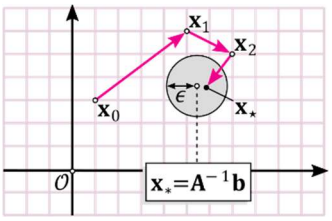
LECTURE

01

Gaussian Elimination



Brief Theory

Classification of methods for linear systems

- Let $m \in \mathbb{N}^*$ and consider a non-singular matrix $\mathbf{A} \in \mathbb{R}^{m\times m}$ and a vector $\mathbf{b} \in \mathbb{R}^m$. Numerical methods for solving the system $\mathbf{Ax} = \mathbf{b}$ are either **direct**, or **iterative**.
  - (a) In the absence of rounding errors, direct methods deliver the exact solution after a finite number of operations.
  - (b) In contrast to direct methods, iterative methods generate a sequence $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_\star$ of improving approximations, given an initial guess $\mathbf{x}_0$ of the solution and appropriate termination criteria, such as $\|\mathbf{x}_* - \mathbf{x}_\star\|_2 \leq \epsilon$, where $\epsilon > 0$.

- As it has been shown in worked example 0.6, when the matrix $\mathbf{A}$ is symmetric and positive definite, then the linear system $\mathbf{Ax} = \mathbf{b}$ is equivalent to the **minimization problem**

$$\min_{\mathbf{x}\in\mathbb{R}^m} f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top\mathbf{Ax} - \mathbf{b}^\top\mathbf{x} + \mathbf{c}.$$

Iterative methods are motivated by methods for solving such minimization problems, while direct methods are constructed by considering the linear system itself.

Unless otherwise marked, $\mathbf{A} \in \mathbb{R}^{m\times m}$ is a non-singular matrix of size $m \in \mathbb{N}^*$ and is associated with a system $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{b} \in \mathbb{R}^m$.

Gaussian elimination

- **Gaussian elimination** is a direct method that consists of two stages, namely, **forward elimination** and **back substitution**.

Gaussian elimination is an instance of the elementary algebraic method for solving a system of m linear equations in m unknowns. In particular, by using the first equation the first unknown from the last $m-1$ equations is eliminated, then the new second equation is used in order to eliminate the second unknown from the last $m-2$ equations, etc.

- In Gaussian elimination, **elementary row operations** are applied to the augmented matrix $[\mathbf{A} \mid \mathbf{b}]$, in order to transform $\mathbf{A}$ to **upper triangular form**, without changing the solution of the original system.

• Let $[\mathbf{A} \mid \mathbf{b}] \in \mathbb{R}^{m\times(m+1)}$ be the augmented matrix associated with a linear system $\mathbf{Ax} = \mathbf{b}$ and let $\mathbf{r}_i = [a_i^1, a_i^2, \dots, a_i^m \mid b_i]$ be the i-th row of $[\mathbf{A} \mid \mathbf{b}]$, where $i \in \{1,2,\dots,m\}$. The allowed elementary row operations are

(a) row switching, denoted by $\mathbf{r}_i \leftrightarrow \mathbf{r}_j$,

(b) row multiplication, denoted by $\mathbf{r}_i \leftarrow \alpha\mathbf{r}_i$ with $\alpha \neq 0$, and

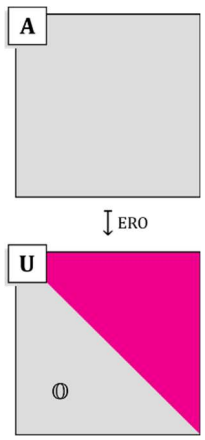
(c) row addition, denoted by $\mathbf{r}_i \leftarrow \mathbf{r}_i + \mathbf{r}_j$.

