Numerical and Symbolic Algorithms Modeling (MODEL, MU4IN901)

Jérémy Berthomieu, Vincent Neiger and Mohab Safey El Din

Introduction

Unless every attending student speaks French, the course will be given in English.

This course is taught on the Friday from 8:30 to 10:30 in different rooms depending on the week. by:

- Jérémy Berthomieu (in charge), jeremy.berthomieu@lip6.fr;

The tutorial are supervised

- on the Monday from 8:30 to 10:30 in different rooms depending on the week and from 10:45 to 12:45 in different rooms depending on the week by
  - Dimitri Lesnoff, dimitri.lesnoff@lip6.fr.
- on the Wednesday from 13:45 to 15:45 in different rooms depending on the week and from 16:00 to 18:00 in different rooms depending on the week by
  - Jérémy Berthomieu, jeremy.berthomieu@lip6.fr;

The evaluation is done through:

- short multiple-choice questions (at least 4 during the semester), evaluating how you learnt the course (20% of the final grade);
- one implementation project (20% of the final grade);
- one mid-term exam evaluating your ability to manipulate the concepts taught during the first half of the semester (30% of the final grade);
- one final exam evaluating your ability to manipulate the concepts taught during the whole semester but with an emphasis on the second half (30% of the final grade).

Objectives of the course and outline.

The course provides an introduction to fundamental linear algebra techniques with *approximate* or *exact* computation. This finds application in many areas of computer science (cryptography, high-performance computing, big data, operational research, imagery, etc.).

1. Linear system solving, Gaussian elimination and PLUQ decomposition
2. Approximate linear system solving and QR decomposition

3. Matrix and vector compression, SVD and FFT algorithms
4. Complexity, non-naive polynomial multiplication algorithms
5. Non-naive matrix multiplication algorithms
6. Reduction of linear algebra operations to PLUQ decomposition
7. Solving sparse and structured linear systems

Contents

I. Exact linear system solving and Gaussian elimination

We first recall floating-point arithmetic. Then, the goal is to solve dense linear systems.

I.1. Computer arithmetics

A normalized floating-point number $x \in \mathbb{F}$ is a number

$$x = \pm x_0.\underbrace{x_1 \dots x_{p-1}}_{\text{mantissa}} \times b^e,\ 0 \le x_i \le b-1,\ x_0 \neq 0$$

where b is the base, p the precision and e the exponent with $e_{\min} \le e \le e_{\max}$.

The machine precision is $\varepsilon = b^{1-p}$. Any $x \in \mathbb{R}$ can be approximated by $\mathrm{fl}(x) = x(1+\delta)$, with $\delta \le \mathbf{u}$. The unit round-off $\mathbf{u}$ equals $\frac{\varepsilon}{2}$ when rounding to nearest.

Arithmetic operations $(+, -, \times, /, \sqrt{})$ are performed as if they were computed with infinite precision before rounding the result. That is $\mathrm{fl}(x \circ y) = (x \circ y)(1+\delta)$ with $\delta \le \mathbf{u}$ for $\circ \in \{+, -, \times, /\}$.

Type	Size	Mantissa	Exponent	Unit round-off	Interval
Simple	32 bits	23 + 1 bits	8 bits	$\mathbf{u} = 2^{-24} \approx 5.96 \times 10^{-8}$	$\approx 10^{\pm 38}$
Double	64 bits	52 + 1 bits	11 bits	$\mathbf{u} = 2^{-53} \approx 1.11 \times 10^{-16}$	$\approx 10^{\pm 308}$

The arithmetic is closed: every operation returns a result. NaN (Not a Number) is generated by computations such as $\frac{0}{0}$, $0 \times \infty$, $\frac{\infty}{\infty}$, $\infty - \infty$ and $\sqrt{-1}$.

Infinities and zeroes satisfies sign rules: $\infty + \infty = \infty$, $(-1) \times \infty = -\infty$, $\frac{1}{\infty} = +0$.

Each time we perform an operation, we lose precision: we say we have rounding errors. The two main sources of rounding errors are cancellation and absorption.

There are four types of rounding modes

- toward 0: truncation, it is similar to the common behavior of float-to-integer conversions, which convert -3.9 to -3 and 3.9 to 3;
- toward $+\infty$: rounding up;
- toward $-\infty$: rounding down;
- to nearest: with ties rounding to nearest even digits in the required position (default).

Problem I.1. *Give the floating-point numbers representing* $1/3$ *and* $-1/5$ *in simple precision for the* 4 *rounding modes.*

Solution.

- In base 2, $1/3 = 1.\underbrace{01010101010101010101}_{23}\ldots \times 2^{-2}$. Towards 0 and $-\infty$, we truncate with 23 digits after the decimal dot so it ends with 010. Towards $+\infty$, we do the same and increase the last digit, hence it ends with 011. Finally, towards the nearest, we need to see what is behind the last 0. We have $101\cdots > 100\ldots$, hence we round towards $+\infty$.
- In base 2, $-1/5 = -1.\underbrace{10011001100110011001100}_{23}\cdots \times 2^{-3}$. Towards 0 and $+\infty$, we truncate with 23 digits after the decimal dot so it ends with 1100. Towards $-\infty$, we do the same and increase the last digit, hence it ends with 1101. Finally, towards the nearest, we need to see what is behind the last 0. We have $1100\cdots > 1000\ldots$, hence we round towards $-\infty$.

Problem I.2. *Let us recall that for x small enough, $\sqrt{1+x} \approx 1 + \frac{x}{2}$, or even $\sqrt{1+x} \approx 1 + \frac{x}{2} - \frac{1}{8}x^2$.*

1. *Let us assume that we have a C function* `double S(double x)` *computing $\sqrt{1+x}$ for a small x using these approximations. We implement the function $f(x) = \sqrt{1+x} - 1$ using* S. *What does the call $f(2^{-53})$ return with rounding towards to nearest?*
2. *Write $f(x)$ as a quotient without any subtraction.*
3. *Compute $f(2^{-53})$ using this new form and the corresponding implementation of the function* S. *What can be noticed?*

Solution.

1. Since $1 + 2^{-53}$ is rounded to 1, S will return 0.
2. $\sqrt{1+x} - 1 = \frac{(\sqrt{1+x}-1)(\sqrt{1+x}+1)}{\sqrt{1+x}+1} = \frac{1}{\sqrt{1+x}+1}$.
3. The denominator is rounded to 2 and the function returns 2^{-54}, which is the floating-point number the closest to the real value.

I.2. Linear system solving

We let $\mathbb{K}$ denote $\mathbb{Q}$, $\mathbb{R}$ or $\mathbb{C}$.

Assume that the linear system $Ax = b$, with $A \in \mathbb{K}^{n\times n}$ and $b \in \mathbb{K}^n$ we want to solve has a upper triangular shape

$$\begin{pmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,n-1} \\ & a_{1,1} & \cdots & a_{1,n-1} \\ & & \ddots & \vdots \\ & & & a_{n-1,n-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{pmatrix}.$$

Then, if $a_{n-1,n-1} \neq 0$, $x_{n-1} = \frac{b_{n-1}}{a_{n-1,n-1}}$.

Theorem I.1. *An upper triangular system $Ax = b$ with $A \in \mathbb{K}^{n\times n}$ and $b \in \mathbb{K}^n$ has a unique solution if, and only if, $a_{0,0} \cdots a_{n-1,n-1} \neq 0$.*

Problem I.3. *Prove this theorem using the reasoning above.*

Solution. The ith equation has a unique solution if, and only if, $a_{i,i} \neq 0$. Hence, we need all of them to be nonzero.

Problem I.4. *Practice solving triangular systems of size 4 or 5 over $\mathbb{Q}$.*

Solution. To be done by the students.

From this reasoning, we get the following iterative algorithm for solving an upper triangular system.

Algorithm 1: UpperTriangularSystemAlgorithm
Input: An upper triangular matrix $A = (a_{i,j})_{0\leq i,j\leq n-1}$ with coefficients in $\mathbb{K}$, with nonzero diagonal coefficients, and a vector $b = (b_i)_{0\leq i\leq n-1}$, with coefficients in $\mathbb{K}$.
Output: The vector $x = (x_i)_{0\leq i\leq n-1}$, with coefficients in $\mathbb{K}$, such that $Ax = b$.
For i **from** $n-1$ **to** 0 **do** $x_i \coloneqq b_i$
For i **from** $n-1$ **to** 0 **do**
  $x_i \coloneqq \frac{x_i}{a_{i,i}}$
  For j **from** $i-1$ **to** 0 **do** $x_j \coloneqq x_j - x_i a_{j,i}$
Return x

Theorem I.2. *Algorithm 1,* UpperTriangularSystemAlgorithm*, is correct and requires at most $\frac{n^2+n}{2}$ multiplications in $\mathbb{K}$.*

The main advantage of upper triangular systems is that solving them comes down to solving, one by one, linear equations in one variable. Indeed, in the last equations $x_0, \ldots, x_{n-2}$ do not appear, allowing us to solve for x_{n-1}. Then, plugging this value in the equation above and using the fact that $x_0, \ldots, x_{n-3}$ do not appear, we only need to solve a linear equation in x_{n-2}. And so on, and so forth.

The idea of *Gaussian elimination* is, at its name suggests, to eliminate variables from some equations to recover such a beneficial situation. That is, we want to reduce the linear system to an upper triangular one. This is done through a type of *elementary operation*: adding to a line the product of another line above with an element of $\mathbb{K}$. This leads to the following algorithm

Theorem I.3. *Algorithm 2,* GaussianEliminationAlgorithm*, is correct and solves the input linear system of size n using approximatively $\frac{n(n-1)(2n-1)}{6} + \frac{n(n-1)}{2}$ operations in $\mathbb{K}$.*

Problem I.5.

Algorithm 2: GaussianEliminationAlgorithm

Input: A matrix $A = (a_{i,j})_{0 \le i,j \le n-1}$, with coefficients in $\mathbb{K}$, and a vector $b = (b_i)_{0 \le i \le n-1}$, with coefficients in $\mathbb{K}$.

Output: An upper triangular matrix and a vector yielding an equivalent system as the input or "No unique solution".

