Numerical Programming 1
(MA 3305)
Winter Term 2023/24

Rainer Callies
Department of Mathematics M2
Technical University of Munich

Error messages/proposed corrections please email to
callies@ma.tum.de

These notes follow Prof. Callies's Numerical Programming course in the Winter term 2021/22 and contain an improved and extended version for the Winter term 2023/24. His course is constructed using many different resources like books, other professor's lecture material or codes, none of which are cited here as these are my personal notes.

Contents

1 Direct Solution of Systems of Linear Equations

Fundamental task

Given : $A = (a_{ik}) \in \mathbb{C}^{n\times n}$, $\vec{b} = (b_i) \in \mathbb{C}^{n}$, A^{-1} exists
Required : $\vec{x} = (x_k) \in \mathbb{C}^{n} \ni A\vec{x} = \vec{b}$

Introductory remark

Direct methods mostly are variants of the so-called Gaussian elimination. Direct methods are applied to dense matrices (i.e. almost all $a_{ik} \neq 0$) or to banded matrices ($a_{ik} = 0$ for $|i-k| > \mathrm{const}$).

For large and sparse matrices iterative solutions are preferred in the numerical computation.

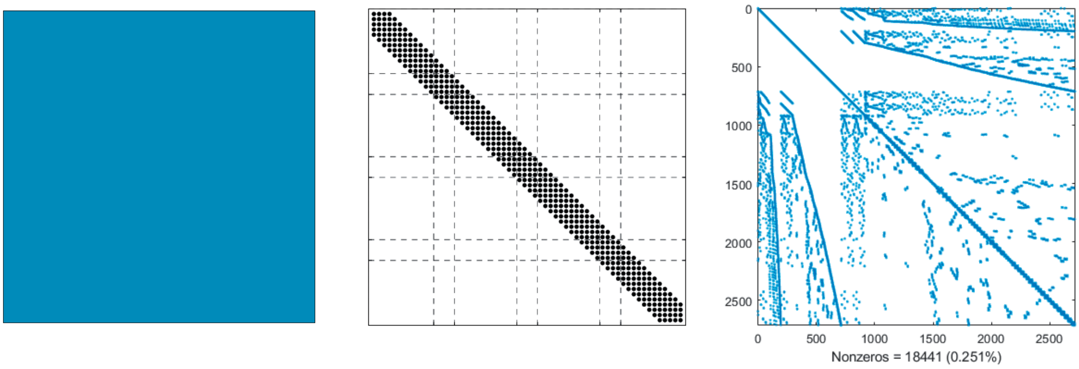


Figure 1: Dense matrix (le.), banded matrix (mi.) and sparse matrix (ri.).

1.1 Repetition from Linear Algebra

1.1.1 Matrices: Examples and Basic Definitions

Basic definitions introduced by examples

- Consider a system of linear equations with two equations and two unknowns $(x_1, x_2) \in \mathbb{R}^2$

$$\begin{array}{rrcr} 2x_1 & +3x_2 & = & 5 \\ -x_1 & +2x_2 & = & -1 \end{array} \quad \Longrightarrow \quad \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 13/7 \\ 3/7 \end{pmatrix}$$

A more convenient notation is the *matrix notation*

$$\underbrace{\begin{pmatrix} 2 & 3 \\ -1 & 2 \end{pmatrix}}_{=:A} \underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_{\vec{x}} = \underbrace{\begin{pmatrix} 5 \\ -1 \end{pmatrix}}_{\vec{b}} \quad \Leftrightarrow \quad A\vec{x} = \vec{b}$$

with the *coefficient matrix* $A \in \mathbb{R}^{2\times 2}$, the *right hand vector* $\vec{b} \in \mathbb{R}^2$ and the *vector of the unknowns* $\vec{x} \in \mathbb{R}^2 = \mathbb{R}^{2\times 1}$.

- *Up to now this is only a new, more concise notation without new functionality.*
- In general a problem of that type can be formulated as

$$\begin{array}{ccc} a_{11}x_1 & +a_{12}x_2 & = & b_1 \\ a_{21}x_1 & +a_{22}x_2 & = & b_2 \end{array} \Leftrightarrow \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \Leftrightarrow A\vec{x} = \vec{b}$$

The vector $\vec{b} = \begin{pmatrix} 5 \\ -1 \end{pmatrix}$ can be understood as a vector, but also as a special matrix with two rows and one column; therefore we write $\vec{b} \in \mathbb{R}^2 = \mathbb{R}^{2\times 1}$.
The first index gives the number of rows, the second the number of columns.

- Let us consider once again the vector $\vec{b} = \begin{pmatrix} 5 \\ -1 \end{pmatrix} \in \mathbb{R}^2 = \mathbb{R}^{2\times 1}$: It is called a *column vector* (matrix with several rows, but only one column), the matrix

$$\vec{d} := (5, -1) \in \mathbb{R}^{1\times 2}$$

is called *row vector* (matrix with only one row, but several columns).

- The transformation of a column vector into a row vector and vice versa is called *transposition*:

$$\vec{b} = \begin{pmatrix} 5 \\ -1 \end{pmatrix} \rightarrow \vec{b}^T = (5, -1) = \vec{d}, \quad \vec{d}^T = \begin{pmatrix} 5 \\ -1 \end{pmatrix} = \vec{b}$$

Convention

The vector symbol (e.g. $\vec{b}$) always represents a column vector. □

Definition: $(\mathbf{m} \times \mathbf{n})$-matrix and system of linear equations

A matrix is a rectangular array of elements (numbers or other mathematical objects) with m rows and n colums for which operations such as addition and multiplication are defined. The elements of a matrix are called *components*. They are from a set K, e.g. $K = \mathbb{R}$ or $K = \mathbb{C}$.

$$A := \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \in K^{m\times n}$$

The position of a particular element a_{ij} of a matrix A is described by two indices, the first index denotes the i-th row and the second the j-th column of A. Sometimes the indices are separated by a comma to improve readability, e.g. $a_{1,11}$ instead of a_{111}.

A system of linear equations consisting of m equations for n unknowns is written in matrix notation

$$\underbrace{\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}}_{=:A} \underbrace{\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}}_{\vec{x}} = \underbrace{\begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}}_{\vec{b}} \quad \Leftrightarrow \quad A\vec{x} = \vec{b}, \quad \vec{x} \in K^n, \vec{b} \in K^m, A \in K^{m \times n}$$

Remark

For such a system of linear equations there may exist exactly one solution, multiple solutions or no solutions at all. □

■ **Example** (system of linear equations in matrix form)

$$\begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 7 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 5 \\ 2 \\ 29 \end{pmatrix}$$

with infinitly many solutions

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ \lambda \\ 4 \end{pmatrix}, \quad \lambda \in \mathbb{R}$$

□

Basic operations applied to modify matrices (explained with examples)

- (Componentwise) addition of two matrices of equal format $A, B \in \mathbb{R}^{m \times n}$

  $$A + B := (a_{ij} + b_{ij})_{i=1,\dots,m;j=1,\dots,n}$$

  or as an example

  $$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} + \begin{pmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} = \begin{pmatrix} 1+4 & 2+5 & 3+6 \\ 4+7 & 5+8 & 6+9 \end{pmatrix} = \begin{pmatrix} 5 & 7 & 9 \\ 11 & 13 & 15 \end{pmatrix}$$

- (Componentwise) multiplication of a matrix $A \in \mathbb{R}^{m \times n}$ with a scalar $\lambda \in \mathbb{R}$

  $$\lambda \cdot A := (\lambda \cdot a_{ij})_{i=1,\dots,m;\ j=1,\dots,n}$$

  or as an example

  $$2 \cdot \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} = \begin{pmatrix} 2 \cdot 1 & 2 \cdot 2 & 2 \cdot 3 \\ 2 \cdot 4 & 2 \cdot 5 & 2 \cdot 6 \end{pmatrix} = \begin{pmatrix} 2 & 4 & 6 \\ 8 & 10 & 12 \end{pmatrix}$$

- Multiplication of a matrix $A \in \mathbb{R}^{p\times m}$ with a matrix $B \in \mathbb{R}^{m\times n}$ according to the rule

  ”row times column”

  results in a matrix $C = A \cdot B \in \mathbb{R}^{p\times n}$ with the components

  $$c_{ij} = \sum_{k=1}^{m} a_{ik} \cdot b_{kj}\,, \quad i = 1,\ldots,p;\, j = 1,\ldots,n$$

  or as an example

  $$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \cdot \begin{pmatrix} 7 & 8 \\ 9 & 4 \\ 2 & 1 \end{pmatrix} = \begin{pmatrix} (1\cdot 7 + 2\cdot 9 + 3\cdot 2) & 19 \\ 85 & 58 \end{pmatrix} = \begin{pmatrix} 31 & 19 \\ 85 & 58 \end{pmatrix}$$

  For matrix multiplication the correct dimensions of both matrices involved are critical

  $$(p \times m)(m \times n) \quad \rightarrow \quad (p \times n)$$

  Frequent error in programming!! □

Example (multiplication of a matrix with a vector)

$$\begin{pmatrix} 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 7 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 5 \\ 2 \\ 29 \\ 3 \end{pmatrix}$$

□

Remark

The multiplication of a matrix $A \in \mathbb{R}^{p\times m}$ with another matrix $B \in \mathbb{R}^{m\times n}$ can also be seen as n matrix-vector-multiplications, if we interpret the columns of B as column vectors

$$C = A \cdot B = (\vec{a}_1, \ldots, \vec{a}_m) \cdot (\vec{b}_1, \ldots, \vec{b}_n) = \left(A\vec{b}_1, \ldots, A\vec{b}_n\right)$$

Example

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \cdot \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \cdot \left(\begin{pmatrix} 5 \\ 7 \end{pmatrix}, \begin{pmatrix} 6 \\ 8 \end{pmatrix} \right) = \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix}$$

□

Special case

A vector can be seen as a matrix with one column and one row, respectively

$$\vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n = \mathbb{R}^{n\times 1}, \quad \vec{x}^T = (x_1, \dots, x_n) \in \mathbb{R}^{1\times n}$$

According to the rules of matrix multiplication we obtain

$$\vec{x} \in \mathbb{R}^n, \vec{y} \in \mathbb{R}^m, A \in \mathbb{R}^{n\times m} \quad \Rightarrow \quad \vec{x}^T Ay \in \mathbb{R} = \mathbb{R}^{1\times 1}$$

The final result is a scalar. □

■ **Example**

$$(1,2)\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = (1,2)\begin{pmatrix} 14 \\ 32 \end{pmatrix} = 78$$

□

Properties of the matrix multiplication

Let $A, B, \dots$ be matrices of the proper dimensions, then

$$\begin{array}{rcll} (A_1 + A_2)B & = & A_1B + A_2B & \text{(distributive)} \\ A(B_1 + B_2) & = & AB_1 + AB_2 & \text{(distributive)} \\ A(BC) & = & (AB)C & \text{(associative)} \\ & & & \\ I^{m\times m}A & = & AI^{n\times n} = A; \quad A \in \mathbb{R}^{m\times n} & (m \neq n\text{: 2 identity matrices)} \\ AB & \overset{mostly}{\neq} & BA & \text{(in general not commutative)} \end{array}$$

Remark

The rules are analogously extended to $A, A_1, A_2 \in \mathbb{C}^{m\times n}$, $B, B_1, B_2 \in \mathbb{C}^{n\times q}$, $C \in \mathbb{C}^{q\times r}$, $m, n, q, r \in \mathbb{N}$. □

1.1.2 Important $(n \times n)$-Matrices

Diagonal matrix D and unit matrix I

$$D := \text{diag}(d_1, \dots, d_n) := \begin{pmatrix} d_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & d_n \end{pmatrix}, \quad d_i \in \mathbb{R}/\mathbb{C} \;\; \forall i \in \{1, \dots, n\}$$

$$I := I^{n\times n} := I_n := \text{diag}(1, \dots, 1)$$

Upper (right) triangular matrix $U \in \mathbb{C}^{n \times n}$ and lower (left) triangular matrix $L \in \mathbb{C}^{n \times n}$

$$U := \begin{pmatrix} r_{11} & \cdots & \cdots & r_{1n} \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & r_{nn} \end{pmatrix}, \quad L := \begin{pmatrix} l_{11} & \cdots & \cdots & 0 \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ l_{n1} & \cdots & \cdots & l_{nn} \end{pmatrix}$$

For $l_{11} = \ldots = l_{nn} = 0$: strictly lower triangular matrix,
for $l_{11} = \ldots = l_{nn} = 1$: normalized lower triangular matrix.

