Numerical linear algebra

Lecture notes

written by Csaba Gáspár, 2020

Contents

1 Vectors, matrices and linear systems

1.1 Motivations

In practice, a huge number of problems lead to linear systems of equations. The numerical solution techniques of these systems is of primary importance. To reflect to the essential differences between the 'pure', 'theoretical' mathematical problems and the numerical mathematical ones, consider the following two examples.

Example: Solve the following system of equations:

$$1000x + 999y = 1$$

$$999x + 998y = 1$$

The determinant of the system equals to $998000 - 999^2$ which is different from zero, therefore exatly one solution exists. It is easy to check that this solution is as follows: $x = 1$, $y = -1$.

Now consider the slightly modified system of equations:

$$1000x + 999y = 1$$

$$999x + 998y = 0.999$$

Intuitively, it is expected that since the change of the data is 'small' (0.1 percent), therefore the change of the solution remains also 'small'. However, now the solution is: $x = 0.001$, $y = 0$, which is completely different from the original solution.

This is the simplest example for the *ill-conditioned systems*, when the solution is very sensitive to the change of the data.

Example: How much time does it take (using the recently fastest computers) to compute the determinant of a 200×200 matrix with the usual recursive definition (i.e. by converting the problem to the calculations of the determinants of smaller matrices)?

Solution: Denote by c_N the neccessary multiplications in the computation of the determinant of a $N \times N$ matrix using the minor expansion formula with respect to the first row. Obviously, $c_N = N \cdot c_{N-1}$, which implies:

$$c_N = N \cdot c_{N-1} = N \cdot (N-1) \cdot c_{N-2} = ... = N!$$

i.e. $c_{200} = 200!$. This number is beyond the comprehension. If we consider a hypothetic computer which is as big as the Earth and its processors have the size in the order of magnitude of an atom, the speed of the information spreading is the speed of light, then the necessary computing time would exceed the time that passed from the 'Big Bang'.

This means that it is not sufficient to know that a method is valid in principle; it is also necessary to be able to estimate both the accuracy and also the computational complexity of the method.

In the remaining part of this introductory section, we briefly recall the main concepts and theorems of the linear algebra which will be important in the numerical methods in the following.

1.2 Vectors and matrices

Denote by $\mathbf{R}^N$ the set of ordered N-tuples of real numbers. Then $\mathbf{R}^N$ is a vector space with respect to the componentwise addition and the componentwise multiplication by scalars.

Let $\mathbf{M}_{N\times M}$ be the set of real-valued matrices with N rows and M columns. Then $\mathbf{M}_{N\times M}$ is a vector space with respect the elementwise addition and elementwise multiplication by scalars. If $A \in \mathbf{M}_{N\times M}$ and $B \in \mathbf{M}_{M\times K}$, then their product AB is defined by

$$(AB)_{kj} = \sum_{i=1}^{M} A_{ki} B_{ij}$$

Thus, $AB \in \mathbf{M}_{N\times K}$.

Unless otherwise specified, a vector $x \in \mathbf{R}^N$ is identified with a *column vector*, i.e. an element of $\mathbf{M}_{N\times 1}$.

A matrix $A \in \mathbf{M}_{N\times M}$ is said to be *regular*, if there exists another matrix $A^{-1} \in \mathbf{M}_{N\times M}$ (*inverse matrix*) for which $A^{-1}A = I$ is valid, where I denotes the $N \times N$ *unit matrix*. If A is regular, then the inverse A^{-1} is unique, and the equality $AA^{-1} = I$ is also valid.

For arbitrary matrices $A \in \mathbf{M}_{N\times M}$, the *adjoint* (or transpose) of A is the matrix $A^* \in \mathbf{M}_{M\times N}$ defined by

$$(A^*)_{kj} = A_{jk}$$

The matrix A is *self-adjoint* (or symmetric), if $A^* = A$. In this case, A is necessarily a square matrix.

A scalar λ (now not necessarily a real number) is said to be the *eigenvalue* of the matrix $A \in \mathbf{M}_{N\times N}$ and the corresponding *eigenvector* is a (column)

vector: $s \in \mathbf{C}^N$, if the equality

$$As = \lambda \cdot s$$

Every square matrix has (at least one) eigenvalue, and the eigenvalues satisfy the *characteristic equation*:

$$\det(A - \lambda \cdot I) = 0,$$

The eigenvalues of even the real-valued matrices are complex numbers in general. However, if the matrix is self-adjoint, then all eigenvalues are real, and the correspondig eigenvectors form an *orthogonal basis* (see later).

Some classes are specified among the self-adjoint matrices with respect to the sign of the *quadratic form* x^*Ax. A self-adjoint matrix A is called

- *positive definite*, if $x^*Ax > 0$ for every vector $x \neq \mathbf{0}$;
- *positive semidefinite*, if $x^*Ax \geq 0$ for every vector x;
- *negative definite*, if $x^*Ax < 0$ for every vector $x \neq \mathbf{0}$;
- *negative semidefinite*, if $x^*Ax \leq 0$ for every vector x;
- *indefinite*, if x^*Ax takes both positive and negative values.

The concept of definiteness is very important in e.g. the extremal value problems. It is of fundamental importance that the definiteness is completely determined by the eigenvalues:

Theorem: The self-adjoint matrix $A \in \mathbf{M}_{N\times N}$ is

- positive definite, if all eigenvalues of A are positive;
- positive semidefinite, if all eigenvalues of A are nonnegative;
- negative definite, if all eigenvalues of A are negative;
- negative semidefinite, if all eigenvalues of A are nonpositive;
- indefinite, if A has both positive and negative eigenvalues.

1.3 Norm and distance in $\mathbf{R}^N$

A frequently appearing problem to 'measure' and to properly define the size and/or the distance of objects which are more complicated than the set of numbers. A typical example: if we want to solve a linear system of equations, in practice, the calculations always contain errors (since the computers work with finite number of digits only and/or the initial data come from measurements). In such cases, it is advantageous to properly define and estimate the distance between the exact and approximate solution. The less this distance, the more accurate the approximation.

Another problem is to properly quantify the speed of convergence of sequences (of numbers, vectors, moreover, matrices). In theoretical mathematics, for example, there is a well known result: the sequence defined by

$$S_n := \sqrt{6 \cdot \left(\frac{1}{1^2} + \frac{1}{2^2} + ... + \frac{1}{n^2} \right)}$$

converges to π. This means that S_n is a 'good' approximation of the number π, provided that n is sufficiently great. However, in order to achieve only a few decimal number accuracy, n should be greater than, say, 100 by orders of magnitude (try it!). Thus, the formula is completely unsuitable to numerical calculations. The same problem occurs in every iterative method, when the exact solution is approximated by a vector sequence.

First, we would like to generalize the concept of 'size' or the 'absolute value' for a vector $x = (x_1, x_2, ..., x_N) \in \mathbf{R}^N$. It can be done in several ways. For instance:

Maximum norm:

$$||x||_{\max} := \max_{1 \le k \le N} |x_k|$$

Sum norm:

$$||x||_1 := \sum_{k=1}^{N} |x_k|$$

Euclidean norm:

$$||x||_2 := \sqrt{\sum_{k=1}^{N} |x_k|^2}$$

It can be checked easily that each of the above norm has the following

essential properties: For any vector norm:

$$||x|| \geq 0, \quad \text{and} \quad ||x|| = 0 \ \text{ if and only if } \ x = (0, 0, ..., 0)$$

$$||\alpha \cdot x|| = |\alpha| \cdot ||x|| \qquad \text{for arbitrary } \alpha \in \mathbf{R}$$

$$||x + y|| \leq ||x|| + ||y|| \qquad \text{(triangle inequality)}$$

(For the Euclidean norm, the triangle inequality is not trivial. It will be proved on the basis of the Cauchy inequality, see later.)

All the above norms generalize to concept of the absolute value. Moreover, the Euclidean norm is a straightforward gereralization of the vector length in the geometry of plane and space.

With the help of the concept of norms, the *distance of the vectors* $x, y \in \mathbf{R}^N$ can be defined as the norm of their difference: $||x - y||$ (for any vector norm). That is, the distance depends on the choice of the norm.

With the help of the concept of distance, the convergence of vector sequences can be defined in a convenient way. We say that the sequence of vectors $(x^{(n)}) \subset \mathbf{R}^N$ is convergent and tends to the vector $x = (x_1, x_2, ..., x_N) \in \mathbf{R}^N$, if the distances of $x^{(n)}$ and x tends to zero, i.e.

$$||x^{(n)} - x|| \to 0$$

in the well known sense of the elementary analysis. Though the distance depends on the choice of the norm in $\mathbf{R}^N$, the convergence does not. It can be shown that $x^{(n)} \to x$ is valid if and only if each sequence of vector component converges to the corresponding component of the limit vector, i.e.

$$x^{(n)} \to x \qquad \Leftrightarrow \qquad x_k^{(n)} \to x_k \quad (k = 1, 2, ..., N)$$

1.4 Inner product in $\mathbf{R}^N$

The *scalar product* (or *inner product*) of the vectors $x := (x_1, x_2, ..., x_N)$, $y := (y_1, y_2, ..., y_N) \in \mathbf{R}^N$ is defined by

$$\langle x, y \rangle := \sum_{j=1}^{N} x_j y_j = x_1 y_1 + ... + x_N y_N.$$

By definition, the inner product induces the Euclidean norm, i.e. $||x|| = \sqrt{\langle x, x \rangle}$. The following statements facilitate the calculations with the inner product and can be proved in an elementary way:

Proposition: For arbitrary vectors $x, y, z \in \mathbf{R}^N$ and scalar $\alpha \in \mathbf{R}$:

- $\langle x, y\rangle = \langle y, x\rangle$
- $\langle \alpha x, y\rangle = \alpha \cdot \langle x, y\rangle$
- $\langle x, y + z\rangle = \langle x, y\rangle + \langle x, z\rangle$

which immediately implies that: $\langle x, \alpha y\rangle = \alpha \cdot \langle x, y\rangle$ and $\langle x + y, z\rangle = \langle x, z\rangle + \langle y, z\rangle$ are also valid.

The following simple equalities are quite useful in the calculations:

$$||x + y||^2 = ||x||^2 + 2\langle x, y\rangle + ||y||^2$$

and

$$||x - y||^2 = ||x||^2 - 2\langle x, y\rangle + ||y||^2$$

The next equality shows the connection between the inner product and the adjoint matrix. Let $A \in \mathbf{M}_{N\times M}$ is an arbitrary matrix. Then for every vectors $x \in \mathbf{R}^M$, $y \in \mathbf{R}^N$:

$$\langle Ax, y\rangle = \langle x, A^* y\rangle,$$

where the inner product in the left-hand side (right-hand side, respectively) is meant in the space $\mathbf{R}^N$ ($\mathbf{R}^M$, respectively).

The following theorem is of special importance:

Theorem (Cauchy inequality): For arbitrary vectors $x, y \in \mathbf{R}^N$:

$$|\langle x, y\rangle| \le ||x|| \cdot ||y||,$$

and the equality is valid if and only if x and y are linearly dependent, i.e. one of them is equal to the other multiplied by a scalar constant.

Proof: For arbitrary scalar $\alpha \in \mathbf{R}$, obviously $||x - \alpha y||^2 \ge 0$, i.e.:

$$\sum_{j=1}^{N}(x_j - \alpha y_j)^2 = \sum_{j=1}^{N}\left(x_j^2 - 2\alpha x_j y_j + \alpha^2 y_j^2\right) =$$

$$= ||x||^2 - 2\alpha\langle x, y\rangle + \alpha^2||y||^2 \ge 0$$

Now define α by $\alpha := \frac{||x||}{||y||}$ (provided that $y \ne \mathbf{0}$; otherwise, the statement is simplified to the trivial equality $0 = 0$). We have:

$$||x||^2 - 2\frac{||x||}{||y||}\langle x, y\rangle + \frac{||x||^2}{||y||^2} \cdot ||y||^2 \ge 0.$$

This can be simplified to the inequality

$$\langle x, y\rangle \leq ||x|| \cdot ||y||$$

This is valid for arbitrary vectors $x, y \in \mathbf{R}^N$. If y is substituted by $(-y)$, the inequality remains valid:

$$\langle x, -y\rangle \leq ||x|| \cdot || - y|| = ||x|| \cdot ||y||,$$

whence $\langle x, y\rangle \geq -||x|| \cdot ||y||$. We have obtained that

$$-||x|| \cdot ||y|| \leq \langle x, y\rangle \leq ||x|| \cdot ||y||,$$

i.e. $|\langle x, y\rangle| \leq ||x|| \cdot ||y||$. Equality is valid only if $||x - \alpha y||^2 = 0$, i.e. $x = \alpha y$.

Remark: From the Cauchy inequality, the triangle inequality simply follows:

$$||x+y||^2 = ||x||^2 + 2\langle x, y\rangle + ||y||^2 \leq ||x||^2 + 2||x|| \cdot ||y|| + ||y||^2 = (||x|| + ||y||)^2.$$

Taking the square root of both sides, we have the triangle inequality.

Remark: If $N = 2$ of 3, the inner product has a geometrical meaning through the well known relation:

$$\langle x, y\rangle = ||x|| \cdot ||y|| \cdot \cos\phi,$$

where ϕ denotes the angle of the vectors x and y.

1.5 Orthogonality in $\mathbf{R}^N$

The vectors $x, y \in \mathbf{R}^N$ are said to be *orthogonal*, if their inner product is zero: $\langle x, y\rangle = 0$. A set of vectors is called *orthogonal system*, if the vectors belonging to this system are pairwise orthogonal, i.e. two different vectors of the system are orthogonal. An orthogonal system is called *orthonormal system*, if the Euclidean norm of each vector belonging to this system is equal to 1. The simplest example for orthonormal systems in $\mathbf{R}^N$ is the system of vectors: $(1, 0, 0, ..., 0)$, $(0, 1, 0, ..., 0)$, $(0, 0, 1, ..., 0)$, ... $(0, 0, 0, ..., 1)$ (the *standard basis* of $\mathbf{R}^N$).

The orthogonality is a generalization of the concept of perpendicular vectors. Orthogonal vectors exhibit a lot of interesting properties. For instance, using the equality showed earlier $||x+y||^2 = ||x||^2 + 2\langle x, y\rangle + ||y||^2$ several times, we obtain the generalized theorem of Pythagoras:

Theorem: Let $x^{(1)}, x^{(2)}, ..., x^{(m)} \in \mathbf{R}^N$ be pairwise orthogonal vectors, then:

$$||\sum_{j=1}^{m} x^{(j)}||_2^2 = \sum_{j=1}^{m} ||x^{(j)}||_2^2.$$

It should be pointed out that any vector can be expressed in an orthonormal basis in an extremely simple way. If $e_1, e_2, ..., e_N$ is an orthonormal basis in $\mathbf{R}^N$, then, for arbitrary vector $x \in \mathbf{R}^N$ can be expressed in the following form:

$$x = \sum_{j=1}^{N} \langle x, e_j \rangle \cdot e_j$$

Note that, if the system $e_1, e_2, ..., e_N$ is not orthonormal, then the coefficients of the linear combination can be determined by solving a linear system of equations for the coefficients.

Orthogonal projection: An immediate consequence of the previous statement is that $X_0 \subset \mathbf{R}^N$ is an M-dimensional subspace spanned by the orthonormal system $e_1, e_2, ..., e_M$ then the vector

$$x_0 := \sum_{j=1}^{M} \langle x, e_j \rangle \cdot e_j \in X_0$$

is the *orthogonal projection* of the vector x to the subspace X_0, i.e. the difference $(x - x_0)$ is orthogonal to X_0. Note that this is the closest vector to x from the subspace X_0. In the one-dimensional special case, when X_0 is spanned by a single vector $e \in \mathbf{R}^N$, then $x_0 = \langle x, e \rangle \cdot e$.

An important technique of several numerical methods that one can construct orthonormal systems from any set of linearly independent vectors. This is the so called *Gram-Schmidt orthogonalization process*. Let $a_1, a_2, ..., a_m$ be linearly independent vectors of $\mathbf{R}^N$, $(m \leq N)$ and define:

$$\tilde{e}_1 := a_1, \quad e_1 := \frac{\tilde{e}_1}{||\tilde{e}_1||},$$

and for $k = 2, 3, ..., m$:

$$\tilde{e}_k := a_k - \sum_{j=1}^{k-1} \langle a_k, e_j \rangle \cdot e_j, \quad e_k := \frac{\tilde{e}_k}{||\tilde{e}_k||}$$

Then, as can be easily checked, e_1, e_2, ..., e_m form an orthonormal system in $\mathbf{R}^N$. Indeed, by definition, $||e_k|| = 1$ and for every $i < k$, the vectors $\tilde{e}_k$ and e_i are orthogonal, which is obvious by calculating the inner product of $\tilde{e}_k$ and e_i.

Orthogonal matrices: A matrix $A \in \mathbf{M}_{N\times N}$ is called *orthogonal*, if its column vectors form an orthonormal system in the vector space $\mathbf{R}^N$.

The most important property of orthogonal matrices is that an orthogonal matrix $A \in \mathbf{M}_{N\times N}$ is always regular, moreover, its inverse is equal to its adjoint: $A^{-1} = A^*$. In addition to it, an orthogonal matrix preserves the norm in the sense that for every column vector $x \in \mathbf{R}^N$, the equality $||Ax|| = ||x||$ holds, where $||.||$ denotes the Euclidean norm. These properties are equivalent to the orthogonality.

Normal matrices: A matrix $A \in \mathbf{M}_{N\times N}$ is said to be a *normal matrix*, if $AA^* = A^*A$. (Obviously, every self-adjoint matrix is normal.)

The most important property of the normal matrices is that if $A \in \mathbf{M}_{N\times N}$ is a normal matrix, then A has a system of eigenvectors which form an orthonormal system. In this basis, A has a diagonal form.

1.6 Norms of matrices

One might think that the proper definition of a matrix norm has been done, since $\mathbf{M}_{N\times M}$ is an $N \cdot M$-dimensional vector space, thus, the matrix norms can be defined similarly to the vector norm. That is, the following matrix norms can be defined without any difficulty:

Maximum norm:

$$||A|| := \max\{|A_{kj}| : \ 1 \le k \le N, \ 1 \le j \le M\}$$

Sum norm:

$$||A|| := \sum_{k=1}^{N}\sum_{j=1}^{M} |A_{kj}|$$

Euclidean norm or *Frobenius norm*:

$$||A|| := \sqrt{\sum_{k=1}^{N}\sum_{j=1}^{M}|A_{kj}|^2}$$

However, these norms (which are similar to the vector norms) are rarely used. The use of the *matrix norm induced by a vector norm* is much more important. The reason is that matrices can be considered also *linear mapping* from $\mathbf{R}^M$ into $\mathbf{R}^N$ by the definition $A(x) := Ax$ (where $x \in \mathbf{R}^M$ is a column vector). Thus an $N \times M$ matrix can be considered a table of numbers having N rows and M columns, and, at the same time, it can be regarded as an $\mathbf{R}^M \to \mathbf{R}^N$ linear mapping. From the point of view of a huge majority of computational methods, the latter interpretation is more important.

The operator norms can be defined for linear operators (mappings) in a quite general way. However, for our purposes, the following definition of matrix norms will be sufficient:

$$||A|| := \max\{||Ax|| : \quad x \in \mathbf{R}^M,\ ||x|| \le 1\}$$

where $||x||$ is a vector norm of x in the space $\mathbf{R}^M$, and similarly, $||Ax||$ is a (possibly different) vector norm of Ax in the space $\mathbf{R}^N$. That is, the matrix norm depends on the choices of the vector norms, therefore it is often referred to as *matrix norm induced by vector norm*.

A crucial theorem concerning the above matrix norm is as follows:

Theorem: Let $A \in \mathbf{M}_{N\times M}$ be an arbitrary matrix. If $C \ge 0$ is a constant such that $||Ax|| \le C \cdot ||x||$ is valid for every $x \in \mathbf{R}^M$, then $||A|| \le C$. Moreover, $||A||$ equals to the least constant C which has this property.

Proof: Let $C \ge 0$ be a constant such that $||Ax|| \le C \cdot ||x||$ holds for every $x \in \mathbf{R}^M$. Then $||Ax|| \le C$ for the vectors x with length of 1. The $\le$ relation remains valid also for their maximum, i.e. $||A|| \le C$. Moreover, the number $||A||$ has also this property, since, by definition: $||A|| \ge ||Ax||$ for arbitrary vectors x with $||x|| \le 1$; therefore $||A|| \ge ||A\frac{x}{||x||}||$, whence $||A|| \cdot ||x|| \ge ||Ax||$.

The induced matrix norm has similar properties to the vector norms.

$$||A|| \ge 0, \quad \text{and} \ \ ||A|| = 0 \ \ \text{if and only if} \ \ A = 0$$

$$||\alpha \cdot A|| = |\alpha| \cdot ||A|| \qquad \text{for arbitrary } \alpha \in \mathbf{R}$$
$$||A + B|| \leq ||A|| + ||B|| \qquad \text{(triangle inequality)}$$

The triangle inequality is a simple consequence of the triangle inequality of vector norm, since for every vector x:

$$||(A + B)x|| = ||Ax + Bx|| \leq ||Ax|| + ||Bx|| \leq (||A|| + ||B||) \cdot ||x||,$$

therefore as a consequence of the previous theorem, $||A+B|| \leq ||A|| + ||B||$.

For the matrix multiplication, a similar inequality holds. If $A \in \mathbf{M}_{N\times M}$ and $B \in \mathbf{M}_{M\times K}$, then:

$$||A \cdot B|| \leq ||A|| \cdot ||B||$$

Indeed, for arbitrary $x \in \mathbf{R}^K$:

$$||ABx|| = ||A(Bx)|| \leq ||A|| \cdot ||Bx|| \leq ||A|| \cdot ||B|| \cdot ||x||,$$

which implies that $||A \cdot B|| \leq ||A|| \cdot ||B||$.