For i **from** 0 **to** $n-1$ **do**
  $j \coloneqq i$
  While $j < n \wedge a_{j,i} = 0$ **do** $j \coloneqq j+1$
  If $j = n$ **then** **Return** "No unique solution".
  $t_{i..n-1} \coloneqq a_{i,i..n-1}$, $u \coloneqq b_i$
  $a_{i,i..n-1} \coloneqq a_{j,i..n-1}$, $b_i \coloneqq b_j$
  $a_{j,i..n-1} \coloneqq t_{i..n-1}$, $b_j \coloneqq u$
  For j **from** $i+1$ **to** $n-1$ **do**
    $b_j \coloneqq b_j - a_{j,i} b_i / a_{i,i}$
    $a_{j,i..n-1} \coloneqq a_{j,i..n-1} - a_{j,i} a_{i,i..n-1} / a_{i,i}$
Return A, b

1. *Check that, for the following input matrix and vector,*

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix}, \quad \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix},$$

   GaussianEliminationAlgorithm *returns*

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 0 & \frac{1}{10} & \frac{2}{5} & \frac{1}{10} \\ 0 & 0 & 2 & 3 \\ 0 & 0 & 0 & \frac{1}{2} \end{pmatrix}, \quad \begin{pmatrix} 32 \\ \frac{3}{5} \\ 5 \\ \frac{1}{2} \end{pmatrix}.$$

2. *Finish solving the system.*

Solution. The intermediate matrices and vectors are

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 0 & \frac{1}{10} & \frac{2}{5} & \frac{1}{10} \\ 0 & \frac{2}{5} & \frac{18}{5} & \frac{17}{5} \\ 0 & 1/10 & 17/5 & 51/10 \end{pmatrix}, \quad \begin{pmatrix} 32 \\ \frac{3}{5} \\ \frac{37}{5} \\ \frac{43}{5} \end{pmatrix},$$

and

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 0 & \frac{1}{10} & \frac{2}{5} & \frac{1}{10} \\ 0 & 0 & 2 & 3 \\ 0 & 0 & 3 & 5 \end{pmatrix}, \quad \begin{pmatrix} 32 \\ \frac{3}{5} \\ 5 \\ 8 \end{pmatrix}.$$

We also have

$$L = \begin{pmatrix} 1 & & & \\ 7/10 & 1 & & \\ 4/5 & 4 & 1 & \\ 7/10 & 1 & 3/2 & 1 \end{pmatrix}.$$

Problem I.6. *Practice solving linear systems.*

Solution. To be done by the students.

I.3. Determinant

Definition I.4. *Let* $A = (a_{i,j})_{0 \le i,j \le n-1} \in \mathbb{K}^{n \times n}$. *The* determinant *of* A *is*

$$\det A = \sum_{\sigma \in \mathfrak{S}_n} (-1)^{\varepsilon(\sigma)} a_{0,\sigma(0)} \cdots a_{n-1,\sigma(n-1)},$$

where $\mathfrak{S}_n$ *is the set of bijection from* $\{0, \dots, n-1\}$ *to itself and for* $\sigma \in \mathfrak{S}_n$,

$$\varepsilon(\sigma) = \#\left\{(i,j) | 0 \le i < j \le n-1,\ \sigma(i) > \sigma(j)\right\}.$$

Problem I.7.

1. *Expand this sum for matrices of sizes* 1, 2, 3 *and* 4.
2. *What about matrices of size* 5*?*
3. *How many terms does this sum have for matrices of size* n*?*

Solution.

1. **Size** 1. $a_{0,0}$.

   Size 2. $a_{0,0}a_{1,1} - a_{0,1}a_{1,0}$.

   Size 3. $a_{0,0}a_{1,1}a_{2,2} - a_{0,1}a_{1,0}a_{2,2} - a_{0,2}a_{1,1}a_{2,0} - a_{0,0}a_{1,2}a_{2,1} + a_{0,1}a_{1,2}a_{2,0} + a_{0,2}a_{1,0}a_{2,1}$.

   Size 4. Too long…
2. Even longer!
3. It has $n!$ terms!!

Proposition I.5. *Let* A *and* B *two matrices in* $\mathbb{K}^{n \times n}$,

1. *then* $\det(AB) = \det A \det B$*;*
2. *let* $A_{i,j}$ *be the submatrix obtained by removing from* A *its* i*th row and* j*th column, then* Laplace's expansion *is*
$$\forall 0 \leq i \leq n-1,\ \det A = \sum_{0 \leq j \leq n-1} (-1)^{i+j} a_j \det A_{i,j};$$
3. *then* $\det A = 0$ *if, and only if,* A *is a singular matrix, that is its rows (resp. columns) are not linearly independent;*
4. *let* A^{T} *be the transposed matrix of* A*, then* $\det A^{\mathrm{T}} = \det A$*.*

Problem I.8. *Prove these properties.*

Problem I.9. *How many recursive calls does Laplace's method perform?*

Theorem I.6. *Let* $T = (t_{i,j})_{0 \leq i,j \leq n-1}$ *be a triangular matrix. Then,*
$$\det T = \prod_{i=0}^{n-1} t_{i,i}.$$

Problem I.10. *Prove Theorem I.6.*

Solution. Expanding the formula of Definition I.4, we see that the only nonzero term is the one given by $\sigma = \mathrm{Id}$, that is the product of the diagonal elements.

Determinants can be useful to solve linear system. Let $b, x \in \mathbb{K}^{n \times 1}$.

Theorem I.7. *The linear system* $Ax = b$ *has a unique solution* x *if, and only if,* $\det A \neq 0$*. In that case,* Cramer's formula *ensures that*
$$x_i = \frac{\det B_i}{\det A}, \quad \text{where } B_i = \begin{pmatrix} a_{0,0} & \cdots & a_{0,i-1} & b_0 & a_{0,i+1} & \cdots & a_{0,n-1} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{n-1,0} & \cdots & a_{n-1,i-1} & b_{n-1} & a_{n-1,i+1} & \cdots & a_{n-1,n-1} \end{pmatrix}.$$

Definition I.8. *Let* $A \in \mathbb{C}^{n \times n}$*. We say that* $\lambda \in \mathbb{C}$ *is an* eigenvalue *of* A *if there exists a nonzero vector* $v \in \mathbb{C}^n$ *such that* $Av = \lambda v$*. In this case,* v *is an* eigenvector associated to λ*. The vector space containing all such* v *is the* eigenspace associated to λ*.*

Problem I.11. *Prove that eigenvalues are exactly the roots of the polynomial in* ℓ*,* $\det(\ell \,\mathrm{Id} - A)$*, which is called the* characteristic polynomial of A*.*

Solution. By definition, λ is an eigenvalue if, and only if, the kernel of $\lambda \,\mathrm{Id} - A$ is not $\{0\}$, i.e. the determinant of the matrix $\lambda \,\mathrm{Id} - A$ is 0. It suffices then to treat λ as an indeterminate.

Problem I.12.

1. *Give a matrix of size* 2 *over* $\mathbb{Q}$ *with* 2 *distinct rational eigenvalues.*

2. *Give a matrix of size 2 over $\mathbb{Q}$ with only one eigenvalue. What are the possible dimensions for the eigenspace associated to this eigenvalue?*

3. *Give a matrix of size 2 over $\mathbb{Q}$ with no rational eigenvalue.*

Solution.

1. $\begin{pmatrix} 1 & \\ & 2 \end{pmatrix}$.

2. $\begin{pmatrix} 1 & \\ & 1 \end{pmatrix}$ with an eigenspace of dimension 2 or $\begin{pmatrix} 1 & 1 \\ & 1 \end{pmatrix}$ with an eigenspace of dimension 1.

3. $\begin{pmatrix} 0 & 1 \\ \pm 2 & 0 \end{pmatrix}$ whose characteristic polynomial is $\lambda^2 \mp 2$ and whose roots are not rational.

Eigenvalues play a crucial role in many areas of computer science: data-mining (PageRank), decision theory, imagery, scientific computing, etc. But first, we need good algorithms for solving linear systems (with polynomial bit complexity) and this leads to good algorithm for computing determinants.

I.4. Gaussian elimination and LU factorization

When performing Gaussian elimination, the goal is to compute linear combinations of the matrix rows to make appear some zeroes and obtain an upper triangular matrix.

These operations can be summed up as a *factorization*, the so-called LU factorization, of the matrix into two: one lower triangular matrix with 1's on the diagonal and one upper triangular matrix.

Example I.9.

$$\begin{pmatrix} 4 & 4 & 8 & 1 \\ 2 & 8 & 7 & 1 \\ 1 & 3 & 6 & 1 \\ -4 & 6 & 5 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{4} & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 & 4 & 8 & 1 \\ 0 & 6 & 3 & \frac{1}{2} \\ 0 & 2 & 4 & \frac{3}{4} \\ 0 & 10 & 13 & 2 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{4} & \frac{1}{3} & 1 & 0 \\ -1 & \frac{5}{3} & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 & 4 & 8 & 1 \\ 0 & 6 & 3 & \frac{1}{2} \\ 0 & 0 & 3 & \frac{7}{12} \\ 0 & 0 & 8 & \frac{7}{6} \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ \frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{4} & \frac{1}{3} & 1 & 0 \\ -1 & \frac{5}{3} & \frac{8}{3} & 1 \end{pmatrix} \begin{pmatrix} 4 & 4 & 8 & 1 \\ 0 & 6 & 3 & \frac{1}{2} \\ 0 & 0 & 3 & \frac{7}{12} \\ 0 & 0 & 0 & -\frac{7}{18} \end{pmatrix}.$$

The LU factorization is not always possible: whenever the pivot is 0, the LU factorization does not exist. In Gaussian elimination, the problem is circumvented by permuting rows. For matrix factorization, this is done by factoring by a permutation matrix yielding a PLU factorization.

Definition I.10. *A permutation matrix is a matrix whose entries are all* 0 *or* 1*. For each row and each column, only one coefficient is nonzero. A permutation matrix P satisfies $P^{-1} = P^{\mathrm{T}}$.*

Gaussian elimination and LU factorization do not behave well with floating-point arithmetic: absorption or cancellation can arise easily. Likewise, a very small pivot will ill behave.

Example I.11.

$$\begin{aligned}
\begin{pmatrix} 4 & 4 & 8 \\ 2 & 2 & 7 \\ 1 & 3 & 6 \end{pmatrix} &= \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{4} & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 & 4 & 8 \\ 0 & 0 & 3 \\ 0 & 2 & 4 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{4} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 4 & 4 & 8 \\ 0 & 0 & 3 \\ 0 & 2 & 4 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 0 & 1 \\ \frac{1}{4} & 1 & 0 \end{pmatrix} \begin{pmatrix} 4 & 4 & 8 \\ 0 & 2 & 4 \\ 0 & 0 & 3 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 & 4 & 8 \\ 0 & 2 & 4 \\ 0 & 0 & 3 \end{pmatrix}.
\end{aligned}$$

What happens for $A = \begin{pmatrix} 2^{-60} & 1 \\ 1 & 1 \end{pmatrix}$*?*

The LU factorization of A is

$$A = \begin{pmatrix} 1 & 0 \\ 2^{60} & 1 \end{pmatrix} \begin{pmatrix} 2^{-60} & 1 \\ 0 & 1 - 2^{60} \end{pmatrix}.$$

In double precision, $1 - 2^{60}$ is represented by -2^{60}, so that the LU factorization is stored as

$$\begin{pmatrix} 1 & 0 \\ 2^{60} & 1 \end{pmatrix} \begin{pmatrix} 2^{-60} & 1 \\ 0 & -2^{60} \end{pmatrix}.$$

Yet, expanding this decomposition using double-precision arithmetic, we obtain

$$\begin{pmatrix} 2^{-60} & 1 \\ 1 & 0 \end{pmatrix} \neq A!$$

If we swap the rows, then we have the following PLU decomposition,

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 2^{-60} & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 2^{-60} & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Problem I.13. *Expand the last decomposition of A using double-precision arithmetic. What can be noticed?*

Solution. We find A again.