Elementary row operations can be used in order to compute the inverse of a non-singular matrix $\mathbf{A} \in \mathbb{R}^{m\times m}$; in particular,

$$[\mathbf{A} \mid \mathbf{I}_m] \overset{\text{ERO}}{\longmapsto} [\mathbf{I}_m \mid \mathbf{A}^{-1}]$$

• **Forward elimination** is an algorithm that exploits elementary row operations in order to transform the matrix $\mathbf{A}$ into an upper triangular matrix $\mathbf{U} \in \mathbb{R}^{m\times m}$ whose diagonal entries are non-vanishing. The resulting system is $\mathbf{Ux} = \mathbf{c}$. Schematically,

$$\underbrace{\begin{bmatrix} \times & \times & \times \\ \otimes & \times & \times \\ \times & \times & \times \end{bmatrix}}_{\mathbf{A}} \overset{\text{ERO}}{\longmapsto} \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ \otimes & \times & \times \end{bmatrix} \overset{\text{ERO}}{\longmapsto} \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \otimes & \times \end{bmatrix} \overset{\text{ERO}}{\longmapsto} \underbrace{\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \end{bmatrix}}_{\mathbf{U}},$$



where ERO stands for «elementary row operations», $\otimes$ is the eliminated element, and $\times$ are the unprocessed entries that need to be eliminated in the next stages.

• Since $\det(\mathbf{A}) \neq 0$, the given equations can be aranged so that $a_1^1 \neq 0$. In the k-th stage of forward elimination, the strictly lower triangular part of $\mathbf{A}$ at column k has to be elimminated. To do so, define the so-called **multipliers**

$$\ell_i^k = a_i^k / a_k^k \quad \forall i \in \{k+1, \dots, m\}$$

and apply the elementary row operations

$$\mathbf{r}_i \leftarrow \mathbf{r}_i - \ell_i^k \mathbf{r}_k \quad \forall i \in \{k+1, \dots, m\}.$$

If the diagonal entry of the remaining (unprocessed) part of the matrix is vanishing, forward elimination fails, since the corresponding multiplier can not be defined. When this is the case in stage k, the k-th row of the augmented matrix can be switched with any subsequent row whose k-th component is non-vanishing. Further, in floating-point arithmetic, to minimize propagation of numerical errors, the row whose k-th component is absolutely larger than the k-th components of the remaining rows is switched with the k-th row. The resulting algorithm is called Gaussian elimination with partial pivoting, while the first non-vanishing entry at each row of the resulting triangular matrix $\mathbf{U}$ is called the pivot of that row.

• **Backward substitution** is an algorithm for solving an upper triangular system $\mathbf{Ux} = \mathbf{c}$, in particular,

$$x_m = \frac{c_m}{u_m^m}, \qquad x_i = \frac{1}{u_i^i}\left(c_i - \sum_{s=i+1}^{m} u_i^s x_k\right) \quad \forall i \in \{1,2,\dots,m-1\}.$$

The cubic complexity of Gaussian elimination limits its usefulness for very large-scale systems. For instance, assuming that the time per flop is 10^{-9} s*, Gaussian elimination delivers the solution of a system with* $m = 10^6$ *after approximately* 21 *years. A common practice to get over this performance bound is to exploit parallelization and the sparsity of the coefficient matrix, whenever possible.*

$$(\mathbf{\Lambda}_k)_i^j = \begin{cases} 1, & i = j; \\ -\ell_i^k, & i \geq k+1; \\ 0, & \text{otherwise.} \end{cases}$$

If $\mathbf{A}$ *is symmetric and positive definite, then it can be factorized according to the Cholesky decomposition* $\mathbf{A} = \mathbf{L}\mathbf{L}^{\mathsf{T}}$*, where the matrix* $\mathbf{L}$ *is lower triangular and has positive entries on its main diagonal.*

- Gaussian elimination for a system of m unknowns requires roughly $2m^3/3$ floating-point operations (flops). The number of flops that are required by a method is called the (arithmetic) **complexity** of that method.

The LU decomposition

- Forward elimination can be written as a sequence of matrix transformations of the form

$$(\mathbf{\Lambda}_{m-1} \cdots \mathbf{\Lambda}_2 \mathbf{\Lambda}_1)\mathbf{A}\mathbf{x} = (\mathbf{\Lambda}_{m-1} \cdots \mathbf{\Lambda}_2 \mathbf{\Lambda}_1)\mathbf{b},$$

such that $(\mathbf{\Lambda}_{m-1} \cdots \mathbf{\Lambda}_2 \mathbf{\Lambda}_1)\mathbf{A} = \mathbf{U}$. Each **elimination matrix** $\mathbf{\Lambda}_k$ is lower triangular with unity entries on its main diagonal, and hence, it is non-singular. By setting $\mathbf{L} = (\mathbf{\Lambda}_{m-1} \cdots \mathbf{\Lambda}_2 \mathbf{\Lambda}_1)^{-1}$ the matrix $\mathbf{A}$ can be written as