1.1.3 Transposed and Conjugate Transposed Matrices

Definition: transposed matrix

The *transpose of a matrix* $A \in \mathbb{R}^{n \times m}$ is an operator which flips a matrix over its diagonal, i.e. the elements of the i-th row of the matrix A are the elements of the i-th column of another matrix written as $A^T \in \mathbb{R}^{m \times n}$

$$A = \begin{pmatrix} a_{11} & \ldots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \ldots & a_{nm} \end{pmatrix} \in \mathbb{R}^{n \times m} \quad \Rightarrow \quad A^T = \begin{pmatrix} a_{11} & \ldots & a_{n1} \\ \vdots & & \vdots \\ a_{1m} & \ldots & a_{nm} \end{pmatrix} \in \mathbb{R}^{m \times n}$$

Example

$$A := \begin{pmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad \Rightarrow \quad A^T = \begin{pmatrix} 4 & 7 \\ 5 & 8 \\ 6 & 9 \end{pmatrix}$$

Properties

Let $A, B \in \mathbb{R}^{n \times m}, C \in \mathbb{R}^{m \times p}$ be matrices and $\lambda \in \mathbb{R}$, then

$$\begin{aligned} (\lambda A + B)^T &= \lambda A^T + B^T \\ (A^T)^T &= A \\ (AC)^T &= C^T A^T \end{aligned}$$

Definition: conjugate transposed matrix

For $A \in \mathbb{C}^{n \times m}$ the *conjugate transpose or adjoint matrix* $A^H \in \mathbb{C}^{m \times n}$ is defined by

$$A = \begin{pmatrix} a_{11} & \ldots & a_{1m} \\ \vdots & & \vdots \\ a_{n1} & \ldots & a_{nm} \end{pmatrix} \in \mathbb{C}^{n \times m} \quad \Rightarrow \quad A^H = \begin{pmatrix} \bar{a}_{11} & \ldots & \bar{a}_{n1} \\ \vdots & & \vdots \\ \bar{a}_{1m} & \ldots & \bar{a}_{nm} \end{pmatrix} \in \mathbb{C}^{m \times n}$$

i.e. the i-th row of A^H is the complex conjugate of the i-th column of A. $\bar{a} \in \mathbb{C}$ denotes the complex conjugate of $a \in \mathbb{C}$. □

Properties

Let $A, B \in \mathbb{C}^{n\times m}, C \in \mathbb{C}^{m\times p}$ be matrices and $\lambda \in \mathbb{C}$, then

$$(\lambda A + B)^H = \bar{\lambda} A^H + B^H, \;\; (A^H)^H = A, \;\; (AC)^H = C^H A^H$$

Definition: symmetric and hermitian matrices

A matrix $A \in \mathbb{R}^{n\times n}$ is called *symmetric*, if $A^T = A$.

A matrix $A \in \mathbb{C}^{n\times n}$ is called *hermitian or self-adjoint*, if $A^H = A$.

Definition: positive definite matrix

$A \in \mathbb{C}^{n\times n}$ is called *positive definite*, if

$$A^H = A \quad \wedge \quad \vec{x}^H A \vec{x} = (\bar{x}_1, \ldots, \bar{x}_n) \cdot A \cdot \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} > 0 \quad \forall \vec{x} \neq \vec{0} \in \mathbb{C}^n .$$

□

1.1.4 Determinant of a Square Matrix

Definition: determinant

For every matrix $A \in \mathbb{R}^{n\times n}$ the determinant is recursively defined

(1) The determinant of a 2×2 matrix is defined by

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \Rightarrow \quad \det A := \begin{vmatrix} a & b \\ c & d \end{vmatrix} := ad - bc$$

(2) We express the determinant of an $n \times n$ matrix, $n \geq 3$, by considering the elements in the first column and the respective submatrices A_{j1}

$$\det A := \begin{vmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \ldots & a_{nn} \end{vmatrix} := \sum_{j=1}^{n} (-1)^{j+1} a_{j1} \cdot \det A_{j1}$$

The matrix $A_{ji} \in \mathbb{R}^{(n-1)\times(n-1)}$ is the matrix obtained from $A \in \mathbb{R}^{n\times n}$ by removing the j-th row and the i-th column.

Attention: Calculating a determinant in that way only is of theoretical importance, but not suited for numerical calculation because of the enormous numerical effort of $\mathcal{O}(n!)$ operations for a dense matrix $A \in \mathbb{R}^{n\times n}$!

■ Example

$$\begin{aligned}\det A &:= \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix} \\ &= (-1)^2 \cdot 1 \cdot \begin{vmatrix} 5 & 6 \\ 8 & 9 \end{vmatrix} + (-1)^3 \cdot 4 \cdot \begin{vmatrix} 2 & 3 \\ 8 & 9 \end{vmatrix} + (-1)^4 \cdot 7 \cdot \begin{vmatrix} 2 & 3 \\ 5 & 6 \end{vmatrix}\end{aligned}$$

□

Remarks

- Instead of expanding in the first column we can apply the same strategy to the i-th column of A

$$\det A = \sum_{j=1}^{n} (-1)^{j+i} a_{ji} \cdot \det A_{ji}$$

- From the properties of the determinant summarized below we obtain that for every $n \times n$ matrix $A \in \mathbb{R}^{n\times n}$ *there exists exactly one* function $\det$

$$\begin{aligned}\det : \quad \mathbb{R}^{n\times n} &\longrightarrow \mathbb{R} \\ A &\longmapsto \det A\end{aligned}$$

The determinant of a real-valued matrix *always* is a real number!

- Analogously: definition of the determinant of a matrix $A \in \mathbb{C}^{n\times n}$

Basic properties of determinants

Let be $A \in \mathbb{R}^{n\times n}$ then

(1) $\det A = \det A^T$, therefore all statements on columns are also valid for rows. Instead of expanding in the j-th column, we also can expand in the j-th row.

(2) $\det$ is linear with respect to each column (row) of A. Example:

$$\det \tilde{A} := \det(\vec{a}_1, \vec{a}_2 + \lambda \vec{c}, \vec{a}_3) = \det(\vec{a}_1, \vec{a}_2, \vec{a}_3) + \lambda \det(\vec{a}_1, \vec{c}, \vec{a}_3)$$

(3) $\det I_n = 1 \qquad (I_n := I \in \mathbb{R}^{n\times n})$

(4) Anti-symmetry: Interchanging two columns of A causes a sign change of the determinant

$$\det(\vec{a}_1, \dots, \vec{a}_i, \dots, \vec{a}_j, \dots, \vec{a}_n) = -\det(\vec{a}_1, \dots, \vec{a}_j, \dots, \vec{a}_i, \dots, \vec{a}_n)$$

(5) If two rows or two columns of A are equal then $\det A = 0$ (from (4)).

(6) The determinant of the product of two matrices is equal to the product of the determinants of these matrices

$$\det(AB) = \det A \cdot \det B$$

(7) The determinant of A remains unchanged, if the λ-fold of the j-th column is added to the i-th column.

(8) Determinants of special matrices:

$$\begin{aligned}
\det(I_n) &= 1 \\
\det(D) &= d_1 \cdot \ldots \cdot d_n = \prod_{i=1}^{n} d_i \\
\det(L) &= l_{11} \cdot \ldots \cdot l_{nn} = \prod_{i=1}^{n} l_{ii} \\
\det(U) &= r_{11} \cdot \ldots \cdot r_{nn} = \prod_{i=1}^{n} r_{ii}
\end{aligned}$$

1.1.5 Invertible $n \times n-$Matrix

Definition: inverse of an $n \times n-$matrix

An $n \times n-$Matrix A is invertible with the inverse matrix A^{-1} if

$$AA^{-1} = A^{-1}A = I_n$$

If the inverse A^{-1} exists, then A is called *nonsingular or invertible*. □

Theorem

The inverse of $A \in \mathbb{R}^{n\times n}$ is uniquely determined.

■ Example

Rotation matrix and inverse rotation through an angle $\alpha \;\rightarrow\; -\alpha$

$$A := \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \rightarrow \quad A^{-1} = \begin{pmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

□

Properties of inverse matrices

$$\begin{aligned}
AA^{-1} &= A^{-1}A = I_n \\
(A^{-1})^{-1} &= A \\
(AB)^{-1} &= B^{-1}A^{-1} \\
(A^T)^{-1} &= (A^{-1})^T
\end{aligned}$$

and A^T is invertible if and only if A is invertible.

Remark: The order of the inverse matrices changes if we calculate $(AB)^{-1}$.

The inverse is only of theoretical interest

Solving systems of linear equations $A\vec{x}=\vec{b}$ by **explicitely calculating** A^{-1} and then multiplying $A^{-1}(A\vec{x})=\vec{x}=A^{-1}b$ is a **severe malpractice**. □

Reversible multiplication with a nonsingular matrix

The multiplication of a system of linear equations $A\vec{x}=\vec{b}$, $A\in\mathbb{R}^{n\times m}, \vec{x}\in\mathbb{R}^m, \vec{b}\in\mathbb{R}^n$, with a nonsingular matrix $C\in\mathbb{R}^{n\times n}$ is possible without losing any information, because another multiplication with C^{-1} restores the original problem

$$A\vec{x}=\vec{b} \;\;\to\;\; CA\vec{x}=C\vec{b} \;\;\to\;\; C^{-1}CA\vec{x}=A\vec{x}=\vec{b}=C^{-1}C\vec{b}$$

□

Definition: rank of a matrix

Consider $A\in\mathbb{R}^{m\times n}$. The rank $\mathrm{rk}\,A$ of a matrix A is the dimension of the vector space generated (or spanned) by its columns; this corresponds to the maximum number of linearly independent columns of A.

Theorem

Consider $A\in\mathbb{R}^{m\times n}$. The number of linearly independent column vectors is equal to the number of linearly independent row vectors. □

■ Example

$$A:=\begin{pmatrix} 1 & 0 & 1 \\ -2 & -3 & 1 \\ 3 & 3 & 0 \end{pmatrix}$$

has rank 2: the first two rows are linearly independent, so the rank is at least 2, but all three rows are linearly dependent (the third is equal to the second subtracted from the first) so the rank must be less than 3. □

Theorem

For a square matrix $A\in\mathbb{R}^{n\times n}$ the *following statements are equivalent*:

- A^{-1} exists
- All columns of $A=(\vec{a}_1,\ldots,\vec{a}_n)$ are linearly independent, i.e.

$$A\vec{x}=\sum_{i=1}^{n}x_i\vec{a}_i=\vec{0} \;\;\Rightarrow\;\; \vec{x}=(x_1,\ldots,x_n)=\vec{0}$$

- $A\vec{x}=\vec{0} \;\;\Rightarrow\;\; \vec{x}=\vec{0}$
- $\mathrm{rk}\,A=n$
- $\det A\neq 0$

□

Application formulae

Determinant of the inverse:

$$1 = \det I_n = \det(AA^{-1}) \quad \Rightarrow \quad \det(A^{-1}) = \frac{1}{\det A}$$

Similarity transformation:

$$B := S^{-1}AS \text{ with } \operatorname{rk} S = n \quad \Rightarrow \quad \det B = \frac{1}{\det S} \cdot \det A \cdot \det S = \det A,$$

In that case the determinant is not changed. This transformation is important e.g. when the basis is changed or for the solution of eigenvalue problems.

Block decomposition:

Suppose $F \in \mathbb{R}^{p\times p}, A \in \mathbb{R}^{n\times n}, D \in \mathbb{R}^{p-n\times p-n}$ with $p > n$ and A invertible (i.e. A^{-1} exists). Let F be the following block matrix

$$F := \begin{pmatrix} A & B \\ C & D \end{pmatrix} \quad \Longrightarrow \quad \det F = \det A \cdot \det(D - CA^{-1}B)$$

This is because

$$\begin{pmatrix} A & 0 \\ C & (D - CA^{-1}B) \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} I & -A^{-1}B \\ 0 & I \end{pmatrix}$$

□

Definition: orthogonal and unitary matrices

$A \in \mathbb{R}^{n\times n}$ *orthogonal* $:\Leftrightarrow$ $A^{-1} = A^T$, that is $AA^T = A^TA = I_n$;

$A \in \mathbb{C}^{n\times n}$ *unitary* $:\Leftrightarrow$ $A^{-1} = A^H$, that is $AA^H = A^HA = I_n$;

$A \in \mathbb{C}^{n\times n}$ *normal* $:\Leftrightarrow$ $AA^H = A^HA$

Every hermitian and every unitary matrix is normal.