Problem I.14. *Solve the following system over $\mathbb{Q}$ using a $A = PLU$ decomposition,*

$$\begin{pmatrix} 2 & 3 & 1 & 5 \\ 6 & 9 & 5 & 19 \\ 2 & 19 & 10 & 23 \\ 8 & 44 & 20 & 76 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 22 \\ 76 \\ 99 \\ 256 \end{pmatrix}.$$

Solution. The PLU decomposition of A is obtained as

$$\begin{aligned} A &= \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix} \begin{pmatrix} 1 & & & \\ 3 & 1 & & \\ 1 & 0 & 1 & \\ 4 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 3 & 1 & 5 \\ 0 & 0 & 2 & 4 \\ 0 & 16 & 9 & 18 \\ 0 & 32 & 16 & 56 \end{pmatrix} \\ &= \begin{pmatrix} 1 & & & \\ & 0 & 1 & \\ & 1 & 0 & \\ & & & 1 \end{pmatrix} \begin{pmatrix} 1 & & & \\ 1 & 1 & & \\ 3 & 0 & 1 & \\ 4 & 2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 3 & 1 & 5 \\ 0 & 16 & 9 & 18 \\ 0 & 0 & 2 & 4 \\ 0 & 0 & -2 & 20 \end{pmatrix} \\ &= \begin{pmatrix} 1 & & & \\ & 0 & 1 & \\ & 1 & 0 & \\ & & & 1 \end{pmatrix} \begin{pmatrix} 1 & & & \\ 1 & 1 & & \\ 3 & 0 & 1 & \\ 4 & 2 & -1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 3 & 1 & 5 \\ 0 & 16 & 9 & 18 \\ 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 24 \end{pmatrix}. \end{aligned}$$

We then find $Ly = P^{\mathrm{T}}b$ with $y = \begin{pmatrix} 22 \\ 77 \\ 10 \\ 24 \end{pmatrix}$ and $Ux = y$ with $x = \begin{pmatrix} 4 \\ 2 \\ 3 \\ 1 \end{pmatrix}$.

I.4.1. Swapping rows and columns

To avoid this kind of problem, it is necessary to swap rows to take the greatest number in the column (below the current row) as the pivot. In fact, it is even better to take the greatest number in the whole submatrix as the pivot. This makes us swapping rows and columns, yielding a LU factorization with permutations matrices on the left and on the right. This is the PLUQ factorization.

The method is straight-forward, at step i (with i from 0 to $n-1$):

- Pick the greatest number in absolute value in the submatrix made from rows i to $n-1$ and column from i to $n-1$ for the pivot.
- Swap the rows and columns, so that the pivot is in position (i, i).
- Update L and U.

Example I.12.

$$A = \begin{pmatrix} 1 & 2^{20} & 2^{40} \\ 2 & 2^{40} & 2^{108} \\ 2^{30} & 2^{54} & 2^{10} \end{pmatrix}$$

$$
\begin{aligned}
&= \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2^{20} & 2^{40} \\ 2 & 2^{40} & 2^{108} \\ 2^{30} & 2^{54} & 2^{10} \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}^{\mathrm{T}} \\
&= \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2^{108} & 2^{40} & 2 \\ 2^{40} & 2^{20} & 1 \\ 2^{10} & 2^{54} & 2^{30} \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \\
&= \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2^{-68} & 1 & 0 \\ 2^{-98} & 0 & 1 \end{pmatrix} \begin{pmatrix} 2^{108} & 2^{40} & 2 \\ 0 & 2^{20} - 2^{-28} & 1 \\ 0 & 2^{54} & 2^{30} \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \\
&= \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2^{-68} & 1 & 0 \\ 2^{-98} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}^{\mathrm{T}} \\
&\quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 2^{108} & 2^{40} & 2 \\ 0 & 2^{20} - 2^{-28} & 1 \\ 0 & 2^{54} & 2^{30} \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \\
&= \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2^{-98} & 1 & 0 \\ 2^{-68} & 0 & 1 \end{pmatrix} \begin{pmatrix} 2^{108} & 2^{40} & 2 \\ 0 & 2^{54} & 2^{30} \\ 0 & 2^{20} - 2^{-28} & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \\
&= \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2^{-98} & 1 & 0 \\ 2^{-68} & 2^{-34} - 2^{-82} & 1 \end{pmatrix} \begin{pmatrix} 2^{108} & 2^{40} & 2 \\ 0 & 2^{54} & 2^{30} \\ 0 & 0 & 1 - 2^{-4} \end{pmatrix} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}
\end{aligned}
$$

.

Problem I.15.

1. *Check all the computations and explain the absorption steps of Example I.12.*
2. *What would be the LU decomposition of the same matrix without any pivoting?*
3. *Do the two decompositions yield the same determinant?*

Solution.

1. The floats cannot be stored in double-precision floating-point numbers and must be rounded.
2. We would have

$$
A = \begin{pmatrix} 1 & & \\ 2 & 1 & \\ 2^{30} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2^{20} & 2^{40} \\ 0 & 2^{40} - 2^{21} & 2^{108} - 2^{41} \\ 0 & 2^{54} - 2^{50} & -2^{70} + 2^{10} \end{pmatrix} = \begin{pmatrix} 1 & & \\ 2 & 1 & \\ 2^{30} & \frac{1-2^{-19}}{1-2^{-4}} \cdot 2^{14} & 1 \end{pmatrix} \begin{pmatrix} 1 & 2^{20} & 2^{40} \\ 0 & 2^{40} - 2^{21} & 2^{108} - 2^{41} \\ 0 & 0 & X \end{pmatrix}.
$$

Note that, as a float, $\frac{1-2^{-19}}{1-2^{-4}}$ is obtained as the rounding of $(1-2^{-19})(1+2^{-4}+2^{-8}+2^{-12}+2^{-16}+\cdots)$.

3. The PLUQ decomposition yields $\det A = 1 \times 1 \times 2^{108} \times 2^{54}(1 - 2^{-4} \times (-1) = -(1 - 2^{-4})2^{162}$.

   The LU factorization yields…

Problem I.16. *Prove that if P and Q are the permutation matrices obtained in the PLUQ decomposition of A, then $\det P = (-1)^p$ and $\det Q = (-1)^q$, where p (resp. q) is the number of swaps for the rows (resp. columns) that have been performed.*

Solution. By definition of a permutation matrix, only 1 term in the determinant appears with a sign, hence it is 1 or -1. Each swap will change the determinant by -1. Hence the formula.

I.4.2. Solving linear systems

Solving a linear system $Ax = b$ comes down to solving $PLUQx = b$. This is equivalent to solving $LUQx = P^{\mathrm{T}}b$ or $LUx' = b'$ with $x' = Qx$ and $b' = P^{\mathrm{T}}b$.

Then, we first solve $Ly = b'$, which is a lower triangular system and then $Ux' = y$, which is an upper one. Finally, since $x' = Qx$, we have $x = Q^{\mathrm{T}}x'$.

Example I.13. *Solving $Ax = \begin{pmatrix} 0 \\ 2^{34} \\ 1 \end{pmatrix}$ in floating-point arithmetic with double precision, gives*

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 2^{-98} & 1 & 0 \\ 2^{-68} & 2^{-34} - 2^{-82} & 1 \end{pmatrix}\begin{pmatrix} 2^{108} & 2^{40} & 2 \\ 0 & 2^{54} & 2^{30} \\ 0 & 0 & 1 - 2^{-4} \end{pmatrix}\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 2^{34} \\ 1 \end{pmatrix}$$

which is equivalent to

$$\begin{pmatrix} 1 & 0 & 0 \\ 2^{-98} & 1 & 0 \\ 2^{-68} & 2^{-34} - 2^{-82} & 1 \end{pmatrix}\begin{pmatrix} 2^{108} & 2^{40} & 2 \\ 0 & 2^{54} & 2^{30} \\ 0 & 0 & 1 - 2^{-4} \end{pmatrix}\begin{pmatrix} x_3 \\ x_2 \\ x_1 \end{pmatrix} = \begin{pmatrix} 2^{34} \\ 1 \\ 0 \end{pmatrix}.$$

Solving $Ly = \begin{pmatrix} 2^{34} \\ 1 \\ 0 \end{pmatrix}$ yields $y = \begin{pmatrix} 2^{34} \\ 1 \\ -2^{-33}+2^{-82} \end{pmatrix}$.

Then, solving $Ux' = \begin{pmatrix} 2^{34} \\ 1 \\ -2^{-33}+2^{-82} \end{pmatrix}$ yields $x' = \begin{pmatrix} 2^{-74}-2^{-122} \\ 2^{-54} \\ 0 \end{pmatrix} = Qx$, hence $x = \begin{pmatrix} 0 \\ 2^{-54} \\ 2^{-74}-2^{-122} \end{pmatrix}$.

I.4.3. Computing determinants

Using Proposition I.5, we can deduce that if $A = PLUQ$, then $\det A = \det P \det L \det U \det Q$.

Problem I.17. *Give an algorithm to compute the determinant of a matrix $A \in \mathbb{C}^m$ using its PLUQ decomposition or during the computation of its PLUQ decomposition.*

Solution.

- We set `det` to 1.
- Each time we perform a swap of rows xor colum, we multiply the `det` by -1 (if we perform both, we multiply twice by -1 so nothing to be done).
- At the end of the PLUQ decomposition, we multiply `det` by diagonal elements of U.

I.5. Cholesky Method

A matrix $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite if $A^{\mathrm{T}} = A$ and for all $x \neq 0$, $x^{\mathrm{T}}Ax > 0$.