$$\mathbf{A} = \mathbf{L}\mathbf{U}.$$

This is the so-called **LU decomoposition** (or factorization) of **A.**

- Let m, n be natural numbers greater than unity and consider a matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$ and a sequence $\mathbf{b}^1$, $\mathbf{b}^2$, ..., $\mathbf{b}^n \in \mathbb{R}^m$ of right-hand side vectors. The n systems $\mathbf{A}\mathbf{x}^k = \mathbf{b}^k$ can be written as $\mathbf{A}\mathbf{X} = \mathbf{B}$, where

$$\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^n] \in \mathbb{R}^{m \times n}, \qquad \mathbf{B} = [\mathbf{b}^1, \mathbf{b}^2, \dots, \mathbf{b}^n] \in \mathbb{R}^{m \times n}.$$

Systems of linear equations with common coefficient matrix and more than one right-hand side vectors are called **multiple right-hand side problems**.

- When a multiple right-hand side problem $\mathbf{A}\mathbf{X} = \mathbf{B}$ has to be solved, the forward elimination stage of Gaussian elimination is common for all right-hand side vectros. To avoid refactorizing the matrix $\mathbf{A}$ for each right-hand side, the LU decomposition of $\mathbf{A}$ is employed and the original problem is written as

$$\mathbf{A}\mathbf{X} = \mathbf{B} \Leftrightarrow \mathbf{L}\underbrace{\mathbf{U}\mathbf{X}}_{\mathbf{V}} = \mathbf{B} \Leftrightarrow \mathbf{L}\mathbf{V} = \mathbf{B}.$$

Then $\mathbf{V}$ is determined by solving $\mathbf{L}\mathbf{V} = \mathbf{B}$ and the sought matrix $\mathbf{X}$ is determined by solving the upper triangular system $\mathbf{U}\mathbf{X} = \mathbf{V}$.

Matrix norms

- A **matrix norm** is defined as a mapping $\|\star\|: \mathbb{R}^{m \times m} \to \mathbb{R}$ that satisfies the following properties;
  - (a) $\|\mathbf{A}\| \geq 0$ for all $\mathbf{A} \in \mathbb{R}^{m \times m}$;
  - (b) $\|\mathbf{A}\| = 0$ if, and only if, $\mathbf{A} = \mathbf{0} \in \mathbb{R}^{m \times m}$;
  - (c) $\|\alpha\mathbf{A}\| = |\alpha| \cdot \|\mathbf{A}\|$ for all $\alpha \in \mathbb{R}$ and all $\mathbf{A} \in \mathbb{R}^{m \times m}$;
  - (d) $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$ for all $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times m}$.

If, in addition, $\|\mathbf{AB}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{B}\|$ for all $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times m}$, then the matrix norm is called **sub-multiplicative**.

- Let $\|\star\|_p \colon \mathbb{R}^m \to \mathbb{R}$ be the ℓ^p vector norm, where $1 \leq p \leq \infty$. It can be shown that there are $m, M > 0$, such that

$$m\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_p \leq M\|\mathbf{x}\|_\infty \; \forall \mathbf{x} \in \mathbb{R}^m,$$

and hence,

$$\frac{\|\mathbf{Ax}\|_p}{\|\mathbf{x}\|_p} \leq \frac{M}{m}\frac{\|\mathbf{Ax}\|_\infty}{\|\mathbf{x}\|_\infty} = \frac{M}{m}\frac{\max\limits_{i\in\{1,\dots,m\}} |\sum_{s=1}^m a_i^s x_s|}{\|\mathbf{x}\|_\infty} \leq \frac{M}{m}\max_{i\in\{1,\dots,m\}}\left(\sum_{s=1}^m |a_i^s|\right) < \infty.$$

The mapping $\|\star\|_p \colon \mathbb{R}^{m\times m} \to \mathbb{R}$ with values

$$\|\mathbf{A}\|_p = \max_{\mathbf{x}\in\mathbb{R}^m\setminus\{\mathbf{0}\}} \frac{\|\mathbf{Ax}\|_p}{\|\mathbf{x}\|_p}$$

is the so-called **induced** (or operator) **norm** of $\mathbf{A}$.