Now let us briefly outline the most frequently used special cases for induced matrix norms. In each case we assume that both in $\mathbf{R}^N$ and in $\mathbf{R}^M$, the same type vector norms are introduced. Let $A \in \mathbf{M}_{N\times M}$ be arbitrary. Then the corresponding induced matrix norms are as follows (without proofs):

Row norm: Suppose that both in $\mathbf{R}^N$ and in $\mathbf{R}^M$, the maximum norm is defined. Then

$$||A|| = \max_{1\leq k\leq N} \sum_{j=1}^{M} |A_{kj}|$$

Column norm: Suppose that both in $\mathbf{R}^N$ and in $\mathbf{R}^M$, the sum norm is defined. Then

$$||A|| = \max_{1\leq j\leq M} \sum_{k=1}^{N} |A_{kj}|$$

Matrix norm induced by the Euclidean vector norm: Suppose that in $\mathbf{R}^N$, the Euclidean norm is given. Then for every matrix $A \in \mathbf{M}_{N\times N}$:

$$||A|| = \max_{1\leq k\leq N} \sqrt{\lambda_k},$$

where λ_1,...,λ_N are the eigenvalues of the (self-adjoint, positive semidefinite) matrix A^*A.

As a consequence, if A is self-adjoint:

$$||A|| = \max_{1 \le k \le N} |\lambda_k|,$$

where now λ_1,...,λ_N are the eigenvalues of the matrix A. In addition to it, if A is positive definite, then

$$||A|| = \max_{1 \le k \le N} \lambda_k, \quad \text{and} \quad ||A^{-1}|| = \frac{1}{\min\limits_{1 \le k \le N} \lambda_k}.$$

In general, for not necessarily self-adjoint (but square) matrices, denote by

$$\rho(A) := \max_{1 \le k \le N} |\lambda_k|.$$

The number $\rho(A)$ is called the *spectral radius* of the matrix A. The last statement can be reformulated as follows: for any self-adjoint matrix, the norm of the matrix induced by the Euclidean vector norm is the spectral radius. For non-selfadjoint matrices, this does not remain the case. What we can state is that the spectral radius is *not greater* than any of induced matrix norm:

$$\rho(A) \le ||A||,$$

since if λ is an eigenvalue of A with eigenvector s, then:

$$|\lambda| \cdot ||s|| = ||As|| \le ||A|| \cdot ||s||,$$

which implies the statement.

1.7 Linear systems of equations

Now let us summarize some concepts and theorems concerning the linear systems of equations.

Let $A \in \mathbf{M}_{N \times N}$ be a given square matrix with the entries $a_{k,j}$ $(k, j = 1, 2, ..., N))$, and let $b \in \mathbf{R}^N$ be a given vector with elements b_1, b_2,...,b_N). Consider the linear system of equations

$$Ax = b \tag{1}$$

where $x \in \mathbf{R}^N$ denotes the solution vector. This compact notation is equivalent to the traditional one:

$$a_{11}x_1 + a_{12}x_2 + ... + a_{1N}x_N = b_1$$

$$a_{21}x_1 + a_{22}x_2 + ... + a_{2N}x_N = b_2$$

.....

$$a_{N1}x_1 + a_{N2}x_2 + ... + a_{NN}x_N = b_N$$

The system is called *homogeneous*, if $b = \mathbf{0}$, i.e. $b_1 = b_2 = ... = b_N = 0$. In this case, the zero vector $x_1 = x_2 = ... = x_N = 0$ always solves the system (*trivial solution*). Any other solution of the homogeneous system is called *nontrivial solution*.

In case of homogeneous equations, the characteristic problem is whether or not a nontrivial solution exists, while in case of nonhomogeneous equations, the problem is whether of not an arbitrary solution exists (and how many).

Recall the theorems which assure the solvability of the system of equations (1):

Theorem: The matrix $A \in \mathbf{M}_{N\times N}$ is regular if and only if the system $Ax = b$ has a unique solution for every right-hand side $b \in \mathbf{R}^N$. In this case, the solution is unique and can be expressed as $x = A^{-1}b$.

Theorem: The matrix $A \in \mathbf{M}_{N\times N}$ is regular if and only if the homogeneous system $Ax = \mathbf{0}$ has nontrivial solution. In this case, the homogeneous system has infinitely many nontrivial solutions.

1.7.1 Perturbed linear systems

Consider the system of linear equations $Ax = b$ and the perturbed system as well:

$$A(x + \Delta x) = b + \Delta b$$

Here the term Δb is interpreted as a perturbation of the right-hand side and Δx is the error of the solution caused by the perturbation. The goal is to characterize the effect of the perturbation.

From the perturbed equation: $Ax + A\Delta x = b + \Delta b$. Taking into account the original equation, we have: $A\Delta x = \Delta b$, i.e. $\Delta x = A^{-1}\Delta b$. This implies that:

$$||\Delta x|| \leq ||A^{-1}|| \cdot ||\Delta b||$$

On the other hand: $||b|| = ||Ax|| \leq ||A|| \cdot ||x||$, therefore

$$\frac{1}{||x||} \cdot \frac{1}{||A||} \leq \frac{1}{||b||}$$

From the last two inequalities:

$$\frac{||\Delta x||}{||x||} \leq ||A|| \cdot ||A^{-1}|| \cdot \frac{||\Delta b||}{||b||}$$

Here the quotient $\frac{||\Delta x||}{||x||}$ is the *relative error* caused by the relative perturbation $\frac{||\Delta b||}{||b||}$. Introducing the *condition number* by $\text{cond}(A) := ||A|| \cdot ||A^{-1}||$, we arrive at the *perturbation lemma*:

$$\frac{||\Delta x||}{||x||} \leq \text{cond}(A) \cdot \frac{||\Delta b||}{||b||}$$

Note that the condition number characterizes how sensitive the solution of the system is to the changes in the right-hand side. If the condition number is under the order of magnitude of 10, the matrix is considered well-conditioned. If it exceeds the order of magnitude of 10^6, the matrix is regarded as an ill-conditioned matrix. In the example of the subsection 'Motivations', the condition number is approximately $4 \cdot 10^6$, thus, the system is really ill-conditioned.

If the matrix A is self-adjoint and positive definite, then:

$$\text{cond}(A) = \frac{\lambda_{\max}}{\lambda_{\min}}$$

2 Direct solution of linear systems

2.1 The Gaussian elimination

Now we outline one of the most important direct method for solving linear systems of equations, the Gaussian elimination.

Consider the following linear system of equations:

$$\begin{gathered}
A_{11}x_1 + A_{12}x_2 + A_{13}x_3 + ... + A_{1N}x_N = b_1 \\
A_{21}x_1 + A_{22}x_2 + A_{23}x_3 + ... + A_{2N}x_N = b_2 \\
A_{31}x_1 + A_{32}x_2 + A_{33}x_3 + ... + A_{3N}x_N = b_3 \\
..... \\
A_{N1}x_1 + A_{N2}x_2 + A_{N3}x_3 + ... + A_{NN}x_N = b_N
\end{gathered}$$

where we suppose that the matrix A is regular. The steps of the Gaussian elimination are as follows:

Step 1. Divide the first equation by the coefficient A_{11}:

$$\begin{gathered}
x_1 + A'_{12}x_2 + A'_{13}x_3 + ... + A'_{1N}x_N = b'_1 \\
A_{21}x_1 + A_{22}x_2 + A_{23}x_3 + ... + A_{2N}x_N = b_2 \\
A_{31}x_1 + A_{32}x_2 + A_{33}x_3 + ... + A_{3N}x_N = b_3 \\
..... \\
A_{N1}x_1 + A_{N2}x_2 + A_{N3}x_3 + ... + A_{NN}x_N = b_N
\end{gathered}$$

Step 2. The first equation multiplied by A_{k1} now should be subtracted from the kth equation ($k = 2, 3, ..., N$). Thus, the unknown x_1 has been eliminated from the 2nd, 3rd, ... Nth equations, which results in the following structure of equations:

$$\begin{gathered}
x_1 + A'_{12}x_2 + A'_{13}x_3 + ... + A'_{1N}x_N = b'_1 \\
A'_{22}x_2 + A'_{23}x_3 + ... + A'_{2N}x_N = b'_2 \\
A'_{32}x_2 + A'_{33}x_3 + ... + A'_{3N}x_N = b'_3 \\
..... \\
A'_{N2}x_2 + A'_{N3}x_3 + ... + A'_{NN}x_N = b'_N
\end{gathered}$$

Step 3. The 2nd, 3rd,..., Nth equations have $(N-1)$ unknowns only: therefore we repeat the above steps: then x_2 has been eliminated from the 3rd, 4th, ..., Nth equations. Similarly, now x_3 is to be eliminated from the 4th, 5th, ..., Nthn equations and so on. Finally, we obtain a system with the

following form:

$$\begin{aligned} x_1 + \tilde{A}_{12}x_2 + \tilde{A}_{13}x_3 + ... + \tilde{A}_{1N}x_N &= \tilde{b}_1 \\ x_2 + \tilde{A}_{23}x_3 + ... + \tilde{A}_{2N}x_N &= \tilde{b}_2 \\ x_3 + ... + \tilde{A}_{3N}x_N &= \tilde{b}_3 \\ ..... & \\ x_N &= \tilde{b}_N \end{aligned}$$

Step 4. From the last equation, x_N in now known. Substituting x_N back into the $(N-1)$th equation, x_{N-1} can now be calculated. Substituting x_N and x_{N-1} back into the $(N-2)$th equation, x_{N-2} can be calculated, and so on. Thus, the unknowns are calculated in reverse order. This steps are called the *back substitution part* of the Gaussian elimination, while the previous steps are called the *elimination part*.

The algorithm is illustrated through the following simple example:

Example: Solve the following system of equations:

$$\begin{array}{rcrcrcr} 2x_1 & - & 6x_2 & + & 10x_3 & = & -12 \\ 2x_1 & - & 5x_2 & + & 3x_3 & = & -4 \\ 3x_1 & - & 2x_2 & + & x_3 & = & 3 \end{array}$$

Solution: Divide the first equation by 2:

$$\begin{array}{rcrcrcr} x_1 & - & 3x_2 & + & 5x_3 & = & -6 \\ 2x_1 & - & 5x_2 & + & 3x_3 & = & -4 \\ 3x_1 & - & 2x_2 & + & x_3 & = & 3 \end{array}$$

Subtract the double of the first equation from the second equation, and then substitute the triple of the first equation from the third equation. Thus, the unknown x_1 has been eliminated from the second and third equations:

$$\begin{array}{rcrcrcr} x_1 & - & 3x_2 & + & 5x_3 & = & -6 \\ & & x_2 & - & 7x_3 & = & 8 \\ & & 7x_2 & - & 14x_3 & = & 21 \end{array}$$

The second equation need not be divided by the coefficient of x_2, since this coefficient equals to 1. Subtract the septuple of the second equation from the third equation, thus, x_3 has been eliminated from the third equation. We have:

$$\begin{array}{rcrcrcr} x_1 & - & 3x_2 & + & 5x_3 & = & -6 \\ & & x_2 & - & 7x_3 & = & 8 \\ & & & & 35x_3 & = & -35 \end{array}$$

Dividing the third equation by 35, the elimination part is finished:

$$\begin{array}{rcrcrcr} x_1 & - & 3x_2 & + & 5x_3 & = & -6 \\ & & x_2 & - & 7x_3 & = & 8 \\ & & & & x_3 & = & -1 \end{array}$$

From the last equation, x_3 has been calculated. Substituting x_3 back into the second equation, x_2 can also be calculated:

$$\begin{array}{rcrcrcr} x_1 & - & 3x_2 & + & 5x_3 & = & -6 \\ & & x_2 & & & = & 1 \\ & & & & x_3 & = & -1 \end{array}$$

Finally, substituting x_2 and x_3 back into the first equation, x_1 can be calculated as well:

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

We have obtained the solution of the system of equations. By substitution, we can easily check that this solution satisfies the original system.

Observe that in order to perform the computations, it is unnecessary to write down the symbols x_1, x_2, x_3 and the equality signs again and again; the calculations are performed on the coefficients only. The above steps of calculation can be rewritten in a more compact form (the vertical line before the last column is for a better transparency only):

$$\left(\begin{array}{ccc|c} 2 & -6 & 10 & -12 \\ 2 & -5 & 3 & -4 \\ 3 & -2 & 1 & 3 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & -3 & 5 & -6 \\ 2 & -5 & 3 & -4 \\ 3 & -2 & 1 & 3 \end{array}\right) \to$$

$$\to \left(\begin{array}{ccc|c} 1 & -3 & 5 & -6 \\ 0 & 1 & -7 & 8 \\ 0 & 7 & -14 & 21 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & -3 & 5 & -6 \\ 0 & 1 & -7 & 8 \\ 0 & 0 & 35 & -35 \end{array}\right) \to$$

$$\to \left(\begin{array}{ccc|c} 1 & -3 & 5 & -6 \\ 0 & 1 & -7 & 8 \\ 0 & 0 & 1 & -1 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & -3 & 5 & -6 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{array}\right) \to$$

$$\to \left(\begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{array}\right)$$

It can be shown that the necessary number of arithmetic operations of the Gaussian elimination is $\mathcal{O}(N^3)$, which shows the from a computational

point of view, the Gaussian elimination is not 'cheap'; if the number of unknowns is doubled, then the number of arithmetic operation is increased by a factor of 8.

The algorithm cannot be performed in the above form; a coefficient, by which an equation should be divided, might be equal to zero. Suppose for instance that $A_{11} = 0$. Now swap the first equation with one of the other equations; it can then be assured that the coefficient of the first unknown x_1 differ from 0, otherwise, the first column of the matrix would consist of zeros, which contradicts the regularity of the matrix. This remains the case also for the further (less and less) systems appearing in the elimination part. From the point of view of accuracy, it would be preferable to keep these divisors (the *pivot elements*) as large as possible (meant in absolute value). Thus, when swapping the kth equation with the rth one, r should be chosen in such a way that $|A_{rk}|$ is maximal for $r = k, k+1, ..., N$. (This is recommended even in the case when $A_{kk} \neq 0$.) This strategy is known as *partial pivoting*, and this works for every regular matrix. A bit more expensive but more accurate strategy is the *full pivoting*, when the maximum of values of $|A_{pq}|$ is to be determined ($p, q = k, k+1, ..., N$). However, not only two equations are swapped, but also the ordering of the unknowns are changed.

A version of the Gaussian elimination is the *Gauss-Jordan* elimination, when with the help of the actual, say, kth equation, the unknown x_k is eliminated not only from the latter equations but also from the previous ones. (Note that, despite its simplicity, the computational cost is higher compared with the Gaussian elimination.)

Example: Consider the system of the previous example. The first two steps of then algorithm are identical to that of the Gaussian elmination; a discrepancy appears from the third step:

$$\left(\begin{array}{ccc|c} 2 & -6 & 10 & -12 \\ 2 & -5 & 3 & -4 \\ 3 & -2 & 1 & 3 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & -3 & 5 & -6 \\ 2 & -5 & 3 & -4 \\ 3 & -2 & 1 & 3 \end{array}\right) \to$$

$$\to \left(\begin{array}{ccc|c} 1 & -3 & 5 & -6 \\ 0 & 1 & -7 & 8 \\ 0 & 7 & -14 & 21 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & 0 & -16 & 18 \\ 0 & 1 & -7 & 8 \\ 0 & 0 & 35 & -35 \end{array}\right) \to$$

$$\to \left(\begin{array}{ccc|c} 1 & 0 & -16 & 18 \\ 0 & 1 & -7 & 8 \\ 0 & 0 & 1 & -1 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{array}\right)$$

The Gaussian elimination is suitable for solving systems with singular matrices as well. In this case, in some step of elimination, all the coefficients of a certain equation become zero. If the corresponding right-hand side differs from zero, then the system contains a contradiction, therefore it has no solution. However, if the corresponding right-hand side is zero, then there is a solution, moreover, there are infinitely many solutions. In this case, one of the unknowns (possibly several ones) can be chosen in an arbitrary way, and the other ones can be expressed as functions of the arbitrarily chosen unknowns.

The above situation is illustrated through an example:

Example: Solve the following system of equations:

$$\begin{array}{rcrcrcl} x_1 & - & 2x_2 & + & x_3 & = & 1 \\ -2x_1 & + & x_2 & + & x_3 & = & 4 \\ x_1 & + & x_2 & - & 2x_3 & = & 1 \end{array}$$

Solution: Using the above compact notations for the steps of the Gaussian elimination:

$$\left(\begin{array}{rrr|r} 1 & -2 & 1 & 1 \\ -2 & 1 & 1 & 4 \\ 1 & 1 & -2 & 1 \end{array}\right) \to \left(\begin{array}{rrr|r} 1 & -2 & 1 & 1 \\ 0 & -3 & 3 & 6 \\ 0 & 3 & -3 & 0 \end{array}\right) \to$$

$$\to \left(\begin{array}{rrr|r} 1 & -2 & 1 & 1 \\ 0 & 1 & -1 & -2 \\ 0 & 3 & -3 & 0 \end{array}\right) \to \left(\begin{array}{rrr|r} 1 & -2 & 1 & 1 \\ 0 & 1 & -1 & -2 \\ 0 & 0 & 0 & 6 \end{array}\right)$$

All the coefficients of the last equation are equal to zero, but the right-hand side differs from zero. This is a contradiction, so that the system has no solution.

Instead of this, consider the corresponding homogeneous system:

$$\begin{array}{rcrcrcl} x_1 & - & 2x_2 & + & x_3 & = & 0 \\ -2x_1 & + & x_2 & + & x_3 & = & 0 \\ x_1 & + & x_2 & - & 2x_3 & = & 0 \end{array}$$

Now we know that there exist nontrivial solutions, since the matrix is singular (check it!). Let us see, how the Gaussian elimination works in this case:

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

$$\rightarrow \left(\begin{array}{ccc|c} 1 & -2 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 3 & -3 & 0 \end{array}\right) \rightarrow \left(\begin{array}{ccc|c} 1 & -2 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right)$$

The last equation has been simplified to the informationless equality $0 = 0$. An unknown (the last one can be recommended) can be defined arbitrarily: $x_3 := t$, where $t \in \mathbf{R}$ is arbitrary. Writing it back in the third equation, the back-substitutions can be performed without difficulty:

$$\left(\begin{array}{ccc|c} 1 & -2 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & t \end{array}\right) \rightarrow \left(\begin{array}{ccc|c} 1 & -2 & 0 & -t \\ 0 & 1 & 0 & t \\ 0 & 0 & 1 & t \end{array}\right) \rightarrow$$

$$\rightarrow \left(\begin{array}{ccc|c} 1 & 0 & 0 & t \\ 0 & 1 & 0 & t \\ 0 & 0 & 1 & t \end{array}\right)$$

We see that there are infinitely many solutions. The general form of them is as follows: $x_1 = t$, $x_2 = t$, $x_3 = t$.

Finally, we show how the Gaussian elimination can be applied to matrix inversion. Let $A \in \mathbf{M}_{N\times N}$ be a regular matrix. Then the matrix equality

$$AA^{-1} = I$$

is valid. Denote by a_1, a_2, ... , a_N the a priori unknown column vectors of the inverse matrix A^{-1}, an let e_1, e_2, ... ,e_N be the column vectors of the unit matrix I (which are actually the element of the standard basis in $\mathbf{R}^N$):

$$A \cdot \left(\begin{array}{c|c|c|c} & & & \\ a_1 & a_2 & ... & a_N \\ & & & \end{array}\right) = \left(\begin{array}{c|c|c|c} & & & \\ e_1 & e_2 & ... & e_N \\ & & & \end{array}\right)$$

By the definition of the multiplication of matrices, this matrix equality is split into N vectorial equalities, namely:

$$Aa_k = e_k \qquad (k = 1, 2, ..., N)$$

Having solved these systems of equations, the solutions as column vectors can be composed to a matrix, which is the inverse of the original matrix. This means that for an inversion, N systems of equations have to be solved. However, the systems has a common matrix, A. This can be performed by a single elimination algorithm with several right-hand sides. The algorithm

is very expressive in the above used compact notations. In the beginning of the algorithm, the left submatrix is the original matrix to be inverted, and the right one is the unit matrix. In the end, the left submatrix becomes the unit matrix and the right one becomes the inverse matrix.

Example: Calculate the inverse of the following matrix:

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

Solution: The steps of the algorithm, using the compact notations:

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

$$\rightarrow \left(\begin{array}{ccc|ccc} 1 & 2/3 & 0 & -1/3 & 0 & 0 \\ 0 & 3 & 2 & 0 & 1 & 0 \\ -2 & 0 & 1 & 0 & 0 & 1 \end{array}\right) \rightarrow$$

$$\rightarrow \left(\begin{array}{ccc|ccc} 1 & 2/3 & 0 & -1/3 & 0 & 0 \\ 0 & 3 & 2 & 0 & 1 & 0 \\ 0 & 4/3 & 1 & -2/3 & 0 & 1 \end{array}\right) \rightarrow$$

$$\rightarrow \left(\begin{array}{ccc|ccc} 1 & 2/3 & 0 & -1/3 & 0 & 0 \\ 0 & 1 & 2/3 & 0 & 1/3 & 0 \\ 0 & 4/3 & 1 & -2/3 & 0 & 1 \end{array}\right) \rightarrow$$

$$\rightarrow \left(\begin{array}{ccc|ccc} 1 & 2/3 & 0 & -1/3 & 0 & 0 \\ 0 & 1 & 2/3 & 0 & 1/3 & 0 \\ 0 & 0 & 1/9 & -2/3 & -4/9 & 1 \end{array}\right) \rightarrow$$

$$\rightarrow \left(\begin{array}{ccc|ccc} 1 & 2/3 & 0 & -1/3 & 0 & 0 \\ 0 & 1 & 2/3 & 0 & 1/3 & 0 \\ 0 & 0 & 1 & -6 & -4 & 9 \end{array}\right) \rightarrow$$

$$\rightarrow \left(\begin{array}{ccc|ccc} 1 & 2/3 & 0 & -1/3 & 0 & 0 \\ 0 & 1 & 0 & 4 & 3 & -6 \\ 0 & 0 & 1 & -6 & -4 & 9 \end{array}\right) \rightarrow$$

$$\rightarrow \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & -3 & -2 & 4 \\ 0 & 1 & 0 & 4 & 3 & -6 \\ 0 & 0 & 1 & -6 & -4 & 9 \end{array}\right)$$

The inverse matrix:

$$A^{-1} = \begin{pmatrix} -3 & -2 & 4 \\ 4 & 3 & -6 \\ -6 & -4 & 9 \end{pmatrix}$$

which can easily be checked by performing the multiplication $A^{-1}A$.