1.1.6 Vector norms

Definition

For $\vec{x} \in \mathbb{C}^n$ we define

$$\begin{aligned} \|\vec{x}\|_1 &:= |x_1| + |x_2| + \ldots + |x_n| \\ \|\vec{x}\|_2 &:= \sqrt{\sum_{i=1}^{n} |x_i|^2} \\ \|\vec{x}\|_\infty &:= \max\{|x_1|, |x_2|, \ldots, |x_n|\} \end{aligned}$$

Properties

For $p = 1, 2, \infty$ the above definitions define norms, i.e. definite, strictly homogeneous and sub-additive mappings $\mathbb{C}^n \to \mathbb{R}^+$ with the following properties

$$\begin{aligned} \|\vec{x}\|_p &> 0 \quad \forall \vec{x} \neq \vec{0} \ \wedge \ \|\vec{x}\|_p = 0 \Leftrightarrow \vec{x} = \vec{0} \\ \|a\vec{x}\|_p &= |a| \cdot \|\vec{x}\|_p \quad \forall a \in \mathbb{C}, \forall \vec{x} \\ \|\vec{x} + \vec{y}\|_p &\leq \|\vec{x}\|_p + \|\vec{y}\|_p \quad \forall \vec{x}, \vec{y} \end{aligned}$$

□

1.2 Gaussian Elimination of Invertible Matrices

Fundamental task

Given : $A = (a_{ik}) \in \mathbb{R}^{n\times n}$, $\vec{b} = (b_i) \in \mathbb{R}^n$, A^{-1} exists
Required : $\vec{x} = (x_k) \in \mathbb{R}^n \ni A\vec{x} = \vec{b}$

■ Example for the solution strategy

$$\begin{aligned} 3x_1 + 2x_2 &= 3 \\ 9x_1 + 8x_2 &= 7 \end{aligned} \quad \Longleftrightarrow \quad \begin{pmatrix} 3 & 2 \\ 9 & 8 \end{pmatrix} \vec{x} = \begin{pmatrix} 3 \\ 7 \end{pmatrix} \quad \Longleftrightarrow: \quad \left(\begin{array}{cc|c} 3 & 2 & 3 \\ 9 & 8 & 7 \end{array}\right)$$

We subtract the threefold of the first line from the second

$$\begin{pmatrix} 3 & 2 \\ 0 & 2 \end{pmatrix} \vec{x} = \begin{pmatrix} 3 \\ -2 \end{pmatrix}$$

The goal is the special (upper) triangular form of the transformed matrix in order to calculate the components of $\vec{x}$ step by step. We start with the last line/row

$$2x_2 = -2 \quad \Rightarrow \quad x_2 = -1$$

$\Rightarrow$ insert into 1st line: $3x_1 = 3 - 2x_2 = 5 \quad \Rightarrow \quad x_1 = 5/3 \quad \Rightarrow \quad \vec{x} = \begin{pmatrix} 5/3 \\ -1 \end{pmatrix}$

Gauss-Jordan algorithm

For the solution of a system of n linear equations for n unknowns by hand we apply the following two-stage solution strategy:

- **Forward elimination**

  The matrix A and the *right hand side* $\vec{b}$ are transformed synchronously until A is reduced to an upper triangular matrix U

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \rightarrow U := \begin{pmatrix} r_{11} & \cdots & \cdots & r_{1n} \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & r_{nn} \end{pmatrix}, \quad \vec{b} \rightarrow \vec{c}$$

  Only transformations are allowed that do not change the solution $\vec{x}$!

- **Back substitution**

  The original system of linear equations is reduced to *row echelon form*

$$U\vec{x} = \vec{c} \quad \in \mathbb{R}^n$$

  The special structure is utilized.

  Starting from the last row (i.e. backwards) we calculate

– x_n from the n-th row,

– insert the result for x_n into the $(n-1)$-th row and determine x_{n-1} from row $n-1$,

– ...

– finally insert the results for $x_n, x_{n-1}, \ldots, x_2$ into the first row and determine x_1 from row 1.

Standard elementary row operations

- Multiply the i-th row by a nonzero scalar $\alpha \in \mathbb{R}$.
  Our abbreviation: $S(\alpha; i)$
- Add the α-multiple ($\alpha \in \mathbb{R}$) of the i-th row to the j-th row.
  Our abbreviation: $N(\alpha; i \to j)$
- Swap the positions of the i-th and the j-th row.
  Our abbreviation: $P(i, j)$

■ Reference example

Consider the system of linear equations

$$\begin{array}{rrrcr} x_1 & -x_2 & & = & 2 \\ -x_1 & +x_2 & +x_3 & = & 1 \\ 2x_1 & & -x_3 & = & 3 \end{array}$$

In matrix notation the *forward elimination* reads as

$$\left(\begin{array}{rrr|r} 1 & -1 & 0 & 2 \\ -1 & 1 & 1 & 1 \\ 2 & 0 & -1 & 3 \end{array}\right) \xrightarrow{N(1;1\to 2)} \left(\begin{array}{rrr|r} 1 & -1 & 0 & 2 \\ 0 & 0 & 1 & 3 \\ 2 & 0 & -1 & 3 \end{array}\right)$$

$$\xrightarrow{N(-2;1\to 3)} \left(\begin{array}{rrr|r} 1 & -1 & 0 & 2 \\ 0 & 0 & 1 & 3 \\ 0 & 2 & -1 & -1 \end{array}\right) \xrightarrow{P(2,3)} \left(\begin{array}{rrr|r} 1 & -1 & 0 & 2 \\ 0 & 2 & -1 & -1 \\ 0 & 0 & 1 & 3 \end{array}\right) =: U$$

Finally we obtain the following system of linear equations

$$\begin{array}{rrrcr} x_1 & -x_2 & & = & 2 \\ & 2x_2 & -x_3 & = & -1 \\ & & x_3 & = & 3 \end{array}$$

Back substitution yields

- from the last row: $x_3 = 3$
- Insertion into the 2nd row:
  $2x_2 - x_3 = 2x_2 - 3 = -1 \;\Rightarrow\; x_2 = 1$

- Insertion into the 1st row:
  $x_1 - x_2 = x_1 - 1 = 2 \;\;\Rightarrow\;\; x_1 = 3$

The *solution vector* is $\vec{x} = \begin{pmatrix} 3 \\ 1 \\ 3 \end{pmatrix}$

□

Details of the forward elimination (c.f. reference example)

- To transform the matrix A and the vector $\vec{b}$ synchronously we write them together into one matrix which is A extended by $\vec{b}$ as an additional column (the so-called *augmented matrix*). In our example

$$\left(\begin{array}{rrr|r} 1 & -1 & 0 & 2 \\ -1 & 1 & 1 & 1 \\ 2 & 0 & -1 & 3 \end{array}\right)$$

- Using the elementary row operations all elements below the diagonal (i.e. in the lower left part of the matrix) are made to zero column by column and starting with the first column. This is possible only if the respective diagonal element is not equal to zero. In the reference example

$$A^{(0)} := A = \left(\begin{array}{rrr|r} 1 & -1 & 0 & 2 \\ -1 & 1 & 1 & 1 \\ 2 & 0 & -1 & 3 \end{array}\right) \longrightarrow \left(\begin{array}{rrr|r} \textcircled{1} & -1 & 0 & 2 \\ 0 & 0 & 1 & 3 \\ 0 & 2 & -1 & -1 \end{array}\right) =: A^{(1)}$$

Completion of the j-th column in this way is called the j*-th principal step* $A^{(j-1)} \to A^{(j)}$.

Here the j-th row is called *pivot row* and the (encircled) diagonal element at the position a_{jj} is called *pivot* (or *leading coefficient*). In the j-th principal step multiples of the j-th row are subtracted from all rows below.

$$\begin{array}{rcccccccl} A =: A^{(0)} & \to & A^{(1)} & \to & \dots & \to & A^{(n-1)} & =: U \\ \vec{b} =: \vec{b}^{(0)} & \to & \vec{b}^{(1)} & \to & \dots & \to & \vec{b}^{(n-1)} & =: \vec{c} \end{array}$$

- It is necessary that the pivot $*$ is not equal to zero.

  If this is not the case, then the position of the j-th row is swaped with the position of one of the rows $j+1, \dots, n$ *below* it. In the reference example this step is rather simple

$$\left(\begin{array}{rrr|r} 1 & -1 & 0 & 2 \\ 0 & \textcircled{0} & 1 & 3 \\ 0 & 2 & -1 & -1 \end{array}\right) \xrightarrow{P(2,3)} \left(\begin{array}{rrr|r} 1 & -1 & 0 & 2 \\ 0 & \textcircled{2} & -1 & -1 \\ 0 & 0 & 1 & 3 \end{array}\right) = U$$

- After the j-th principal step the rows $1, \dots, j$ of the matrix are finalized and remain unchanged during the following steps.

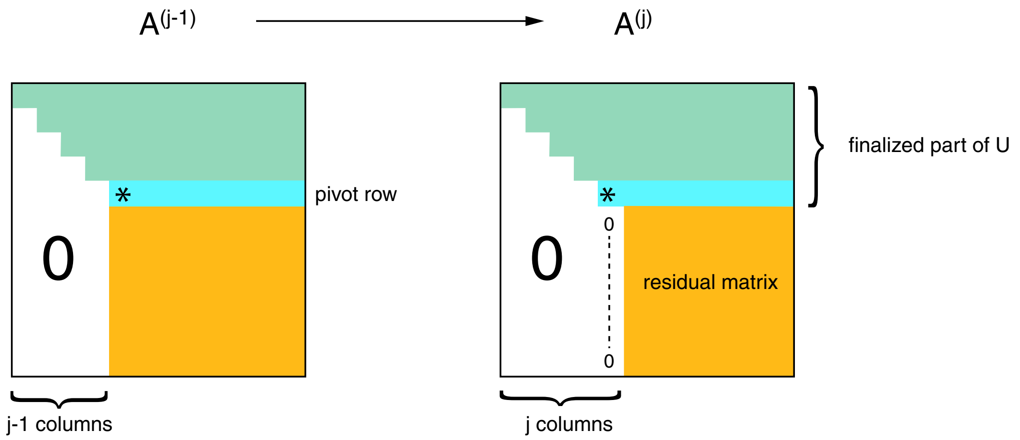


Figure 2: j-th principal step $A^{(j-1)} \to A^{(j)}$, $*$ marks the pivot.

- Determination of the rank of the upper triangular matrix U:

  For the upper triangular matrix U we immediately obtain the rank defined as the number of linearly independent column vectors. In our example

$$\operatorname{rk} U = \operatorname{rk}\left(\begin{array}{rrr|r} 1 & -1 & 0 & 2 \\ 0 & 2 & -1 & -1 \\ 0 & 0 & 1 & 3 \end{array}\right) = 3 \quad \Rightarrow \quad U^{-1} \text{ exists}$$

  and therefore a unique solution of the system of linear equations exists

$$\text{formally:} \quad U\vec{x} = \vec{c} \;\Rightarrow\; \vec{x} = U^{-1}(U\vec{x}) = U^{-1}\vec{c}$$

  We never calculate the inverse explicitely, but prefer the much cheaper back substitution.

The forward elimination presented in detail above can also be carried out using *elementary transformation matrices*. New: As a by-product we then obtain the determinant of the original matrix $A \in \mathbb{R}^{n\times n}$.

The total picture: Gauss-Jordan algorithm in matrix notation

An arbitrary matrix $A \in \mathbb{R}^{n\times n}$ is reshaped by multiplying the complete linear system with a non-singular elementary transformation matrix $Q \in \mathbb{R}^{n\times n}$ from the left, the solution vector $\vec{x}$ of the transformed system remains unchanged.

$$A\vec{x} = \vec{b} \quad \Leftrightarrow \quad (Q\cdot A)\vec{x} = (Q\cdot\vec{b})\,, \quad A, Q \in \mathbb{R}^{n\times n} \;\wedge\; \det(Q) \neq 0\,,\; \vec{x}, \vec{b} \in \mathbb{R}^n$$

The goal of the reshaping process is to transform the linear system to another one, which can be more easily solved (because e.g. QA is a lower triangular matrix). Because $\det(Q) \neq 0$ there is no information lost.

However, the determinant may change

$$\det(QA) = \det(Q)\cdot\det(A)$$

□

Definition: elementary transformation matrices

with the following effect by multiplication to a matrix A from the left.
Multiplication from the right has the respective effect on the columns!