- The definition of the induced norm implies that the inequality $\|\mathbf{Ax}\|_p \leq \|\mathbf{A}\|_p\|\mathbf{x}\|_p$ holds for all $\mathbf{x} \in \mathbb{R}^m$.

The induced norms give the maximal amplification factor that is caused when a matrix is applied to a vector.

- The most commonly used induced norms are

$$\|\mathbf{A}\|_1 = \max_{j\in\{1,\dots,m\}}\left(\sum_{s=1}^m |a_s^j|\right), \quad \|\mathbf{A}\|_2 = \sqrt{\max(\sigma(\mathbf{A}^\top\mathbf{A}))},$$
$$\|\mathbf{A}\|_\infty = \max_{i\in\{1,\dots,m\}}\left(\sum_{s=1}^m |a_i^s|\right)$$

for $p \in \{1,2,\infty\}$, respectively.

In addition to the induced norms, the entry-wise norms

$$[\![\mathbf{A}]\!]_p = \left(\sum_{i=1}^m\sum_{j=1}^m |a_i^j|^p\right)^{1/p}$$

are commonly employed. The entry-wise norm for $p = 2$ is called the Frobenius norm and is often denoted by $\|\star\|_F$.

Recall that $\sigma(\star)$ is the set of eigenvalues.

Stability and conditioning

- Consider the linear system $\mathbf{Ax} = \mathbf{b}$ and let $\delta\mathbf{b}$ be a change in the right-hand side vector $\mathbf{b}$ that gives rise to a variation $\delta\mathbf{x}$ in the solution $\mathbf{x}$. Then,

$$\mathbf{A}(\mathbf{x}+\delta\mathbf{x}) = \mathbf{b}+\delta\mathbf{b} \Leftrightarrow \mathbf{Ax}+\mathbf{A}\delta\mathbf{x} = \mathbf{b}+\delta\mathbf{b} \Leftrightarrow \mathbf{A}\delta\mathbf{x} = \delta\mathbf{b},$$

where the first equivalence follows from the linearity of $\mathbf{A}$, while the second equivalence follows from the linear system itself. If $\mathbf{A}$ is non-singular, then $\delta\mathbf{x} = \mathbf{A}^{-1}\delta\mathbf{b}$, and hence,

$$\|\delta\mathbf{x}\|_p = \|\mathbf{A}^{-1}\delta\mathbf{b}\|_p \leq \|\mathbf{A}^{-1}\|_p\|\delta\mathbf{b}\|_p. \text{ (I)}$$

Further,

$$\mathbf{Ax} = \mathbf{b} \Rightarrow \|\mathbf{Ax}\|_p = \|\mathbf{b}\|_p \Rightarrow \|\mathbf{A}\|_p \|\mathbf{x}\|_p \geq \|\mathbf{b}\|_p. \text{ (II)}$$

The so-called **sensitivity** $\|\delta\mathbf{x}\|_p / \|\mathbf{x}\|_p$ is then bounded as

$$\frac{\|\delta\mathbf{x}\|_p}{\|\mathbf{x}\|_p} \overset{\text{(I)}}{\leq} \frac{\|\mathbf{A}^{-1}\|_p \|\delta\mathbf{b}\|_p}{\|\mathbf{x}\|_p} \overset{\text{(II)}}{\leq} \|\mathbf{A}\|_p \|\mathbf{A}^{-1}\|_p \frac{\|\delta\mathbf{b}\|_p}{\|\mathbf{b}\|_p},$$

where $\kappa_p(\mathbf{A}) = \|\mathbf{A}\|_p \|\mathbf{A}^{-1}\|_p$ is the **condition number** of $\mathbf{A}$.

Small values of $\kappa_p(\mathbf{A})$ indicate a weak sensitivity of the solution to changes in the data and the corresponding system is said to be well-conditioned. On the other hand, if $\kappa_p(\mathbf{A}) \gg 1$, then the solution may be sensitive to changes in the data and the corresponding linear system is called ill-conditioned or «close to ill-posed».

- A linear system that
  - (a) has (existence)
  - (b) exactly one solution (uniqueness)
  - (c) that is not significantly affected by small changes in the right-hand side vector (stability)

is said to be **well-posed**.

Worked Examples

1.1 Determine a condition among the scalars α, β, γ so that the system $x + 4y - 3z = \alpha$, $3x - 2y + 2z = \beta$, $x - 10y + 8z = \gamma$ has a solution.