2.2 Solution of three-diagonal systems of equations by recursion

In some applications the systems of equations, the matrix of which has a three-diagonal form, have special importance. Assume that the matrix has the form:

$$A := \begin{pmatrix} B_1 & C_1 & 0 & 0 & ... & 0 \\ A_2 & B_2 & C_2 & 0 & ... & 0 \\ 0 & A_3 & B_3 & C_3 & ... & 0 \\ ... & ... & ... & ... & ... & ... \\ 0 & ... & 0 & A_{N-1} & B_{N-1} & C_{N-1} \\ 0 & ... & 0 & 0 & A_N & B_N \end{pmatrix},$$

and consider the system of equations $Ax = b$.

Try to find the solution in the form:

$$x_k = m_{k+1}x_{k+1} + n_{k+1}$$

('backward' recursion). Then

$$x_{k-1} = m_k x_k + n_k = m_k \cdot (m_{k+1}x_{k+1} + n_{k+1}) + n_k =$$

$$= m_k m_{k+1} x_{k+1} + (m_k n_{k+1} + n_k)$$

Substituting into the kth equation ($k = 2, 3, ..., N-1$):

$$A_k x_{k-1} + B_k x_k + C_k x_{k+1} =$$

$$= A_k[m_k m_{k+1} x_{k+1} + (m_k n_{k+1} + n_k)] + B_k[m_{k+1}x_{k+1} + n_{k+1}] + C_k x_{k+1} =$$

$$= (A_k m_k m_{k+1} + B_k m_{k+1} + C_k)x_{k+1} + (A_k m_k n_{k+1} + A_k n_k + B_k n_{k+1}) = b_k$$

The equality is obviously valid, if

$$A_k m_k m_{k+1} + B_k m_{k+1} + C_k = 0$$

and

$$A_k m_k n_{k+1} + A_k n_k + B_k n_{k+1} = b_k$$

that is, if the numbers m_k, n_k satisfy the 'forward' recursions:

$$m_{k+1} = -\frac{C_k}{A_k m_k + B_k}, \qquad n_{k+1} = \frac{b_k - A_k n_k}{A_k m_k + B_k}$$

Define $m_1 := 0$, $n_1 := 0$, then $m_2 = -\frac{C_1}{B_1}$, $n_2 = \frac{b_1}{B_1}$, and $x_1 = m_2 x_2 + n_2 = -\frac{C_1}{B_1} x_2 + \frac{b_1}{B_1}$, whence the first equation is also satisfied, since

$$B_1 x_1 + C_1 x_2 = -C_1 x_2 + b_1 + C_1 x_2 = b_1.$$

In the backward recursion, define: $x_N := n_{N+1}$, then $x_{N-1} = m_N x_N + n_N$. Thus, the last equation is also satisfied, since:

$$A_n x_{N-1} + B_N x_N = A_N \cdot (m_N x_N + n_N) + B_N x_N = (A_N m_N + B_N) \cdot x_N + A_N n_N =$$

$$= (A_N m_N + B_N) \cdot n_{N+1} + A_N n_N = b_N - A_N n_N + A_N n_N = b_N.$$

Thus, the complete algorithm of the solution:

Forward step, 2 recursions:

$$m_1 := 0, \qquad m_{k+1} := -\frac{C_k}{A_k m_k + B_k} \qquad (k = 1, 2, ..., N-1)$$

$$n_1 := 0, \qquad n_{k+1} := \frac{b_k - A_k n_k}{A_k m_k + B_k} \qquad (k = 1, 2, ..., N)$$

Backward step, 1 recursion:

$$x_N := n_{N+1}, \qquad x_{k-1} := m_k x_k + n_k \qquad (k = N, N-1, ..., 2)$$

The total number of the necessary algebraic operations is clearly $\mathcal{O}(N)$ only. Moreover, the algorithm is numerically stable.

2.3 The LU decomposition of matrices

Let $A = [a_{kj}] \in \mathbf{M}_{N \times N}$ be a regular matrix, for which the Gaussian elimination can be performed without swapping rows.

Theorem: The matrix A can be decomposed uniquely in the form

$$A = LU$$

where L is a *normed lower triangular matrix* (i.e. $L_{kk} = 1$, and $L_{kj} = 0$, if $j > k$), and U is an *upper triangular matrix* (i.e. $U_{kj} = 0$, if $j < k$).

Proof: The proof of the theorem and the validation of the following algorithm are not difficult but lengthy, so they are omitted. Only the uniqueness is proved. We will utilize the easily verifiable facts:

- the product of (normed) lower triangular matrices is (normed) lower triangular;
- the product of (normed) upper triangular matrices is (normed) upper triangular;
- the inverse of (normed) lower triangular matrix is (normed) lower triangular (provided that it exists);
- the inverse of (normed) upper triangular matrix is (normed) upper triangular (provided that it exists);

Now suppose that there exist two decompositions: $A = L_1U_1 = L_2U_2$. Then $L_2^{-1}L_1 = U_2U_1^{-1}$. The matrix on the left-hand side is normed lower triangular, while the matrix on the right-hand side is upper triangular. Consequently, both sides are equal to the unit matrix, which implies the uniqueness.

The LU decomposition is often called LU factorization as well.

The practical importance of the LU decomposition is as follows. If a lot of systems of equations $Ax = b$ are to be solved with different right-hand sides but with a common matrix A, then it is worth performing an LU decomposition only once; after this one has to solve the equations $L(Ux) = b$ i.e. the pairs of equations $Ly = b$, $Ux = y$. From computational point of view, it is much cheaper than solving the equations independently, since forward and back substitutions have to be performed without performing elimination steps.

It can be shown that the computational complexity of the LU decomposition is $\mathcal{O}(N^3)$. Having performed the decomposition, the computational complexity of each pair of equations $Ly = b$, $Ux = y$ is at most $\mathcal{O}(N^2)$. Is special cases, this may be even less, when e.g. L and U are sparse matrices.

LU decomposition by Gaussian elimination: When eliminating with the help of the kth equation, the kth equation multiplied by the factor $l_{m,k} :=$

$A_{m,k}/A_{k,k}$ is subtracted from the mth equation ($m = k+1, ..., N$). From these factors $l_{m,k}$, the matrix L is built up:

$$L = \begin{pmatrix} 1 & 0 & 0 & ... & 0 \\ l_{2,1} & 1 & 0 & ... & 0 \\ l_{3,1} & l_{3,2} & 1 & ... & 0 \\ ... & ... & ... & ... & ... \\ l_{N,1} & l_{N,2} & l_{N,3} & ... & 1 \end{pmatrix}$$

After the elimination steps, we obtain the matrix U.

Example: Compute the LU decomposition of the matrix

$$A := \begin{pmatrix} 2 & -6 & 10 \\ 2 & -5 & 3 \\ 3 & -2 & 1 \end{pmatrix}.$$

Solution:

$$\begin{pmatrix} 2 & -6 & 10 \\ 2 & -5 & 3 \\ 3 & -2 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 \\ . & 1 & 0 \\ . & . & 1 \end{pmatrix}$$

$$\begin{pmatrix} 2 & -6 & 10 \\ 0 & 1 & -7 \\ 3 & -2 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ . & . & 1 \end{pmatrix}$$

$$\begin{pmatrix} 2 & -6 & 10 \\ 0 & 1 & -7 \\ 0 & 7 & -14 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ \frac{3}{2} & . & 1 \end{pmatrix}$$

$$U = \begin{pmatrix} 2 & -6 & 10 \\ 0 & 1 & -7 \\ 0 & 0 & 35 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ \frac{3}{2} & 7 & 1 \end{pmatrix} = L$$

Another example, for which two different solutions is shown. The first technique is essentially identical to the previous technique, while the second one is perhaps even simpler.

Compute the LU decomposition of the following matrix:

$$A := \begin{pmatrix} 2 & -2 & 4 \\ -2 & -1 & -1 \\ 4 & -1 & 3 \end{pmatrix}$$

Solution: First of all, we fill the matrix L with the entries which are immediately known. In the main diagonal, the entries are equal to 1, and above the main diagonal, all entries are zero:

$$A = \begin{pmatrix} 2 & -2 & 4 \\ -2 & -1 & -1 \\ 4 & -1 & 3 \end{pmatrix} \qquad L = \begin{pmatrix} 1 & 0 & 0 \\ . & 1 & 0 \\ . & . & 1 \end{pmatrix}$$

Now start the elimination: the first row multiplied by $\frac{-2}{2}$, $\frac{4}{2}$ are subtracted from the second and the third row, respectively. The factors $\frac{-2}{2}$ és $\frac{4}{2}$ are written in the first column (to the second and third entry).

$$\begin{pmatrix} 2 & -2 & 4 \\ 0 & -3 & 3 \\ 0 & 3 & -5 \end{pmatrix} \qquad L = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & . & 1 \end{pmatrix}$$

The elimination is continued; the second row of the matrix multiplied by the factor $\frac{3}{-3}$ is subtracted from the third row. The factor $\frac{3}{-3}$ is written in the second column (to the third entry). Thus, we have obtained both the matrix U and L:

$$U = \begin{pmatrix} 2 & -2 & 4 \\ 0 & -3 & 3 \\ 0 & 0 & -2 \end{pmatrix} \qquad L = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & -1 & 1 \end{pmatrix}$$

(Check the result by computing the matrix product LU directly.)

Another solution: Fill in the entries of the matrices L and U which are a priori known:

- all entries of the main diagonal of L are equal to 1;
- all entries above the main diagonal of L are equal to 0;
- the entries of the first column of L are: A_{21}/A_{11}, A_{31}/A_{11}, A_{41}/A_{11}, ...
- all entries below the main diagonal of U are equal to 0;
- the first row of U always equals to the first row of A.

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & \ell_{32} & 1 \end{pmatrix} \qquad U = \begin{pmatrix} 2 & -2 & 4 \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}$$

Multiplying the kth row of L by the jth colums of U, the entry A_{kj} of the original matrix A has to be obtained. A smart choice of the order of the row-column multiplications makes it possible that only one unknown entry appears in every equation, which can be substituted into the corresponding entry of L or U, respectively.

For instance, multiplying the second row of L by the second column of U, we have: $(-1) \cdot (-2) + 1 \cdot u_{22} + 0 = -1$, whence u_{22} can be calculated: $u_{22} = -3$:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & \ell_{32} & 1 \end{pmatrix} \qquad U = \begin{pmatrix} 2 & -2 & 4 \\ 0 & -3 & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}$$

Now multiplying the third row of L by the second column of U, we have: $2 \cdot (-2) + \ell_{32} \cdot (-3) + 0 = -1$, whence ℓ_{32} can be calculated: $\ell_{32} = -1$:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & -1 & 1 \end{pmatrix} \qquad U = \begin{pmatrix} 2 & -2 & 4 \\ 0 & -3 & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}$$

Now the matrix L has been computed. Multiplying the second row of L by the third column of U, we have: $(-1) \cdot 4 + 1 \cdot u_{23} + 0 = -1$, from which u_{23} can be calculated: $u_{23} = 3$:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & -1 & 1 \end{pmatrix} \qquad U = \begin{pmatrix} 2 & -2 & 4 \\ 0 & -3 & 3 \\ 0 & 0 & u_{33} \end{pmatrix}$$

Finally, multiplying the third row of L by the third column of U: $2 \cdot 4 + (-1) \cdot 3 + 1 \cdot u_{33} = 3$, whence u_{33} can be calculated: $u_{33} = -2$:

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & -1 & 1 \end{pmatrix} \qquad U = \begin{pmatrix} 2 & -2 & 4 \\ 0 & -3 & 3 \\ 0 & 0 & -2 \end{pmatrix}$$

Thus, the desired LU decomposition has been computed.

Remark: The LU decomposition gives us also a simple method for calculating the determinant of the matrix, which is much cheaper than the recursive

definition. Indeed, if $A = LU$, then $\det(A) = \det(L) \cdot \det(U)$. Both of the determinants in the right-hand side can be calculated easily, they are the product of the diagonal entries (why?). Thus, $\det(L) = 1$, and

$$\det(A) = \det(U) = U_{11} \cdot U_{22} \cdot ... \cdot U_{NN}.$$

2.4 LDL^* decomposition of self-adjoint matrices

Let $A = [a_{kj}] \in \mathbf{M}_{N\times N}$ be self-adjoint. For the time being, assume that it is positive definite. Then the Gaussian elimination can be performed without swapping rows (since all leading principal minors are positive). That is, A can be uniquely decomposed in the form:

$$A = LU$$

Denote by D the diagonal matrix, the diagonal entries of which are composed by the diagonal entries of the matrix U:

$$D = \begin{pmatrix} U_{11} & 0 & 0 & ... & 0 \\ 0 & U_{22} & 0 & ... & 0 \\ 0 & 0 & U_{33} & ... & 0 \\ ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & U_{NN} \end{pmatrix}.$$

Due to the regularity of A, all diagonal entries differ from zero. Express U as a product: $U = DU'$. Then U' is a *normed* upper triangular matrix, and $A = LDU'$. Since A is self-adjoint, therefore $A = A^* = (U')^* DL^*$. Here $(U')^*$ is a *normed* lower triangular matrix and DL^* is an upper triangular matrix. The uniqueness of the LU decomposition implies that $(U')^* = L$ and $DL^* = U$, whence: $A = LDL^*$.

This decomposition is called the LDL^* decomposition (or LDL^* factorization) of the matrix A.

Proposition: If A is self-adjoint and positive definite, then the LDL^* decomposition is uniquely determined.

Proof: Assume that A has two such decompositions:
$A = L_1 D_1 L_1^* = L_2 D_2 L_2^*$, then $A = L_1(L_1 D_1)^* = L_2(L_2 D_2)^*$. Both of them are an LU decomposition. The uniqueness of the LU decomposition implies that $L_1 = L_2$ and $L_1 D_1 = L_2 D_2$, whence: $D_1 = D_2$.

Remark: If A is self-adjoint not necessarily positive definite but regular, then the Gaussian elimination can be performed without swapping rows (all leading principal minors differ from zero). Thus, the LU decomposition still exists, and the main diagonal of U contains no zeros (due to the regularity of A). Therefore the construction of the LDL^* decomposition and also the uniqueness remain valid.

Example: Calculate the LDL^* decomposition of the matrix:

$$A = \begin{pmatrix} 2 & -2 & 4 \\ -2 & -1 & -1 \\ 4 & -1 & 3 \end{pmatrix}$$

Solution: The LU decomposition of the matrix is: $A = LU$, where

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & -1 & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} 2 & -2 & 4 \\ 0 & -3 & 3 \\ 0 & 0 & -2 \end{pmatrix}$$

(check it!). Hence $A = LDL^*$, where

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

Observe that among the diagonal entries of D, there are negative numbers as well, since A is not positive definite (though it is self-adjoint and regular).

2.5 Cholesky decomposition of self-adjoint, positive definite matrices

Let the matrix $A = [a_{kj}] \in \mathbf{M}_{N\times N}$ be self-adjoint, positive definite. Then the Gaussian elimination can be performed without swapping rows, therefore A can be uniquely decomposed in a LU as well as in a LDL^* form:

$$A = LDL^*$$

Observe that the diagonal entries of D are positive. Let $x \neq \mathbf{0}$ an arbitrary vector, then the positive definiteness implies that $x^*Ax > 0$, therefore $x^*LDL^*x = (L^*x)^*D(L^*x) > 0$. Since L is regular, therefore so is L^*. Due to the regularity, for every index $k = 1, 2, ..., N$ there exists a vector $x_k \neq \mathbf{0}$

such that $L^*x_k = e_k$, where e_k denotes the kth standard basis vector. Hence: $(L^*x_k)^*D(L^*x_k) = e_k^*De_k = D_{kk} > 0$, as stated earlier.

Consequently, the matrix $\sqrt{D}$ does make sense, it is a diagonal matrix and $(\sqrt{D})_{kk} = \sqrt{D_{kk}}$ $(k = 1, 2, ..., N)$. This implies that

$$A = LDL^* = L\sqrt{D}\sqrt{D}L^* = \tilde{L}\tilde{L}^*,$$

where $\tilde{L} := L\sqrt{D}$. Here $\tilde{L}$ is a not necessarily normed lower triangular matrix, and its diagonal entries (which are equal to the numbers $\sqrt{D_{kk}}$) are positive.

The decomposition $A = \tilde{L}\tilde{L}^*$ is called the *Cholesky decomposition* (or Cholesky factorization) of A.

Remark: When solving systems of equations $Ax = b$ *with a lot of different right-hand sides*, the computational advantage of the use of Cholesky decomposition is as follows. Once the Cholesky decomposition has been calculated, then, instead of the systems $Ax = b$, the pairs of the systems $\tilde{L}y = b$, $\tilde{L}^*x = y$ have to be solved. These systems requires no eliminations, only forward and back substitutions, which reduces the computational cost from $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2)$ per system. It can be shown, that the error propagation is more moderate than in the case of the LU decomposition. In addition to it, the Cholesky decomposition gives a very good tool to compute all eigenvalues of a self-adjoint, positive definite matrix, see later.

Proposition: The Cholesky decomposition is unique, i.e. if A is self-adjoint and positive definite, then there exists a uniquely determined $\tilde{L}$ lower triangular matrix with positive diagonal entries such that $A = \tilde{L}\tilde{L}^*$.

Proof: Suppose that A that two such decompositions: $A = \tilde{L}_1\tilde{L}_1^* = \tilde{L}_2\tilde{L}_2^*$. Then there exist diagonal matrices D_1, D_2 with positive diagonal entries and normed lover triangular matrices L_1, L_2 such that $\tilde{L}_1 = L_1D_1$ and $\tilde{L}_2 = L_2D_2$. Consequently:

$$A = \tilde{L}_1\tilde{L}_1^* = L_1D_1D_1L_1^* = L_1(D_1^2L_1^*)$$

and

$$A = \tilde{L}_2\tilde{L}_2^* = L_2D_2D_2L_2^* = L_2(D_2^2L_2^*)$$

The right-hand sides are LU decompositions of the same matrix A. Due to the uniqueness of the LU decomposition, $L_1 = L_2$, and $D_1^2L_1^* = D_2^2L_2^* =$

$D_2^2 L_1^*$, whence $D_1^2 = D_2^2$ (since L_1^* is invertible). Finally, since the diagonal entries of D_1 and D_2 are positive, we have that $D_1 = D_2$, therefore $\tilde{L}_1 = \tilde{L}_2$.

In practical computations, the Cholesky decomposition can be performed through the above outlined LU and LDL^* decompositions, but the technique based on the matrix multiplication can also be applied, thus avoiding the elimination. Seek the matrix $\tilde{L}$ in the form:

$$\tilde{L} = \begin{pmatrix} \ell_{11} & 0 & 0 & ... & 0 \\ \ell_{21} & \ell_{22} & 0 & ... & 0 \\ \ell_{31} & \ell_{32} & \ell_{33} & ... & 0 \\ ... & ... & ... & ... & ... \\ \ell_{N1} & \ell_{N2} & \ell_{N3} & ... & \ell_{NN} \end{pmatrix}$$

Then, performing the multiplication $\tilde{L}\tilde{L}^*$, we have:

$$A_{ij} = \sum_{k=1}^{\min(i,j)} \ell_{ik}\ell_{jk}$$

We have obtained the following, recursively defined formulations. For the diagonal entries:

$$A_{jj} = \sum_{k=1}^{j} \ell_{jk}^2 = \ell_{jj}^2 + \sum_{k=1}^{j-1} \ell_{jk}^2,$$

whence:

$$\ell_{jj} = \sqrt{A_{jj} - \sum_{k=1}^{j-1} \ell_{jk}^2} \qquad (j = 1, 2, ..., N).$$

For the off-diagonal entries i.e. for the row indices $i = j+1, j+2, ..., N$:

$$A_{ij} = \sum_{k=1}^{j} \ell_{ik}\ell_{jk} = \ell_{ij}\ell_{jj} + \sum_{k=1}^{j-1} \ell_{ik}\ell_{jk},$$

whence:

$$\ell_{ij} = \frac{1}{\ell_{jj}} \cdot \left(A_{ij} - \sum_{k=1}^{j-1} \ell_{ik}\ell_{jk} \right) \qquad (i = j+1, ..., N)$$

It is recommended to perform the computations columnwise: first, the numbers ℓ_{11}, ℓ_{21}, ℓ_{31}, ... ℓ_{N1} are computed. After this, the numbers ℓ_{22}, ℓ_{32}, ... ℓ_{N2} are computed and so forth.