(A) **Multiply the i-th row by a factor $\alpha \in \mathbb{R}$.**

$$S(\alpha;i) \;:=\; \text{diag}\,(d_1,\ldots,d_n) = \begin{pmatrix} d_1 & & 0 \\ & \ddots & \\ 0 & & d_n \end{pmatrix} \in \mathbb{R}^{n\times n}$$

$$\text{with} \quad d_j \;:=\; \begin{cases} 1 & , \; j \neq i \\ \alpha & , \; j = i \end{cases} \;, \; j \in \{1,\ldots,n\}\,, \; \text{we choose:} \;\; \alpha \neq 0$$

Inverse: $S^{-1}(\alpha;i) = \text{diag}\,(1/d_1,\ldots,1/d_n)$
$\Rightarrow$ $S^{-1} \neq S^T$ (not orthogonal), $S^T S = S S^T$ (normal)
Determinant: $\det S(\alpha;i) = \alpha$

(B) **Swap the positions of the i-th and the j-th row.**

The permutation matrix $P(i,j)$ differs from the identity matrix $I_n = \text{diag}\,(1,\ldots,1) \in \mathbb{R}^{n\times n}$ only in four elements that are separately listed

$$\begin{aligned} P(i,j)_{ii} &= 0\,, & P(i,j)_{ij} &= 1 \\ P(i,j)_{ji} &= 1\,, & P(i,j)_{jj} &= 0 \end{aligned}$$

Inverse: $P(i,j)^{-1} = P(i,j)$
$\Rightarrow$ $P^{-1} = P^T = P$ (orthogonal and therefore normal)
Determinant: $\det P(i,j) = -1$

(C) **Add the α-multiple of the i-th row to the j-th row.**

The transformation matrix $N(\alpha; i \to j)$ differs from the identity matrix $I_n = \text{diag}\,(1,\ldots,1) \in \mathbb{R}^{n\times n}$ only in one single element listed below

$$\big(N(\alpha; i \to j)\big)_{ji} = \alpha$$

Inverse: $N(\alpha; i \to j)^{-1} = N(-\alpha; i \to j)$
$\Rightarrow$ $N^{-1} \neq N^T$ (not orthogonal), $N^T N \neq N N^T$ (not normal)
Determinant: $\det N(\alpha; i \to j) = +1$

■ **Example**

$$S(8.0;2) \cdot \begin{pmatrix} 1 & 2 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 8.0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 56 & 64 \end{pmatrix}$$

□

■ **Example**

$$P(2,3) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{4\times 4} \;\Rightarrow\; P(2,3) \cdot \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 3 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 7 & 8 & 9 \\ 4 & 5 & 6 \\ 3 & 2 & 1 \end{pmatrix}$$

□

■ Example

$$N(-3;1\to 3) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -3 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{4\times 4}$$

$$\Rightarrow\ N(-3;1\to 3)\cdot \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 3 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 4 & 2 & 0 \\ 3 & 2 & 1 \end{pmatrix}$$

□

■ Example: forward elimination with pivoting

$$\begin{pmatrix} 0 & 3 & 1 & 1 \\ 2 & 2 & -1 & -2 \\ 4 & 4 & -3 & -1 \\ -2 & 4 & 4 & 2 \end{pmatrix} \qquad A \qquad \text{(pivot} = 0\,!)$$

$$\begin{pmatrix} 2 & 2 & -1 & -2 \\ 0 & 3 & 1 & 1 \\ 4 & 4 & -3 & -1 \\ -2 & 4 & 4 & 2 \end{pmatrix} \qquad A^{(0)} = P(1,2)\cdot A$$

$$\begin{pmatrix} 2 & 2 & -1 & -2 \\ 0 & 3 & 1 & 1 \\ 0 & 0 & -1 & 3 \\ 0 & 6 & 3 & 0 \end{pmatrix} \qquad A^{(1)} = N(1;1\to 4)\cdot N(-2;1\to 3)\cdot A^{(0)}$$

$$\begin{pmatrix} 2 & 2 & -1 & -2 \\ 0 & 3 & 1 & 1 \\ 0 & 0 & -1 & 3 \\ 0 & 0 & 1 & -2 \end{pmatrix} \qquad A^{(2)} = N(-2;2\to 4)\cdot A^{(1)}$$

$$\begin{pmatrix} 2 & 2 & -1 & -2 \\ 0 & 3 & 1 & 1 \\ 0 & 0 & -1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad U = A^{(3)} = N(1;3\to 4)\cdot A^{(2)}$$

For the determinant we obtain

$$\begin{aligned} \det U &= 2\cdot 3\cdot(-1)\cdot 1 = -6 \\ &= \det(N(1;3\to 4))\cdot\det(A^{(2)}) = \det(A^{(2)}) \\ \det(A^{(2)}) &= \det(N(-2;2\to 4))\cdot\det(A^{(1)}) \\ \det(A^{(1)}) &= \det(N(1;1\to 4))\cdot\det(N(-2;1\to 3))\cdot\det(A^{(0)}) = \det(A^{(0)}) \\ \det(A^{(0)}) &= \det P(1,2)\cdot\det A = -\det A \ \Rightarrow\ \det A = 6 \end{aligned}$$

□

Because of the special structure of U, its determinant can be calculated easily using the recursive definition of the determinant.

■ Example

$$\begin{vmatrix} 2 & 2 & -1 & -2 \\ 0 & 3 & 1 & 1 \\ 0 & 0 & -1 & 3 \\ 0 & 0 & 0 & 1 \end{vmatrix} = (-1)^{1+1} \cdot 2 \cdot \begin{vmatrix} 3 & 1 & 1 \\ 0 & -1 & 3 \\ 0 & 0 & 1 \end{vmatrix} = 2 \cdot (-1)^{1+1} \cdot 3 \cdot \begin{vmatrix} -1 & 3 \\ 0 & 1 \end{vmatrix} = -6$$

□

The prototype algorithm

Of course, we do not program expensive matrix-matrix-multiplications, but skillfully omit unnecessary operations.

Algorithm 1: Gauss-Jordan algorithm without pivoting

```
■ Input:   $A \in \mathbb{C}^{n \times n}, b \in \mathbb{C}^n$

■ Step 1.  Forward elimination
*Transition:*  $A^{(j)} \to A^{(j+1)}, b^{(j)} \to b^{(j+1)}$
for $j = 1$ to $n-1$ do
    *Store the elements already finalized: $r_{jj}$ is called pivot*
    for $k = j$ to $n$ do
        $r_{jk} = a_{jk}^{(j)}$
    $c_j = b_j^{(j)}$

    *Elimination of the element at position $(i,j)$, elimination factor $l_{ij}$:*
    for $i = j+1$ to $n$ do
        $l_{ij} = a_{ij}^{(j)} / r_{jj}$

        for $k = j+1$ to $n$ do
            $a_{ik}^{(j+1)} = a_{ik}^{(j)} - l_{ij} r_{jk}$

        $b_i^{(j+1)} = b_i^{(j)} - l_{ij} c_j$
$c_n = b_n^{(n)}$ ;  $r_{nn} = a_{nn}^{(n)}$

■ Step 2.  Back substitution

for $i = n$ to $1$ do
    $x_i = \left( c_i - \sum_{j=i+1}^{n} r_{ij} x_j \right) / r_{ii}$
```

The algorithm is for demonstration purposes only. For production code use existing software packages (e.g. MatLab).

Remark to storage organization:

Total storage requirements can be reduced to $A(1:n,1:n)$, $b(1:n)$; for that we substitute in the *final* algorithm

$$l_{ij} \to a_{ij},\, r_{jk} \to a_{jk},\, a_{ik}^{(j)} \to a_{ik},\, a_{ik}^{(j+1)} \to a_{ik}, \quad c_j \to b_j,\, b_i^{(j)} \to b_i,\, b_i^{(j+1)} \to b_i$$

So we skillfully use storage cells the information in which is no longer needed. Such a program is difficult to read and to debug!

Computational effort for the Gauss-Jordan algorithm:

$$\mathcal{O}(n^3/3) \text{ subtractions and multiplications } + \mathcal{O}(n^2/2) \text{ divisions}$$

As a by-product we obtain the determinant almost for free (if we solve a linear system) or by $\mathcal{O}(2n^3/3)$ operations (if we are only interested in the determinant). Compare this with $\mathcal{O}(n!)$ operations necessary for the theoretically-based approach in chap. 1.1.4. □

1.3 LU-factorization

The $N(\alpha; i \to j)$ with $i < j$ are normalized lower triangular matrices. The product of two lower triangular matrices again is a lower triangular matrix (i.e. $L_1 \cdot L_2 = L_3$), schematically

$$\begin{pmatrix} * & & 0 \\ \vdots & \ddots & \\ * & \dots & * \end{pmatrix} \cdot \begin{pmatrix} * & & 0 \\ \vdots & \ddots & \\ * & \dots & * \end{pmatrix} = \begin{pmatrix} * & & 0 \\ \vdots & \ddots & \\ * & \dots & * \end{pmatrix}$$

In our last example we had

$$\begin{aligned} U &= N(1;3 \to 4) \cdot A^{(2)} \\ &= N(1;3 \to 4) \cdot N(-2;2 \to 4) \cdot A^{(1)} \\ &= N(1;3 \to 4) \cdot N(-2;2 \to 4) \cdot N(1;1 \to 4) \cdot N(-2;1 \to 3) \cdot P(1,2) \cdot A \end{aligned}$$

or after multiplication with the inverse elementary matrices

$$\Rightarrow \quad \underbrace{N(+2;1 \to 3)N(-1;1 \to 4)N(+2;2 \to 4)N(-1;3 \to 4)}_{=:L} \cdot U = P(1,2) \cdot A$$

$$\Rightarrow \quad L \cdot U = PA$$

Therefore Gauss elimination without pivoting is shown to be a factorization of A into a product of a normalized (!) lower triangular matrix L and an upper triangular matrix U. This idea can be directly used to construct the algorithm:

$$A = L \cdot U \quad \Leftrightarrow \quad a_{ik} = \sum_{j=1}^{\min\{i,k\}} l_{ij} r_{jk} \quad \Leftrightarrow \quad a_{ik} = \begin{cases} \sum_{j=1}^{k-1} l_{ij} r_{jk} + l_{ik} r_{kk}\,, & i > k \\ \sum_{j=1}^{i-1} l_{ij} r_{jk} + \underbrace{l_{ii}}_{=1} r_{ik}\,, & i \le k \end{cases}$$

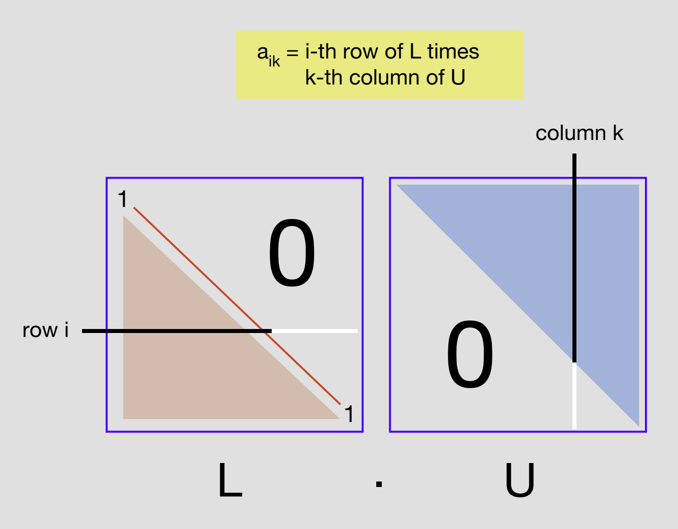


Figure 3: Basic idea of the LU algorithm

■ Example

We start with a given matrix A and want to factorize it according to the scheme in Fig. 3.

$$\begin{pmatrix} 1 & 3 \\ 2 & 8 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ l_{21} & 1 \end{pmatrix} \cdot \begin{pmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{pmatrix}$$

We compare the elements of the resulting matrix $L \cdot U$ after multiplication with those of $A = (a_{ij})$ at the same positions. We proceed row by row:

$$\begin{array}{llcllcl} a_{ij}, & i=1, j=1 & : & 1 & = & 1 \cdot r_{11} & \Rightarrow & r_{11} & = & 1 \\ & i=1, j=2 & : & 3 & = & 1 \cdot r_{12} & \Rightarrow & r_{12} & = & 3 \\ & i=2, j=1 & : & 2 & = & l_{21} \cdot r_{11} & \Rightarrow & l_{21} & = & 2 \\ & i=2, j=2 & : & 8 & = & l_{21} \cdot r_{12} + 1 \cdot r_{22} & \Rightarrow & r_{22} & = & 2 \end{array}$$

In total we get

$$\begin{pmatrix} 1 & 3 \\ 2 & 8 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 3 \\ 0 & 2 \end{pmatrix}$$

□

Algorithm 2: LU-factorization without pivoting

```
■ Input:   $A \in \mathbb{C}^{n\times n}$
for $i = 1$ to $n$ do
    for $k = 1$ to $i-1$ do
        $l_{ik} = \left( a_{ik} - \sum_{j=1}^{k-1} l_{ij} r_{jk} \right) / r_{kk}$
    $l_{ii} = 1$
    for $k = i$ to $n$ do
        $r_{ik} = a_{ik} - \sum_{j=1}^{i-1} l_{ij} r_{jk}$
```

Comparison between Gauss-Jordan algorithm and LU-factorization:

Gauss-Jordan (parallel):

$$U = L^{-1}A \quad \longleftrightarrow \quad \vec{y} = L^{-1}\vec{b} \quad \text{simultaneously}$$

$$U\vec{x} = \vec{y} \quad \text{backward substitution}$$

LU-factorization (sequentially):

$$\begin{aligned} A &= L \cdot U && \text{decomposition} \\ L\vec{y} &= b && \text{forward substitution} \\ U\vec{x} &= \vec{y} && \text{backward substitution} \end{aligned}$$

Exactly the same operations, but in different order. Because the respective operations are independent, the results are exactly the same (even if numerical rounding errors are taken into account).

Benefit of the LU-factorization:

A is factorized once, the triangular matrices are stored efficiently. If $A\vec{x} = \vec{b}$ has to be solved for a new right hand side $\vec{b}$, only a cheap forward and a cheap backward substitution have to be performed. □

Conclusion

We step by step have developed a numerical algorithm for the solution of linear systems $A\vec{x} = \vec{b}$, starting with the calculation by hand.