SOLUTION

Suppose that the given system has a solution. The coefficient and the augmented matrices of the given system are

$$\mathbf{A} = \begin{bmatrix} 1 & 4 & -3 \\ 3 & -2 & 2 \\ 1 & -10 & 8 \end{bmatrix}, \qquad [\mathbf{A} \mid \mathbf{b}] = \left[\begin{array}{ccc|c} 1 & 4 & -3 & \alpha \\ 3 & -2 & 2 & \beta \\ 1 & -10 & 8 & \gamma \end{array}\right].$$

Since the given system is assumed to have a solution (not necessarily unique), the rank of the coefficient matrix coincides with the rank of the augmented matrix, that is,

$$\text{rank}(\mathbf{A}) = \text{rank}([\mathbf{A} \mid \mathbf{b}]).$$

The matrix $\mathbf{A}$ is transformed into an upper triangular matrix by the elementary row operations that have been defined in the forward elimination algorithm; in particular,

$$\left[\begin{array}{ccc|c} 1 & 4 & -3 & \alpha \\ 3 & -2 & 2 & \beta \\ 1 & -10 & 8 & \gamma \end{array}\right] \overset{\mathbf{r}_2 \leftarrow \mathbf{r}_2 - 3\mathbf{r}_1}{\longmapsto} \left[\begin{array}{ccc|c} 1 & 4 & -3 & \alpha \\ 0 & -14 & 11 & \beta - 3\alpha \\ 1 & -10 & 8 & \gamma \end{array}\right] \overset{\mathbf{r}_3 \leftarrow \mathbf{r}_3 - \mathbf{r}_1}{\longmapsto}$$

$$\left[\begin{array}{ccc|c} 1 & 4 & -3 & \alpha \\ 0 & -14 & 11 & \beta - 3\alpha \\ 0 & -14 & 11 & \gamma - \alpha \end{array}\right] \overset{r_3 \leftarrow r_3 - r_2}{\mapsto} \left[\begin{array}{ccc|c} 1 & 4 & -3 & \alpha \\ 0 & -14 & 11 & \beta - 3\alpha \\ 0 & 0 & 0 & 2\alpha - \beta + \gamma \end{array}\right].$$

A pivot is the first non-zero entry in a row. In row-echelon form, the number of pivots equals the rank of a matrix.

If $2\alpha - \beta + \gamma \neq 0$, then due to the third equation, the given system has no solution, and hence, it must be $2\alpha - \beta + \gamma = 0$.

1.2 Estimate the arithmetic complexity of forwad eliminnation.

SOLUTION

Let $\mathbf{A} \in \mathbb{R}^{m \times m}$ and consider the well posed system $\mathbf{Ax} = \mathbf{b}$, which we intend to solve with the Gaussian elimination algorithm. To count the required flops, we need the following results. Let $S_m = 1 + 2 + \cdots + m$ and mark that S_m can be written as

$$S_m = m + (m-1) + (m-2) + \cdots + 1.$$

Adding the two expressions for S_m yields

$$2S_m = m(m+1) \Leftrightarrow \sum_{n=1}^{m} n = \frac{m(m+1)}{2}.$$

Then, write

$$(n-1)^3 = n^3 - 3n^2 + 3n - 1 \Leftrightarrow n^3 - (n-1)^3 = 3n^2 - 3n + 1.$$

Summing the left side from $n = 1$ to m yields m^3, since the sum telescopes, that is,

$$1^3 - 0^3 + 2^3 - 1^3 + 3^3 - 2^3 + \cdots + (m-1)^3 - (m-2)^3 + m^3 - (m-1)^3 = m^3.$$

Thus, from $n^3 - (n-1)^3 = 3n^2 - 3n + 1$ we obtain

$$m^3 = \sum_{n=1}^{m} 3n^2 - 3n + 1 \Leftrightarrow m^3 = 3\sum_{n=1}^{m} n^2 - 3\sum_{n=1}^{m} n + \sum_{n=1}^{m} 1 \Leftrightarrow$$
$$m^3 = 3\sum_{n=1}^{m} n^2 - 3S_m + m \Leftrightarrow \sum_{n=1}^{m} n^2 = \frac{m(m+1)(2m+1)}{6}.$$