Example: Determine the Cholesky decomposition of the matrix

$$A = \begin{pmatrix} 4 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 4 \end{pmatrix}.$$

Solution: The LU decomposition of the matrix: $A = LU$, where

$$L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{4} & \frac{1}{2} & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} 4 & 2 & 1 \\ 0 & 3 & \frac{3}{2} \\ 0 & 0 & 3 \end{pmatrix}$$

(check it!). Hence $A = LDL^*$, where

$$D = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{pmatrix}$$

Therefore

$$\sqrt{D} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & \sqrt{3} & 0 \\ 0 & 0 & \sqrt{3} \end{pmatrix}$$

We have obtained the Cholesky decomposition: $A = \tilde{L}\tilde{L}^*$, where

$$\tilde{L} = L\sqrt{D} = \begin{pmatrix} 2 & 0 & 0 \\ 1 & \sqrt{3} & 0 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} & \sqrt{3} \end{pmatrix}.$$

Note that using the above recursive formulas, the elimination as well as the LU decomposition can be avoided. Performing the computations columnwise:

$$\ell_{11} = \sqrt{A_{11}} = 2$$

$$\ell_{21} = \frac{1}{\ell_{11}} \cdot A_{21} = \frac{1}{2} \cdot 2 = 1, \qquad \ell_{31} = \frac{1}{\ell_{11}} \cdot A_{31} = \frac{1}{2} \cdot 1 = \frac{1}{2}$$

$$\ell_{22} = \sqrt{A_{22} - \ell_{21}^2} = \sqrt{4 - 1^2} = \sqrt{3}$$

$$\ell_{32} = \frac{1}{\ell_{22}} \cdot (A_{32} - \ell_{31}\ell_{21}) = \frac{1}{\sqrt{3}} \cdot \left(2 - \frac{1}{2} \cdot 1\right) = \frac{\sqrt{3}}{2}$$

$$\ell_{33} = \sqrt{A_{33} - \ell_{31}^2 - \ell_{32}^2} = \sqrt{4 - \frac{1}{4} - \frac{3}{4}} = \sqrt{3},$$

in accordance with the previous result.

2.6 QR decomposition of square matrices

Let $A \in \mathbf{M}_{N\times N}$ be a *regular* matrix. Denote by a_1, a_2, ..., a_N the column vectors of A (they are obviously linearly independent). Formally:

$$A = (\ a_1 \mid a_2 \mid ... \mid a_N)$$

From the vectors a_1, a_2, ..., a_N, construct the orthonormal system of vectors e_1, e_2, ..., e_N with the help of the *Gram-Schmidt orthogonalization* algorithm:

$$\tilde{e}_1 := a_1, \quad e_1 := \frac{\tilde{e}_1}{||\tilde{e}_1||},$$

and for $k = 2, 3, ..., N$:

$$\tilde{e}_k := a_k - \sum_{j=1}^{k-1} \langle a_k, e_j \rangle \cdot e_j, \quad e_k := \frac{\tilde{e}_k}{||\tilde{e}_k||}$$

From these equations we obtain:

$$a_1 = \tilde{e}_1 = e_1 \cdot ||\tilde{e}_1||$$

and for $k = 2, 3, ..., N$:

$$a_k = \sum_{j=1}^{k-1} \langle a_k, e_j \rangle \cdot e_j + \tilde{e}_k = \sum_{j=1}^{k-1} \langle a_k, e_j \rangle \cdot e_j + e_k \cdot ||\tilde{e}_k||$$

The pth component of the qth vector in the left-hand side equals to the matrix element A_{pq} $(p, q = 1, 2, ..., N)$. From the definition of the matrix multiplication it follows that the pth component of the qth vector in the right-hand side equals to the (p, q)th element of the following product matrix:

$$Q \cdot R :=$$

$$:= (\ e_1 \mid e_2 \mid ... \mid e_N) \cdot \begin{pmatrix} ||\tilde{e}_1|| & \langle e_1, a_2 \rangle & \langle e_1, a_3 \rangle & ... \\ 0 & ||\tilde{e}_2|| & \langle e_2, a_3 \rangle & ... \\ 0 & 0 & ||\tilde{e}_3|| & ... \\ ... & ... & ... & ... \end{pmatrix},$$

that is, $A = QR$.

Since the vectors e_1, e_2, ..., e_N form an orthonormal system, the matrix Q is an orthogonal matrix, and R is an upper triangular matrix with positive diagonal entries. The above decomposition is called the QR *decomposition* (or QR factorization) of the matrix A.

Remark: From the equality $a_1 = e_1 \cdot ||\tilde{e}_1||$, we have: $\langle e_1, a_1 \rangle = ||\tilde{e}_1||$. Similarly, the equalities

$$a_k = \sum_{j=1}^{k-1} \langle a_k, e_j \rangle \cdot e_j + e_k \cdot ||\tilde{e}_k||$$

imply that: $\langle e_k, a_k \rangle = ||\tilde{e}_k||$. Thus, the entries of R can be rewritten in the following, more uniform way:

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} & ... \\ 0 & r_{22} & r_{23} & ... \\ 0 & 0 & r_{33} & ... \\ ... & ... & ... & ... \end{pmatrix},$$

where $r_{k,j} := \langle e_k, a_j \rangle$ $(k = 1, 2, ..., N; j = k, k+1, ..., N)$

Note that having performed the Gram-Schmidt orthogonalization algorithm, and the matrix Q has been computed, the orthogonality of the matrix Q implies that the matrix R can be performed in the following way: $R = Q^{-1}A = Q^*A$.

If the system $Ax = b$ has to be solved with *a lot of different right-hand sides*, then, with the help of the QR decomposition, these equations can be solved in a way that is simpler than even the LU or the Cholesky decomposition. Namely, $Ax = b$ is equivalent to $QRx = b$, whence $Rx = Q^*b$. This system has an upper triangular matrix, therefore it can be solved economically (from computational point of view).

Proposition: The QR decomposition is uniquely determined, i.e. if $A \in \mathbf{M}_{N \times N}$ is regular then there exists a unique orthogonal matrix Q and a unique upper triangular matrix R with positive diagonal entries such that $A = QR$.

Proof: Suppose that A has two such decompositions: $A = Q_1R_1 = Q_2R_2$. Then $Q_2^{-1}Q_1 = R_2R_1^{-1}$. The left-hand side is an orthogonal matrix, therefore $R_2R_1^{-1}$ is orthogonal *and* upper triangular matrix, and its inverse, $(R_2R_1^{-1})^{-1}$ is also orthogonal and upper triangular matrix. Consequently:

$$(R_2R_1^{-1})^{-1} = (R_2R_1^{-1})^* = (R_1^*)^{-1}R_2^*,$$

which is a lower triangular matrix, therefore $R_2R_1^{-1}$ is also a lower triangular matrix. We have obtained that $R_2R_1^{-1}$ is an upper and lower triangular

matrix at the same time, that is, it is diagonal matrix. But it is also an orthogonal matrix, therefore the absolute values of the diagonal entries are equal to 1. Since the diagonal entries of R_1 and R_2 are positive, this remains the case also for R_1^{-1}. Therefore $R_2 R_1^{-1} = I$, i.e. $R_1 = R_2$, which implies also $Q_1 = Q_2$.

Example: Determine the QR decomposition of the matrix:

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

Proof: The column vectors of the matrix A are as follows:

$$a_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \quad a_2 = \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix}, \quad a_3 = \begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix}.$$

Performing a Gram-Schmidt orthogonalization:

$$\tilde{e}_1 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \qquad e_1 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}.$$

$$\tilde{e}_2 = \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} - \langle \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix}, \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} \rangle \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix}, \qquad e_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

$$\tilde{e}_3 = \begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix} - \langle \begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix}, \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} \rangle \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix} - \langle \begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \rangle \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix},$$

that is:

$$\tilde{e}_3 = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}, \qquad e_3 = \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{pmatrix}.$$

Hence:

$$Q = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{pmatrix}$$

The matrix R can be determined componentwise:

$$r_{11} = ||\tilde{e}_1|| = \sqrt{2}, \qquad r_{12} = \langle e_1, a_2 \rangle = 0, \qquad r_{13} = \langle e_1, a_3 \rangle = 2\sqrt{2}$$

$$r_{22} = ||\tilde{e}_2|| = 2, \qquad r_{23} = \langle e_2, a_3 \rangle = 0$$

$$r_{33} = ||\tilde{e}_3|| = \sqrt{2}$$

Hence:

$$R = \begin{pmatrix} \sqrt{2} & 0 & 2\sqrt{2} \\ 0 & 2 & 0 \\ 0 & 0 & \sqrt{2} \end{pmatrix}$$

2.7 QR decomposition by Householder transformation

For an arbitrary nonzero vector $\mathbf{0} \neq u \in \mathbf{R}^N$, define the following matrix:

$$Q := Q_u := I - \frac{2}{||u||^2} \cdot uu^*$$

(u is meant to be a column vector, so that uu^* is a dyadic matrix.)

The matrix transformation defined by $\mathbf{M}_{N\times N} \to \mathbf{M}_{N\times N}$, $A \to QA$ is called Householder transformation.

Proposition: The above matrix Q is orthogonal *and* self-adjoint.

Proof:

$$Q^* = I - \frac{2}{||u||^2} \cdot (uu^*)^* = I - \frac{2}{||u||^2} \cdot u^{**}u^* = Q,$$

thus, Q is self-adjoint. On the other hand:

$$QQ^* = Q^2 = \left(I - \frac{2}{||u||^2} \cdot uu^* \right)^2 = I - \frac{4}{||u||^2} uu^* + \frac{4}{||u||^4} uu^*uu^* = I,$$

since in the right-hand side: $u^*u = ||u||^2$, therefore $uu^*uu^* = ||u||^2 \cdot uu^*$. That is, $Q^{-1} = Q^*$, which means that the matrix Q is orthogonal.

Remark: The matrix Q the matrix of *reflection* to the subspace with the normal vector $\frac{u}{||u||}$, since: $I - \frac{1}{||u||^2} \cdot uu^*$ is the matrix of projection to the same subspace. The Q matrix is *involutory mapping* (or *involution*), i.e. it is the inverse of itself. Indeed, $Q^2 = QQ^* = I$, that is: $Q^{-1} = Q$.

Now let $A \in \mathbf{M}_{N\times N}$ be a regular matrix and denote by a_1, a_2, ..., a_N the column vectors of the matrix A:

$$A = (\ a_1 \mid a_2 \mid ... \mid a_N)$$

Denote by e_1, e_2, ... e_N the standard basis of the space $\mathbf{R}^N$.

Define

$$u := a_1 + \tau \cdot e_1,$$

and try to choose τ in such a way that $Q_u a_1 = const. \cdot e_1$ is valid. If it has been managed, then:

$$Q_u A = Q_u\,(\ a_1 \mid a_2 \mid ... \mid a_N) = \begin{pmatrix} const. & ... & ... & ... \\ 0 & - - - & - - - & - - - \\ 0 & \mid & & \\ 0 & \mid & A_1 & \\ ... & \mid & & \\ 0 & \mid & & \end{pmatrix}$$

Repeating the procedure for the smaller matrix $A_1 \in \mathbf{M}_{(N-1)\times(N-1)}$ and the even smaller matrix $A_2 \in \mathbf{M}_{(N-2)\times(N-2)}$ etc., we arrive at the QR decomposition.

Now let us try to properly choose the parameter τ.

$$Q_u a_1 = a_1 - \frac{2}{||u||^2} uu^* a_1 = a_1 - \frac{2\langle u, a_1\rangle}{||u||^2} u = a_1 - \frac{2\langle u, a_1\rangle}{||u||^2} a_1 - \frac{2\tau\langle u, a_1\rangle}{||u||^2} e_1 =$$

$$= (1 - \frac{2\langle u, a_1\rangle}{||u||^2}) \cdot a_1 - \frac{2\tau\langle u, a_1\rangle}{||u||^2} e_1$$

The right-hand side certainly equals to $const. \cdot e_1$, if $1 - \frac{2\langle u,a_1\rangle}{||u||^2} = 0$. With equivalent transformations:

$$1 - \frac{2\langle u, a_1\rangle}{||u||^2} = 0 \qquad \Leftrightarrow \qquad 2\langle u, a_1\rangle = ||u||^2 \qquad \Leftrightarrow$$

$$\Leftrightarrow \qquad 2\langle a_1 * \tau e_1, a_1\rangle = ||a_1||^2 + 2\tau\langle a_1, e_1\rangle + \tau^2 \qquad \Leftrightarrow$$

$$\Leftrightarrow \qquad 2||a_1||^2 + 2\tau\langle e_1, a_1\rangle = ||a_1||^2 + 2\tau\langle a_1, e_1\rangle + \tau^2 \qquad \Leftrightarrow$$

$$\Leftrightarrow \qquad \tau^2 = ||a_1||^2$$

This gives two possible values for τ, which differ from each other in sign only. Define:

$$\tau := (\text{sign}\langle a_1, e_1\rangle) \cdot ||a_1||$$

Then

$$||u||^2 = ||a_1||^2 + 2\tau\langle a_1, e_1\rangle + \tau^2$$

With the above definition of τ, the three terms in the right-hand side will be positive, therefore $u \neq \mathbf{0}$. This assures that the Householder transformation defined by the vector u does make sense (the divisor is not equal to zero), and $Q_u a_1 = const. \cdot e_1$. Moreover, the *const.* in the right-hand side can be calculated easily:

$$const. = -\frac{2\tau\langle u, a_1\rangle}{||u||^2} = -2\tau\frac{\langle a_1 + \tau e_1, a_1}{||a_1 + \tau e_1||^2} =$$

$$= -2\tau\frac{||a_1||^2 + \tau\langle e_1, a_1\rangle, a_1}{||a_1||^2 + 2\tau\langle a_1, e_1\rangle + \tau^2} = -2\tau\frac{||a_1||^2 + \tau\langle e_1, a_1\rangle, a_1}{2||a_1||^2 + 2\tau\langle a_1, e_1\rangle} = -\tau,$$

that is, $Q_u a_1 = -\tau \cdot e_1$

Example: Determine the QR decomposition of the matrix

$$A = \begin{pmatrix} 3 & 4 \\ 4 & 0 \end{pmatrix}$$

by Householder transformation.

Solution: Here $a_1 = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$, and $e_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

Define $\tau := (\text{sign}\langle a_1, e_1\rangle) \cdot ||a_1|| = 5$, and $u = a_1 + \tau e_1 = \begin{pmatrix} 8 \\ 4 \end{pmatrix}$.

Moreover, $||u||^2 = 80$, whence

$$Q_u = I - \frac{2}{80} \cdot \begin{pmatrix} 8 \\ 4 \end{pmatrix} (8 \quad 4) = \begin{pmatrix} -\frac{3}{5} & -\frac{4}{5} \\ -\frac{4}{5} & \frac{3}{5} \end{pmatrix}$$

This implies that:

$$Q_u A = \begin{pmatrix} -5 & -\frac{12}{5} \\ 0 & -\frac{16}{5} \end{pmatrix} =: R,$$

i.e. the desired QR decomposition is as follows:

$$A = Q_u^* R = \begin{pmatrix} -\frac{3}{5} & -\frac{4}{5} \\ -\frac{4}{5} & \frac{3}{5} \end{pmatrix} \cdot \begin{pmatrix} -5 & -\frac{12}{5} \\ 0 & -\frac{16}{5} \end{pmatrix}$$

2.8 Exercises

1. Solve the following system of equations by Gaussian elimination:

$$\begin{array}{rcrcrcr} x & + & 4y & + & 2z & = & 5 \\ -3x & + & 2y & + & z & = & -1 \\ 4x & - & y & - & z & = & 2 \end{array}$$

Solution: The scheme of computation is as follows:

$$\left(\begin{array}{rrr|r} 1 & 4 & 2 & 5 \\ -3 & 2 & 1 & -1 \\ 4 & -1 & -1 & 2 \end{array}\right) \to \left(\begin{array}{rrr|r} 1 & 4 & 2 & 5 \\ 0 & 14 & 7 & 14 \\ 0 & -17 & -9 & -18 \end{array}\right) \to$$

$$\to \left(\begin{array}{rrr|r} 1 & 4 & 2 & 5 \\ 0 & 1 & \frac{1}{2} & 1 \\ 0 & -17 & -9 & -18 \end{array}\right) \to \left(\begin{array}{rrr|r} 1 & 4 & 2 & 5 \\ 0 & 1 & \frac{1}{2} & 1 \\ 0 & 0 & -\frac{1}{2} & -1 \end{array}\right) \to$$

$$\to \left(\begin{array}{rrr|r} 1 & 4 & 2 & 5 \\ 0 & 1 & \frac{1}{2} & 1 \\ 0 & 0 & 1 & 2 \end{array}\right) \to \left(\begin{array}{rrr|r} 1 & 4 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \end{array}\right) \to$$

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

The solution is: $x = 1$, $y = 0$, $z = 2$.

2. Solve the following linear system of equations by Gauss-Jordan elimination:

$$\begin{array}{rcrcrcr} 2x_1 & - & 3x_2 & + & x_3 & = & -1 \\ x_1 & - & 2x_2 & - & 3x_3 & = & 6 \\ 2x_1 & + & x_2 & + & x_3 & = & 3 \end{array}$$

Solution: Swapping the first and the second rows, and subtraction the double of the (new) first row to the second and the third rows:

$$\left(\begin{array}{rrr|r} 2 & -3 & 1 & -1 \\ 1 & -2 & -3 & 6 \\ 2 & 1 & 1 & 3 \end{array}\right) \quad \to \quad \left(\begin{array}{rrr|r} 1 & -2 & -3 & 6 \\ 2 & -3 & 1 & -1 \\ 2 & 1 & 1 & 3 \end{array}\right) \quad \to$$

$$\left(\begin{array}{ccc|c} 1 & -2 & -3 & 6 \\ 0 & 1 & 7 & -13 \\ 0 & 5 & 7 & -9 \end{array}\right)$$

Let us eliminate in forward direction with the second row:

$$\left(\begin{array}{ccc|c} 1 & -2 & -3 & 6 \\ 0 & 1 & 7 & -13 \\ 0 & 0 & -28 & 56 \end{array}\right)$$

Continue the elimination also in the backward direction. (This is the essential difference from the Gaussian elimination.) Adding the double of the second row to the first row, we obtain:

$$\left(\begin{array}{ccc|c} 1 & 0 & 11 & -20 \\ 0 & 1 & 7 & -13 \\ 0 & 0 & -28 & 56 \end{array}\right) \quad \rightarrow \quad \left(\begin{array}{ccc|c} 1 & 0 & 11 & -20 \\ 0 & 1 & 7 & -13 \\ 0 & 0 & 1 & -2 \end{array}\right)$$

With the third row, we eliminate in backward direction:

$$\left(\begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -2 \end{array}\right)$$

Finally we have:

$$x_1 = 2, \qquad x_2 = 1, \qquad x_3 = -2$$

3. Solve the following system of linear equations by Gaussian elimination:

$$\begin{array}{rcrcrcl} 2x_1 & - & x_2 & + & x_3 & = & 3 \\ 2x_1 & + & 2x_2 & - & 4x_3 & = & 4 \\ x_1 & - & 2x_2 & + & 3x_3 & = & 1 \end{array}$$

Solution: First of all, now the matrix of the system is singular (its determinant is zero; check it!). This means that the solvability of the system depends on the righ-hand side. Either a solution exists (and in this case, infinitely many solutions exist), or no solution exists. The scheme of the computation is as follows:

$$\left(\begin{array}{ccc|c} 2 & -1 & 1 & 3 \\ 2 & 2 & -4 & 4 \\ 1 & -2 & 3 & 1 \end{array}\right) \rightarrow \left(\begin{array}{ccc|c} 1 & -2 & 3 & 1 \\ 2 & 2 & -4 & 4 \\ 2 & -1 & 1 & 3 \end{array}\right) \rightarrow$$

$$\left(\begin{array}{ccc|c} 1 & -2 & 3 & 1 \\ 0 & 6 & -10 & 2 \\ 0 & 3 & -5 & 1 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & -2 & 3 & 1 \\ 0 & 1 & -\frac{5}{3} & \frac{1}{3} \\ 0 & 0 & 0 & 0 \end{array}\right)$$

At the last elimination step, the third row consists of zeros. One of the unknowns, say x_3 can be chosen in an arbitrary way: $x_3 := t$ (where $t \in \mathbf{R}$ is arbitrary). Thus, the third row will change in the following way:

$$\left(\begin{array}{ccc|c} 1 & -2 & 3 & 1 \\ 0 & 1 & -\frac{5}{3} & \frac{1}{3} \\ 0 & 0 & 1 & t \end{array}\right)$$

Continuing the elimination:

$$\left(\begin{array}{ccc|c} 1 & -2 & 0 & 1-3t \\ 0 & 1 & 0 & \frac{1}{3}+\frac{5}{3}t \\ 0 & 0 & 1 & t \end{array}\right) \quad \to \quad \left(\begin{array}{ccc|c} 1 & 0 & 0 & \frac{5}{3}+\frac{1}{3}t \\ 0 & 1 & 0 & \frac{1}{3}+\frac{5}{3}t \\ 0 & 0 & 1 & t \end{array}\right)$$

The system of equations has infinitely many solutions, and for arbitrary $t \in \mathbf{R}$:

$$x_1 = \frac{5}{3} + \frac{1}{3}t, \quad x_2 = \frac{1}{3} + \frac{5}{3}t, \quad x_3 = t$$

4. Compute the inverse of the following matrix by Gaussian elimination:

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

Solution: The scheme of computation is as follows:

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

Swap the first two rows and eliminate in the first column:

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

$$\left(\begin{array}{ccc|ccc} 1 & -1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 2 & 1 \end{array}\right) \to \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1 & 2 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 2 & 1 \end{array}\right)$$

We have obtained that:

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

5. Compute the inverse of the following matrix by Gaussian elimination:

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

Solution: The scheme of computation is as follows:

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

$$\to \left(\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 1 & 0 \\ 0 & 1 & 1 & \frac{1}{3} & 0 & \frac{1}{3} \end{array}\right)$$

Now swap the second and third rows in order that we can continue the elimination:

$$\left(\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 2 & 0 & 1 & 0 \end{array}\right) \to \left(\begin{array}{ccc|ccc} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & 0 & 1 & 0 & \frac{1}{2} & 0 \end{array}\right) \to$$

$$\to \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & -\frac{1}{2} & 0 \\ 0 & 1 & 0 & \frac{1}{3} & -\frac{1}{2} & \frac{1}{3} \\ 0 & 0 & 1 & 0 & \frac{1}{2} & 0 \end{array}\right)$$

The inverse matrix:

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

6. Determine the LU decomposition of the following matrix, and with the help of decomposition, compute the determinant:

$$A := \begin{pmatrix} 4 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 4 \end{pmatrix}$$

Solution: Fill the entries of the matrix L by the a priori known elements. The main diagonal consists of ones, above the diagonal, all entries are zero:

$$A = \begin{pmatrix} 4 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 4 \end{pmatrix} \qquad L = \begin{pmatrix} 1 & 0 & 0 \\ . & 1 & 0 \\ . & . & 1 \end{pmatrix}$$

Now start the elimination. The first row of A multiplied by the factor $\frac{2}{4}$ is subtracted from the second row. Similarly, the first row of A multiplied by the factor $\frac{1}{4}$ is subtracted from the third row. The above factors ($\frac{2}{4}$ and $\frac{1}{4}$) are inserted in the first column of L, to the second and the third element, respectively:

$$\begin{pmatrix} 4 & 2 & 1 \\ 0 & 3 & \frac{3}{2} \\ 0 & \frac{3}{2} & \frac{15}{4} \end{pmatrix} \qquad L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{4} & . & 1 \end{pmatrix}$$

Continue the elimination. The second row of the matrix multiplied by the factor $\frac{1}{2}$ is subtracted from the third row. The factor $\frac{1}{2}$ is inserted to the $(3,2)$th element of the matrix L. Thus, we have obtained an upper triangular matrix; this is the matrix U in the LU decomposition. At the same time, the lower triangular matrix L is also obtained:

$$U = \begin{pmatrix} 4 & 2 & 1 \\ 0 & 3 & \frac{3}{2} \\ 0 & 0 & 3 \end{pmatrix} \qquad L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{4} & \frac{1}{2} & 1 \end{pmatrix}$$

(Check the result by the direct calculation of the product $L \cdot U$. Try to use the method based on matrix multiplication which was shown earlier.)