The idea of using elementary transformation matrices formalizes the procedure and leads to an efficient algorithm. □

1.4 Row Pivoting

In the course of the elimination process it may happen that the *pivot element* is equal to zero. Then a row permutation becomes necessary. This corresponds to the multiplication from the left with a matrix $P(i,j)$.

As the new pivot element we choose one the absolute value of which is ”not too small” (different and mostly heuristic strategies).

Combining all the permutation matrices into a single matrix P and performing the permutation steps (formally) at the very beginning yields the following structure of the Gaussian decomposition or the LU factorization

$$PA = LU\,.$$

Remark

Sometimes also column pivoting is performed. This corresponds to the multiplication with a matrix $P(i,j)$ from the right. Multiplication with elementary transformation matrices from the right has the same effect on the columns as multiplication from the left has on the rows.

By this, also the components of $\vec{x}$ are reordered:

$$\begin{aligned} A\vec{x} = \vec{b} \quad\longrightarrow\quad A(P(i,j)P(i,j)^{-1})\vec{x} &= (AP(i,j))(P(i,j)^{-1}\vec{x}) \\ &= (AP(i,j))(P(i,j)\vec{x}) = \vec{b}\,. \end{aligned}$$

Example

$$\left(\begin{array}{cc|c} 1 & 2 & 3 \\ 4 & 5 & 6 \end{array}\right) \;\Leftrightarrow\; \begin{aligned} 1x_1 + 2x_2 &= 3 \\ 4x_1 + 5x_2 &= 6 \end{aligned}$$

$$\rightarrow\; \begin{aligned} 2x_2 + 1x_1 &= 3 \\ 5x_2 + 4x_1 &= 6 \end{aligned} \;\Leftrightarrow\; \begin{pmatrix} 2 & 1 \\ 5 & 4 \end{pmatrix} \begin{pmatrix} x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} 3 \\ 6 \end{pmatrix}$$

□

Gaussian elimination without pivoting preserves matrix symmetry

Theorem (proof in (A7)):

Let be $A \in \mathbb{R}^{n\times n}$ a symmetric or $A \in \mathbb{C}^{n\times n}$ a hermitian or A a positive/negative definite matrix. After each step of the Gaussian elimination process without pivoting, the residual matrix has the same properties.

Conclusion:

For positive and negative definite matrices, Gaussian elimination without pivoting is always possible, because the residual matrix is also positive and negative definite and $a_{11} > 0, \ldots$

Remark:

To preserve the matrix structure, for symmetric or hermitian matrices only the diagonal elements of the respective residual matrix are candidates for pivot elements (row *and* column permutation necessary!). □

1.5 Iterative Refinement

Iterative refinement is a technique for improving the accuracy of a solution obtained by a direct method. Suppose that the linear system has been solved by the LU factorization.

Denote by $\vec{x}$ the exact solution

$$\vec{x} = A^{-1}\vec{b}$$

and by $\vec{x}^{(0)}$ the computed solution with finite precision (→ next chapter).

The iterative refinement is done as follows: For $i = 0, 1, \dots$ do

(1) Compute the residual $\vec{r}^{(i)} = \vec{b} - A\vec{x}^{(i)}$ *in double precision* (→ next chapter).

(2) Solve the linear system $A\vec{z} = \vec{r}^{(i)}$ using the LU factorization of A (cheap, factorization already computed).

(3) Update the solution setting $\vec{x}^{(i+1)} = \vec{x}^{(i)} + \vec{z}$.

(4) Stop the iteration, if $\|\vec{z}\|/\|\vec{x}^{(i+1)}\| < Tol$.

Because of rounding errors (→ next chapter), $\vec{z}$ is not the exact correction vector.

Why does this technique work:

$$\begin{aligned} \vec{x}^{(i+1)} &= \vec{x}^{(i)} + \vec{z} \\ &= \vec{x}^{(i)} + A^{-1}\vec{r}^{(i)} \\ &= \vec{x}^{(i)} + A^{-1}(\vec{b} - A\vec{x}^{(i)}) = \vec{x}^{(i)} + \vec{x} - \vec{x}^{(i)} = \vec{x} \end{aligned}$$

in absence of rounding errors in step (1,2)! □

1.6 Application of Gauss-Jordan Algorithm to General $n \times n-$Matrices

The same algorithm can be applied to any $n \times n-$matrix – singular or non-singular.

■ **Example** (scaling to the 1st row is done in the beginning)

$$A = \begin{pmatrix} 3 & 0 & 6 \\ 2 & 1 & 4 \\ 3 & 5 & 6 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 0 \\ 0 & 5 & 0 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

The rank of A is 2, because only 2 column vectors are linearly independent; thus the inverse does not exist. □

Table of possible cases for $A\vec{x} = \vec{b}$, $A \in \mathbb{R}^{n \times n}$, $\vec{x}, \vec{b} \in \mathbb{R}^n$

$\vec{b} = \vec{0}$,	$\text{rk}\, A = n$	:	only trivial solution $\vec{x} = \vec{0}$
$\vec{b} = \vec{0}$,	$\text{rk}\, A = r < n$	:	solution with $n - r$ parameters
$\vec{b} \neq \vec{0}$,	$\text{rk}\, A = n$	:	exactly one solution $\vec{x}$
$\vec{b} \neq \vec{0}$,	$n > r = \text{rk}\, A = \text{rk}\,(A\vert\vec{b})$	:	solution with $n - r$ parameters
$\vec{b} \neq \vec{0}$,	$n > r = \text{rk}\, A < \text{rk}\,(A\vert\vec{b})$	:	no solution

The rules can be immediately obtained from the Gauss-Jordan algorithm. □

■ **Example**

$$A\vec{x} = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \vec{x} = \begin{pmatrix} 1 \\ 2 \\ t \end{pmatrix} = \vec{b} \quad \Rightarrow \quad (A|\vec{b}) = \left(\begin{array}{ccc|c} 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & t \end{array}\right)$$

For $t \neq 0$ no solution exists. In this case $2 = \text{rk}\, A < \text{rk}\,(A|\vec{b}) = 3$ and the last row contains a contradiction

$$0 \cdot x_1 + 0 \cdot x_2 + 0 \cdot x_3 = t \neq 0 \qquad \lightning$$

For $t = 0$ we obtain

$$\begin{aligned} x_1 + 2x_3 &= 1 \\ x_2 &= 2 \end{aligned}$$

These are two equations for three unknowns. For every value of one of the unknowns – w.l.o.g. of x_3 – the resulting system can be solved

$$\begin{aligned} x_1 &= 1 - 2x_3 \\ x_2 &= 2 \end{aligned} \quad \Rightarrow \quad \vec{x} = \begin{pmatrix} 1 - 2x_3 \\ 2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix} + \lambda \begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix}, \ \lambda \in \mathbb{R} \text{ mit } \lambda := x_3$$

We obtain a solution with one free parameter (here: x_3). □

How is the forward elimination modified, if $\operatorname{rk} A < n$?

Almost no change! Apply the algorithm as usual. If the first column of a residual matrix is equal to the zero vector and therefore there exists no pivot, then simply jump to the next column and proceed as usual.

■ Example

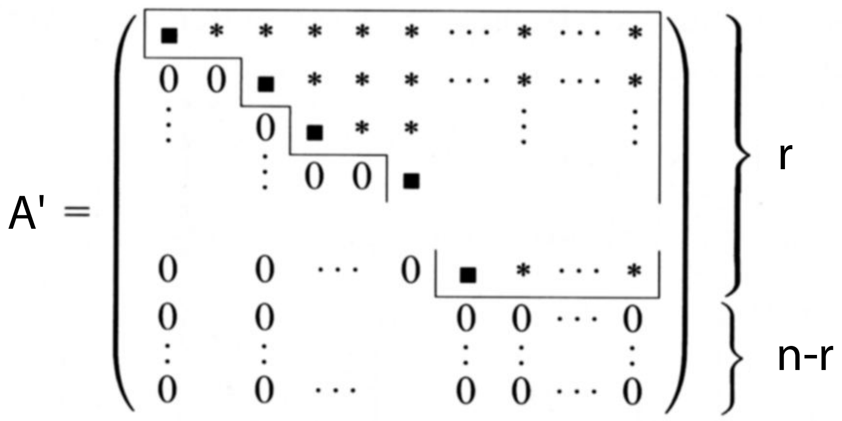
$$\begin{pmatrix} \enclose{circle}{1} & 2 & 3 & 4 \\ 2 & 4 & 3 & 5 \\ -1 & -2 & 3 & 4 \\ -3 & -6 & 0 & 3 \end{pmatrix} \to \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 0 & \enclose{circle}{-3} & -3 \\ 0 & 0 & 6 & 8 \\ 0 & 0 & 9 & 15 \end{pmatrix} \to \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 0 & -3 & -3 \\ 0 & 0 & 0 & \enclose{circle}{2} \\ 0 & 0 & 0 & 6 \end{pmatrix}$$

$$\to \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & 0 & -3 & -3 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

□

Definition: row echelon form

For each row in a matrix $A \in \mathbb{R}^{n\times n}$, if the row does not consist of only zeros, then the left-most non-zero entry is called the leading coefficient (or pivot) of that row. By elementary transformations, one can always order the rows so that for every non-zero row, the leading coefficient is at least on position to the right of the leading coefficient of the row above. If this is the case, then the matrix is said to be in *row echelon form*. The lower left part of the matrix contains only zeros, all of the zero rows are below the non-zero rows: schematically $A \to A'$



The last $n-r$ rows of the matrix in the figure contain zeros only. Black squares are non-zero elements, entries marked by "$*$ " may contain any number. □

Determination of the rank

From the row echelon form we immediately obtain the rank of A: The rank is the number of *row vectors* of A that are non-zero vectors, because all non-zero rows are linearly independent in row echelon form.

2 Error Analysis

2.1 Introduction of Machine Numbers

The system $\mathbb{R}$ of real numbers is unbounded, continuous (without gaps) and uncountable: For a distinct real number there exist other real numbers which are larger and smaller, between two distinct real numbers there are other real numbers.

On a computer only a finite subset $\mathbb{M} \subset \mathbb{R}$ of the set of real numbers $\mathbb{R}$ can be represented exactly. That is the reason why almost any machine operation is affected by rounding errors.

Definition: machine numbers

The set $\mathbb{G}$ of normalized floating-point numbers with t significant digits and the base B is defined by

$$\mathbb{G} = \{g | g := M \cdot B^E \text{ with } M = 0 \text{ or } B^{t-1} \leq |M| < B^t\},$$

$\mathbb{G}$ is unbounded. $\mathbb{M}$ is a subset of $\mathbb{G}$ defined by

$$\mathbb{M} := \{g \in G | \alpha \leq E \leq \beta\}.$$

$\mathbb{M}$ is described by 4 parameters: the *base* $B > 1$ (often: B=2,10), the *number of significant digits* $t \geq 0$ and the *limits* α and β. $M \in \mathbb{Z}$ is called *mantissa* and $E \in \mathbb{Z}$ *exponent*.

■ Example

$B = 10, t = 3, \alpha = -8, \beta = 2: \quad a = 15.8 \quad \rightarrow \quad g = 158 \cdot 10^{-1}, E = -1$ □

Remark

Often for the normalized floating point representation the modified form $a = 0.158 \cdot 10^2$ is used, i.e. instead of the mantissa M the expression $M \cdot B^{-t}$ is used.

■ Example (cont.)

$B = 10, t = 3, \alpha = -8, \beta = 2$

$$\begin{aligned} \mathbb{G} &:= \{g | g := M \cdot 10^E, \ M = 0 \text{ oder } 10^2 \leq |M| < 10^3\} \\ \mathbb{M} &:= \{g \in G | -8 \leq E \leq 2\} \end{aligned}$$

The real numbers below are represented by the following machine numbers

$a = 15.8 \rightarrow g = 158 \cdot 10^{-1}, E = -1 \rightarrow a \in \mathbb{M}$
$b = 15.83977 \rightarrow g = 158 \cdot 10^{-1}, E = -1 \rightarrow b \notin \mathbb{M}$ (rounding error)
$c = 15800000 \rightarrow g = 158 \cdot 10^5, E = 5 \rightarrow c \notin \mathbb{M}$ (overflow) □

Remark

Moreover, machine numbers are not equally spaced along the real line, but they accumulate close to the smallest representable number. In our example, this is $B^{t-1}B^{\alpha} = 10^{-6}$. There are gaps between floating-point numbers. As the numbers get larger, so do the gaps. Moreover, there is a gap around the zero.

Exponent over-/underflow ($E \notin [\alpha, \beta]$) is not often critical nowadays (for reasonably programmed code). □

2.2 Rounding and Rounding Errors

Definition: correct rounding

Correct rounding is the mapping

$$\mathrm{fl} : \mathbb{R} \to \mathbb{G}\,,$$

which assigns to every $r \in \mathbb{R}$ the "nearest" $g \in \mathbb{M}$

$$|\mathrm{fl}(r) - r| \leq |g - r| \;\; \forall g \in \mathbb{G}$$

□

Remarks

A computer not necessarily does correct rounding!