At stage k (column) of forward elimination, each new zero – out of the $m - k$ that are to be eliminated – requires one division, $m - (k-1)$ multiplications and equally many subtractions, that is, the total number flops is

$$(m-k)(2m-2k+3) = 2m^2 - 4km + 3m + 2k^2 - 3k.$$

Summation from $k = 1$ up to $m - 1$ results in

$$\frac{2m^3}{3} + \frac{m^2}{2} - \frac{7m}{6} = \mathcal{O}\left(\frac{2m^3}{3}\right).$$

Recall that $f(x) = \mathcal{O}(g(x))$, if, and only if, there exists a positive real number M and a real number x_0 such that

$$|f(x)| \leq Mg(x) \quad \forall x \geq x_0.$$

1.3 **Let $\mathbf{A} \in \mathbb{R}^{m\times m}$. Prove that $\|\mathbf{Ax}\|_p \leq \|\mathbf{A}\|_p\|\mathbf{x}\|_p$ for all $\mathbf{x} \in \mathbb{R}^m$.**

SOLUTION

If $\mathbf{x} = \mathbf{0}$, then $\|\mathbf{A0}\|_p = \|\mathbf{0}\|_p = 0$ and the equality holds in the given inequality. If $\mathbf{x} \neq \mathbf{0}$, then $\|\mathbf{x}\|_p \neq 0$, and hence,

$$\left\|\mathbf{A}\frac{\mathbf{x}}{\|\mathbf{x}\|_p}\right\|_p = \left\|\frac{1}{\|\mathbf{x}\|_p}\mathbf{Ax}\right\|_p = \frac{\|\mathbf{Ax}\|_p}{\|\mathbf{x}\|_p} \overset{\text{(I)}}{\leq} \max_{\mathbf{x}\in\mathbb{R}^m\setminus\{\mathbf{0}\}} \frac{\|\mathbf{Ax}\|_p}{\|\mathbf{x}\|_p} = \|\mathbf{A}\|_p.$$

where inequality (I) follows from the definition of the maximum.

1.4 **Let $\mathbf{A} \in \mathbb{R}^{m\times m}$. Prove that the induced matrix norms can be written as $\|\mathbf{A}\|_p = \max\{\|\mathbf{Ax}\|_p \mid \mathbf{x} \in \mathbb{R}^m \text{ and } \|\mathbf{x}\|_p = 1\}$.**

SOLUTION

We have

$$\|\mathbf{A}\|_p = \max_{\mathbf{x}\in\mathbb{R}^m\setminus\{\mathbf{0}\}} \frac{\|\mathbf{Ax}\|_p}{\|\mathbf{x}\|_p} = \max_{\mathbf{x}\in\mathbb{R}^m\setminus\{\mathbf{0}\}} \left\|\mathbf{A}\frac{\mathbf{x}}{\|\mathbf{x}\|_p}\right\|_p = \max_{\|\mathbf{y}\|_p=1} \|\mathbf{Ay}\|_p,$$

where in the last equality we define the vector $\mathbf{y} = \mathbf{x}/\|\mathbf{x}\|_p$ whose ℓ^p vector norm is unity.

Computing Lab

Problem description

- Let $(x_i, y_i) \in \mathbb{R}^2$, with $i \in \{1,2,\ldots,m\}$, be m measurements that approximately pass through a line $f(x) = \alpha x + \beta$, with $\alpha, \beta \in \mathbb{R}$; that is,

$$\alpha_1 x_1 + \alpha_0 \cong y_1, \qquad \alpha_1 x_2 + \alpha_0 \cong y_2, \qquad \ldots, \qquad \alpha_1 x_m + \alpha_0 \cong y_m.$$

Using matrix notation, these equations can be written as

$$\underbrace{\begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_m & 1 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \alpha_1 \\ \alpha_0 \end{bmatrix}}_{\mathbf{x}} \cong \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}}_{\mathbf{b}} \Leftrightarrow \mathbf{Ax} \cong \mathbf{b}.$$