The determinant equals to the product of the diagonal entries of the matrix U:

$$\det(A) = \det(U) = 4 \cdot 3 \cdot 3 = 36.$$

7. Determine the LU decomposition of the following matrix, and with the help of decomposition, compute the determinant:

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

Solution: Fill the entries of the matrix L by the a priori known elements. The main diagonal consists of ones, above the diagonal, all entries are zero:

$$A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix} \qquad L = \begin{pmatrix} 1 & 0 & 0 \\ . & 1 & 0 \\ . & . & 1 \end{pmatrix}$$

Now start the elimination. The first row of A multiplied by the factor $\frac{1}{2}$ is subtracted from the second row. The third row remains unchanged, since its firs element is zero. The above factors ($\frac{1}{2}$ and 0) are inserted in the first column of L, to the second and the third element, respectively:

$$\begin{pmatrix} 2 & 1 & 0 \\ 0 & \frac{3}{2} & 1 \\ 0 & 1 & 2 \end{pmatrix} \qquad L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ 0 & . & 1 \end{pmatrix}$$

Continue the elimination. The second row of the matrix multiplied by the factor $\frac{2}{3}$ is subtracted from the third row. The factor $\frac{2}{3}$ is inserted to the $(3,2)$th element of the matrix L. Thus, we have obtained an upper triangular matrix; this is the matrix U in the LU decomposition. At the same time, the lower triangular matrix L is also obtained:

$$U = \begin{pmatrix} 2 & 1 & 0 \\ 0 & \frac{3}{2} & 1 \\ 0 & 0 & \frac{4}{3} \end{pmatrix} \qquad L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ 0 & \frac{2}{3} & 1 \end{pmatrix}$$

(Check the result by the direct calculation of the product $L \cdot U$. Try to use the method based on matrix multiplication which was shown earlier.)

The determinant equals to the product of the diagonal entries of the matrix U:

$$\det(A) = \det(U) = 2 \cdot \frac{3}{2} \cdot \frac{4}{3} = 4.$$

8. Determine the LDL^* and Cholesky factorization of the following self-adjoint matrix:

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

Solution: The LU decomposition is: $A = LU$, where

$$L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ 0 & \frac{2}{3} & 1 \end{pmatrix}, \qquad U = \begin{pmatrix} 2 & 1 & 0 \\ 0 & \frac{3}{2} & 1 \\ 0 & 0 & \frac{4}{3} \end{pmatrix}$$

(check it!). Hence $A = LDL^*$, where

$$D = \begin{pmatrix} 2 & 0 & 0 \\ 0 & \frac{3}{2} & 0 \\ 0 & 0 & \frac{4}{3} \end{pmatrix}$$

Therefore

$$\sqrt{D} = \begin{pmatrix} \sqrt{2} & 0 & 0 \\ 0 & \sqrt{\frac{3}{2}} & 0 \\ 0 & 0 & \frac{2}{\sqrt{3}} \end{pmatrix}$$

Thus, the Cholesky factorization has the form $A = \tilde{L}\tilde{L}^*$, where:

$$\tilde{L} = L\sqrt{D} = \begin{pmatrix} \sqrt{2} & 0 & 0 \\ \frac{\sqrt{2}}{2} & \sqrt{\frac{3}{2}} & 0 \\ 0 & \sqrt{\frac{2}{3}} & \frac{2}{\sqrt{3}} \end{pmatrix}$$

(Hint: Perform the computation by applying the recursive formulations as well.)

3 Iterative solution of linear systems

Though in principle, a direct solution method results in the exact solution (apart from computational errors due to the finite accurary of the computers), in a lot of cases, direct methods cannot be applied. For instance, if the number of unknowns is extremely great, but the matrix is sparse, then the Gaussian elimination results in the filling of the matrix, which causes serious numerical problems.

Iterative methods basically differ from the direct methods. Here we always have an *approximate solution* only, and the aim of the method is to improve the approximation. Thus, every iterative method generates a vector sequence $(x^{(n)}) \subset \mathbf{R}^N$ which is expected to converge to the exact solution $x^* \in \mathbf{R}^N$ of the original problem. The analysis of an iterative method should aim not only the existence of convergence but also the speed of convergence.

3.1 The fixed point theorem

First, one of the most important iterations of the numerical analysis is presented. The following theorem is valid in much more general context, however, we need only a restricted version as follows.

Theorem (fixed point theorem of Banach): Let $F : \mathbf{R}^N \to \mathbf{R}^N$ be a *contractive mapping* (or contraction), i.e. assume that there exists a constant $0 \leq q < 1$ such that for every $x, y \in \mathbf{R}^N$, the inequality

$$||F(x) - F(y)|| \leq q \cdot ||x - y||$$

holds (with an arbitrary vector norm $||.||$). Then the mapping F has exactly one fixed point $x^* \in \mathbf{R}^N$, for which $F(x^*) = x^*$. Moreover, for any starting approximation $x^{(0)} \in \mathbf{R}^N$, the recursively defined sequence

$$\boxed{x^{(n+1)} := F(x^{(n)}) \qquad (n = 0, 1, 2, ...)}$$

converges to the fixed point x^*.
Proof: First we note that the contractivity implies the continuity of F. Indeed, if $x^{(n)} \to x$ a convergent sequence, then

$$||F(x^{(n)}) - F(x)|| \to q \cdot ||x^{(n)} - x|| \to 0$$

i.e. $F(x^{(n)}) \to F(x)$ also holds.

Next, let us estimate the distance of two consecutive element of the recursively defined sequence $(x^{(n)})$:

$$||x^{(n)} - x^{(n+1)}|| = ||F(x^{(n-1)}) - F(x^{(n)})|| \leq q \cdot ||x^{(n-1)} - x^{(n)}||$$

Similar estimates can be derived for the lower indices:

$$||x^{(n)} - x^{(n+1)}|| \leq q \cdot ||x^{(n-1)} - x^{(n)}|| \leq q^2 \cdot ||x^{(n-2)} - x^{(n-1)}|| \leq ...$$

$$\leq q^n \cdot ||x^{(0)} - x^{(1)}||$$

Now we estimate the distance of two, net necessarily consecutive elements. Let n be an arbitrary index and $m \geq n$. Then:

$$||x^{(n)} - x^{(m)}|| = ||x^{(n)} - x^{(n+1)} + x^{(n+1)} - x^{(n+2)} + x^{(n+2)} - ... - x^{(m)}|| \leq$$

$$\leq ||x^{(n)} - x^{(n+1)}|| + ||x^{(n+1)} - x^{(n+2)}|| + ... + ||x^{(m-1)} - x^{(m)}|| \leq$$

$$\leq (q^n + q^{n+1} + ... + q^{m-1}) \cdot ||x^{(0)} - x^{(1)}||$$

where we have utilized the previous estimation. Summing up the geometric series of the powers of q, we have:

$$q^n + q^{n+1} + ... + q^{m-1} \leq \frac{q^n}{1-q}$$

We have obtained that:

$$||x^{(n)} - x^{(m)}|| \leq \frac{q^n}{1-q} \cdot ||x^{(0)} - x^{(1)}|| \to 0$$

as $n \to \infty$. This means that the sequence $(x^{(n)})$ is a *Cauchy sequence* in $\mathbf{R}^N$. Due to an essential theorem of the elementary analysis, the sequence is also *convergent*: $x^{(n)} \to x^* \in \mathbf{R}^N$. Combined this with the continuity of F, the recursive definition $x^{(n+1)} := F(x^{(n)})$ implies that $x^* = F(x^*)$, i.e. x^* is a fixed point of F. Finally, if there were two fixed points x^* and x^{**}, then

$$||x^* - x^{**}|| = ||F(x^*) - F(x^{**})|| \leq q \cdot ||x^* - x^{**}||$$

Since $q < 1$, this implies that $x^* = x^{**}$. Theorem is proved.

For the speed of convergence, one can derive an immediate a priori estimate. From the recursive definition:

$$||x^{(n)} - x^*|| = ||F(x^{(n-1)}) - F(x^*)|| \leq q \cdot ||x^{(n-1)} - x^*|| \leq ...$$

whence

$$\boxed{||x^{(n)} - x^*|| \leq q^n \cdot ||x^{(0)} - x^*||}$$

This means that the speed of convergence is the speed of convergence of a geometrical sequence q^n. That is, if q is close to 1, then the convergence is slow.

3.2 Fixed point iteration for linear systems

In the iterative methods of linear system of equations, the fixed point theorem can be applied in the following way. Suppose that the system

$$Ax = b$$

is to be solved. By equivalent transformations, this system can be (not uniquely) rewritten as

$$x = Bx + f,$$

which is a fixed point problem. The matrix B is often called *transition matrix*. The role of the function F is played by the function $x \to Bx + f$. This immediately implies the following corollary:

Theorem: If $||B|| < 1$ (with respect to any matrix norm induced by vector norm), then the system $x = Bx + f$ has a unique solution x^*, and for any starting approximation $x^{(0)} \in \mathbf{R}^N$, the recursively defined sequence

$$\boxed{x^{(n+1)} := Bx^{(n)} + f \qquad (n = 0, 1, 2, ...)}$$

converges to x^*. Moreover, the error of the nth approximation can be estimated by:

$$\boxed{||x^{(n)} - x^*|| \leq ||B||^n \cdot ||x^{(0)} - x^*||.}$$

In practice, it is often more convenient to use an *a posteriori error estimate* constructed in the following way. Starting from the recursive definition

$$x^{(n+1)} := Bx^{(n)} + f$$

and the exact solution of the fixed point problem

$$x^* = Bx^* + f$$

we have (by subtracting the two equalities):

$$x^{(n+1)} - x^* = Bx^{(n)} + Bx^*$$

Subtracting $B(x^{(n+1)} - x^*)$ from both sides:

$$(I - B)(x^{(n+1)} - x^*) = -B(x^{(n+1)} - x^{(n)}),$$

whence

$$x^{(n+1)} - x^* = -(I - B)^{-1}B(x^{(n+1)} - x^{(n)})$$

Taking the norm of both sides:

$$||x^{(n+1)} - x^*|| \leq ||(I - B)^{-1}|| \cdot ||B|| \cdot ||x^{(n+1)} - x^{(n)}||$$

But since $||B|| < 1$, the inverse $(I-B)^{-1}$ can be expressed in a matrix series (*Neumann series*)

$$(I - B)^{-1} = I + B + B^2 + B^3 + ...$$

(prove it!), we have the estimation

$$||(I - B)^{-1}|| \leq 1 + ||B|| + ||B||^2 + ||B||^3 + ... = \frac{1}{1 - ||B||}$$

from which we obtain the a posteriori estimation:

$$\boxed{||x^{(n+1)} - x^*|| \leq \frac{||B||}{1 - ||B||} \cdot ||x^{(n+1)} - x^{(n)}||}$$

This estimation might be useful when calculating the newer and newer approximations of the fixed point provided that $||B||$ is small enough.

Example: Consider the fixed point problem $x = Bx + f$ with

$$B := \begin{pmatrix} 0.00 & 0.50 & 0.25 \\ 0.50 & 0.00 & 0.25 \\ 0.50 & 0.25 & 0.00 \end{pmatrix}, \quad f := \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}.$$

The column sum norm of B equals to 1, but the row sum norm is 0.75. Thus, the fixed point theorem is applicable, and the fixed point iteration sequence converges to the (unique) fixed point, which is: $x^* = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$) (check it!)

Note that the condition of the above theorem (based on the fixed point theorem) is sometimes too strong. It should be pointed out that the necessary and sufficient condition of the convergence is as follows (without proof):

Theorem: Consider the fixed point problem $x = Bx+f$, and the recursively defined sequence:

$$x^{(n+1)} := Bx^{(n)} + f \qquad (n = 0, 1, 2, ...)$$

Starting from any vector $x^{(0)}$, this sequence converges to the fixed point x^* if and only if $\rho(B) < 1$, i.e. the absolute values of all eigenvalues of B are less than 1.

In practice, however, the application of this thoerem might generate serious computational problems as illustrated through the following example:

Example: Consider the fixed point problem $x = Bx + f$ with $f = \mathbf{0}$ and

$$B := \begin{pmatrix} 0.5 & 100 & 0 & 0 \\ 0 & 0.5 & 100 & 0 \\ 0 & 0 & 0.5 & 100 \\ 0 & 0 & 0 & 0.5 \end{pmatrix}$$

Obviously, the exact solutions is: $x^* = \mathbf{0}$. Note that all eigenvalues of B are equal to 0.5, thus, the above theorem is applicable.

Let the starting approximation be: $x^{(0)} := \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$.

Then the first 7 approximate solutions are: $x^{(1)} = \begin{pmatrix} 0 \\ 0 \\ 100.0 \\ 0.5 \end{pmatrix}$,

$$x^{(2)} = \begin{pmatrix} 0 \\ 10000 \\ 100 \\ 0 \end{pmatrix}, \quad x^{(3)} = \begin{pmatrix} 1000000 \\ 15000 \\ 100 \\ 0 \end{pmatrix}, \quad x^{(4)} = \begin{pmatrix} 2000000 \\ 15000 \\ 100 \\ 0 \end{pmatrix},$$

$$x^{(5)} = \begin{pmatrix} 2500000 \\ 12500 \\ 0 \\ 0 \end{pmatrix}, \quad x^{(6)} = \begin{pmatrix} 2500000 \\ 9400 \\ 0 \\ 0 \end{pmatrix}, \quad x^{(7)} = \begin{pmatrix} 2187500 \\ 6600 \\ 0 \\ 0 \end{pmatrix}$$

For larger matrices which have the same structure, this necessarily causes overflow in spite of the fact that the sequence converges to $\mathbf{0}$.

3.3 The Richardson iteration

Let $A \in \mathbf{M}_{N\times N}$ be a regular matrix. Let $b \in \mathbf{R}^N$ be a given vector. The linear system of equations:

$$Ax = b$$

is equivalent to the system of equations

$$x = x - \omega \cdot (Ax - b) = (I - \omega A)x + \omega b$$

(where $\omega > 0$ is a temporarily arbitrary parameter).

Now the original problem has been deduced to a fixed point problem with the transition matrix $B := (I - \omega A)$. The corresponding fixed point iteration (*Richardson iteration)* is as follows:

$$\boxed{x^{(n+1)} := (I - \omega A)x^{(n)} + \omega b \quad (n = 0, 1, 2, ...)}$$

We have the following convergence theorem:

Theorem: If $A \in \mathbf{M}_{N\times N}$ is a **self-adjoint and positive definite** matrix, then for each parameter $0 < \omega < \frac{2}{||A||}$, (where $||A||$ is an arbitrary matrix norm induced by a vector norm):

$$\rho(I - \omega A) < 1,$$

consequently, the Richardson iteration is convergent.
Proof: The transition matrix B is obviously self-adjoint. If the eigenvalues of A are $\lambda_j > 0$ $(j = 1, 2, ..., N)$, then the eigenvalues of B are of the form $(1 - \omega \cdot \lambda_j)$. Since A is positive definite, therefore $(1 - \omega \cdot \lambda_j)$ is always less that 1. On the other hand, since $\rho(A) \leq ||A||$, therefore $\omega < \frac{2}{\rho(A)}$, which implies that $\omega \cdot \lambda_j < \frac{2\lambda_j}{\rho(A)} \leq 2$, consequently $1 - \omega\lambda_j > -1$. That is:

$$||B|| = \rho(B) < 1,$$

and the iteration is convergent.

Optimal choice of the parameter. Assume for simplicity that the eigenvalues of A are indexed in such a way that

$$0 < \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_N$$

Since B is also self-adjoints and the eigenvalues of B are $(1 - \omega \cdot \lambda_j)$ $(j = 1, 2, ..., N)$, therefore

$$||B|| = \max(|1 - \omega \cdot \lambda_1|, |1 - \omega \cdot \lambda_N|)$$

The smaller the norm of B, the faster the convergence. In the optimal case (when $||B||$ is as small as possible):

$$1 - \omega \cdot \lambda_1 = -(1 - \omega \cdot \lambda_N),$$

whence we obtain the optimal value of the parameter ω:

$$\boxed{\omega_{opt} = \frac{2}{\lambda_1 + \lambda_N}}$$

This shows that in order to optimalize the value of the parameter, on needs some information about the eigenvalues of A.

Using this optimal value, the norm of the transition matrix:

$$||B|| = 1 - \frac{2\lambda_1}{\lambda_1 + \lambda_N} = \frac{\lambda_N - \lambda_1}{\lambda_N + \lambda_1} = \frac{\text{cond}(A) - 1}{\text{cond}(A) + 1},$$

since $\text{cond}(A) = \frac{\lambda_N}{\lambda_1}$. This shows that for ill-conditioned matrices, the convergence may be slow.

3.4 The Jacobi iteration

Let $A \in \mathbf{M}_{N \times N}$ be a regular matrix. Let $b \in \mathbf{R}^N$ be a given vector. Consider the linear system of equations:

$$Ax = b$$

Let us decompose the original matrix A into the sum of three matrices: $A = L + D + U$, where L is a strictly lower triangular, D is a diagonal and U is a strictly upper diagonal matrix. Suppose that the diagonal elements of A contain no zeros. Then the original problem has the form:

$$(L + D + U)x = b$$

which is equivalent to:

$$Dx = -(L+U)x + b,$$

whence we obtain a fixed point form:

$$x = -D^{-1}(L+U)x + D^{-1}b$$

The transition matrix is $B = -D^{-1}(L+U)$. This generates the fixed point iteration (the *Jacobi iteration*):

$$\boxed{x^{(n+1)} := -D^{-1}(L+U)x^{(n)} + D^{-1}b}$$

From computational point of view, it is more convenient to express this formula in a componentwise form:

$$x_k^{(n+1)} := \frac{1}{A_{kk}} \cdot \left(-\sum_{j=1}^{k-1} A_{kj}x_j^{(n)} - \sum_{j=k+1}^{N} A_{kj}x_j^{(n)} + b_k \right)$$

($k = 1, 2, ..., N$) Practically this means that from the kth equation, we express the kth unknown with the help of the other unknowns and thus we update the actual approximations.

Theorem: If the matrix A is *diagonally dominant* i.e.

$$|A_{kk}| > \sum_{j \neq k} |A_{kj}|$$

holds for $k = 1, 2, ..., N$, then the Jacobi iteration is convergent.
Proof: Compute the norm of the transition matrix B induced by the maximum norm. Since for every $x \in \mathbf{R}^N$:

$$(Bx)_k = \frac{1}{A_{kk}} \cdot \left(-\sum_{j=1}^{k-1} A_{kj}x_j - \sum_{j=k+1}^{N} A_{kj}x_j \right) = -\frac{1}{A_{kk}} \cdot \sum_{j \neq k} A_{kj}x_j,$$

therefore (with respect to the maximum norm):

$$||Bx|| = \max_k \frac{1}{|A_{kk}|} \cdot \left(\sum_{j \neq k} |A_{kj}| \cdot |x_j| \right) \leq \left[\max_k \frac{1}{|A_{kk}|} \cdot \left(\sum_{j \neq k} |A_{kj}| \right) \right] \cdot ||x||,$$

which implies that $||B|| < 1$, therefore the iteration is convergent.