The LU factorization of a symmetric positive definite matrix is always possible. Furthermore, since $A = \begin{pmatrix} \alpha^2 & \omega^{\mathrm{T}} \\ \omega & K \end{pmatrix}$, we have

$$A = \begin{pmatrix} 1 & 0 \\ \frac{\omega}{\alpha^2} & \mathrm{Id} \end{pmatrix} \begin{pmatrix} \alpha^2 & \omega \\ 0 & K - \frac{\omega\omega^{\mathrm{T}}}{\alpha^2} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \frac{\omega}{\alpha^2} & \mathrm{Id} \end{pmatrix} \begin{pmatrix} \alpha^2 & 0 \\ 0 & K - \frac{\omega\omega^{\mathrm{T}}}{\alpha^2} \end{pmatrix} \begin{pmatrix} 1 & \frac{\omega}{\alpha^2} \\ 0 & \mathrm{Id} \end{pmatrix} = LDL^{\mathrm{T}}$$
$$= \begin{pmatrix} \alpha & 0 \\ \frac{\omega}{\alpha} & \mathrm{Id} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & K - \frac{\omega\omega^{\mathrm{T}}}{\alpha^2} \end{pmatrix} \begin{pmatrix} \alpha & \frac{\omega}{\alpha} \\ 0 & \mathrm{Id} \end{pmatrix}$$

For symmetric positive definite matrix, Cholesky decomposition gives $A = LL^{\mathrm{T}}$, with L lower triangular or $A = LDL^{\mathrm{T}}$, with L lower triangular with 1's on the diagonal and D diagonal.

Example I.14. *The Cholesky decomposition of the following matrix is*

$$A = \begin{pmatrix} 9 & 3 & 12 \\ 3 & 5 & -6 \\ 12 & -6 & 105 \end{pmatrix}$$
$$= \begin{pmatrix} 3 & 0 & 0 \\ 1 & 1 & 0 \\ 4 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & -10 \\ 0 & -10 & 89 \end{pmatrix} \begin{pmatrix} 3 & 1 & 4 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 3 & 0 & 0 \\ 1 & 1 & 0 \\ 4 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & -5 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 64 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & -5 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 & 1 & 4 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 3 & 0 & 0 \\ 1 & 2 & 0 \\ 4 & -5 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 64 \end{pmatrix} \begin{pmatrix} 3 & 1 & 4 \\ 0 & 2 & -5 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 0 & 0 \\ 1 & 2 & 0 \\ 4 & -5 & 8 \end{pmatrix} \begin{pmatrix} 3 & 1 & 4 \\ 0 & 2 & -5 \\ 0 & 0 & 8 \end{pmatrix}.$$

Problem I.18. *Compute the Cholesky decomposition of*

$$A = \begin{pmatrix} 1 & -1 & 2 \\ -1 & -3 & 2 \\ 2 & 2 & 7 \end{pmatrix}.$$

Solution. We have

$$L = \begin{pmatrix} 1 & & \\ -1 & 1 & \\ 2 & -1 & 1 \end{pmatrix}, \; D = \begin{pmatrix} 1 & & \\ & -4 & \\ & & 7 \end{pmatrix}, \; A = LDL^{\mathrm{T}}.$$

Implementation I.1. *Implement the LU and the PLUQ factorization and linear system solving over double-precision floating-point numbers.*

Implementation I.2. *Install the MPFR library*[1] *and implement the LU and the PLUQ factorization over multi-precision floating-point numbers (*`mpfr_t`*).*

Compare the results with the previous implementation.

Implementation I.3. *Implement the Cholesky decomposition for symmetric matrix. Compare its efficiency with the LU decomposition.*

[1] https://www.mpfr.org/

II. Approximate solving of over-determined linear systems

II.1. QR Decomposition

Whenever a matrix A has size $m \times n$ with $m \geq n$, even if it has full rank, a solution of the linear system $Ax = b$ may not exist. The least-square method, then, tries to minimize the error for the Euclidean norm. That is find x' such that $\|Ax' - b\|_2$ is minimal.

The QR decomposition factors A into QR with Q unitary and R upper triangular.

II.1.1. Euclidean norm, orthogonal and unitary matrices

We start by recalling the definition of the *Euclidean norm* and of the classical *scalar product* in dimension m. These extends the known definitions in dimensions 1, 2 or 3.

Definition II.1 (Scalar product)**.** *Let x and y be two vectors in $\mathbb{R}^m$. Then, their* scalar product *is the scalar $\langle x \mid y \rangle$ defined as*

$$\langle x \mid y \rangle = x^{\mathrm{T}} \cdot y = \sum_{i=1}^{m} x_i y_i.$$

This definition can be extended to complex vectors $x, y \in \mathbb{C}^m$ as follows using the notation $x^\star = \bar{x}^{\mathrm{T}}$:

$$\langle x \mid y \rangle = x^\star \cdot y = \sum_{i=1}^{m} \bar{x}_i y_i.$$

Definition II.2 (Euclidean norm)**.** *Let x be a vector in $\mathbb{R}^m$. Then, its* Euclidean norm*, or* norm *if there is no ambiguity, is the scalar $\|x\|_2$, or $\|x\|$ again if there is no ambiguity, defined as*

$$\|x\|_2 = \sqrt{\langle x \mid x \rangle} = \sqrt{\sum_{i=1}^{m} x_i^2}.$$

This can be extended to a complex vector $x \in \mathbb{C}^m$ as follows

$$\|x\|_2 = \sqrt{\langle x \mid x \rangle} = \sqrt{\sum_{i=1}^{m} |x_i|^2}.$$

Problem II.1. *Let $m, n \in \mathbb{N}$ with $m > n > 0$. Let $x \in \mathbb{C}^m$ and let us denote $y = x_{1,\ldots,n} \in \mathbb{C}^n$ the vector formed by the first n rows of x. Likewise, let us denote $z = x_{n+1,\ldots,m} \in \mathbb{C}^{m-n}$ the vector formed by the last $m - n$ rows of x.*

Show that $\|x\|_2^2 = \|y\|_2^2 + \|z\|_2^2$.

Solution. We have $\|x\|_2^2 = \sum_{i=1}^m |x_i|^2 = \sum_{i=1}^n |x_i|^2 + \sum_{i=n+1}^m |x_i|^2 = \|y\|_2^2 + \|z\|_2^2$.

Orthogonal and unitary matrices are the one that preserve this scalar product, or this norm.

Definition II.3 (Orthogonal or unitary matrix)**.** *A matrix* $Q \in \mathbb{R}^{m\times m}$ *is* orthogonal *if one of the following equivalent conditions is fulfilled*

- $QQ^{\mathrm{T}} = Q^{\mathrm{T}}Q = \mathrm{Id}$*;*
- *for all* $x \in \mathbb{R}^m$, $\|Qx\|_2 = \|x\|_2$*;*
- *for all* $x, y \in \mathbb{R}^m$, $\langle Qx \mid Qy\rangle = \langle x \mid y\rangle$.

A matrix $Q \in \mathbb{C}^{m\times m}$ *is* unitary *if one the following equivalent conditions is fulfilled*

- $QQ^\star = Q^\star Q = \mathrm{Id}$*;*
- *for all* $x \in \mathbb{C}^m$, $\|Qx\|_2 = \|x\|_2$*;*
- *for all* $x, y \in \mathbb{C}^m$, $\langle Qx \mid Qy\rangle = \langle x \mid y\rangle$.

Problem II.2.

1. *Prove that in the orthogonal or unitary cases, the three conditions are indeed equivalent.*
2. *Show that the set of orthogonal matrices of size* n *is a group, that is it satisfies the three following conditions:*
   - Id *is orthogonal;*
   - *if* A *is orthogonal, then so is* A^{-1}*;*
   - *if* A *and* B *are orthogonal, then so is* AB.
3. *Show that the set of unitary matrices of size* n *is a group.*

Solution.

1. Let us consider the complex case.

   (iii) $\Rightarrow$ **(ii)** It suffices to take $y = x$.

   (iii) $\Rightarrow$ **(i)** Let x be the ith vector of the canonical basis and y be the jth one. Then $\langle x \mid y\rangle$ is 1 if $i = j$ and 0 otherwise. Furthermore, since $\langle Qx \mid Qy\rangle = x^\star Q^\star Qy$ and $\langle Qx \mid Qy\rangle = \langle x \mid y\rangle$, it is also the coefficient in position (i, j) of $Q^\star Q$. Hence $Q^\star Q = \mathrm{Id}$.

   (i) $\Rightarrow$ **(iii)** $\langle Qx \mid Qy\rangle = x^\star Q^\star Qy = x^\star y = \langle x \mid y\rangle$.

   (ii) $\Rightarrow$ **(iii)** This is done by expanding $\|Q(x + y)\|_2^2$ (and $\|Q(x + \mathrm{i}y)\|^2$ in the unitary case).

2.
   - Clearly $\mathrm{Id}^{\mathrm{T}}\,\mathrm{Id} = \mathrm{Id}$.
   - Since $A^{\mathrm{T}} = A^{-1}$, then the first condition implies that A^{-1} is orhogonal.
   - $(AB)^{\mathrm{T}}(AB) = B^{\mathrm{T}}A^{\mathrm{T}}AB = B^{\mathrm{T}}\,\mathrm{Id}\,B = B^{\mathrm{T}}B = \mathrm{Id}$, hence AB is orthogonal.
3. Exactly the same, just replacing $^{\mathrm{T}}$ by *.

Example II.4. *The following matrices A and B are respectively orthogonal and unitary.*

$$A = \begin{pmatrix} \frac{3}{5} & -\frac{4}{5} \\ \frac{4}{5} & \frac{3}{5} \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 1 & 0 \\ \frac{\mathrm{i}}{2} & 0 & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & 0 & -\frac{\mathrm{i}}{2} \end{pmatrix}.$$

Problem II.3. *Is the matrix A of Example II.4 unitary? Is the matrix B of Example II.4 orthogonal?*

Solution. Since real matrices are their own conjugate, if $A \in \mathbb{R}^{m\times m}$, then $A^{\star} = A^{\mathrm{T}}$. Hence, real orthogonal matrices are unitary. However, this is not true for complex not real matrices.

A is unitary but B is not orthogonal (even if we consider a complex matrice Q to be orthogonal if $Q^{\mathrm{T}}Q = \mathrm{Id}$.

Problem II.4. *Let $Q \in \mathbb{C}^{m\times m}$ be unitary. Let $n \in \mathbb{N}$, $m > n > 0$ and let $\tilde{Q}$ be the matrix formed by the first n columns of Q.*

1. *Show that $\tilde{Q}^{\star}\tilde{Q} = \mathrm{Id}$.*
2. *What can be said about $\tilde{Q}\tilde{Q}^{\star}$?*

Solution.

1. $\tilde{Q}^{\star}\tilde{Q}$ is the top $n \times n$-bloc of $Q^{\star}\tilde{Q}$, hence it is the identity of size n.
2. Not much!

II.1.2. Solving a least-square problem

Theorem II.5. *For any full-rank matrix $A \in \mathbb{C}^{m\times n}$ with $m \geq n$, a QR decomposition of A with Q unitary and R upper triangular exists.*

Proof. We denote $u_1, \dots, u_n$, the column vectors of A. We let $q_1 = \frac{u_1}{\|u_1\|_2}$. This is an orthonormal basis of the vecyor space spanned by u_1.