For every $r \in \mathbb{R}$ there exists (neglecting over-/underflows) an M and an E such that

$$M \cdot B^E \leq r \leq (M+1) \cdot B^E$$

□

Conclusion

From the remark we obtain the following limits for the *absolute* and the *relative rounding error* in case of correct (!!) rounding

$$\begin{aligned} |\mathrm{fl}(r) - r| &\leq 0.5 \cdot B^E \\ \left| \frac{\mathrm{fl}(r) - r}{r} \right| &\leq \frac{1}{2|M|} \leq \frac{B}{2} B^{-t}, \quad \text{because} \quad |M| \geq B^{t-1} \end{aligned}$$

□

Definition: machine precision

The *machine precision* is defined as $\varepsilon_0 := \varepsilon_t := B^{1-t}$. □

With that definition we obtain

$$fl(r) = r \cdot (1 + \varepsilon)\,, \quad |\varepsilon| \leq \varepsilon_0/2\,.$$

It can be proven: For correct rounding we need $t+2$ digits on the computer, if the result has to be accurate to t digits!

■ **Example**

For floating point numbers according to IEEE 754: $B = 2$ and

single (32 bit): $\varepsilon_{t=24} \approx 10^{-7}$, double (64 bit): $\varepsilon_{t=53} \approx 2 \cdot 10^{-16}$

The absolute value of the largest number that can be represented using $\beta = 104$ and $\beta = 971$, respectively is:

$g_{max} = B^{t+\beta} \approx 3 \cdot 10^{38}$ and $2 \cdot 10^{308}$, respectively.

The absolute value of the smallest number that can be represented using $\alpha = -149$ and $\beta = -1074$, respectively is:

$g_{min} = B^{t-1+\alpha} \approx 1 \cdot 10^{-38}$ and $2 \cdot 10^{-308}$, respectively.

Single precision hence corresponds to approx. 6-7 decimal digits, double precision to 14-15 decimal digits. □

Correct rounding is so important that we want to have this property fulfilled also for the basic operations $* \in \{+, -, \cdot, /\}$.

Definition

Let be $* \in \{+, -, \cdot, /\}$. An arithmetic is called *ideal*, if

$$\mathrm{fl}(a * b) = (a * b)(1 + \varepsilon) \quad \forall a, b \in \mathbb{M}, \quad |\varepsilon| = |\varepsilon(a, b, *)| \leq \varepsilon_t / 2$$

A real arithmetic satisfies the *strong hypothesis*, if

$$\mathrm{fl}(a * b) = (a * b)(1 + \varepsilon) \quad \forall a, b \in \mathbb{M}, \quad |\varepsilon| = |\varepsilon(a, b, *)| \leq \mathcal{O}(\varepsilon_t / 2)$$

A real arithmetic satisfies the *weak hypothesis*, if

$$\mathrm{fl}(a * b) = a \cdot (1 + \varepsilon_1) * b \cdot (1 + \varepsilon_2) \quad \forall a, b \in \mathbb{M}, \quad |\varepsilon_i| = |\varepsilon_i(a, b, *)| \leq \varepsilon_t / 2$$

Weak arithmetics e.g. show problems when subtracting almost equal numbers (zero or non-zero, wrong sign?).

Ideal rounding exists (IEEE standard) on some machines. The *strong hypothesis* applies for most computers. Fulfillment of the strong hypothesis often is the prerequisite for the mathematical analysis of rounding errors of algorithms. Rounding errors obtained by the above estimates are worst-case estimates and therefore often too pessimistic.

2.3 Error Propagation – Basics

The errors that are investigated in this context are unavoidable and problem-induced, even if a perfect numerical algorithm is chosen. To reduce the errors one has to reformulate the underlying mathematical problem.

Only input errors are considered, other error sources like rounding errors are neglected at this stage.

Input errors may be measurement errors (mostly dominate!), but also all errors (e.g. rounding and discretization errors) made in previous calculation steps: The total numerical problem is often decomposed into several subproblems – see below –, which are solved step by step.

Formulation of the general error propagation problem

Consider the normed vector spaces $(X, \|\cdot\|_X)$ and $(Y, \|\cdot\|_Y)$ and a mapping $f : X \to Y$ (often called ”problem” or ”subproblem”). Examples are $X \in \mathbb{R}^n$ and $Y \in \mathbb{R}^m$.

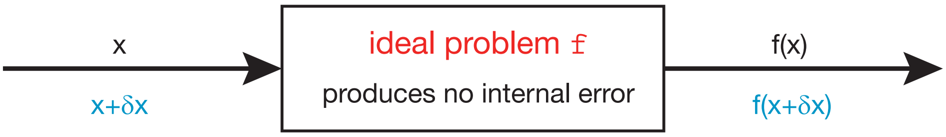


Figure 4: Ideal problem amplifies/attenuates the effect of an input error δx on the result, but produces no internal error itself.

For $x, \delta x \in X$ we define

$$\delta f := f(x + \delta x) - f(x)$$

Important question

How large is the perturbation δf of the solution compared to the perturbation δx of the input data?

Definition (absolute and relative error)

Let be $A := (A_1, \ldots, A_n)$ a tupel of matrices subject to perturbations

$$A \quad \longrightarrow \tilde{A} := (A_1 + \delta A_1, \ldots, A_n + \delta A_n)$$

The *absolute error* E_{abs} and the *relative error* E_{rel} are defined by

$$E_{abs}(A) := \max_{1 \le i \le n} \|\delta A_i\|\,, \quad E_{rel}(A) := \max_{1 \le i \le n} \frac{\|\delta A_i\|}{\|A_i\|}$$

Remark

Vectors and scalars are treated as special cases of matrices. The error definition is rather flexible, but not unique. For instance, one can decompose a vector in different ways into a tuple of subvectors.

Special cases e.g. for a perturbed vector $\vec{x} + \delta\vec{x} \in \mathbb{R}^n$ are the norm error

$$E_{abs}(x) := \|\delta\vec{x}\|\,, \quad E_{rel}(x) := \frac{\|\delta\vec{x}\|}{\|\vec{x}\|}$$

and the componentwise error

$$E_{abs}(x) := \max_{1 \le i \le n} |\delta x_i|\,, \quad E_{rel}(x) := \max_{1 \le i \le n} \frac{|\delta x_i|}{|x_i|}$$

Please notice that in finite dimensional spaces all norms are equivalent! □

Definition (absolute and relative condition for the norm error)

The *absolute condition number* of f in $x \in X$ is

$$\kappa_{abs}(f,x) := \lim_{\delta \to 0} \left(\sup_{\|\delta x\| < \delta} \frac{\|\delta f\|_Y}{\|\delta x\|_X} \right), \qquad \delta f = f(x+\delta x) - f(x)$$

Using the norm allows an approach from different directions!

The *relative condition number* of f in $x \in X$ is

$$\kappa_{rel}(f,x) := \lim_{\delta \to 0} \left(\sup_{\|\delta x\| < \delta} \frac{\|\delta f\|_Y / \|f(x)\|_Y}{\|\delta x\|_X / \|x\|_X} \right)$$

Definition (absolute and relative condition in general)

Norm and componentwise errors are special cases of this general definition.

The *absolute condition number* of f in $x \in X$ is

$$\kappa_{abs}(f,x) := \lim_{\delta \to 0} \left(\sup_{\|\delta x\| < \delta} \frac{E_{abs}(f(x))}{E_{abs}(x)} \right)$$

The *relative condition number* of f in $x \in X$ is

$$\kappa_{rel}(f,x) := \lim_{\delta \to 0} \left(\sup_{\|\delta x\| < \delta} \frac{E_{rel}(f(x))}{E_{rel}(x)} \right)$$

Definition (ill-conditioned problem)

A problem is *ill-conditioned*, if $\kappa_{abs}(f,x) \gg 1$ or $\kappa_{rel}(f,x) \gg 1$.

A problem is *ill-posed*, if $\kappa_{abs}(f,x) \to \infty$ or $\kappa_{rel}(f,x) \to \infty$.

Remark

The condition number characterizes the worst case error!

If the condition number is large, a small perturbation in the input data (e.g. small error in the measured values) may cause large perturbations in the final result.

The condition numbers compress the information about error amplification into one scalar. This often gives helpful information for the optimization of a complete algorithm. □

Remark

Introducing condition numbers can be seen as a linearization of the original error propagation problem. E.g. for the norm error, $\kappa_{abs}(f,x) \geq 0$ is the number for which

$$\|\delta f\| \leq \kappa_{abs}(f,x)\|\delta x\| \;+\; \mathcal{O}\left(\|\delta x\|^2\right) \quad \text{for} \quad \|\delta x\| \to 0$$

□

■ **Example** (one-dimensional case - norm error = component error)

Let be $f \in \mathcal{C}^{n+1}([a,b],\mathbb{R})$ and $a < x < x+h < b$. Taylor's theorem tells us, that there exists the following expansion with at least one $\xi \in [x,x+h]$ such that

$$\begin{aligned} f(x+h) &= f(x)+f'(x)h+\ldots+\frac{1}{n!}f^{(n)}(x)h^n+\frac{1}{(n+1)!}f^{(n+1)}(\xi)h^{n+1} \\ &= f(x)+f'(x)h+\frac{1}{2}f''(x)h^2+\ldots+\frac{1}{n!}f^{(n)}(x)h^n+\mathcal{O}(|h^{n+1}|) \end{aligned}$$

For $f \in \mathcal{C}^2(\mathbb{R},\mathbb{R})$ we especially get $\|\cdot\| = |\cdot|$ and thus from the definition

$$\kappa_{abs}(f,x) = \lim_{\delta x \to 0}\left|\frac{\delta f}{\delta x}\right| = |f'(x)|\,, \quad \kappa_{rel}(f,x) := \frac{|f'(x)|}{|f(x)|/|x|}$$

Now let us investigate two simple functions

$$\begin{aligned} f(x) := 2x \quad &\Rightarrow \quad \kappa_{abs}(f,x) = 2,\, \kappa_{rel}(f,x) = 1 \\ &\qquad \text{input error amplified by factor 2, not critical} \\ f(x) := \sqrt{x},\, x \geq 0 \quad &\Rightarrow \quad \kappa_{abs}(f,x) = \frac{1}{2\sqrt{x}},\, \kappa_{rel}(f,x) = \frac{1}{2} \\ &\qquad \kappa_{abs} \to \infty \text{ for } x \to 0, \text{ but } \kappa_{rel} \text{ not critical} \end{aligned}$$

□

■ **Example** (roots of a polynomial)

Calculate the roots of the polynomial $y(x) = x^2 - 2x + q \;\Rightarrow\; x_{1,2} = 1 \pm \sqrt{1-q}$; we consider $f(q) := x_1 = 1 + \sqrt{1-q}$ for $q := 1-\varepsilon,\, 1 > \varepsilon > 0$ and obtain

$$\begin{aligned} \kappa_{abs}(f,q) &= \frac{1}{2\sqrt{1-q}} = \frac{1}{2\sqrt{\varepsilon}} \to \infty \quad \text{for} \quad \varepsilon \to 0 \\ \kappa_{rel}(f,q) &= \frac{1/(2\sqrt{\varepsilon})}{(1+\sqrt{\varepsilon})/(1-\varepsilon)} = \frac{1-\sqrt{\varepsilon}}{2\sqrt{\varepsilon}} \to \infty \quad \text{for} \quad \varepsilon \to 0 \end{aligned}$$

Therefore it is not a good idea e.g. to calculate the eigenvalues λ of a matrix $A \in \mathbb{R}^{n\times n}$ numerically as the zeros of the characteristic polynomial (equation) $\det(A - \lambda I_n) = 0$. □

■ **Example** (norm error)

Consider $\vec{f} \in \mathcal{C}^2(D,\mathbb{R}^m),\, D \subseteq \mathbb{R}^n$. Multidimensional Taylor expansion with truncation after the linear term yields

$$\delta\vec{f} \doteq Df(\vec{x}) \cdot \delta\vec{x}\,, \quad Df(\vec{x}) \text{ Jacobian}$$

For the calculation of the condition numbers for the norm errors of this problem we use matrix norms (see below) and obtain

$$\|\delta\vec{f}\| \,\dot{\leq}\, \|Df(\vec{x})\| \cdot \|\delta\vec{x}\| \;\Rightarrow\; \kappa_{abs}(\vec{f},\vec{x}) = \|Df(\vec{x})\|\,, \quad \kappa_{rel}(\vec{f},\vec{x}) := \frac{\|Df(\vec{x})\|}{\|\vec{f}(\vec{x})\|/\|\vec{x}\|}$$

□

■ **Example** (relative and absolute condition for the componentwise error)

Consider $f \in \mathcal{C}^2(D, \mathbb{R}),\ D \subseteq \mathbb{R}^n$.