3.5 The Seidel iteration

Let $A \in \mathbf{M}_{N\times N}$ be a regular matrix again. Let $b \in \mathbf{R}^N$ be a given vector. Consider the linear system of equations:

$$Ax = b$$

Let us decompose the original matrix A into the sum of three matrices: $A = L + D + U$ as earlier, where L is a strictly lower triangular, D is a diagonal and U is a strictly upper diagonal matrix. Then the original problem has the form:

$$(L + D + U)x = b$$

which is equivalent to:

$$(L + D)x = -Ux + b,$$

whence we obtain a fixed point form:

$$x = -(L + D)^{-1}Ux + (L + D)^{-1}b$$

The transition matrix is $B = -(L + D)^{-1}U$. This generates the fixed point iteration (the *Seidel iteration*):

$$\boxed{x^{(n+1)} := -(L + D)^{-1}Ux^{(n)} + (L + D)^{-1}b}$$

From computational point of view, it is much more convenient to express this formula in a componentwise form. This is very similar to the Jacobi iteration, with the only difference that any component of the updated approximation is immediately utilized in the expressions of the latter components:

$$x_k^{(n+1)} := \frac{1}{A_{kk}} \cdot \left(-\sum_{j=1}^{k-1} A_{kj}x_j^{(n+1)} - \sum_{j=k+1}^{N} A_{kj}x_j^{(n)} + b_k \right)$$

$(k = 1, 2, ..., N)$ Practically this means that from the kth equation, we express the kth unknown with the help of the other unknowns and thus we update the actual approximations; any component is immediately utilized when computing the remaining components of the approximation.

The following convergence result is presented without proof:

Theorem: If the matrix A is diagonally dominant or A is self-adjoint and positive definite, then the Seidel iteration is convergent.

3.6 Relaxation principle

Consider a fixed point problem

$$x = Bx + f$$

and the corresponding iteration:

$$x^{(n+1)} := Bx^{(n)} + f,$$

which is equivalent to:

$$x^{(n+1)} := x^{(n)} - x^{(n)} + Bx^{(n)} + f = x^{(n)} + (B - I)x^{(n)} + f =: x^{(n)} + w^{(n)},$$

where the correcting term $w^{(n)}$ is defined by $w^{(n)} := (B - I)x^{(n)} + f$

The main idea of the relaxation methods is that the correction is performed by adding the correction term multiplied by a (small) scalar ω:

$$x^{(n+1)} := x^{(n)} + \omega \cdot w^{(n)} = x^{(n)} + \omega \cdot ((B - I)x^{(n)} + f)$$

i.e.

$$\boxed{x^{(n+1)} := ((1 - \omega) \cdot I + \omega \cdot B)x^{(n)} + \omega \cdot f}$$

This is another fixed point iteration with the transition matrix

$$(1 - \omega) \cdot I + \omega \cdot B.$$

Sometimes it exhibits more advantageous properties than the original iteration.

The relaxation is called *underrelaxation*, if $0 < \omega \leq 1$ and *overrelaxation*, if $\omega > 1$.

Proposition: If the original fixed point iteration is convergent (i.e. $\rho(B) < 1$) and $0 < \omega \leq 1$, then the correspondig (under)relaxation is also convergent.
Proof: Let λ_j be an eigenvalue of the original transition matrix B. Then the eigenvalues of the underrelaxed transition matrix are of the form $1-\omega+\omega\lambda_j$. The square of their absolute value can be estimeted by:

$$|(1 - \omega) + \omega\lambda_j|^2 \leq (|1 - \omega| + |\omega\lambda_j|)^2 < (1 - \omega + \omega)^2 = 1,$$

that is, the relaxed iteration is convergent.

As a consequence, the underrelaxed Jacobi iteration is convergent, provided that the original matrix A is diagonally dominant.

3.7 Variational methods

Let $A \in \mathbf{M}_{N\times N}$ be a self-adjoint, positive definite matrix. Let $b \in \mathbf{R}^N$ be a given vector. Consider the linear system of equations:

$$Ax = b$$

and denote by x^* its (unique) solution.
Introduce the following definitions:

- $\langle x, y\rangle_A := \langle Ax, y\rangle$ (*energetic inner product*)
- $||x||_A := \sqrt{\langle x, x\rangle}$ (*energetic norm*)
- $F(x) := \langle Ax, x\rangle - 2\langle x, b\rangle$ (*energetic functional*)

Denote by $0 < \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_N$ the eigenvalues of the matrix A, and let $s_1, ..., s_N$ be the corresponding orthonormal eigenvectors. Then for every vector x can be expressed in the form

$$x = \sum_{j=1}^{N} \alpha_j s_j,$$

whence

$$Ax = \sum_{j=1}^{N} \alpha_j \lambda_j s_j$$

This implies that:

$$\lambda_1 \cdot ||x||^2 \leq ||x||_A^2 \leq \lambda_N \cdot ||x||^2,$$

which gives us a relationship between the Euclidean norm and the energetic norm.

It is easy to check that the energetic inner product is indeed an inner product in $\mathbf{R}^N$, and the energetic norm is the norm induced by the energetic inner product. By definition:

$$F(x) = ||x||_A^2 - 2\langle x, b\rangle$$

However, since $Ax^* = b$, therefore

$$F(x) = ||x||_A^2 - 2\langle x, Ax^*\rangle = ||x||_A^2 - 2\langle x, x^*\rangle_A = ||x - x^*||_A^2 - ||x^*||_A^2$$

This implies the following important observation:

Proposition: The energetic functional F has exactly one location where it takes its minimal value and this is the (unique) solution x^* of the original problem $Ax = b$.

This means that the original problem is converted to the solution of an extremal value problem. The approximate solution techniques of the latter problem are called *variational methods*.

The simplest variational method is the *minimization along a direction*. Let $e \in \mathbf{R}^N$ be a given directional vector. Let x be an approximation of the minimizing vector. The minimization along the direction e means the minimization along a line that passes x and has the direction vector e, i.e. the minimization of the univariate function:

$$f(t) := F(x + t \cdot e)$$

Obviously:

$$f(t) = ||x+te-x^*||_A^2 - ||x^*||_A^2 = ||x-x^*||_A^2 + 2t\langle e, x-x^*\rangle_A + t^2||e||_A^2 - ||x^*||_A^2$$

Differentiating f with respect to t, we obtain:

$$f'(t) = 2\langle e, x - x^*\rangle_A + 2t||e||_A^2$$

At the location, where f takes its extremal value, the derivative of f is zero. Hence:

$$t = -\frac{\langle e, x - x^*\rangle_A}{||e||_A^2} = -\frac{\langle Ax - b, e\rangle}{\langle Ae, e\rangle} = -\frac{\langle r, e\rangle}{\langle Ae, e\rangle}$$

This means that x is an approximate location of minimum, then the improved location can be expressed as:

$$\boxed{\tilde{x} := x - \frac{\langle r, e\rangle}{\langle Ae, e\rangle} \cdot e}$$

where $r := Ax - b$, the residual vector. The improved location $\tilde{x}$ is not worse that the previous location x in the sense that the energetic functional F takes a value here wich is not greater than at the location x.

The problem is now, how to properly choose the direction vector e.

It can be easily seen that if the directions are defined to be the standard basis vectors in a circular way, then we obtain the familiar Seidel iteration.

But there is a better choice, and this is the method of 'steepest descent' or the 'gradient method', which will be outlined in the next subsection.

3.8 The gradient method

This method is a special minimization technique, when the energetic function F is always minimized in the direction of the greatest change, i.e. in the direction of the gradient vector. Standard calculations show that for every $h \in \mathbf{R}^N$:

$$F(x+h) = \langle Ax + Ah, x + h\rangle - 2\langle x + h, b\rangle =$$

$$= \langle Ax, x\rangle + 2\langle Ax, h\rangle + \langle Ah, h\rangle - 2\langle x, h\rangle - 2\langle h, b\rangle =$$

$$= F(x) + 2\langle Ax - b, h\rangle + \langle Ah, h\rangle$$

The last term in the right-hand side is of order $\mathcal{O}(||h||^2)$, which implies that the gradient vector of F at the location x is as follows:

$$\operatorname{grad} F(x) = 2 \cdot (Ax - b).$$

Now let x be an approximate solution and define the direcion vector e as well as the residual vector r by $e := r := Ax - b$. Then, according to the previous subsection, the improved solution $\tilde{x}$ is expressed as:

$$\tilde{x} = x - \frac{||r||^2}{\langle Ar, r\rangle} \cdot r$$

The procedure is repeated in each iteration step. This results in the following iteration (the gradient method). Starting from an arbitraty vector $x^{(0)}$, consider the recursively defined vector sequence:

$$x^{(n+1)} := x^{(n)} - \frac{||r^{(n)}||^2}{\langle Ar^{(n)}, r^{(n)}\rangle} \cdot r^{(n)}, \quad \text{where } r^{(n)} := Ax^{(n)} - b.$$

The error estimation is not easy. Here we show a theorem without proof.

Theorem: The error of the nth approximate solution with respect to the energetic norm can be estimated as follows:

$$||x^{(n)} - x^*||_A^2 \leq \left(1 - \frac{\lambda_1}{\lambda_N}\right)^n \cdot ||x^{(0)} - x^*||_A^2$$

With respect to the Euclidean norm, a similar estimation can be deduced:

$$||x^{(n)} - x^*||^2 \leq \frac{\lambda_N}{\lambda_1} \cdot \left(1 - \frac{\lambda_1}{\lambda_N}\right)^n \cdot ||x^{(0)} - x^*||^2$$

Note that the speed of convergence is more or less the same than that of the Richardson iteration with optimal parameter. However, in the case of the gradient method, the eigenvalues occur only in the error estimation formula and not in the definition of the algorithm. The gradient method can be built up and applied without any a priori piece of information about the eigenvalues.

From the error estimations, it can be seen that the speed of convergence is that of a geometrical sequence with quotient

$$1 - \frac{\lambda_1}{\lambda_N} = 1 - \frac{1}{\mathrm{cond}(A)}.$$

That is, if A is ill-conditioned, the convergence may be (extremely) slow. Generally, the first one or two iteration steps make a significant reduction of the error of the approximate solution, and the further steps do not (try it!).

3.9 The Krylov subspace methods - an outlook

The Krylov subspace methods are known to be the most efficient solvers of linear systems of equations. However, the structure of these methods is a bit more complicated than that of the above classical methods (except for the conjugate gradient method, which is in fact a special kind of Krylov subspace methods). The Krylov subspace methods are special *projection type methods*, which are briefly outlined as follows.

Consider a linear system of equations of the form:

$$Ax = b,$$

where the matrix $A \in \mathbf{M}_{N\times N}$ is assumed to be regular. Denote by x^* the (unique) solution of this system.

3.9.1 Projection methods

The projection methods convert the above problem to a sequence of another, smaller problems. Define two n-dimensional subspaces $\mathcal{S}_n$ (*search space*) and $\mathcal{C}_n$ (*constraints space*). The approximate solution x_n is defined in $\mathcal{S}_n$ in such a way that the residual $r_n := b - Ax_n$ is orthogonal to the subspace $\mathcal{C}_n$.

Let $s_1, s_2, ..., s_n \in \mathcal{S}_n$ and $c_1, c_2, ..., c_n \in \mathcal{C}_n$ be (not necessarily orthogonal) bases in $\mathcal{S}_n$ and in $\mathcal{C}_n$, respectively. Then the approximate solution x_n is sought in the form:

$$x_n := \sum_{j=1}^{n} \alpha_j s_j,$$

and the orthogonality constraints imply that the a priori unknown coefficients α_j satisfy the conditions $b - \sum_{j=1}^{n} \alpha_j A s_j \perp \mathcal{C}_n$ i.e.

$$\sum_{j=1}^{n} \alpha_j \cdot \langle A s_j, c_k \rangle = \langle b, c_k \rangle \qquad (k = 1, 2, ..., n)$$

The method is often referred to as the *method of moments*.

If $n = N$, this results in an exact method with the computational complexity $\mathcal{O}(N^3)$. If $n < N$ or $n \ll N$, this method results in a dimension reduction. However, the accuracy as well as the computational complexity of the method highly depend on the particular choices of the subspaces $\mathcal{S}_n$ and $\mathcal{C}_n$ and also on the choice of the bases in these subspaces. Here two popular special cases are outlined, where the projected problems have unique solutions and they are close to the exact solution x^* in some sense.

A special case: Galerkin's method. Suppose now that A is self-adjoint and positive definite. Define

$$\mathcal{C}_n := \mathcal{S}_n$$

Let $s_1, s_2, ..., s_n \in \mathcal{S}_n$ be a (not necessarily orthogonal) basis in the subspace $\mathcal{S}_n$, and denote by $c_k := s_k$ $(k = 1, 2, ..., n)$. The system of equations of the corresponding projection method is:

$$\sum_{j=1}^{n} \alpha_j \cdot \langle A s_j, s_k \rangle = \langle b, s_k \rangle \qquad (k = 1, 2, ..., n)$$

Having solved this linear system of equations, the approximate solution is given by:

$$x_n := \sum_{j=1}^{n} \alpha_j s_j,$$

Theorem: The error (measured in A-norm) $||x^* - x_n||_A$ is minimal among the vectors $y_n := \sum_{j=1}^{N} \alpha_j s_j \in \mathcal{S}_n$.

Proof: If $||x^* - y_n||_A$ is minimal, then y_n is the best approximation of x^* with respect to the A-norm. This implies that $(x^* - y_n)$ is A-orthogonal to $\mathcal{S}_n$, i.e.

$$\langle A(x^* - y_n), s_k \rangle = 0 \qquad (k = 1, 2, ..., n)$$

from which:

$$\langle Ax^* - Ay_n, s_k \rangle = \langle b - Ay_n, s_k \rangle = 0 \qquad (k = 1, 2, ..., n)$$

We have obtained that the coefficients α_j satisfy the same system of equations:

$$\sum_{j=1}^{n} \alpha_j \cdot \langle As_j, s_k \rangle = \langle b, s_k \rangle \qquad (k = 1, 2, ..., n)$$

which implies that, the approximate solution x_n minimizes the error with respect to the A-norm.

Another special case: the method of least squares. Suppose that A is nonsingular (but not necessarily self-adjoint). Define

$$\mathcal{C}_n := A(\mathcal{S}_n)$$

Let $s_1, s_2, ..., s_n \in \mathcal{S}_n$ be a (not necessarily orthogonal) basis in the subspace $\mathcal{S}_n$, and denote by $c_k := As_k$ $(k = 1, 2, ..., n)$. Then $c_1, c_2, ..., c_n \in \mathcal{C}_n$ form a (not necessarily orthogonal) basis in the subspace $\mathcal{C}_n$. The system of equations of the corresponding projection method is:

$$\sum_{j=1}^{n} \alpha_j \cdot \langle As_j, As_k \rangle = \langle b, As_k \rangle \qquad (k = 1, 2, ..., n)$$

Having solved this linear system of equations, the approximate solution is given by:

$$x_n := \sum_{j=1}^{n} \alpha_j s_j,$$

Theorem: The norm of the residual $||b - Ax_n||$ is minimal among the vectors $y_n := \sum_{j=1}^{N} \alpha_j s_j \in \mathcal{S}_n$.

Proof:

$$||b - Ay_n||^2 = ||Ax^* - Ay_n||^2 = ||A(x^* - y_n)||^2 = ||x^* - y_n||^2_{A^*A}$$

If $||b - Ay_n||^2$ is minimal, then y_n is the best approximation of x^* with respect to the A^*A-norm. This implies that $(x^* - y_n)$ is A^*A-orthogonal to $\mathcal{S}_n$, i.e.

$$\langle A^*A(x^* - y_n), s_k \rangle = 0 \qquad (k = 1, 2, ..., n)$$

from which:

$$\langle Ax^* - Ay_n, As_k \rangle = \langle b - Ay_n, As_k \rangle = 0 \qquad (k = 1, 2, ..., n)$$

We have obtained that the coefficients α_j satisfy the same system of equations:

$$\sum_{j=1}^{n} \alpha_j \cdot \langle As_j, As_k \rangle = \langle b, As_k \rangle \qquad (k = 1, 2, ..., n)$$

Thus, the approximate solution x_n minimizes the residual, as stated.

Note that the minimality of the residual $||b - Ax_n||$ means that the error of the approximate solution $(x^* - x_n)$ is also minimal with respect to the A^*A-norm, since

$$||b - Ax_n||^2 = ||Ax^* - Ax_n||^2 = \langle A(x^* - x_n), A(x^* - x_n) \rangle =$$

$$= \langle A^*A(x^* - x_n), (x^* - x_n) \rangle = ||x^* - x_n||^2_{A^*A}.$$

Remark: The above outlined projection methods can be written in an iterative form as follows. Consider the problem $Ax = b$, where $A \in \mathbf{M}_{N \times N}$ is regular, and denote by x^* the exact solution. Now let x_0 be a starting approximate solution (it can be defined arbitrarily, e.g. $x_0 := \mathbf{0}$). The approximate solution x_n can be written in the form $x_n = x_0 + w_n$, where the correcting term $w_n := \sum_{j=1}^{n} \alpha_j s_j \in \mathcal{S}_n$ is to be defined in such a way that the residual r_n is orthogonal to the subspace $\mathcal{C}_n$. Since

$$r_n = b - Ax_n = b - Ax_0 - Aw_n = r_0 - Aw_n,$$

therefore in order to enforce the orthogonality conditions, the modified system

$$\sum_{j=1}^{n} \alpha_j \langle As_j, c_k \rangle = \langle r_0, c_k \rangle \qquad (k = 1, 2, ..., n)$$

has to be solved, and then $x_n := x_0 + w_n$ is the improved approximation. Using exact arithmetic, the approximate solution x_n is equal to the exact solution x^* after n steps, where n is at most N. In practice, this is not the case, and it is worth restarting the projection algorithm from time to time by choosing the next starting approximation x_0 to be the just computed x_n for some n.

3.9.2 Orthogonalization techniques

The number of necessary arithmethic operations can be often reduced by orthogonalizing a basis in the subspace $\mathcal{S}_n$.

Let $c_1, c_2, ..., c_n \in \mathcal{C}_n$ be a (not necessarily orthogonal) basis in the subspace $\mathcal{C}_n$, and denote by $s_k := A^* c_k$ $(k = 1, 2, ..., n)$. Then $s_1, s_2, ..., s_n \in \mathcal{S}_n$ form a (not necessarily orthogonal) basis in the subspace $\mathcal{S}_n$. Perform a Gram-Schmidt ortogonalization procedure for the vectors s_1, s_2, ..., s_n. We obtain the orthonormal system e_1, e_2, ..., e_n in $\mathcal{S}_n$ via the recursions:

$$\tilde{e}_1 := s_1, \qquad e_1 := \frac{\tilde{e}_1}{||\tilde{e}_1||},$$

$$\tilde{e}_k := s_k - \sum_{j=1}^{k-1} \langle s_k, e_j \rangle \cdot e_j, \qquad e_k := \frac{\tilde{e}_k}{||\tilde{e}_k||}$$

for $k = 2, 3, ..., n$.

The numbers $\langle x^*, e_k \rangle$ can be computed recursively:

$$\langle x^*, e_1 \rangle = \langle x^*, \frac{A^* c_1}{||A^* c_1||} \rangle = \langle A x^*, \frac{c_1}{||A^* c_1||} \rangle = \langle b, \frac{c_1}{||A^* c_1||} \rangle,$$

and

$$\langle x^*, e_k \rangle = \langle x^*, A^* c_k \rangle - \sum_{j=1}^{k-1} \langle A^* c_k, e_j \rangle \cdot \langle x^*, e_j \rangle$$

for $k = 2, 3, ..., n$.

Having computed the numbers $\langle x^*, e_k \rangle$, the approximate solution is defined in terms of finite Fourier series:

$$x_n := \sum_{j=1}^{n} \langle x^*, e_j \rangle \cdot e_j$$

Remark: The above definition results in a special projection method. Indeed, x_n is the orthogonal projection of x^* to the subspace $\mathcal{S}_n$, therefore the vector $(x^* - x_n)$ is orthogonal to the subspace $\mathcal{S}_n$. This means that $(x^* - x_n) \perp A^* c_k$ for $k = 1, 2, ..., n$. Hence:

$$\langle x^* - x_n, A^* c_k \rangle = \langle A x^* - A x_n, c_k \rangle = \langle r_n, c_k \rangle = 0$$

i.e. $r_n \perp \mathcal{C}_n$.

Now suppose that A is a self-adjoint, positive definite matrix. If – by orthogonalization or not – an A-orthonormal system e_1, e_2, ..., e_N is available, and $\mathcal{S}_n = \text{span}(e_1, ..., e_n)$, moreover, $\mathcal{C}_n = \mathcal{S}_n$, then the situation becomes simpler. In this case:

$$x^* = \sum_{j=1}^{N} \langle x^*, e_j \rangle_A \cdot e_j,$$

and

$$x_n = \sum_{j=1}^{n} \langle x^*, e_j \rangle_A \cdot e_j = \sum_{j=1}^{n} \langle b, e_j \rangle \cdot e_j$$

The residual $r_n = b - Ax_n = A(x^* - x_n)$ is orthogonal to the subspace $\mathcal{C}_n$, since:

$$\langle r_n, e_k \rangle = \langle A(x^* - x_n), e_k \rangle = \langle x^* - x_n, e_k \rangle_A = \sum_{j=n+1}^{N} \langle x^*, e_j \rangle_A \cdot \langle e_j, e_k \rangle_A = 0$$

That is, the resulting method is a special projection method.