Assuming $q_1, \dots, q_i$ form an orthonormal basis of the vector space spanned by $u_1, \dots, u_i$, we let q_{i+1} be a vector in $\mathrm{Span}(u_1, \dots, u_{i+1})$ such that $q_1, \dots, q_{i+1}$ is an orthonormal family.

The family $q_1, \dots, q_n$ can be extended into an orthonormal family $q_1, \dots, q_m$ (for instance by first extending the free family $u_1, \dots, u_n$ into a basis $u_1, \dots, u_m$ and applying the same process).

Then, the matrix Q whose columns are $q_1, \dots, q_m$ is unitary. Furthermore, for all i, there exist $r_{i,1}, \dots, r_{i,i}$ such that $u_i = \sum_{j=1}^{i} r_{i,j}q_j$, hence $R = (r_{i,j})_{1\leq i,j\leq n}$ is upper triangular and $A = QR$. □

First, it is important to notice that minimizing $\|Ax' - b\|_2$ is equivalent to minimizing

$$\|Ax' - b\|_2^2 = \|Q^\star(Ax' - b)\|_2^2 = \|Rx' - c\|_2^2, \quad c = Q^\star b.$$

Now, if $m > n$, then R has $m - n$ rows of zeroes and

$$\|Ax' - b\|_2^2 = \|R_{1,\dots,n}x' - c_{1,\dots,n}\|_2^2 + \|c_{n+1,\dots,m}\|_2^2.$$

If A has full rank, then so is $R_{1,\dots,n}$ and x' is found by solving the triangular system $R_{1,\dots,n}x' = c_{1,\dots,n}$, where $R_{1,\dots,n}$ is the matrix formed with the n first rows of R and $c_{1,\dots,n}$ (resp. $c_{n+1,\dots,m}$) is the vector form with the n first (resp. $m - n$ last) rows of c.

Example II.6. *A QR decomposition of*

$$A = \begin{pmatrix} 3 & -3 \\ 4 & -4 \\ 0 & 40 \end{pmatrix} = \begin{pmatrix} \frac{3}{5} & 0 & \frac{4}{5} \\ \frac{4}{5} & 0 & -\frac{3}{5} \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 5 & -5 \\ 0 & 40 \\ 0 & 0 \end{pmatrix}.$$

Thus, finding x' such that $\|Ax' - b\|_2$ is minimal with $b = \begin{pmatrix} 5 \\ 10 \\ 2 \end{pmatrix}$ comes down to solving

$$\begin{pmatrix} 5 & -5 \\ 0 & 40 \end{pmatrix} x' - \begin{pmatrix} \frac{3}{5} & \frac{4}{5} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 5 \\ 10 \\ 2 \end{pmatrix} = 0 \iff \begin{pmatrix} 5 & -5 \\ 0 & 40 \end{pmatrix} x' - \begin{pmatrix} 11 \\ 2 \end{pmatrix} = 0.$$

Thus $x' = \begin{pmatrix} \frac{9}{4} \\ \frac{1}{20} \end{pmatrix}$.

Problem II.5. *Let $A \in \mathbb{C}^{m\times n}$, $Q \in \mathbb{C}^{m\times m}$ and $R \in \mathbb{C}^{m\times n}$ such that $A = QR$, Q is unitary and R is upper triangular.*

Let $Q' \in \mathbb{C}^{m\times n}$ made from the first n columns of Q and $R' \in \mathbb{C}^{n\times n}$ made from the first n rows of R.

Show that $A = Q'R'$.

Solution. The $m - n$ last rows of R are only made of 0. These coefficients are multiplied by the coefficients of the last $m - n$ columns of Q. Therefore, these columns of Q and these rows of R are useless in the product to obtain A.

Problem II.6. *Find $x' \in \mathbb{C}^3$ such that $\|QRx' - b\|_2$ is minimal for*

$$Q = \begin{pmatrix} \frac{5}{13} & 0 & \frac{12}{13} & 0 \\ 0 & -\frac{3}{5} & 0 & \frac{4}{5}\mathrm{i} \\ \frac{12}{13} & 0 & -\frac{5}{13} & 0 \\ 0 & -\frac{4}{5}\mathrm{i} & 0 & \frac{3}{5}\mathrm{i} \end{pmatrix}, \; R = \begin{pmatrix} 1 & -3 & 0 \\ & -1 & 4 \\ & & 140 \\ & & 0 \end{pmatrix}, \; b = \begin{pmatrix} 130 \\ 130 \\ 130 \\ 130 \end{pmatrix}.$$

Solution. We solve $R'x' = Q'^\star b$, where Q' and R' are defined as in Problem II.5. We have

$$Q'b = \begin{pmatrix} 170 \\ -78 + 104I \\ 70 \end{pmatrix}, \; x = \begin{pmatrix} 410 - 312I \\ 80 - 104I \\ \frac{1}{2} \end{pmatrix}.$$

II.2. Computing a QR Decomposition

II.2.1. Givens' method

As in the Gaussian elimination, the goal is to make some zeroes appear under the diagonal of the matrix. However, given a column, we cannot make all the zeroes appear in this column in one round.

The idea to put a 0 in position (i, j) in the matrix R, with $i > j$, is to multiply the matrix by a $m \times m$ rotation matrix $G_{i,j}$ on the left. Over $\mathbb{R}$, $G_{i,j}$ is a matrix with 0 coefficients everywhere except:

- $g_{k,k} = 1$ for all $1 \leq k \leq m$, $k \neq i, j$;
- $g_{i,i} = g_{j,j} = c$, $g_{j,i} = -g_{i,j} = s$ with $c^2 + s^2 = 1$.

$$G_{i,j} = \begin{pmatrix} c & 0 & s & 0 \\ 0 & 1 & 0 & 0 \\ -s & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, i = 3, j = 1$$

for some c and s. The coefficients c and s are directly given by $r_{j,j}$ and $r_{i,j}$ with

$$c = \frac{r_{j,j}}{\sqrt{r_{j,j}^2 + r_{i,j}^2}}, \quad s = \frac{r_{i,j}}{\sqrt{r_{j,j}^2 + r_{i,j}^2}}.$$

Then, the algorithm is

Algorithm 3: Givens' algorithm

Input: A real matrix A of size $m \times n$.
Output: Its QR Decomposition.
$Q \leftarrow \mathrm{Id}_m$
$R \leftarrow A$
For j **from** 1 **to** n **do**
  For i **from** $j + 1$ **to** m **do**
    $R \leftarrow G_{i,j} R$
    $Q \leftarrow Q G_{i,j}^{\mathrm{T}}$
Return Q, R

Problem II.7. *Using Givens' method, compute a QR decomposition of*

$$\begin{pmatrix} 3 & -3 & -2 \\ 4 & -4 & 14 \\ 12 & -12 & 24 \\ 0 & 3 & -5 \end{pmatrix}.$$

Solution.

$$\begin{pmatrix} 3/13 & 0 & -4/5 \\ 4/13 & 0 & 3/5 \\ 12/13 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 13 & -13 & 26 \\ 0 & 3 & -5 \\ 0 & 0 & 10 \end{pmatrix}$$

Problem II.8.

1. *Show that if $A \in \mathbb{R}^{m\times m}$ is orthogonal, then* $\det A = \pm 1$.
2. *Show that at the end of Givens' method,* $\det Q = 1$.

Solution.

1. Since $AA^{\mathrm{T}} = \mathrm{Id}$, then $\det A \det A^{\mathrm{T}} = (\det A)^2 = \det \mathrm{Id} = 1$, hence $\det A = \pm 1$.
2. Rotation matrices, as defined above, have determinant $c^2 + s^2 = 1$. Hence, the orthogonal matrix obtained with Givens' method has determinant 1.

II.2.2. Gram–Schmidt's method

The main idea of the Gram–Schmidt method is to construct an orthonormal basis from the column vector of the input matrix A. More precisely, if $a_1, \dots, a_n$ are the columns of A, we want $q_1, \dots, q_n$ to satisfy

1. $(q_1, \dots, q_n)$ is an orthonormal family;
2. for all i, $q_1, \dots, q_i$ span the same vector space as $a_1, \dots, a_i$.

In the following algorithm, $r_{i,j}$ is the coefficient of R in position (i, j) while q_j (resp. a_j) is the jth column vector of Q (resp. A).

Problem II.9. *Using Gram–Schmidt's method, compute a QR decomposition of*

$$\begin{pmatrix} -7 & 21 \\ -4 & 26 \\ -4 & -2 \\ 0 & 7 \end{pmatrix}.$$

Solution.

$$\begin{pmatrix} -7/9 & 0 \\ -4/9 & 2/3 \\ -4/9 & -2/3 \\ 0 & 1/3 \end{pmatrix} \begin{pmatrix} 9 & -27 \\ 0 & 21 \end{pmatrix}.$$

Problem II.10. 1. *Compute a QR decomposition of*

$$A = \begin{pmatrix} 3 & 2 & 16 \\ 4 & 11 & 13 \\ 0 & 0 & 12 \\ 0 & 0 & 9 \end{pmatrix}.$$

Algorithm 4: Gram–Schmidt's algorithm

Input: A matrix A of size $m \times n$.
Output: Its QR Decomposition.

$r_{1,1} \leftarrow \|a_1\|_2$
$q_1 \leftarrow \frac{a_1}{r_{1,1}}$
For j **from** 2 **to** n **do**
  $q_j = a_j$
  For i **from** 1 **to** $j-1$ **do**
    $r_{i,j} \leftarrow q_i^\star q_j$
    $q_j \leftarrow q_j - r_{i,j} q_i$
  $r_{j,j} \leftarrow \|q_j\|_2$
  $q_j \leftarrow \frac{q_j}{r_{j,j}}$
Return Q, R

2. *Compute the vector x' such that $\|Ax' - b\|_2$ is minimal for*

$$b = -\begin{pmatrix} 21 \\ 3 \\ 33 \\ 6 \end{pmatrix}.$$

Solution.

$$A = \begin{pmatrix} 3/5 & -4/5 & 0 \\ 4/5 & 3/5 & 0 \\ 0 & 0 & 4/5 \\ 0 & 0 & 3/5 \end{pmatrix} \begin{pmatrix} 5 & 10 & 20 \\ & 5 & -5 \\ & & 15 \end{pmatrix}, \quad x' = \begin{pmatrix} 3 \\ 1 \\ -2 \end{pmatrix}.$$

Problem II.11 (Householder transformation). *The goal is to study another QR decomposition method.*

1. *Let z be a vector of size m. Let $v = z - \alpha e_1$ with $\alpha = \|z\|$ and e_1 the first vector of the canonical basis and let $u = v/\|v\|$. Let*

$$Q = \mathrm{Id} - 2uu^*.$$

Show that Q is unitary and that $Qz = \alpha e_1$. The matrix Q is a Householder transformation.