When the number of equations is greater than the number of the unknowns, then the system is said to be **overdetermined**. Since $\mathbf{Ax} \cong \mathbf{b}$, we introduce a residual vector $\mathbf{r} = \mathbf{Ax} - \mathbf{b} \cong \mathbf{0}$, which we aim to minimize; that is, we consider the solution to the problem

$$\min_{\mathbf{x}\in\mathbb{R}^m} \mathcal{f}(\mathbf{x}) = \frac{1}{2}\|\mathbf{r}\|_2^2 = \frac{1}{2}\|\mathbf{Ax} - \mathbf{b}\|_2^2$$

to be the so-called **least square** solution of the overdetermined system. It can be shown that $\mathcal{f}$ attains a minimum at a point $\mathbf{x} \in \mathbb{R}^m$ that satisfies the system

$$\mathbf{A}^\top\mathbf{A}\mathbf{x} = \mathbf{A}^\top\mathbf{b},$$

which is called the system of **normal equations**. The system of normal equations always has a solution, but not necessarily a unique one. One way to prove this existence result is by using the **singular value decomposition (SVD)**. The proof is left as an exercise.

Any matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ *can be factorized as*

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top = [\mathbf{U}_1 \quad \mathbf{U}_2]\begin{bmatrix}\mathbf{D} & \mathbf{O}\\ \mathbf{O} & \mathbf{O}\end{bmatrix}\begin{bmatrix}\mathbf{V}_1^\top\\ \mathbf{V}_2^\top\end{bmatrix},$$

where the matrices $\mathbf{U} \in \mathbb{R}^{m\times m}$ *and* $\mathbf{V} \in \mathbb{R}^{n\times n}$ *are orthogonal,*

$$\mathbf{U}^\top\mathbf{U} = \mathbf{U}\mathbf{U}^\top = \mathbf{I}_m, \qquad \mathbf{V}^\top\mathbf{V} = \mathbf{V}\mathbf{V}^\top = \mathbf{I}_n,$$

and the matrix $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ *is diagonal. The diagonal entries of the* $r \times r$ *block* $\mathbf{D}$ *are* $\sigma_1 > \sigma_2 > \cdots > \sigma_r > 0$*, with* $r \leq \min(m,n)$ *being the rank of* $\mathbf{A}$*, and are called the* ***singular values*** *of* $\mathbf{A}$*.*

- Similarly, a polynomial fit of degree $n > 1$ can be obtained by assuming a function $f: \mathbb{R} \to \mathbb{R}$ whose values are

$$f(x) = \alpha_n x^n + \alpha_{n-1}x^{n-1} + \cdots + \alpha_1 x + \alpha_0.$$

To determine the least squares solution for the real coefficients α_n, α_{n-1}, ..., α_1, α_0, the system of normal equations $\mathbf{A}^\top\mathbf{A}\mathbf{x} = \mathbf{A}^\top\mathbf{b}$ needs to be solved for

$$\mathbf{A} = \begin{bmatrix} x_1^n & x_1^{n-1} & \cdots & x_1 & 1\\ x_2^n & x_2^{n-1} & \cdots & x_2 & 1\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ x_m^n & x_m^{n-1} & \cdots & x_m & 1\end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix}\alpha_n\\ \alpha_{n-1}\\ \vdots\\ \alpha_0\end{bmatrix}.$$

Brief python implementation

- The following code lines introduce the required tools and perform some aesthetic modifications.

```
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams['font.size'] = 18
plt.rcParams['text.usetex'] = True
```

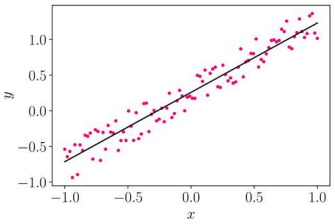
- Here, we generate a data set, by adding some noise to the line that is defined by $f(x) = x$ for all $x \in [-1,1]$, while we also construct the matrix $\mathbf{A}^\top$ with the function `vstack`.

```
m = 100
x = np.linspace(-1.0, 1.0, m)
e = np.ones(x.size)
y = x + 0.5*np.random.rand(x.size)
AT = np.vstack((x, e))
```

- The system of normal equations is solved with the standard NumPy solver. Alternatively, the SciPy solver `scipy.linalg.solve` can be used, since SciPy provides a bigger set of optimized linear algebra routines.

```
a = np.linalg.solve(AT@AT.T, AT@y)
```

- A plot of the data and the linear fit can be obtained and saved with Matplotlib, as depicted in the code snippet below.




```
plt.plot(x, y, '.', color=[240/255,0,120/255])
plt.plot(x, a[0]*x+a[1], 'k')
plt.xlabel(r'$x$')
plt.ylabel(r'$y$')
plt.savefig('lsplot.pdf', bbox_inches='tight')
```