Similarly, supposing that A is regular only (not necessarily self-adjoint and positive definite), let an A^*A-orthonormal system e_1, e_2, ..., e_N be available. Define $\mathcal{S}_n := \text{span}(e_1, ..., e_n)$, and $\mathcal{C}_n := \text{span}(Ae_1, ..., Ae_n)$. Then:

$$x^* = \sum_{j=1}^{N} \langle x^*, e_j \rangle_{A^*A} \cdot e_j,$$

and

$$x_n = \sum_{j=1}^{n} \langle x^*, e_j \rangle_{A^*A} \cdot e_j = \sum_{j=1}^{n} \langle b, e_j \rangle \cdot e_j$$

The residual $r_n = b - Ax_n = A(x^* - x_n)$ is orthogonal to the subspace $\mathcal{C}_n$, since:

$$\langle r_n, Ae_k \rangle = \langle A(x^* - x_n), Ae_k \rangle = \langle x^* - x_n, e_k \rangle_{A^*A} =$$

$$= \sum_{j=n+1}^{N} \langle x^*, e_j \rangle_{A^*A} \cdot \langle e_j, e_k \rangle_{A^*A} = 0$$

That is, the resulting method is also a special projection method.

3.9.3 Krylov subspaces

Let $A \in \mathbf{M}_{N\times N}$ be a regular matrix, and let $\mathbf{0} \neq v \in \mathbf{R}^N$ be a vector. The rth *Krylov subspace* generated by A and v is defined as follows:

$$\mathcal{K}_r(A, v) := \text{span}(v, Av, A^2v, ..., A^{r-1}v)$$

Obviously $\mathcal{K}_1(A, v) \subset \mathcal{K}_2(A, v) \subset \mathcal{K}_3(A, v) \subset ... \subset \mathbf{R}^N$.

The projection methods based on Krylov subspaces are called *Krylov subspace methods*.

Remark: The vectors v, Av, ..., $A^N v$ are clearly linearly dependent, therefore there exists a nontrivial linear combination of these vectors which equals to zero:

$$\alpha_0 v + \alpha_1 Av + ... + \alpha_N A^N v = \mathbf{0}$$

Let $k \geq 0$ be the minimal index for which $\alpha_k \neq 0$. Then:

$$\alpha_k A^k v + ... + \alpha_N A^N v = \mathbf{0},$$

whence

$$A^k v = -\frac{1}{\alpha_k}\left(\alpha_{k+1}A^{k+1}v + ... + \alpha_N A^N v\right),$$

which implies that

$$A^{-1}v = -\frac{1}{\alpha_k}\left(\alpha_{k+1}v + \alpha_{k+2}Av... + \alpha_N A^{N-k-1}v\right),$$

This means that $A^{-1}v \in \mathcal{K}_N(A, v)$, i.e. the solution of the system $Ax = v$ can be expressed as a linear combination of the vectors $A^j v$ ($j = 0, 1, ..., N-1$). *If the matrix A is sparse*, then the calculation of the vectors $A^j v$ is cheap (from computational point of view). This observation anticipates the computational role of the Krylov subspace methods.

Now we summarize the main properties of Krylov subspaces. The proofs are more or less easy, so they are left as exercises.

Denote by $1 \leq d \leq N$ the number for which v, Av, ..., $A^{d-1}v$ are still linearly independent but v, Av, ..., $A^d v$ are not. (In fact, $A^d v$ linearly depends on the vectors v, Av, ..., $A^{d-1}v$.) This d is called the *grade of v with respect to A*. Then

$$\mathcal{K}_1(A, v) \subset \mathcal{K}_2(A, v) \subset ... \subset \mathcal{K}_r(A, v),$$

but

$$\mathcal{K}_r(A,v) = \mathcal{K}_{r+1}(A,v) = \mathcal{K}_{r+2}(A,v) = ...$$

$\mathcal{K}_r(A,v)$ is an invariant subspace if A, i.e. $A(\mathcal{K}_r(A,v)) \subset \mathcal{K}_r(A,v)$, moreover, the equality sign is also valid, since A is regular.

The essential idea of the Krylov subspace methods is as follows. Suppose that the equation

$$Ax = b$$

is to be solved. Denote by x^* the exact solution and let x_0 be an initial approximation. Denote by $r_0 := b - Ax_0$ the residual, and look for the nth approximate solution in the form

$$x_n := x_0 + w_n,$$

where

$$w_n \in \mathcal{K}_n(A, r_0)$$

is defined in such a way that the corresponding residual $r_n := b - Ax_n$ is orthogonal to the constraints space $\mathcal{C}_n$.

In short, every Krylov subspace method is a special projection method with $\mathcal{S}_n := \mathcal{K}_n(A, r_0)$ and with some constraints space $\mathcal{C}_n$. It should be pointed out that (in exact arithmetic) the exact solution x^* is obtained after the dth projection step, where d the the grade of r_0 with respect to A.

Now we derive some expressions for the residuals as well as the errors.

Since obviously $Ax_n - b = Ax_0 - b + Aw_n$, therefore

$$r_n = r_0 - Aw_n.$$

And, since w_n has the form:

$$w_n = \alpha_0 r_0 + \alpha_1 A r_0 + ... + \alpha_{n-1} A^{n-1} r_0,$$

therefore

$$r_n = (I - \alpha_0 A - \alpha_1 A^2 - ... - \alpha_{n-1} A^n) r_0 =: p_n(A) r_0$$

where p_n is a polynomial of degree at most n and $p_n(0) = 1$.

For the error vectors $e_n := x^* - x_n$ obviously: $Ae_n = r_n$, therefore the similar equality holds:

$$e_n = p_n(A) e_0$$

The concrete form of this polynomial depends of the concrete Krylov subspace method. The above equalities make it possible to derive error estimations. Without going into details we note that in a lot of cases, it is sufficient to estimate the norm of the matrix polynomial $p_n(A)$ (taking into account that p_n is a polynomial of degree at most n, $p_n(0) = 1$ and also the minimal property of the applied projection method).

Special cases:

1. *The conjugate gradient method (CG)*: Assume that $A \in \mathbf{M}_{N\times N}$ is a self-adjoint, positive definite matrix. Let x_0 be an arbitrary starting approximate solution and $r_0 := b - Ax_0$. Define

$$\mathcal{S}_n := \mathcal{C}_n := \mathcal{K}_n(A, r_0)$$

Define the *searching directions* $p_0, p_1, ...$ as well as the approximate solutions $x_1, x_2, ...$ recursively. Let $p_0 := r_0$ and:

$$\boxed{r_n := b - Ax_n}$$

$$\boxed{x_{n+1} := x_n + \frac{\langle r_n, p_n\rangle}{\langle Ap_n, p_n\rangle} \cdot p_n}$$

$$\boxed{r_{n+1} := b - Ax_{n+1}}$$

$$\boxed{p_{n+1} := r_{n+1} - \frac{\langle Ar_{n+1}, p_n\rangle}{\langle Ap_n, p_n\rangle} \cdot p_n}$$

It can be shown that the vectors $p_0, p_1, ..., p_{n-1}$ form and A-orthogonal basis in the Krylov subspace $\mathcal{K}_n(A, r_0)$, and the method is a Krylov subspace method.

In principle, the method is exact (after at most N iteraton steps, it gives the exact solution provided that the computations are performed in exact arithmetic). In practice, however, it is worth using as an iterative method (restarting from time to time). As an iterative algorithm, the conjugate gradient method is much faster than the gradient method. It can be proved that the error after the nth iteration can be estimated as:

$$||x_n - x^*|| \leq const. \cdot \frac{\sqrt{\text{cond}(A)} - 1}{\sqrt{\text{cond}(A)} + 1}$$

where the constant is independent of n.

If the original matrix A is a sparse matrix (which is often the case when solving partial differential equations numerically), then the computational complexity of each iteration step remains low, which makes the conjugate gradient method quite economical from computational point of view.

2. *The generalized minimal residual method (GMRES)*: Assume that $A \in \mathbf{M}_{N\times N}$ is nonsingular (not necessarily self-adjoint) matrix. Let x_0 be an arbitrary starting approximate solution and $r_0 := b - Ax_0$. Define

$$\mathcal{S}_n := \mathcal{K}_n(A, r_0), \qquad \mathcal{C}_n := A(\mathcal{S}_n)$$

   - Perform a Gram-Schmidt orthogonalization for the Krylov basis r_0, Ar_0, ..., $A^{n-1}r_0$. This process results in the orthonormal basis v_1, v_2, ..., v_n. Define the matrix formed by the vectors v_j as column vectors

$$V_n := (\ v_1 \mid v_2 \mid ... \mid v_n)$$

     Then the linear combinations $\alpha_1 v_1 + \alpha_2 v_2 + ...\alpha_n v_n$ can be expressed in the form $V_n\alpha$ where $\alpha \in \mathbf{R}^n$.

   - Compute the vector of coefficients α by minimizing the residual

$$||b - A(x_0 + V_n\alpha|| = ||r_0 - AV_n\alpha||,$$

     and define the improved approximation:

$$x_n := x_0 + V_m\alpha$$

The method is obviously a Krylov subspace method. Note that the special form of V_n makes it possible to solve the n-dimensional extremum problem arising in the second step of the algorithm by a reasonable number of arithmetic operations.

4 Calculation of eigenvalues

The eigenvalue problem is of primary importance in engineering applications. As is well known, the eigenvalues of a matrix $A \in \mathbf{M}_{N\times N}$ satisfy the *characteristic equation*:

$$\det(A - \lambda I) = 0$$

From numerical point of view, this is a really challenging task, since the algebraic equations of high degree have no exact formula to determine the roots. Sometimes there is no need to determine the exact eigenvalues, it is sufficient to roughly localize them. In some applications, it is sufficient to determine the extremal eigenvalues, but in general, there is no way to avoid the computation of the complete system of eigenvalues.

4.1 Localization of the eigenvalues by Gershgorin circles

Let $A \in \mathbf{M}_{N\times N}$ be a square matrix (with possibly complex entries). The simplest localization theorem can be expressed with the help of the matrix norm. If λ is an eigenvalue of A with eigenvector $s \neq \mathbf{0}$, then:

$$|\lambda| \cdot ||s|| = ||\lambda \cdot s|| = ||As|| \leq ||A|| \cdot ||s||,$$

which immediately implies the following statement:

Proposition: All eigenvalues of the matrix A are located in the closed disk centered at the origin with radius $||A||$, where $||.||$ is an arbitrary matrix norm induced by an arbitrary vector norm. In other words, if $\lambda \in \mathbf{C}$ is an eigenvalue of A, then

$$|\lambda| \leq ||A||,$$

or, equivalently: $\rho(A) \leq ||A||$.

Another simple but useful thorem for localization of the eigenvalues is as follows. Define the numbers $r_k := \sum_{j\neq k} |A_{kj}| \quad (k = 1, 2, ..., N)$, and consider the closed disks B_k in the complex plane centered at A_{kk} with radius r_k (*Gershgorin disks*).

Theorem (Gershgorin): All eigenvalues of the matrix A are located in the union of the Gershgorin disks.
Proof: Let $\lambda \in \mathbf{C}$ be an eigenvalue of A and denote by s the corresponding

eigenvector. Then $As = \lambda \cdot s$. Let k be the index for which $|s_k| = ||s||_{\max}$. Then:

$$\lambda \cdot s_k = \sum_{j=1}^{N} A_{kj} s_j = A_{kk} s_k + \sum_{j \neq k} A_{kj} s_j,$$

whence

$$(\lambda - A_{kk}) \cdot s_k = \sum_{j \neq k} A_{kj} s_j$$

This implies that

$$|\lambda - A_{kk}| \cdot ||s||_{\max} \leq \sum_{j \neq k} |A_{kj}| \cdot |s_j| \leq \sum_{j \neq k} |A_{kj}| \cdot ||s||_{\max}$$

from which the statement follows:

$$|\lambda - A_{kk}| \leq \sum_{j \neq k} |A_{kj}| = r_k.$$

4.2 Determination of the eigenvalue with maximal absolute value. The power iteration

Let $A \in \mathbf{M}_{N \times N}$ be a *normal matrix*. Here complex matrix entries (as well as vector components) are allowed. Recall that the vector space $\mathbf{C}^N$ has an orthonormal basis formed by the eigenvectors of A. It should also be pointed out that the inner product in $\mathbf{C}^N$ is defined with a slight modification compared with the space $\mathbf{R}^N$, namely, if $x := (x_1, x_2, ..., x_N)$, $y := (y_1, y_2, ..., y_N) \in \mathbf{C}^N$ are arbitrary vectors (with complex components), then their inner product is defined by

$$\langle x, y \rangle := \sum_{j=1}^{N} x_j \overline{y}_j = x_1 \overline{y}_1 + ... + x_N \overline{y}_N.$$

where the overbar denotes the complex conjugate. As a consequence, $\langle x, y \rangle = \overline{\langle y, x \rangle}$, and $\langle x, \alpha y \rangle = \overline{\alpha} \cdot \langle x, y \rangle$.

Assume that the eigenvalues of A are indexed as follows: $|\lambda_1| \leq |\lambda_2| \leq ... \leq |\lambda_{N-1}| < |\lambda_N|$ (which means that the dominant eigenvalue λ_N is a *simple eigenvalue*), and denote by $s_1, s_2, ..., s_N$ the corresponding orthonormal eigenvectors.

The power iteration: Let $x_0 \in \mathbf{R}^N$ be a vector such that $\langle x_0, s_N \rangle \neq 0$, and define:

$$\boxed{x_{n+1} := Ax_n \qquad (n = 0, 1, 2, ...)}$$

Then the sequence of the *Rayleigh quotients* converges to λ_N:

$$\boxed{\frac{\langle Ax_n, x_n\rangle}{||x_n||^2} = \frac{\langle x_{n+1}, x_n\rangle}{||x_n||^2} \to \lambda_N \quad (n \to \infty)}$$

Proof: It is easy to verify that $x_n = A^n x_0$. Let us write the vector x_0 in the basis formed by the orthonormal eigenvectors of A:

$$x_0 = \sum_{j=1}^{N} \alpha_j s_j,$$

where $\alpha_N \neq 0$ due to the assumptions. Then:

$$x_n = A^n x_0 = \sum_{j=1}^{N} \lambda_j^n \alpha_j s_j$$

The terms appearing in the expression of the Rayleigh quotient can be determined by straightforward calculations, exploiting the fact that the eigenvectors form an orthonormal system. We obtain:

$$||x_n||^2 = \sum_{j=1}^{N} |\lambda_j|^{2n} \cdot |\alpha_j|^2 = |\lambda_N|^{2n} \cdot |\alpha_N|^2 + \sum_{j=1}^{N-1} |\lambda_j|^{2n} \cdot |\alpha_j|^2$$

$$\langle Ax_n, x_n\rangle = \sum_{j=1}^{N} \lambda_j \cdot |\lambda_j|^{2n} \cdot |\alpha_j|^2 = \lambda_N \cdot |\lambda_N|^{2n} \cdot |\alpha_N|^2 + \sum_{j=1}^{N-1} \lambda_j \cdot |\lambda_j|^{2n} \cdot |\alpha_j|^2$$

The Rayleigh quotient:

$$\frac{\langle Ax_n, x_n\rangle}{||x_n||^2} = \frac{\lambda_N \cdot |\lambda_N|^{2n} \cdot |\alpha_N|^2 + \sum_{j=1}^{N-1} \lambda_j \cdot |\lambda_j|^{2n} \cdot |\alpha_j|^2}{|\lambda_N|^{2n} \cdot |\alpha_N|^2 + \sum_{j=1}^{N-1} |\lambda_j|^{2n} \cdot |\alpha_j|^2},$$

whence:

$$\frac{\langle Ax_n, x_n\rangle}{||x_n||^2} = \frac{\lambda_N + \sum_{j=1}^{N-1} \lambda_j \cdot \left|\frac{\lambda_j}{\lambda_N}\right|^{2n} \cdot \left|\frac{\alpha_j}{\alpha_N}\right|^2}{1 + \sum_{j=1}^{N-1} \left|\frac{\lambda_j}{\lambda_N}\right|^{2n} \cdot \left|\frac{\alpha_j}{\alpha_N}\right|^2} \to \lambda_N$$

as $n \to \infty$, since λ_N is the dominant eigenvalue, therefore $\frac{\lambda_j}{\lambda_N} \to 0$ for each index $1 \leq j \leq N-1$.

For the speed of convergence, a similar calculation shows the validity of the following estimation (details are omitted):

$$\left| \frac{\langle Ax_n, x_n \rangle}{||x_n||^2} - \lambda_N \right| \leq const. \cdot \left| \frac{\lambda_{N-1}}{\lambda_N} \right|^{2n}$$

4.3 Determination of the eigenvalue with minimal absolute value. The inverse iteration

Let $A \in \mathbf{M}_{N \times N}$ be a normal, regular matrix with the eigenvalues: $0 < |\lambda_1| < |\lambda_2| \leq \lambda_3| \leq ... \leq |\lambda_N|$, and denote by $s_1, s_2, ..., s_N$ the corresponding othonormal eigenvectors.

Apply the power iteration to the matrix A^{-1} (its eigenvalues are $\frac{1}{\lambda_1}$, ..., $\frac{1}{\lambda_N}$).

Let $x_0 \in \mathbf{R}^N$ be a vector such that $\langle x_0, s_1 \rangle \neq 0$, and define:

$$x_{n+1} := A^{-1} x_n \qquad (n = 0, 1, 2, ...)$$

Then the sequence of the Rayleigh quotients converges to $\frac{1}{\lambda_1}$:

$$\boxed{\frac{\langle A^{-1} x_n, x_n \rangle}{||x_n||^2} = \frac{\langle x_{n+1}, x_n \rangle}{||x_n||^2} \to \frac{1}{\lambda_1} \quad (n \to \infty)}$$

Instead of computing the inverse matrix A^{-1}, it is worth calculating the vector x_{n+1} by solving the equation

$$Ax_{n+1} = x_n$$

and applying an LU (or QR) decomposition.

4.4 Determination of the intermediate eigenvalues. The shifted inverse power method

Let $A \in \mathbf{M}_{N \times N}$ be a normal matrix with eigenvalues: $\lambda_1, \lambda_2, ...\lambda_N$, and denote by $s_1, s_2, ..., s_N$ the corresponding orthonormal eigenvectors.

Assume that λ_k is a single eigenvalue, and λ_0 is a sufficiently good approximation of λ_k, such that the nearest eigenvalue to λ_0 is λ_k, i.e. for every $j \neq k$, the inequality $|\lambda_j - \lambda_0| > |\lambda_k - \lambda_0|$ holds.

Then the number $(\lambda_k - \lambda_0)$ is an eigenvalue of the matrix $(A - \lambda_0 I)$ with minimal absolute value, so that the inverse iteration is applicable.

Let $x_0 \in \mathbf{R}^N$ be a vector such that $\langle x_0, s_k \rangle \neq 0$, and define:

$$x_{n+1} := (A - \lambda_0 I)^{-1} x_n \qquad (n = 0, 1, 2, ...)$$

Then the sequence of the Rayleigh quotients converges to $\frac{1}{\lambda_k - \lambda_0}$:

$$\boxed{\frac{\langle (A - \lambda_0 I)^{-1} x_n, x_n \rangle}{||x_n||^2} = \frac{\langle x_{n+1}, x_n \rangle}{||x_n||^2} \to \frac{1}{\lambda_k - \lambda_0} \quad (n \to \infty)}$$

Again, the computation of the inverse of $(A - \lambda_0 I)$ should be avoided. Instead, it is worth calculating the vector x_{n+1} by solving the equation

$$(A - \lambda_0 I) x_{n+1} = x_n$$

and applying an LU (or QR) decomposition.

4.5 Determination of all eigenvalues

Now we outline three methods which are able to compute all the eigenvalues of a matrix, under some conditions. It should be pointed out that the determination of the eigenvalues is *not* based on the characteristic equation. The proofs of the following statements are rather lengthy, so they are omitted.

First, let $A \in \mathbf{M}_{N \times N}$ be a self-adjoint matrix.

Method of Jacobi: Denote by $A_0 := A$. Define the pair of indices (p, q) $(p < q)$, for which $|A_{pq}|$ is maximal above the main diagonal. Define

$$\cot 2t := \frac{A_{qq} - A_{pp}}{2A_{pq}},$$

and define the (orthogonal) matrix Q_n as follows:

$$Q_n := \begin{pmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & \cos t & ... & ... & ... & \sin t & \\ & & ... & 1 & & & ... & \\ & & ... & & ... & & ... & \\ & & ... & & & 1 & ... & \\ & & -\sin t & ... & ... & ... & \cos t & \\ & & & & & & & 1 \\ & & & & & & & & 1 \end{pmatrix},$$

Now define the the matrix A_{n+1} by:

$$\boxed{A_{n+1} := Q_n^* A_n Q_n}$$

and repeat the procedure. If the eigenvalues of A are distinct, then the matrix sequence defined above tends to a diagonal matrix, the main diagonal of which contains the eigenvalues of A.

Now let $A \in \mathbf{M}_{N\times N}$ be a self-adjoint positive definite matrix.

Method based on the Cholesky decomposition: Denote by $A_0 := A$, and for $n = 0, 1, 2, ...$ define

$$\boxed{A_{n+1} := L_n^* L_n}$$

where $L_n L_n^*$ is the Cholesky decomposition of A_n. Then the sequence of matrices (A_n) elementwise converges to a diagonal matrix, the diagonal entries of which are the eigenvalues of the matrix A.

Method based of the QR decomposition: Denote by $A_0 := A$, and for $n = 0, 1, 2, ...$ define

$$\boxed{A_{n+1} := R_n Q_n}$$

where $Q_n R_n$ is the QR decomposition of A_n. Then the sequence of matrices (A_n) elementwise converges to a diagonal matrix, the diagonal entries of which are the eigenvalues of the matrix A.

From computational point of view, the above two algorithms are rather expensive, since in *each* iteration step, a Cholesky (resp. a QR) decomposition has to be performed.

5 The method of least squares

Strictly speaking, the method of least squares is not a concrete, special method. It is a type of approaches to handle problems which may be quite far from each other. A common property of this approach is to convert the original problem to a *minimization problem*.