2. *Let R be a matrix of size $m \times n$. Let z be the first column vector of R and let Q be defined as above. Show that the first column of QR has a nonzero first coefficient and only zeroes below.*

3. *Let R_0 be a matrix of size $m \times n$, we know, using the previous question, how to determine a Householder transformation Q_1 such that $R_1 = Q_1 R_0$ only has zeroes under the diagonal of*

its first column. Explain how to determine

$$Q_2 = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{Id} - 2u_2 u_2^* \end{pmatrix}$$

so that $R_2 = Q_2 R_1$ only has zeroes under the diagonal of its first and second columns.

4. *Give an algorithm iterating this process in order to obtain an upper triangular matrix through multiplications by Householder matrices.*

Solution.

1. The vector u has norm 1 so $u^\star u = 1$, hence $Q^\star Q = (\mathrm{Id} - 2uu^\star)^\star(\mathrm{Id} - 2uu^\star) = (\mathrm{Id} - 2uu^\star)^2 = \mathrm{Id} - 4uu^\star + 4uu^\star uu^\star = \mathrm{Id}$.

   We have $\|v\|^2 = \|z\|^2 + 2\,\mathfrak{Re}(\langle z, \alpha e_1\rangle) + \|\alpha e_1\|^2 = 2\|z\|^2 + 2\,\mathfrak{Re}(\langle z, \alpha e_1\rangle) = 2\,\mathfrak{Re}(\langle z, z + \alpha e_1\rangle) = 2\,\mathfrak{Re}(\langle z, v\rangle)$.

   Hence, $Qz = z - 2\frac{vv^*}{\|v\|^2} z = \alpha e_1$.

2. By construction, the first column of QR is the vector αe_1 defined above.

3. Same, with the second vector.

4. –

II.3. Diagonalization

Definition II.7. *A $n \times n$ matrix A is diagonalizable if there exists P invertible such that $A = PDP^{-1}$ with D diagonal. The coefficients of D are the* eigenvalues *and the column vectors of P are the* eigenvectors *of A.*

We know that A is diagonalizable if

- A is real symmetric ($A = A^{\mathrm{T}}$) or complex Hermitian ($A^\star = \bar{A}^{\mathrm{T}} = A$), the eigenvalues are then real;
- A is normal ($AA^\star = A^\star A$);
- the eigenvalues of A are all distinct.

Problem II.12. *Show whether the matrices*

$$A_\varepsilon = \begin{pmatrix} 1 & 1 \\ 0 & 2+\varepsilon \end{pmatrix}, \quad B_\varepsilon = \begin{pmatrix} 1 & 1 \\ 0 & 1+\varepsilon \end{pmatrix}$$

are diagonalizable for $\varepsilon \geq 0$.

Solution. For A_ε, the eigenvalues are 1 and $2 + \varepsilon$ with multiplicity 1, so A_ε is diagonalizable. For $\varepsilon > 0$, B_ε has eigenvalues 1 and $1 + \varepsilon$ with multiplicity 1 so B_ε is diagonalizable. For $\varepsilon = 0$, B_0 has eigenvalue 1 with multiplicity 2 but the associated eigenspace has dimension 1, B_0 is not diagonalizable.

II.3.1. Computing the eigenvalues

Numerically, it is possible to compute the eigenvalues and eigenvectors of a $n \times n$ matrix A without computing its characteristic polynomial.

1. Find V unitary such that $V^\star AV = H$ with H
   - tridiagonal if A is Hermitian;
   - upper Hessenberg (the coefficients below the first subdiagonal are 0).

   Notice that if $H = PDP^{-1}$, then $A = VPDP^{-1}V^\star = (VP)D(VP)^{-1}$.
2. Compute the diagonalization of H iterating about $5n$ times the following:
   a. Compute the QR decomposition of $H = QR$;
   b. Replace H by RQ. Since $RQ = Q^\star QRQ = Q^\star HQ$, the new H has the same eigenvalues as the former H.
3. The elements outside the diagonal vanish allowing us to read the eigenvalues of H.

Implementation II.1. *Implement the three QR decompositions algorithms over double-precision floating-point numbers.*
Compare their efficiency.

Implementation II.2. *Implement the three QR decompositions algorithms over multi-precision floating-point numbers.*
Compare their efficiency.

III. Matrix and vector compression, SVD and FFT

III.1. Singular Value Decomposition

The Singular Value Decomposition is a decomposition that exists for any $m \times n$ complex matrix. It is related to the diagonalization of matrices. Furthermore, if a grayscale picture is seen as a matrix, its SVD allows us to compress the picture.

Definition III.1. *Let $A \in \mathbb{C}^{m\times n}$ with $m \geq n$. There exists U, V, Σ such that*

- $A = U\Sigma V^\star$;
- *U has size $m \times m$ and $U^\star U = \mathrm{Id}$;*
- *$\Sigma \in \mathbb{R}^{m\times n}$ and its nonzero elements are on the diagonal and satisfy $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$;*
- *V has size $n \times n$ and $V^\star V = \mathrm{Id}$.*

If $A = U\Sigma V^\star$, then $A^\star A = V\Sigma^\star\Sigma V^\star$. Therefore, $\sigma_1^2, \ldots, \sigma_n^2$ are the eigenvalues of $A^\star A$ and the columns of V are their eigenvectors.

Example III.2. *The SVD of the matrix*

$$M = \begin{pmatrix} 2 & 4 & 4 \\ 0 & 8 & 3 \\ 12 & 16 & 11 \end{pmatrix}$$
$$= \begin{pmatrix} -2.368 & -6.928 \cdot 10^{-1} & -9.691 \\ -2.984 & -9.441 & 1.404 \\ -9.246 & 3.224 & 2.029 \end{pmatrix} \begin{pmatrix} 2.463 \cdot 10^1 & & \\ & 4.609 & \\ & & 1.409 \end{pmatrix} \begin{pmatrix} -4.696 & -7.359 & -4.877 \\ 8.094 & -5.795 & 9.492 \\ 3.524 & 3.502 & -8.678 \end{pmatrix} \cdot 10^{-2}.$$

By setting to 0, the last two singular values, we can compress the representation into

$$\begin{pmatrix} -2.368 \\ -2.984 \\ -9.246 \end{pmatrix} \left(2.463 \cdot 10^1\right) \begin{pmatrix} -4.696 & -7.359 & -4.877 \end{pmatrix} \cdot 10^{-2} = \begin{pmatrix} 2.740 & 4.293 & 2.845 \\ 3.452 & 5.409 & 3.585 \\ 1.070 \cdot 10^1 & 1.676 \cdot 10^1 & 1.111 \cdot 10^1 \end{pmatrix}.$$

III.1.1. Computing the SVD

The computation is straightforward:

1. Compute $A^\star A$.
2. Compute its diagonalization $A^\star A = VDV^\star$.
3. Let Σ be the $m \times n$ matrix whose diagonal elements are the non-negative square roots of the diagonal elements of D in decreasing order.
4. Solve $U\Sigma = AV$ with U unitary.

III.1.2. Properties of the SVD

Problem III.1. *Let $A = U\Sigma V^{\mathrm{T}}$ be the SVD of a matrix A of size $m \times n$ with $m \geq n$.*

1. *Show that if A has full rank, then the solution to $\min_x \|Ax - b\|_2$ is $x = V\Sigma^{-1}U^{\mathrm{T}}b$.*
2. *Let us recall that*
$$\|A\|_2 = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \max_{\|x\|_2 = 1} \|Ax\|_2.$$
Show that $\|A\|_2 = \sigma_1$ and that if A is an invertible square matrix, then $\|A^{-1}\|_2 = \sigma_n^{-1}$ and $\|A\|_2 \cdot \|A^{-1}\|_2 = \frac{\sigma_1}{\sigma_n}$.
3. *Let us write $U = [u_1, u_2, \ldots, u_n]$ and $V = [v_1, v_2, \ldots, v_n]$, with $u_1, \ldots, u_n, v_1, \ldots, v_n$ column-vectors. We have*
$$A = U\Sigma V^{\mathrm{T}} = \sum_{i=1}^{n} \sigma_i u_i v_i^{\mathrm{T}}.$$
Show that the closest matrix of rank $k < n$ (for $\|\cdot\|_2$) to A is $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^{\mathrm{T}}$ and that $\|A - A_k\|_2 = \sigma_{k+1}$.

Solution.

1. As V is an orthogonal matrix, solving $\min_x \|Ax-b\|_2$ is equivalent to solving $\min_x \|AV(V^{\mathrm{T}}x)-b\|_2$. Since A has full rank, then $AV = U\Sigma$ and this is a QR decomposition, hence the solution is exactly $V^{\mathrm{T}}x = \Sigma^{-1}U^{\mathrm{T}}b$, which is equivalent to $x = V\Sigma^{-1}U^{\mathrm{T}}b$.
2. Let $y = Ax$ with x of norm 1. Since $A = U\Sigma V^{\mathrm{T}}$, then $x' = V^{\mathrm{T}}x$ has norm 1 as well and $y' = U^{\mathrm{T}}y$ has the same norm as y. We want $\max_{\|x'\|_2} \|\Sigma x'\|_2$. Clearly, this max is achieved for $x' = (1, 0, \ldots, 0)^{\mathrm{T}}$.

   Conversely, if A is invertible, its inverse satisfies $A^{-1} = V\Sigma^{-1}U^{\mathrm{T}}$ and its largest singular value is the inverse of the smallest of A.
3. By construction, A_k has rank at most k.
$$\|A - A_k\|_2 = \left\|\sum_{i=k+1}^{n} \sigma_i u_i v_i^{\mathrm{T}}\right\|_2 = \left\| U \begin{pmatrix} 0 & & \\ & \sigma_{k+1} & \\ & & \ddots \\ & & & \sigma_n \end{pmatrix} V^{\mathrm{T}} \right\|_2 = \sigma_{k+1}$$

Let us show that no matrix of rank k or less is closer. Let B be a matrix of rank at most k, then its kernel has rank at least $n-k$. The vector space spanned by $v_1, \dots, v_{k+1}$ has dimension $k+1$ and the intersection of these two subspaces is non trivial. Let w be in this intersection, of norm 1, then

$$\begin{aligned}\|A-B\|_2 &\geqslant \|(A-B)w\|_2 = \|Aw\|_2 = \|U\Sigma V^{\mathrm{T}}w\|_2 \\ &\geqslant \|\Sigma(V^{\mathrm{T}}w)\|_2 \geqslant \sigma_{k+1}\|V^{\mathrm{T}}w\|_2 \geqslant \sigma_{k+1}.\end{aligned}$$

Observe that this yields a compression algorithm with loss if we see a grayscale as a matrix of pixels.

The SVD can also be used for computing the pseudo-inverse of a matrix.