$$\begin{aligned}
\kappa_{rel}(f,\vec{x}) \quad := \quad & \lim_{\delta \to 0} \left(\sup_{\|\delta\vec{x}\|<\delta} \frac{E_{rel}(f(\vec{x}))}{E_{rel}(\vec{x})} \right) = \lim_{\delta \to 0} \left(\sup_{\|\delta\vec{x}\|<\delta} \frac{|f(\vec{x}+\delta\vec{x}) - f(\vec{x})|}{|f(\vec{x})| \cdot E_{rel}(\vec{x})} \right) \\
= \quad & \lim_{\delta \to 0} \left(\sup_{\|\delta\vec{x}\|<\delta} \frac{|\nabla f(\vec{x})^T \cdot \delta\vec{x}|}{|f(\vec{x})| \cdot E_{rel}(\vec{x})} \right) \\
\doteq \quad & \lim_{\delta \to 0} \left(\sup_{\|\delta\vec{x}\|<\delta} \frac{|\nabla f(\vec{x})^T \cdot \delta\vec{x}|}{|f(\vec{x})| \cdot \left(\max\limits_{1 \le i \le n} \dfrac{|\delta x_i|}{|x_i|} \right)} \right)
\end{aligned}$$

Because numerator and denominator are homogeneous in $\delta\vec{x}$, we can simplify

$$\kappa_{rel}(f,\vec{x}) = \sup_{\|\delta\vec{x}\| \neq 0} \frac{|\nabla f(\vec{x})^T \cdot \delta\vec{x}|}{|f(\vec{x})| \cdot \left(\max\limits_{1 \le i \le n} \dfrac{|\delta x_i|}{|x_i|} \right)} \overset{(*)}{=} \frac{\sum\limits_{i=1}^{n} |f_{x_i}(\vec{x})| \cdot |x_i|}{|f(\vec{x})|}$$

How do we get rid of the supremum in step $(*)$? This is mathematically tricky and is given here only for those interested in:

We define componentwise two new auxiliary vectors $\vec{w}$ by $w_i := \delta x_i / x_i$ and $\vec{y}$ by $y_i := f_{x_i}(\vec{x}) \cdot x_i$ for $i = 1, \ldots, n$. We obtain

$$\sup_{\|\delta\vec{x}\| \neq 0} \frac{|\nabla f(\vec{x})^T \cdot \delta\vec{x}|}{\left(\max\limits_{1 \le i \le n} \dfrac{|\delta x_i|}{|x_i|} \right)} = \sup_{\vec{w} \neq \vec{0}} \frac{|\vec{y}^T \cdot \vec{w}|}{\|\vec{w}\|_\infty} \overset{(**)}{=} \|\vec{y}\|_1 = \sum_{i=1}^{n} |f_{x_i}(\vec{x}) \cdot x_i|$$

For $(**)$ we used Hölder's inequality: $\left|\vec{u}^T \cdot \vec{v}\right| \le \|\vec{u}\|_1 \cdot \|\vec{v}\|_\infty$.

For the absolute condition for the componentwise error we get analogously using Hölder's inequality again

$$\begin{aligned}
\kappa_{abs}(f,\vec{x}) \quad := \quad & \lim_{\delta \to 0} \left(\sup_{\|\delta\vec{x}\|<\delta} \frac{E_{abs}(f(\vec{x}))}{E_{abs}(\vec{x})} \right) = \lim_{\delta \to 0} \left(\sup_{\|\delta\vec{x}\|<\delta} \frac{|f(\vec{x}+\delta\vec{x}) - f(\vec{x})|}{\max_{1 \le i \le n} |\delta x_i|} \right) \\
\doteq \quad & \lim_{\delta \to 0} \left(\sup_{\|\delta\vec{x}\|<\delta} \frac{|\nabla f(\vec{x})^T \cdot \delta\vec{x}|}{\max_{1 \le i \le n} |\delta x_i|} \right) \\
= \quad & \sup_{\|\delta\vec{x}\| \neq 0} \frac{|\nabla f(\vec{x})^T \cdot \delta\vec{x}|}{\|\delta\vec{x}\|_\infty} \overset{(**)}{=} \left\| \nabla f(\vec{x})^T \right\|_1 = \sum_{i=1}^{n} |f_{x_i}(\vec{x})|
\end{aligned}$$

□

■ **Example**

Let us now analyze the subtraction of two floating point numbers x_1, x_2

$$z := f(x_1,x_2) := x_1 - x_2\,, \quad f \in \mathcal{C}^\infty(\mathbb{R}^2,\mathbb{R})\,, \quad \vec{x} = (x_1,x_2)$$

The total derivative is

$$\mathrm{d}z = \frac{\partial f}{\partial x_1}(\vec{x})\,\mathrm{d}x_1 + \frac{\partial f}{\partial x_2}(\vec{x})\,\mathrm{d}x_2 \quad \longleftrightarrow \quad \delta z \doteq \frac{\partial f}{\partial x_1}(\vec{x})\,\delta x_1 + \frac{\partial f}{\partial x_2}(\vec{x})\,\delta x_2$$

Thus we obtain

$$\begin{aligned} \delta z &= \delta x_1 - \delta x_2 \\ \frac{\delta z}{z} &= \frac{\delta x_1}{x_1 - x_2} - \frac{\delta x_2}{x_1 - x_2} = \left(\frac{x_1}{x_1 - x_2}\right)\frac{\delta x_1}{x_1} - \left(\frac{x_2}{x_1 - x_2}\right)\frac{\delta x_2}{x_2} \end{aligned}$$

If the input data x_1, x_2 are correct up to an error $\delta x_1, \delta x_2$, then the absolute error δz of the result z is of the same order of magnitude as the absolute input errors.

For $x_1 \approx x_2 \wedge x_1 \cdot x_2 > 0$ the *relative* error $\delta z / z$ is much larger than the relative input error.

We now apply the results of the example before and obtain

$$\kappa_{abs}(f,\vec{x}) = 1\,, \quad \kappa_{rel}(f,\vec{x}) = \frac{|x_1| + |x_2|}{|x_1 - x_2|}$$

The problem is *(possibly) ill-posed*. □

In case of vectors and matrices it is often helpful to analyze the errors in single components. We have done that already in the example above.

Definition (error amplification factors)

Let be $\vec{f} \in \mathcal{C}^2(D,\mathbb{R}^m)$, $D \subseteq \mathbb{R}^n$ with $\vec{f}(\vec{x}) := (f_1(x_1,\ldots,x_n),\ldots,f_m(x_1,\ldots,x_n))$.

The *absolute error of a single component* is defined by

$$\delta f_i \doteq \sum_{j=1}^{n} \hat{\kappa}_{ij}(\vec{f},\vec{x}) \cdot \delta x_j\,, \quad \hat{\kappa}_{ij} := \frac{\partial f_i(\vec{x})}{\partial x_j}\,, \quad i = 1,\ldots,m$$

and $|\hat{\kappa}_{ij}|$ is called *absolute amplification factor*.

The *relative error of a single component* is

$$\varrho f_i := \frac{\delta f_i}{f_i} \doteq \sum_{j=1}^{n} \kappa_{ij} \cdot \frac{\delta x_j}{x_j} = \sum_{j=1}^{n} \kappa_{ij} \cdot \varrho x_j\,, \quad \kappa_{ij} := \frac{x_j}{f_i(\vec{x})} \cdot \frac{\partial f_i(\vec{x})}{\partial x_j}\,, \quad i = 1,\ldots,m$$

and $|\kappa_{ij}|$ is called *relative amplification factor*; moreover, $\varrho x_j := \dfrac{\delta x_j}{x_j}$. □

■ Example: summary of errors in basic operations

$$\begin{array}{rclrcl}
\delta(a\pm b) & = & \delta a \pm \delta b & \varrho(a\pm b) & = & \varrho a \dfrac{a}{a\pm b} \pm \varrho b \dfrac{b}{a\pm b} \\
\delta(a\cdot b) & = & b\cdot\delta a + a\cdot\delta b & \varrho(a\cdot b) & = & \varrho a + \varrho b \\
\delta(a/b) & = & \delta a/b - a\delta b/b^2 & \varrho(a/b) & = & \varrho a - \varrho b \\
\delta(\sqrt{a}) & = & \delta a/(2\sqrt{a}) & \varrho(\sqrt{a}) & = & \varrho(a)/2
\end{array}$$

The absolute errors in division and root finding and the relative errors in the subtraction of positive numbers may become large, the respective problems are (potentially) ill-posed. The *propagation* of the errors may became dangerous. □

■ Example: loss of significance

Let be $t = 7$, $B = 10$ and consider the two numbers

$$\begin{array}{rcl}
a & = & 1234567\cdot 10^2 \\
b & = & 1234569\cdot 10^2
\end{array}$$

Subtraction yields $c := a - b = 2000000\cdot 10^{-4}$. If in case of an input error, let $b \to \tilde{b} = 1234568\cdot 10^2$, then $\tilde{c} := a - \tilde{b} = 1000000\cdot 10^{-4}$. The relative change in b is 10^{-7}, this leads to a relative change in c of

$$\left|\frac{c-\tilde{c}}{c}\right| = 0.5 \quad \to \quad \text{error amplification: } 5\cdot 10^6$$

Conclusion: Do not subtract positive numbers of almost equal size! □

2.4 Application to Linear Systems

Definition (matrix norm)

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

$$\begin{array}{lcll}
\|A\| & > & 0 \quad \forall A \neq 0 \quad \wedge & (\|A\| = 0 \Leftrightarrow A = 0) \\
\|\alpha A\| & = & |\alpha|\|A\| & \text{(homogenity)} \\
\|A+B\| & \leq & \|A\| + \|B\| & \text{(triangular inequality)}
\end{array}$$

for all $A, B \in \mathbb{R}^{n\times m}, \alpha \in \mathbb{R}$.

A matrix norm is called *sub-multiplicative*, if

$$\|C\cdot D\| \leq \|C\|\cdot\|D\| \quad \forall C \in \mathbb{R}^{n\times m}, D \in \mathbb{R}^{m\times p}$$

A matrix norm is called *compatible or consistent* with the vector norm $\|\cdot\|$, if

$$\|Ax\| \leq \|A\|\|\vec{x}\| \quad \forall A \in \mathbb{R}^{n\times m}, \vec{x} \in \mathbb{R}^m$$

Definition

Let be $A \in \mathbb{R}^{n\times m}$, then the vector norm $\|\cdot\|_p$ can be used to define the following matrix norm

$$\|A\|_p := \sup_{\vec{x}\neq 0} \frac{\|A\vec{x}\|_p}{\|\vec{x}\|_p} = \sup_{\|\vec{x}\|_p=1} \|A\vec{x}\|_p$$

This matrix norm is called *induced matrix norm.* □

Induced matrix norms are compatible. Among all compatible matrix norms, the induced matrix norm is the smallest one

$$\|A\vec{x}\|_p \leq \|A\|\|\vec{x}\|_p \quad \forall \vec{x} \quad \Rightarrow \quad \|A\| \geq \|A\|_p$$

■ **Example** (induced matrix norms)

Suppose $A \in \mathbb{R}^{n\times m}$. Then we get the following induced matrix norms

$$\begin{array}{lcl}
\|\vec{x}\|_1 = |x_1|+\ldots+|x_n| & \Rightarrow & \|A\|_1 = \max\limits_{j=1\ldots m}\left(\sum\limits_{i=1}^{n}|a_{ij}|\right) \\
\|\vec{x}\|_2 = \sqrt{\sum\limits_{i=1}^{n}|x_i|^2} & \Rightarrow & \|A\|_2 = \sqrt{\lambda_{max}(A^H A)} \\
 & & = \max\limits_{\vec{x}\neq 0}\sqrt{\vec{x}^H A^H A\vec{x}/(\vec{x}^H\vec{x})} \\
\|\vec{x}\|_\infty = \max\{|x_1|,\ldots,|x_n|\} & \Rightarrow & \|A\|_\infty = \max\limits_{i=1\ldots n}\left(\sum\limits_{j=1}^{m}|a_{ij}|\right)
\end{array}$$

$\|A\|_\infty$ is the maximum absolute row sum of the matrix and is called *row sum norm*, $\|A\|_1$ is the maximum absolute column sum of the matrix and is called *column sum norm*, $\|A\|_2$ is the *spectral norm.* □

■ **Example**

All induced norms are sub-multiplicative, the matrix norm $\|A\| := \max_{i,j}|a_{ij}|$ is not. The *Frobenius-norm* is a sub-multiplicative norm compatible with – but not induced by – the vector norm $\|\cdot\|_2$

$$\|A\|_F := \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m}|a_{ij}|^2}$$

□

Perturbed linear system

Let us consider the perturbation

$$A\vec{x} = b \quad \rightarrow \quad (A+\delta A)(\vec{x}+\delta\vec{x}) = \vec{b}+\delta\vec{b}, \quad A \in \mathbb{R}^{n\times n}, \vec{x},\vec{b} \in \mathbb{R}^n$$

When does a unique solution of the perturbed system exist?

We assume that the matrix A is non-singular and that the perturbation δA small enough such that also $\det(A+\delta A) \neq 0$. Under which conditions is the latter assumption valid? For that we analyze the kernel of $A+\delta A$.