Evaluation Quiz

Double-choice questions

Instructions. *Characterize the following statements as true (T) or false (F) and justify your choice.*

1.1 If **A**, **B** are upper triangular, then **AB** is upper triangular.

1.2 An upper triangular matrix that has a zero entry on its main diagonal is singular.

1.3 The inverse of a non-singular upper triangular matrix is upper triangular.

1.4 If $\mathbf{Ax} = \mathbf{b}$ has a unique solution, $\text{rank}([\mathbf{A} \mid \mathbf{b}]) = \text{rank}(\mathbf{A})$.

1.5 In Gaussian elimination with pivoting, the magnitude of the multipliers is less than unity.

1.6 A singular matrix does not admit an LU decomposition.

1.7 If the induced norm of a matrix is vanishing, then all entries of that matrix are vanishing.

1.8 If **A** is symmetric, then $\|\mathbf{A}\|_1 = \|\mathbf{A}\|_\infty$.

1.9 $\|\mathbf{A}\|_1 = \|\mathbf{A}^\top\|_\infty$ for all matrices **A**.

Exercises and problems

Instructions. *Solve the following tasks and justify each step of the solution process, in detail.*

1.10 Prove that for any non-singular matrix $\mathbf{A} \in \mathbb{R}^{m\times m}$ the condition number is $\kappa_2(\mathbf{A}) = \lambda_m/\lambda_1$, where λ_m and λ_1 are the largest and smallest eigenvalues of $\mathbf{A}^\top\mathbf{A}$.

1.11 Show that Gaussian elimination for tridiagonal $m \times m$ matrices requires $\mathcal{O}(m)$ operations.

1.12 Let $\mathbf{A} \in \mathbb{R}^{m\times m}$. Prove that if $\|\mathbf{A}\|_p < 1$ then $\mathbf{I}_m - \mathbf{A}$ is non-singular. Then, show that $\|(\mathbf{I}_m - \mathbf{A})^{-1}\|_p \leq \left(1 - \|\mathbf{A}\|_p\right)^{-1}$.

1.13 Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times m}$. Prove that if $\mathbf{AB}$ is non-singular, then $\mathbf{A}$ and $\mathbf{B}$ are non-singular.

1.14 Prove that $\kappa_2(\mathbf{A}^2) = [\kappa_2(\mathbf{A})]^2$ for symmetric and positive definite matrices $\mathbf{A} \in \mathbb{R}^{m\times m}$.

1.15 Prove that if a matrix $\mathbf{A} \in \mathbb{R}^{m\times m}$ admits a decomposition $\mathbf{A} = \mathbf{B}\mathbf{B}^\mathsf{T}$ with $\mathbf{B}$ being non-singular, then $\mathbf{A}$ is symmetric and positive definite.

1.16 Prove that $\|\mathbf{A}\|_p = \max_{\mathbf{x}\in\mathbb{R}^m, \|\mathbf{x}\|_p \leq 1} \|\mathbf{A}\mathbf{x}\|_p = \max_{\mathbf{x}\in\mathbb{R}^m, \|\mathbf{x}\|_p = 1} \|\mathbf{A}\mathbf{x}\|_p$.

1.17 Prove that $\kappa_p(\alpha\mathbf{A}) = \kappa_p(\mathbf{A})$ for all $\alpha \in \mathbb{R} \setminus \{0\}$ and all non-singular matrices $\mathbf{A} \in \mathbb{R}^{m\times m}$.

1.18 Prove that the condition number of a matrix is greater than or equal to one

1.19 Prove that the least squares solution of an overdetermined system satisfies the normal equations.

1.20 Prove that the normal equations always have a solution.

Mini project

Instructions. *Work the following task and report by providing detailed procedures, results, code, and graphical representations as part of your resolution.*

1.21 In this computer lab, you will work with the data sets Norris and Pontius from NIST. For each data set,

(a) load the observations,

(b) compute the coefficients of the associated polynomials, and

(c) compare the coefficients with the certified values.

(d) Make plots similar to the NIST plots of both the fit and the residuals.

(e) Compare your results with those obtained with `polyfit`.

To obtain the NIST data sets visit www.itl.nist.gov/div898/strd/ and go to Linear Regression under Dataset Archives