiplications and divisions to execute the algorithm is equal to $\frac{n(n+1)}{2}$, while the number of sums and subtractions is $\frac{n(n-1)}{2}$. The global operation count for the forward substitution is n^2.

Similar conclusions can be drawn for a linear system $U\boldsymbol{x} = \boldsymbol{b}$, with U being a nonsingular upper triangular matrix of order $n(n \geq 2)$. In this case the algorithm is called *backward substitution* and in the general case can be written as:

$$\begin{aligned} x_n &= \frac{b_n}{u_{nn}} \\ x_i &= \frac{1}{u_{ii}} \left(b_i - \sum_{j=i+1}^{n} u_{ij}x_j \right), i = n-1, ..., 1 \end{aligned} \tag{8.4}$$

Its computational cost is the same as that of the forward substitution.

8.2 Guassian Elimination and LU Factorization

Consider a nonsingular matrix $A \in \mathbb{R}^{n \times n}$, and suppose that the diagonal entries a_{ii} is nonzero. Introducing the *multiplers*:

$$m_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}, i = 2, ..., n \tag{8.5}$$

where $a_{i1}^{(1)}$ denote the elements of $A^{(1)}$, it is possible to eliminate the unknown x_1 from the rows other than the first one by simply subtracting from row i, with $i = 2, ..., n$, the first row multiplied by m_{i1} and doing the same on the right side. If we now define

$$\begin{aligned} a_{ij}^{(2)} &= a_{ij}^{(1)} - m_{i1} a_{1j}^{(1)}, i, j = 2, ..., n \\ b_i^{(2)} &= b_i^{(1)} - m_{i1} b_1^{(1)}, i = 2, ..., n \end{aligned} \tag{8.6}$$

where $b_i^{(1)}$ denotes the elements of $\boldsymbol{b}^{(1)}$, we have the following system:

$$\begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & ... & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & ... & a_{2n}^{(2)} \\ ... & & & \\ 0 & 0 & ... & a_{nn}^{(n)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ ... \\ b_n^{(n)} \end{bmatrix} \tag{8.7}$$

which we denote by $A^{(2)}\boldsymbol{x} = \boldsymbol{b}^{(2)}$. that is equivalent to the starting one. Similarly, we can transform the system in such a way that the unknown x_2 is eliminated from rows $3, ..., n$,.In general, we end up with the finite sequence of systems

$$A^{(k)}\boldsymbol{x} = \boldsymbol{b}^{(k)}, k = 1, ..., n \tag{8.8}$$

where, for $k \geq 2$, matrix $A^{(k)}$ takes the following form:

$$A^{(k)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & ... & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & ... & a_{2n}^{(2)} \\ ... & & & \\ 0 & 0 & ... & a_{kk}^{(k)} \end{bmatrix} = L^{(k)}U^{(k)} \tag{8.9}$$

It is clear that for $k = n$ we obtain the upper triangular system $U\boldsymbol{x} = \boldsymbol{b}^{(n)}$ which can be solved by backward substitution.

Thm Existence and Uniqueness **theorem 8.2.1**

Let $A \in \mathbb{R}^{n \times n}$. The LU factorization of A with $l_{ii} = 1$ for $i = 1, ..., n$ exists and is unique iff the principal submatrices A_i of A of order $i = 1, ..., n - 1$ are nonsingular.

Thm Sufficient Condition for Gaussian Elimination **theorem 8.2.2**

Let $A \in \mathbb{R}^{n \times n}$ be a nonsingular matrix. The LU factorization of A exists and is unique if A follows the below two conditions:

1. A is strictly diagonally dominant by rows / columns
2. A is symmetric and positive definite

8.3 Pivoting techniques

As previously pointed out, the GEM process breaks down as soon as a zero pivotal entry is computed. In such case, one needs to resort to the so-called *pivoting techniques*, which amounts to

exchanging rows(columns) of the system in such a way that nonzero pivotal elements are always available. So the LU factorization becomes:

$$PA = LU \tag{8.10}$$

where P is a permutation matrix. To solve linear system $\boldsymbol{Ax} = \boldsymbol{b}$, we solve the equivalent system $P\boldsymbol{Ax} = P\boldsymbol{b}$, which can be solved by the following two triangular systems:

$$\begin{aligned} L\boldsymbol{y} &= P\boldsymbol{b} \\ U\boldsymbol{x} &= \boldsymbol{y} \end{aligned} \tag{8.11}$$

Moreover, the piovtal element should be as large as possible to avoid round-off errors. In practice:

1. doing pivoting even when it is not strictly needed.
2. Swap the row k with the row i, where i is the row with the largest pivotal element in the k-th column.