If $(A+\delta A)$ is singular, then there exists an $\vec{x} \neq 0$ (non-zero kernel) such that

$$\begin{aligned} (A+\delta A)\vec{x} = 0 \quad &\Rightarrow \quad \vec{x} = -A^{-1}\delta A\vec{x} \quad \Rightarrow \quad \|\vec{x}\| \leq \|A^{-1}\| \cdot \|\delta A\| \cdot \|\vec{x}\| \\ &\Rightarrow \quad (1 - \|A^{-1}\| \cdot \|\delta A\|) \cdot \|\vec{x}\| \leq 0 \quad \Rightarrow \quad \|\delta A\| \geq \frac{1}{\|A^{-1}\|} \end{aligned}$$

If

$$\|\delta A\| < \frac{1}{\|A^{-1}\|}$$

holds, the norm estimate in the first line can be valid only for $\vec{x} = \vec{0}$. Thus $\vec{x} = 0$ is the only solution of $(A+\delta A)\vec{x} = 0$ and therefore $A+\delta A$ is non-singular. □

Calculation of the condition numbers for norm errors

We do this in 2 steps. First for an ideal A and a perturbed $\vec{b}$ and second vice versa.

- First we consider the vector function $\vec{f}: \mathbb{R}^n \to \mathbb{R}^n$ with $\vec{f}(\vec{b}) := A^{-1}\vec{b}$; using the definition of the total derivative we get

$$\begin{aligned} \kappa_{abs}(\vec{f},\vec{b}) &= \|\vec{f}'(b)\| = \|A^{-1}\| \\ \kappa_{rel}(\vec{f},\vec{b}) &= \frac{\|A^{-1}\|}{\|A^{-1}\vec{b}\|/\|\vec{b}\|} = \|A^{-1}\|\frac{\|A\vec{x}\|}{\|\vec{x}\|} \leq \|A^{-1}\|\|A\| \end{aligned}$$

- Next we analogously investigate the case $\vec{g}: \{A \in \mathbb{R}^{n\times n} \,|\, \det A \neq 0\} \to \mathbb{R}^n$ with $\vec{g}(A) := A^{-1}\vec{b}$ in the second step: Let us consider the perturbed matrix $\tilde{A} = A + \delta A$ with $\|\delta A\| \leq \varepsilon \|A\|$ and the unperturbed right hand side $\vec{b}$.

  That leads to $(A+\delta A)(\vec{x}+\delta x) = (A\vec{x} + \delta A\vec{x} + A\delta\vec{x} + \delta A\delta\vec{x}) = \vec{b}$ or (neglecting terms of order higher than linear)

$$\delta\vec{x} \doteq -A^{-1} \cdot \delta A \cdot \vec{x}$$

  Choosing a vector norm together with an *induced matrix norm* we get

$$\|\delta\vec{x}\| \overset{\cdot}{\leq} \|A^{-1}\| \cdot \|\delta A\| \cdot \|\vec{x}\| \leq \varepsilon \|A^{-1}\| \cdot \|A\| \cdot \|\vec{x}\|$$

  or

$$\frac{\|\delta\vec{x}\|}{\|\vec{x}\|} \leq \kappa_{rel}(\vec{g},A) \cdot \varepsilon \quad \text{with} \quad \kappa_{rel}(\vec{g},A) = \|A^{-1}\|\|A\|$$

- Thus the same expression appears in both cases. It is therefore defined as the ”condition of the linear system” and denoted by $\kappa_{rel}(A) := \|A^{-1}\|\|A\|$.

For an induced matrix norm, we may further rewrite this

$$\begin{aligned}\|A\|_p\|A^{-1}\|_p &= \max_{\|\vec{x}\|_p\neq 0}\frac{\|A\vec{x}\|_p}{\|\vec{x}\|_p}\cdot\max_{\|\vec{y}\|_p\neq 0}\frac{\|A^{-1}\vec{y}\|_p}{\|\vec{y}\|_p} = \max_{\|\vec{x}\|_p\neq 0}\frac{\|A\vec{x}\|_p}{\|x\|_p}\cdot\max_{\|\vec{z}\|_p\neq 0}\frac{\|\vec{z}\|_p}{\|A\vec{z}\|_p}\\ &= \max_{\|\vec{x}\|_p\neq 0}\frac{\|A\vec{x}\|_p}{\|x\|_p}\cdot\frac{1}{\min_{\|\vec{z}\|_p=1}\|A\vec{z}\|_p} = \frac{\max_{\|\vec{x}\|_p=1}\|A\vec{x}\|_p}{\min_{\|\vec{z}\|_p=1}\|A\vec{z}\|_p}\end{aligned}$$

Properties of $\kappa_{rel}(A)$

- $\kappa_{rel}(A) \geq 1$.
- $\kappa_{rel}(A) = \kappa_{rel}(\alpha A) \quad \forall \alpha \in \mathbb{R},\, \alpha \neq 0$.
- $\kappa_{rel}(A) = \infty \quad \Leftrightarrow \quad \det(A) = 0$

Unlike the determinant, the condition of a matrix is invariant under scaling. □

■ Example

We calculate the condition numbers for a special linear system

$$A := \begin{pmatrix} 0 & 1 \\ -\varepsilon & 1 \end{pmatrix},\; \vec{b} := \begin{pmatrix} 1 \\ 1 \end{pmatrix} \;\Rightarrow\; A^{-1} = \begin{pmatrix} 1/\varepsilon & -1/\varepsilon \\ 1 & 0 \end{pmatrix}$$

$$\Rightarrow\; \|A^{-1}\|_\infty\|A\|_\infty = 2/\varepsilon\cdot(1+\varepsilon) \approx 2/\varepsilon \;\to\; \infty \quad \text{für} \quad \varepsilon \to 0$$

The exact solution of the linear system is

$$\vec{x} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \text{ for } \varepsilon \neq 0 \quad \text{and} \quad \vec{x} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \lambda\begin{pmatrix} 1 \\ 0 \end{pmatrix},\, \lambda \in \mathbb{R} \text{ for } \varepsilon = 0$$

□

■ Example (advantage of unitary transformations)

The LU factorization can be seen as the multiplication of the initial matrix $A \in \mathbb{R}^{n\times n}$ by a series of elementary transformation matrices $Q \in \mathbb{R}^{n\times n}$.

After each step $A \to QA$ the condition number of the new and transformed matrix QA may increase.

If we only use unitary transformation matrices Q, this effect cannot happen.

Let be $Q^H = Q^{-1}$ a unitary matrix used as an elementary transformation matrix, then we get with the $p = 2$-norm

$$\|Q\vec{y}\|_2^2 = (Q\vec{y})^H(Q\vec{y}) = \vec{y}^H Q^H Q\vec{y} = \|\vec{y}\|_2^2 \;\; \forall \vec{y} \in \mathbb{R}^n$$

and for the induced matrix norm

$$\begin{aligned}
\|A\|_2 &= \max_{\vec{x}\neq 0}\sqrt{\frac{\vec{x}^H A^H A\vec{x}}{\vec{x}^H\vec{x}}} \\
&= \max_{\vec{x}\neq 0}\sqrt{\frac{\vec{x}^H A^H Q^H Q A\vec{x}}{\vec{x}^H\vec{x}}} = \|QA\|_2 \\
&= \max_{\vec{y}=Q\vec{x}\neq 0}\sqrt{\frac{(\vec{x}^H Q^H)A^H A(Q\vec{x})}{(\vec{x}^H Q^H)(Q\vec{x})}} = \max_{\vec{x}\neq 0}\sqrt{\frac{\vec{x}^H Q^H A^H A Q\vec{x}}{\vec{x}^H\vec{x}}} = \|AQ\|_2 \\
\|A^{-1}Q^H\|_2 &= \max_{\vec{x}\neq 0}\sqrt{\frac{\vec{x}^H Q(A^{-1})^H(A^{-1})Q^H\vec{x}}{\vec{x}^H(QQ^H)\vec{x}}} = \max_{\vec{y}=Q^H\vec{x}\neq 0}\sqrt{\frac{\vec{y}^H(A^{-1})^H(A^{-1})\vec{y}}{\vec{y}^H\vec{y}}} \\
&= \|A^{-1}\|_2
\end{aligned}$$

By that we have proven

$$\kappa_{rel}(QA) = \|QA\|_2 \cdot \|(QA)^{-1}\|_2 = \|A\|_2 \cdot \|A^{-1}Q^H\|_2 = \kappa_{rel}(A)$$

□

2.5 Rounding Errors and Discretization Errors

Up to now we have treated ideal problems only and have analyzed, how the output error is amplified compared to the input error. That amplification cannot be avoided. In real problems additional errors are induced by the numerical solution.

One source are rounding errors in each step of the calculations. Another contribution to the total errors are discretization errors resulting from the fact that functions of continuous variables are represented in the computer by a finite number of evaluations, e.g. on a grid or by truncated iterations. Discretization errors can usually be reduced by using more finely spaced grids or more iterations, but only at the price of increased computational cost.

Unfortunately means to decrease the rounding errors often increase the discretization errors and vice versa. The main goal is to reduce the

total error = discretization error + rounding error

■ **Example** (minimizing the total error in differentiation)

Let be $f \in \mathcal{C}^3([a,b],\mathbb{R})$ and $a < x < x+h < b$.

Consider the following approximation formulae for the first derivative

$$\begin{aligned}
D^+ f(x,h) &:= \frac{f(x+h)-f(x)}{h} \\
D^\circ f(x,h) &:= \frac{f(x+h)-f(x-h)}{2h}
\end{aligned}$$

$D^+ f(x,h)$ is called *forward finite difference* and $D^\circ f(x,h)$ is called *centered finite difference*.

From the Taylor expansion we derive the following approximation formulae for the first derivatives

$$\begin{aligned} D^+ f(x,h) &:= \frac{f(x+h)-f(x)}{h} && \doteq f'(x) + \frac{1}{2} f''(\xi) h \\ D^\circ f(x,h) &:= \frac{f(x+h)-f(x-h)}{2h} && \overset{\cdot\cdot}{=} f'(x) + \frac{1}{6}\left(f^{(3)}(\xi_1) + f^{(3)}(\xi_2)\right) h^2 \end{aligned}$$

with $\xi, \xi_1 \in [x, x+h]$, $\xi_2 \in [x-h, x]$.
We define $M_1 := \max\limits_{|t| \le h} |f^{(2)}(x+t)|$ and $M_2 := \max\limits_{|t| \le h} |f^{(3)}(x+t)|$.

Instead of $f(x-h), f(x), f(x+h)$ the rounded (or measured) values F_1, F_2, F_3 are available with an error of ε (often $\varepsilon \gg \varepsilon_0$)

$$|F_1 - f(x-h)| \le \varepsilon, \;\; |F_2 - f(x)| \le \varepsilon, \;\; |F_3 - f(x+h)| \le \varepsilon\,.$$

Estimation of the total error yields using the triangle inequality several times

$$\begin{aligned} err_1 &:= \left|\frac{F_3 - F_2}{h} - f'(x)\right| \\ &= \left|\frac{F_3 - f(x+h) + f(x+h) - (F_2 - f(x)) - f(x)}{h} - f'(x)\right| \\ &\le \left|\frac{1}{h}[(F_3 - f(x+h)) - (F_2 - f(x))]\right| + \left|\frac{f(x+h)-f(x)}{h} - f'(x)\right| \\ &\le \frac{1}{h}(|F_3 - f(x+h)| + |F_2 - f(x)|) + \left|D^+(f(x,h) - f''(x)\right| \\ &\le \frac{2\varepsilon}{h} + \frac{h}{2} M_1 \end{aligned}$$

Minimum total error obtained for $err_1'(h) = -\dfrac{2\varepsilon}{h^2} + \dfrac{M_1}{2} \overset{!}{=} 0$. From that we get

$$h_{1,opt} = 2\sqrt{\frac{\varepsilon}{M_1}} \quad \Rightarrow \quad err_{1,opt} = 2\sqrt{\varepsilon M_1}$$

For the centered finite difference we obtain analogously

$$err_2 := \left|\frac{F_3 - F_1}{2h} - f'(x)\right| = \ldots \le \frac{\varepsilon}{h} + \frac{h^2}{3} M_2$$

Minimum total error obtained for $err_2'(h) = -\dfrac{\varepsilon}{h^2} + \dfrac{2hM_2}{3} \overset{!}{=} 0$. From that we get

$$h_{2,opt} = \sqrt[3]{\frac{3\varepsilon}{2M_2}} \quad \Rightarrow \quad err_{2,opt} = \sqrt[3]{\frac{2\varepsilon^2 M_2}{3}} + \sqrt[3]{\frac{2\varepsilon^2 M_2}{3}}$$

For decreasing h, the amplification of the input errors and the "rounding errors" increase (e.g. because of division by h and $f(x+h) \approx f(x)$ in $(F_3 - F_2)/h$), whereas the "discretization errors" $\frac{1}{2} f''(\xi) h$ and $\frac{1}{6}\left(f^{(3)}(\xi_1) + f^{(3)}(\xi_2)\right) h^2$ decrease. □