Definition III.3 (Pseudoinverse)**.** *Let $A \in \mathbb{C}^{m\times n}$ and $A = U\Sigma V^\star$ be its SVD. Denote $\Sigma = (\sigma_{i,j})_{\substack{0\leq i<m\\0\leq j<n}}$. Let $T = (\tau_{i,j})_{\substack{0\leq i<n\\0\leq j<m}} \in \mathbb{C}^{n\times m}$ defined by $\tau_{i,j} = 0$ if $i \neq j$, $\tau_{i,i} = \sigma_{i,i}^{-1}$ if $\sigma_{i,i} \neq 0$ and $\tau_{i,i} = 0$ otherwise.*

Then, the pseudoinverse *of A is $A^\dagger = VTU^\star$.*

Problem III.2.

1. *Show that if Σ only has nonzero coefficients on its diagonal, then $\Sigma^\dagger = T$, as defined in Definition III.3.*
2. *Show that if $A \in \mathbb{C}^{m\times m}$ is invertible, then $A^\dagger = A^{-1}$.*

Solution.

1. It suffices to apply the definition, since $U = V = \mathrm{Id}$.
2. $A^{-1} = V\Sigma^{-1}U^\star$ but by construction $T = \Sigma^{-1}$.

III.2. Fast Fourier Transform

The FFT was invented by Carl Friedrich Gauß in 1866. The modern FFT algorithm is due to James W. Cooley and John W. Tukey in 1965.

The FFT is a bijective linear map on vectors in $\mathbb{C}^n$. It can be used to multiply fast polynomials, by identifying a polynomial of degree at most $n-1$ with its vector of coefficients of size n. Technically, the FFT is the transformation and the inverse FFT is the inverse transformation. We shall see that the inverse FFT is a kind of FFT itself.

III.2.1. Definition

Definition III.4. *For n a positive integer, the nth roots of unity are all the roots of the polynomial $z^n - 1$. They form exactly the set $\left\{ \mathrm{e}^{\frac{2\mathrm{i}\ell\pi}{n}} \,\middle|\, \ell \in \{0,\dots,n-1\} \right\}$ in $\mathbb{C}$.*

An nth root of unity is primitive *if it is not a kth root of unity for $1 \leq k \leq n-1$. In $\mathbb{C}$, $\mathrm{e}^{\pm\frac{2\mathrm{i}\pi}{n}}$ are always primitive.*

Furthermore, if z is a (primitive) 2^kth root of unity, then z^2 is a (primitive) 2^{k-1}th root of unity.

Definition III.5. *Let $n \in \mathbb{N}$, $n > 0$, and ω be a primitive nth root of unity in $\mathbb{C}$. Let $v \in \mathbb{C}^n$. The FFT of v is the vector $\Omega_n v$, where*

$$\Omega_n = \left(\omega^{ij}\right)_{0\leq i,j<n} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \omega & \cdots & \omega^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & \omega^{n-1} & \cdots & \omega^{(n-1)^2} \end{pmatrix}.$$

This matrix of the Vandermonde matrix of $1, \omega, \ldots, \omega^{n-1}$.

Problem III.3.

1. *What are the primitive nth root of unity in $\mathbb{C}$ for $n = 1, 2, 3, 4, 5, 6$?*
2. *Let $n \in \mathbb{N}^*$, what is the sum of all the nth roots of unity?*
3. *What is their product?*
4. *What is the FFT of $(1, 0, 0, 0)^{\mathrm{T}}$, assuming $\omega = \mathrm{i}$?*
5. *Which vector has $(1, 0, 0, 0)^{\mathrm{T}}$ as its FFT with the same assumption?*

Solution.

1. $n = 1$. 1;
   $n = 2$. -1;
   $n = 3$. j and j^2, where $\mathrm{j} = \mathrm{e}^{2\mathrm{i}\pi/3}$.
   $n = 4$. i and $-\mathrm{i}$.
   $n = 5$. $\mathrm{e}^{2\mathrm{i}k\pi/5}$ for $1 \leq k \leq 4$.
   $n = 6$. $\mathrm{e}^{\mathrm{i}k\pi/3}$ for $k = 1$ and $k = 5$.
2. $S = \sum_{k=0}^{n-1} \mathrm{e}^{2\mathrm{i}k\pi/n}$. If $n = 1$, then $S = 1$. Otherwise, $S = \frac{1-\mathrm{e}^{2\mathrm{i}n\pi/n}}{1-\mathrm{e}^{2\mathrm{i}\pi/n}} = 0$.
3. The product is $P = \prod_{k=0}^{n-1} \mathrm{e}^{2\mathrm{i}k\pi/n} = (-1)^{n+1}$.
4. This is the evaluation of the polynomial $1 + 0x + 0x^2 + 0x^3$ in $1, \mathrm{i}, -1, -\mathrm{i}$. Hence it is $(1, 1, 1, 1)$.
5. This is the evaluation of the same polynomial in $1, -\mathrm{i}, -1, \mathrm{i}$ divided by 4. Hence it is $(1/4, 1/4, 1/4, 1/4)$.

Problem III.4.

1. *Compute the inverse of the matrix Ω_n for $n = 1, 2, 3, 4$.*
2. *Let Ω_2 and Ω_4 be the matrix of Definition III.5 for respectively $n = 2$ and $n = 4$.*
   a. *Let Ω'_4 be the matrix obtained from Ω_4 after permuting the columns 1 and 2 (numbered from 0 to 3). Give Ω'_4.*
   b. *Show that Ω'_4 is a (2×2)-block matrix whose blocks are derived from Ω_2.*

Lemma III.6. *Let $\omega \in \mathbb{C}$ be a primitive nth root of unity and let $p = (p_0, \dots, p_{n-1})^{\mathrm{T}} \in \mathbb{C}^n$. Let $\Omega_n = \left(\omega^{ij}\right)_{0 \le i,j < n}$ be the Vandermonde matrix of $1, \omega, \dots, \omega^{n-1}$ and $P = p_{n-1}x^{n-1} + \cdots + p_0$ be the polynomial whose vector of coefficients is p.*

Then, $\Omega_n p$ is the vector $\left(P(1), P(\omega), \dots, P(\omega^{n-1})\right)^{\mathrm{T}}$.

Problem III.5. *Prove Lemma III.6.*

Solution. Expand the product.

To perform this matrix-vector efficiently, we will rely polynomial evaluation.

III.2.2. Evaluation by Divide and Conquer

We now assume that $n = 2^k$. Let us notice that since ω is a primitive nth root of unity, then for all i, ω^i and $-\omega^i$ are also nth roots of unity. Indeed, the powers of ω are in fact $1, \omega, \omega^2, \dots, \omega^{\frac{n}{2}-1}, \omega^{\frac{n}{2}} = -1, \omega^{\frac{n}{2}+1} = -\omega, \omega^{\frac{n}{2}+2} = -\omega^2, \dots, \omega^{n-1} = -\omega^{\frac{n}{2}-1}$.

Lemma III.7. *Let $P = p_{n-1}x^{n-1} + \cdots + p_0$ be a polynomial of degree $n-1$. Let P_o and P_e be polynomials of degree $\frac{n}{2} - 1$ defined by*

$$P = P_e(x^2) + xP_o(x^2).$$

Then,

$$P_e = p_{n-2}x^{\frac{n-2}{2}} + p_{n-4}x^{\frac{n-4}{2}} + \cdots + p_2 x + p_0$$
$$P_o = p_{n-1}x^{\lfloor \frac{n-1}{2} \rfloor} + p_{n-3}x^{\lfloor \frac{n-3}{2} \rfloor} + \cdots + p_3 x + p_1.$$

Furthermore, evaluating P in $1, \omega, \omega^2, \dots, \omega^{n-1}$ comes down to evaluating P_e and P_o in $1, \omega^2, \omega^4, \dots, \omega^{n-2}$.

Proof. Since $P(\omega^i) = P_e(\omega^{2i}) + \omega^i P_o(\omega^{2i})$ and $P(-\omega^i) = P_e(\omega^{2i}) - \omega^i P_o(\omega^{2i})$, we can evaluate P in ω^i and $-\omega^i$ by evaluating P_e and P_o in ω^{2i} plus one multiplication by ω^i and two additions or subtractions. □

Example III.8. *$P = 10x^5 + x^4 + 2x^3 + 6x^2 + 4x + 3 = (x^4 + 6x^2 + 3) + x(10x^4 + 2x^2 + 4)$ so that $P_e = x^2 + 6x + 3$ and $P_o = 10x^2 + 2x + 4$.*

To evaluate P in $1, \omega = \mathrm{i}, \omega^2 = -1, \omega^3 = -\mathrm{i}$,

- *we evaluate* P_e *and* P_o *in* $1, \omega^2 = -1$
  - $P_e(1) = 10$ *and* $P_e(-1) = -2$*;*
  - $P_o(1) = 16$ *and* $P_o(-1) = 12$*;*
- *we recombine these evaluations*
  - $P(1) = P_e(1) + 1 \cdot P_o(1) = 10 + 1 \cdot 16 = 26$*;*
  - $P(\mathrm{i}) = P_e(-1) + \mathrm{i} \cdot P_o(-1) = -2 + \mathrm{i} \cdot 12 = -2 + 12\mathrm{i}$*;*
  - $P(-1) = P_e(1) - 1 \cdot P_o(1) = 10 - 1 \cdot 16 = -6$*;*
  - $P(-\mathrm{i}) = P_e(-1) - \mathrm{i} \cdot P_o(-1) = -2 - \mathrm{i} \cdot 12 = -2 - 12\mathrm{i}$.

Since $\frac{n}{2} = 2^{k-1}$, we can reapply this process on P_e and P_o to evaluate them in $1, \omega^2, \ldots, \omega^{n-2}$ by building $P_{ee}, P_{eo}, P_{oe}, P_{oo}$ and evaluating them in $1, \omega^4, \ldots, \omega^{n-4}$ until we end up on the base case: evaluating a polynomial in 1.

Example III.9 (Continuation of Example III.8)**.** *To evaluate* $P_e = x^2 + 6x + 3$ *in* $1, \omega^2 = -1$*, we split it into* $P_{ee} = x + 3$ *and* $P_{eo} = 6$*. Likewise, we split* $P_o = 10x^2 + 2x + 4$ *into* $P_{oe} = 10x + 4$ *and* $P_{oo} = 2$*.*

- *We evaluate* P_{ee}*,* P_{oe}*,* P_{oe} *and* P_{oo} *in* 1
  - $P_{ee}(1) = 4$ *and* $P_{eo}(1) = 6$*;*
  - $P_{oe}(1) = 14$ *and* $P_{oo}(1) = 2$*.*
- *We recombine these evaluations*
  - $P_e(1) = P_{ee}(1) + 1 \cdot P_{eo}(1) = 4 + 1 \cdot 6 = 10$*;*
  - $P_e(-1) = P_{ee}(1) - 1 \cdot P_{eo}(1) = 4 - 1 \cdot 6 = -2$*;*
  - $P_o(1) = P_{oe}(1) + 1 \cdot P_{oo}(1) = 14 + 1 \cdot 2 = 16$*;*
  - $P_o(-1) = P_{oe}(1) - 1 \cdot P_{oo}(1) = 14 - 1 \cdot 2 = 12$*.*

This yields the following algorithm.

Example III.10. *Assume, we want to evaluate* $P = 32x^5 + 16x^4 + 8x^3 + 4x^2 + 2x + 1$ *with the FFT algorithm. The polynomial has size* 6*, so a primitive* 8*th root of unity* ω *is needed. We can choose* $\omega = \mathrm{e}^{\frac{\mathrm{i}\pi}{4}} = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}\mathrm{i}$*, we then have* $\omega^2 = \mathrm{i}$*,* $\omega^4 = -1$ *and* $\omega^8 = 1$*.*

1. *We call FFT*(P, ω)*.*
   a. *We split it into* $P_e = 16x^2 + 4x + 1$ *and* $P_o = 32x^2 + 8x + 2$*.*
   b. *We call FFT*(P_e, i)*.*
      i. *We split into* $P_{ee} = 16x + 1$ *and* $P_{eo} = 4$*.*
      ii. *We call FFT*$(P_{ee}, -1)$*.*

Algorithm 5: FFT

Input: A polynomial P of degree strictly less than $n = 2^k$ and ω a primitive nth root of unity.

Output: The evaluation of P in $1, \omega, \ldots, \omega^{n-1}$

If $\omega = 1$ **then Return** $(P(1))$

Split P into P_e and P_o.

Call FFT on P_e and $\tau = \omega^2$ to compute $(P_e(1), P_e(\tau), \ldots, P_e(\tau^{\frac{n}{2}-1}))$.

Call FFT on P_o and $\tau = \omega^2$ to compute $(P_o(1), P_o(\tau), \ldots, P_o(\tau^{\frac{n}{2}-1}))$.

For j **from** 0 **to** $\frac{n}{2} - 1$ **do**

$P(\omega^j) = P_e(\omega^{2j}) + \omega^j P_o(\omega^{2j}) = P_e(\tau^j) + \omega^j P_o(\tau^j)$.

$P(\omega^{\frac{n}{2}+j}) = P_e(\omega^{2j}) - \omega^j P_o(\omega^{2j}) = P_e(\tau^j) - \omega^j P_o(\tau^j)$.

Return $(P(1), P(\omega), \ldots, P(\omega^{n-1}))$.

. *We split into* $P_{eee} = 1$ *and* $P_{eeo} = 16$.

. *We call FFT*$(P_{eee}, 1)$ *and it returns* (1).

. *We call FFT*$(P_{eeo}, 1)$ *and it returns* (16).

. *It returns* $(1 + 16, 1 - 16) = (17, -15)$.

iii. We call FFT$(P_{eo}, -1)$.

. *We split into* $P_{eoe} = 4$ *and* $P_{eoo} = 0$.

. *We call FFT*$(P_{eoe}, 1)$ *and it returns* (4).

. *We call FFT*$(P_{eoo}, 1)$ *and it returns* (0).

. *It returns* $(4 + 0, 4 - 0) = (4, 4)$.

iv. It returns $(17 + 4, -15 + 4\mathrm{i}, 17 - 4, -15 - 4\mathrm{i}) = (21, -15 + 4\mathrm{i}, 13, -15 - 4\mathrm{i})$.

c. We call FFT(P_o, i).

i. We split into $P_{oe} = 32x + 2$ *and* $P_{oo} = 8$.

ii. We call FFT$(P_{oe}, -1)$.

. *We split into* $P_{oee} = 2$ *and* $P_{oeo} = 32$.

. *We call FFT*$(P_{oee}, 1)$ *and it returns* (2).

. *We call FFT*$(P_{oeo}, 1)$ *and it returns* (32).

. *It returns* $(2 + 32, 2 - 32) = (34, -30)$.

iii. We call FFT$(P_{oo}, -1)$.

. *We split into* $P_{ooe} = 8$ *and* $P_{ooo} = 0$.

. *We call FFT*$(P_{ooe}, 1)$ *and it returns* (8).

. *We call FFT*$(P_{ooo}, 1)$ *and it returns* (0).

. *It returns* $(8+0, 8-0) = (8, 8)$.

iv. It returns $(34+8, -30+8\mathrm{i}, 34-8, -30-8\mathrm{i}) = (42, -30+8\mathrm{i}, 26, -30-8\mathrm{i})$.

d. It returns

$$\begin{aligned}&(21+42, -15+4\mathrm{i}+(-30+8\mathrm{i})\omega, 13+26\mathrm{i}, -15-4\mathrm{i}+(-30-8\mathrm{i})\omega^3,\\ &\quad 21-42, -15+4\mathrm{i}-(-30+8\mathrm{i})\omega, 13-26\mathrm{i}, -15-4\mathrm{i}-(-30-8\mathrm{i})\omega^3)\\ &= (63, -15-19\sqrt{2}+4\mathrm{i}-11\mathrm{i}\sqrt{2}, 13+26\mathrm{i}, -15+19\sqrt{2}-4\mathrm{i}-11\mathrm{i}\sqrt{2}\\ &\quad -21, -15+19\sqrt{2}+4\mathrm{i}+11\mathrm{i}\sqrt{2}, 13-26\mathrm{i}, -15-19\sqrt{2}-4\mathrm{i}+11\mathrm{i}\sqrt{2}).\end{aligned}$$

This algorithm uses the structure of the Vandermonde matrix Ω_n made from all the nth roots of unity.

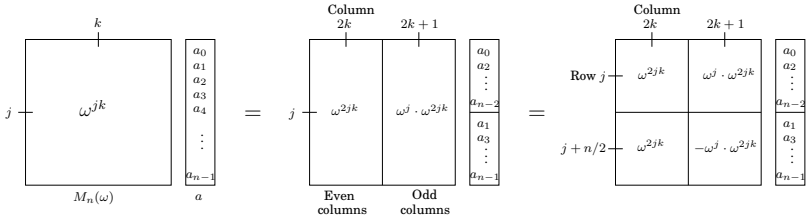

Figure III.1.. Recursive submatrix product

Example III.11. *The product*

$$V_{(1,\mathrm{i},-1,-\mathrm{i}),4}\begin{pmatrix}p_0\\p_1\\p_2\\p_3\end{pmatrix} = \begin{pmatrix}1&1&1&1\\1&\mathrm{i}&-1&-\mathrm{i}\\1&-1&1&-1\\1&-\mathrm{i}&-1&\mathrm{i}\end{pmatrix}\begin{pmatrix}p_0\\p_1\\p_2\\p_3\end{pmatrix}$$

can be rewritten

$$\left(\begin{array}{cc|cc}1&1&1&1\\1&-1&\mathrm{i}&-\mathrm{i}\\\hline 1&1&-1&-1\\1&-1&-\mathrm{i}&\mathrm{i}\end{array}\right)\left(\begin{array}{c}p_0\\p_2\\\hline p_1\\p_4\end{array}\right).$$

III.2.3. Interpolating by Divide and Conquer

The inverse operation of the FFT is the inverse FFT. It consists in multiplying a vector by the inverse matrix of Ω_n. Since multiplying a vector by Ω_n consists in evaluating a polynomial in the powers of ω, the inverse operation is an *interpolation*: from a vector $(y_0, \ldots, y_{n-1})^{\mathrm{T}} \in \mathbb{C}^n$, we want to find the unique polynomial P of degree at most $n-1$ such that $P(\omega^i) = y_i$ for all $0 \leq i < n$.

Proposition III.12. *Let $\omega \in \mathbb{C}$ be a primitive nth root of unity. Let $\Omega_n = \left(\omega^{ij}\right)_{0\le i,j<n}$. Then,*

$$\Omega_n^{-1} = \frac{1}{n}\left(\omega^{-ij}\right)_{0\le i<n} = \frac{1}{n}\bar{\Omega}_n.$$

In other words, the inverse FFT is an FFT with $\omega^{-1} = \bar{\omega}$ as the primitive nth root of unity followed by a division by n.

Problem III.6. *Prove this statement.*

Solution. The coefficient in position (i, j) is $\frac{1}{n}\sum_{k=0}^{n-1}\omega^{-ik}\omega^{kj} = \frac{1}{n}\sum_{k=0}^{n-1}\omega^{k(j-i)}$.

This is a geometric series. If $j - i = 0$, then the result is 1. Otherwise, as previously, the sum is 0.

Example III.13. *If we have found that $P(1) = 10, P(\mathrm{i}) = -2-2\mathrm{i}, P(-1) = -2$ and $P(-\mathrm{i}) = -2+2\mathrm{i}$, then its inverse FFT is the FFT of the polynomial $S = s_3x^3 + s_2x^2 + s_1x + s_0 = (-2+2\mathrm{i})x^3 - 2x^2 - (2+2\mathrm{i})x + 10$ with $\omega = -\mathrm{i}$ divided by 4. The FFT of S is $(4, 8, 12, 16)$ so that $P = 4x^3 + 3x^2 + 2x + 1$.*

Problem III.7 (Polynomial multiplication)**.** *Let P and Q be two polynomials over $\mathbb{C}$ of respective degrees $\ell - 1$ and $m - 1$. We shall use the FFT to multiply them.*

1. *Show that $R = PQ$ has exactly $n = \ell + m - 1$ coefficients.*
2. *Let p, q and r be the vectors of coefficients of P, Q and R seen as polynomials of degree at most $n - 1$. Let ω be a primitive nth root of unity in $\mathbb{C}$. What is the relation between the FFT of p and q on the one hand and the FFT of r on the other hand.*
3. *Propose a multiplication algorithm for P and Q, i.e. that computes R, using the FFT.*

Solution.

1. R has degree $\ell - 1 + m - 1 = \ell + m - 2 = n - 1$, hence it has n coefficients.
2. The FFTs of P, Q and R are their evaluations in the powers of ω. If $R = PQ$, then $R(\omega^i) = P(\omega^i)Q(\omega^i)$.

   Hence, if $u = (u_0, \ldots, u_{n-1})$, $v = (v_0, \ldots, v_{n-1})$ and $w = (w_0, \ldots, w_{n-1})$ are the FFTs of respectively P, Q and R, then $w_i = u_iv_i$.
3. a. Compute the FFTs of P and Q with ω a primitive nth root of unity, where $n > \deg P + \deg Q$.
   b. Multiply coefficient by coefficient these FFTs to get the FFT of R, where $R = PQ$.
   c. Compute the inverse FFT of the FFT of R to retrieve R.