5.1 Approximation of linear systems by the method of least squares

For the time being, let $A \in \mathbf{M}_{N\times N}$ be a *regular* matrix, but not necessarily self-adjoint. Let $b \in \mathbf{R}^N$ be a vector and consider the linear system of equations:

$$Ax = b$$

Denote by x^* the (uniquely determined) exact solution which obviously minimizes the square of the Euclidean norm of the residual vector: minimizes the following functional:

$$F(x) := ||Ax - b||^2,$$

since $F(x^*) = 0$, and due to the regularity of A, if $x \neq x^*$, then $F(x) > 0$.

With straightforward calculations:

$$F(x) = ||Ax-b||^2 = \langle Ax-b, Ax-b\rangle = \langle Ax, Ax\rangle - \langle Ax, b\rangle - \langle b, Ax\rangle + \langle b, b\rangle =$$

$$= \langle A^*Ax, x\rangle - 2\langle x, A^*b\rangle + ||b||^2$$

i.e. the minimizing vector x^* minimizes also the energetic functional of the *Gaussian normal equation*:

$$A^*Ax = A^*b$$

(note that A^*A is always self-adjoint and it is positive definite since A is assumed to be regular).

Remark: The Gaussian normal equation can be obtained from the original equation $Ax = b$ in a quite simple way, by multiplying the adjoint matrix A^*. In other words, this is a *symmetrization* of the original equation. Since the matrix of the new equation is self-adjoint and positive definite, the variational methods e.g. the gradient method can be applied without difficulty: let $x^{(0)}$ be a starting approximation and:

$$x^{(n+1)} := x^{(n)} - \frac{||r^{(n)}||^2}{\langle A^*Ar^{(n)}, r^{(n)}\rangle} \cdot r^{(n)}$$

whence

$$\boxed{x^{(n+1)} = x^{(n)} - \frac{||r^{(n)}||^2}{||Ar^{(n)}||^2} \cdot r^{(n)} \qquad \text{where } r^{(n)} := A^*Ax^{(n)} - A^*b.}$$

This is the unsymmetric version of the gradient method. Note however, that the symmetrization is often disadvantagous from computational point of view, since the matrix A^*A may be seriously ill-conditioned.

The above idea (to minimize the norm of the residual vector instead of solving the original system of equations) can be extensively generalized e.g. for systems which have no solution at all in classical sense. As a very simple example, consider the *linear regression problem.*

5.1.1 Linear regression

Let $x_1, x_2, ..., x_M$ be given numbers and let $y_1, y_2, ..., y_M$ be some associated values. Find a polynomial $p(x) = a_0 + a_1 x$ such that it fits to the given data 'as exactly as possible'.

The exact fitting would be as follows:

$$a_0 + a_1 x_k = y_k$$

for all $k = 1, 2, ..., M$, which can be written in the following form:

$$Aa = y,$$

where

$$A := \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ ... & ... \\ 1 & x_M \end{pmatrix}, \qquad a := \begin{pmatrix} a_0 \\ a_1 \end{pmatrix}, \qquad y := \begin{pmatrix} y_1 \\ y_2 \\ ... \\ y_M \end{pmatrix}$$

There is no exact solution in general. It is worth defining the 'best fitting' by minimizing the norm of the residual:

$$F(a_0, a_1) := \sum_{k=1}^{M} (a_0 + a_1 x_k - y_k)^2 = ||Aa - y||^2 \to \min!$$

At the location of the minimum, the partial derivatives vanish:

$$\frac{\partial F}{\partial a_0} = \sum_{k=1}^{M} 2 \cdot (a_0 + a_1 x_k - y_k) \cdot 1 = 0$$

$$\frac{\partial F}{\partial a_1} = \sum_{k=1}^{M} 2 \cdot (a_0 + a_1 x_k - y_k) \cdot x_k = 0$$

whence a_0, a_1 can be computed by solving the system of equations:

$$a_0 \cdot \sum_{k=1}^{M} 1 + a_1 \cdot \sum_{k=1}^{M} x_k = \sum_{k=1}^{M} y_k$$

$$a_0 \cdot \sum_{k=1}^{M} x_k + a_1 \cdot \sum_{k=1}^{M} x_k^2 = \sum_{k=1}^{M} x_k y_k$$

By direct calculations, it can be easily checked that the latter system can be expressed in the form:

$$A^* A a = A^* y$$

i.e. the Gaussian normal equation of the original system.

5.1.2 Overdetermined linear systems

Now let $A \in \mathbf{M}_{N \times M}$ be a necessarily square matrix with $N > M$ (or even $N \gg M$). Let $b \in \mathbf{R}^N$ be a vector and consider the linear system of equations (*overdetermined system*):

$$Ax = b$$

This system has no solution in general in classical sense. The vector $x^* \in \mathbf{R}^M$ is called a least squares solution or *generalized solution* which minimizes the square of the Euclidean norm of the residual vector i.e. minimizes the functional

$$F(x) := ||Ax - b||^2$$

If x happens to satisfy the original system, obviously $F(x) = 0$. Otherwise, the philosophy is as follows: the smaller the norm of the residual, the more 'exact' the solution. In this sense, the generalized solution is the 'most exact'.

Theorem: The generalized solution $x^* \in \mathbf{R}^M$ satisfies the *Gaussian normal equation*:

$$A^* A x = A^* b$$

Proof: We have seen that

$$F(x) = ||Ax - b||^2 = \langle A^* A x, x \rangle - 2\langle x, A^* b \rangle + ||b||^2$$

Straightforward calculations show that the gradient vector of F at x is as follows:

$$\operatorname{grad} F(x) = 2A^*Ax - 2A^*b$$

At the location of minimum, the gradient vector vanishes, which implies the theorem.

Remark: The minimization problem can also be handled by the traditional tools of the multivariate extremal value problems. Obviously

$$F(x_1, x_2, ..., x_M) = ||Ax - b||^2 = \sum_{k=1}^{N} \left(\sum_{j=1}^{M} A_{kj}x_j - b_k \right)^2 ,$$

whence, for all indices $m = 1, 2, ..., N$:

$$\frac{\partial F}{\partial x_m} = \sum_{k=1}^{N} 2 \cdot \left(\sum_{j=1}^{M} A_{kj}x_j - b_k \right) \cdot A_{km} = 0$$

Changing the order of summations in the left-hand side:

$$\sum_{j=1}^{M} \left(\sum_{k=1}^{N} A_{kj}A_{km} \right) \cdot x_j = \sum_{k=1}^{N} A_{km}b_k$$

In the left-hand side: $A_{kj} = A^*_{jk}$. Similarly, in the right-hand side: $A_{km} = A^*_{mk}$. We have obtained that

$$\sum_{j=1}^{M} \left(\sum_{k=1}^{N} A^*_{jk}A_{km} \right) \cdot x_j = \sum_{k=1}^{N} A^*_{mk}b_k$$

for all indices $m = 1, 2, ..., N$. By definition of the matrix multiplication, this is equivalent to the Gaussian normal equation

$$A^*Ax = A^*b$$

Obviously $A^*A \in \mathbf{M}_{M\times M}$, therefore the size of the matrix of the Gaussian normal equation is much less than that of the original system, provided that $N \gg M$. Note however, that the Gaussian normal equation may be severely ill-conditioned, which causes computational problems.

It should be pointed out that the method of least squares remains applicable also in the case when $N < M$ (underdetermined systems). It is typical that the Gaussian normal equation (and possibly also the origial system) has several solutions. Sometimes it is worth finding the solution with minimal Eucliean norm. Details are omitted.

5.1.3 Approximation of functions

Let $[a,b] \subset \mathbf{R}$ be a finite interval and let $f : [a,b] \to \mathbf{R}$ be a given, say, continuous function. Our aim is to approximate the – possibly complicated – function f by simpler functions e.g. polynomial. Assume that p is a polynomial of degree at most N:

$$p(x) := a_0 + a_1 x + a_2 x^2 + ... + a_N x^N$$

How to define the accuracy of approximation?

One possibility is to define a set of points in the interval $[a,b]$: $a \le x_1 < x_2 < ... < x_M \le b$. The a priori unknown coefficients a_0, a_1, ..., a_N are chosen in such a way that the sum of squares of the deviations is minimal:

$$F(a_0, a_1, ..., a_N) := \sum_{k=1}^{M} (p(x_k) - f(x_k))^2 = \sum_{k=1}^{M} \left(\sum_{j=0}^{N} a_j x_k^j - f(x_k) \right)^2 \to \min!$$

This is a *polynomial regression* problem. One can easly check that the unknown coefficients satisfy the following system of equations:

$$\sum_{j=0}^{N} a_j \cdot \left(\sum_{k=1}^{M} x_k^{j+m} \right) = \sum_{k=1}^{M} x_k^m f(x_k) \qquad (m = 0, 1, ..., N)$$

Another possibility is to choose the coefficients in such a way that the square integral of the difference of the functions f and p is minimal:

$$G(a_0, a_1, ..., a_N) := \int_a^b (p(x) - f(x))^2 \, dx = \int_a^b \left(\sum_{j=0}^{N} a_j x^j - f(x) \right)^2 dx \to \min!$$

Now the accuracy of approximation is characterized by the above square integral.

At the location of minimum, all partial derivatives vanish, i.e. for all indices $m = 0, 1, ..., N$:

$$\frac{\partial G}{\partial a_m} = \int_a^b 2 \cdot \left(\sum_{j=0}^{N} a_j x^j - f(x) \right) \cdot x^m \, dx = 0$$

Thus, we obtain the following system of equations for the unknown coefficients:

$$\sum_{j=0}^{N} a_j \left(\int_a^b x^{j+m} \, dx \right) = \int_a^b f(x) \cdot x^m \, dx \qquad (m = 0, 1, ..., N)$$

The matrix entries can be calculated easily:

$$A_{mj} = \int_a^b x^{j+m}\, dx = \frac{b^{j+m+1} - a^{j+m+1}}{j+m+1}$$

In particular, if $[a, b] = [0, 1]$, then the form of the matrix is extremely simple:

$$A_{mj} = \frac{1}{j+m+1} \qquad (j, m = 0, 1, ..., N)$$

This matrix is called *Hilbert matrix* of order $(N + 1)$.

The integrals appearing in the right-hand side can be computed *in principle*; in practice, however, one has to apply some approximate method only.

Remark: The Hilbert matrices are self-adjoint and positive definite, however, they are extremely ill-conditioned. For this reason, they are often preferred when testing algorithms for solving linear systems.

6 The Singular Value Decomposition

6.1 SVD for square, regular matrices

Let $A \in \mathbf{M}_{N\times N}$ be a regular matrix. Denote by $\lambda_j > 0$ the eigenvalues of the (self-adjoint, positive definite) matrix A^*A. Let v_j be the corresponding orthonormal eigenvectors ($j = 1, 2, ..., N$).

The numbers $\sigma_k := \sqrt{\lambda_k}$ ($k = 1, ..., N$) are called the *singular values* of A. Assume that they are ordered in descending order: $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_N > 0$. Note that all singular values are positive, since due to the regularity of A, the matrix A^*A is nonsingular, therefore positive definite.

Define the orthogonal matrix

$$V := \left(\begin{array}{c|c|c|c} & & & \\ v_1 & v_2 & ... & v_N \\ & & & \end{array} \right).$$

Introduce the vectors

$$u_k := \frac{Av_k}{\sigma_k} \qquad (k = 1, 2, ..., N)$$

Then u_1, u_2, ..., u_N form an *orthonormal system*. Indeed:

$$\langle u_k, u_j \rangle = \frac{\langle Av_k, Av_j \rangle}{\sigma_k \sigma_j} = \frac{\langle v_k, A^*Av_j \rangle}{\sigma_k \sigma_j} =$$

$$= \frac{\lambda_j}{\sigma_k \sigma_j},$$

which equals to $\frac{\lambda_j}{\sigma_j^2} = 1$, if $k = j$; otherwise it is zero.

Thus, the following matrix is also orthogonal:

$$U := \left(\begin{array}{c|c|c|c} & & & \\ u_1 & u_2 & ... & u_N \\ & & & \end{array} \right)$$

Introduce the diagonal matrix:

$$S := \begin{pmatrix} \sigma_1 & 0 & ... & 0 \\ 0 & \sigma_2 & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & \sigma_N \end{pmatrix}.$$

It can be easily checked that the matrix A is decomposed in the form :

$$A = USV^*$$

This decomposition is called *singular value decomposition* of A.

Remarks:

- The decomposition is not unique, e.g. the unit matrix can be decomposed in the form
$$I = QIQ^*$$
for arbitrary othogonal matrix Q.
- The SVD requires computing a complete eigensystem of the matrix A^*A, which is expensive from computational point of view.
- Once the SVD has been computed, the solution of the system $Ax = b$ is cheap from computational point of view ($\mathcal{O}(N^2)$ operations are needed), since
$$x = VS^{-1}U^*.$$

6.2 SVD for non-square matrices

Let $A \in \mathbf{M}_{M\times N}$ be a matrix, where $M \geq N$. Denote by $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_N \geq 0$ the eigenvalues of the (self-adjoint, positive semidefinite) matrix A^*A. Assume that they are ordered in descending order. Let v_j be the corresponding orthonormal eigenvectors ($j = 1, 2, ..., N$).

The numbers $\sigma_k := \sqrt{\lambda_k}$ ($k = 1, ..., r$) are called the *singular values* of A, where $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq ...$ and r is the last index for which $\sigma_r > 0$

Define the orthogonal matrix

$$V := \left(\begin{array}{c|c|c|c} & & & \\ v_1 & v_2 & ... & v_N \\ & & & \end{array} \right).$$

Introduce the vectors

$$u_k := \frac{Av_k}{\sigma_k} \qquad (k = 1, 2, ..., r)$$

Then u_1, u_2, ..., u_r form an *orthonormal system*. Indeed:

$$\langle u_k, u_j \rangle = \frac{\langle Av_k, Av_j \rangle}{\sigma_k \sigma_j} = \frac{\langle v_k, A^* A v_j \rangle}{\sigma_k \sigma_j} =$$

$$= \frac{\lambda_j}{\sigma_k \sigma_j},$$

which equals to $\frac{\lambda_j}{\sigma_j^2} = 1$, if $k = j$; otherwise it is zero.

Now complete the set of vectors $u_1, ..., u_r$ by the vectors $u_{r+1}, u_{r+2}, ..., u_M$ in such a way that $u_1, u_2, ..., u_M$ form an orthonormal system (by e.g. a Gram-Schmidt orthogonalization procedure). Then the following matrix is orthogonal:

$$U := \left(\begin{array}{c|c|c|c} & & & \\ u_1 & u_2 & ... & u_M \\ & & & \end{array} \right)$$

Define the M-by-N 'diagonal' matrix:

$$S := \begin{pmatrix} \sigma_1 & 0 & ... & 0 & 0 & ... & 0 \\ 0 & \sigma_2 & ... & 0 & 0 & ... & 0 \\ ... & ... & ... & ... & 0 & ... & 0 \\ 0 & 0 & ... & \sigma_r & 0 & ... & 0 \\ 0 & 0 & ... & 0 & 0 & ... & 0 \\ ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & ... & 0 & 0 & ... & 0 \end{pmatrix}.$$

After a bit lengthy but straightforward calculations, we obtain that the matrix A is decomposed in the form

$$A = USV^*,$$

where S is the above M-by-N quasidiagonal matrix formed by the nonzero singular values, $U \in \mathbf{M}_{M \times M}$, $V \in \mathbf{M}_{N \times N}$ are orthogonal matrices. This decomposition is called *singular value decomposition* of A. Note that the decomposition is not unique.

Example: Let $\mathbf{0} \neq A := a \in \mathbf{M}_{M \times 1}$ be a column vector. Let $u_1 := \frac{a}{||a||}$ and complete u_1 by the vectors $u_2, ..., u_M$ such that they form an orthonormal system. The SVD of the matrix A is as follows:

$$A = \left(\begin{array}{c|c|c|c} & & & \\ u_1 & u_2 & ... & u_M \\ & & & \end{array} \right) \cdot \begin{pmatrix} ||a|| \\ 0 \\ 0 \\ ... \\ 0 \end{pmatrix} \cdot (1)$$

Example: Let $\mathbf{0} \neq a \in \mathbf{M}_{M\times 1}$ be a column vector and

$$A := \left(\begin{array}{c|c} a & a \end{array} \right).$$

Let $u_1 := \frac{a}{||a||}$ and complete u_1 by the vectors $u_2, ..., u_M$ such that they form an orthonormal system. The SVD of the matrix A is as follows:

$$A = \left(\begin{array}{c|c|c|c} u_1 & u_2 & ... & u_M \end{array} \right) \cdot \begin{pmatrix} \sqrt{2}||a|| & 0 \\ 0 & 0 \\ 0 & 0 \\ ... & ... \\ 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

6.3 The generalized inverse

Let $A \in \mathbf{M}_{M\times N}$ be a matrix with the SVD $A = USV^*$, where $U \in \mathbf{M}_{M\times M}$, $V \in \mathbf{M}_{N\times N}$ are orthogonal matrices and S is the quasidiagonal matrix formed by the positive singular values:

$$S := \begin{pmatrix} \sigma_1 & 0 & ... & 0 & 0 & ... & 0 \\ 0 & \sigma_2 & ... & 0 & 0 & ... & 0 \\ ... & ... & ... & ... & 0 & ... & 0 \\ 0 & 0 & ... & \sigma_r & 0 & ... & 0 \\ 0 & 0 & ... & 0 & 0 & ... & 0 \\ ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & ... & 0 & 0 & ... & 0 \end{pmatrix} \in \mathbf{M}_{M\times N}$$

Denote by

$$S^+ := \begin{pmatrix} \sigma_1^{-1} & 0 & ... & 0 & 0 & ... & 0 \\ 0 & \sigma_2^{-1} & ... & 0 & 0 & ... & 0 \\ ... & ... & ... & ... & 0 & ... & 0 \\ 0 & 0 & ... & \sigma_r^{-1} & 0 & ... & 0 \\ 0 & 0 & ... & 0 & 0 & ... & 0 \\ ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & ... & 0 & 0 & ... & 0 \end{pmatrix} \in \mathbf{M}_{N\times M}$$

The matrix $A^+ := VS^+U^*$ is called the *generalized inverse* of A (*Moore-Penrose inverse* or *pseudoinverse*).

If the matrix $A \in \mathbf{M}_{N\times N}$ is regular, then $A^+ = A^{-1}$. The matrices AA^+ and A^+A are self-adjoint. Moreover, $AA^+A = A$ and $A^+AA^+ = A^+$.

The pseudoinverse is uniquely determined by the above properties.

6.4 Generalized solution of linear systems

Let $A \in \mathbf{M}_{M\times N}$ be a matrix, $b \in \mathbf{R}^M$. The vector

$$x^+ := A^+b \in \mathbf{R}^N$$

is called the *generalized solution* of the system $Ax = b$.

The generalized solution $x^+ = A^+b$ always uniquely exists, and this is the solution in the sense of least squares, i.e. it minimizes the functional $||Ax - b||^2$. If several minimizing vectors exist, then x^+ is the one that has the minimal Euclidean norm. Furthermore, the generalized solution satisfies the Gaussian normal equations, i.e. $A^*Ax^+ = A^*b$. This is a close connection between the SVD and the least squares.

6.5 An application – image compression using SVD

An image can be considered a matrix of pixels. For simplicity, we restrict ourselves to black-and-white (greyscale) images. Every pixel has an intensity value which varies between zero (black) and a maximal value (white). Thus, a greyscale image can be considered a (not necessarily square) matrix.

Computing the SVD of an N-by-N image, the distribution of the singular values of the image matrix is generally not uniform. A typical phenomenon is that there are a lot of singular values which are nearly equal to zero. Let us partition the matrices appearing in the SVD formulation $A = USV^*$ in the following way:

$$U_{11} \in \mathbf{M}_{m\times m},\quad U_{12} \in \mathbf{M}_{m\times(N-m)},\quad U_{21} \in \mathbf{M}_{(N-m)\times m},\quad U_{22} \in \mathbf{M}_{(N-m)\times(N-m)}$$

$$S_{11} \in \mathbf{M}_{m\times m},\quad S_{22} \in \mathbf{M}_{(N-m)\times(N-m)}$$

$$V_{11}^* \in \mathbf{M}_{m\times m},\quad V_{12}^* \in \mathbf{M}_{m\times(N-m)},\quad V_{21}^* \in \mathbf{M}_{(N-m)\times m},\quad V_{22}^* \in \mathbf{M}_{(N-m)\times(N-m)}$$

Then the SVD has the form:

$$\left(\begin{array}{c|c} U_{11} & U_{12} \\ \hline U_{21} & U_{22} \end{array}\right) \cdot \left(\begin{array}{c|c} S_{11} & 0 \\ \hline 0 & S_{22} \end{array}\right) \cdot \left(\begin{array}{c|c} V_{11}^* & V_{12}^* \\ \hline V_{21}^* & V_{22}^* \end{array}\right)$$

If $S_{22} \approx \mathbf{0}$, (where m is possibly much less than N) then the matrix product USV^* approximately equals to the product

$$\left(\begin{array}{c|c} U_{11} & U_{12} \\ \hline U_{21} & U_{22} \end{array}\right) \cdot \left(\begin{array}{c|c} S_{11} & 0 \\ \hline 0 & 0 \end{array}\right) \cdot \left(\begin{array}{c|c} V_{11}^* & V_{12}^* \\ \hline V_{21}^* & V_{22}^* \end{array}\right),$$

Using the definition of the multiplication of matrices, it can be easily checked, that this matrix product is equal to:

$$\left(\begin{array}{c|c} U_{11}S_{11}V_{11}^* & U_{11}S_{11}V_{12}^* \\ \hline U_{21}S_{11}V_{11}^* & U_{21}S_{11}V_{12}^* \end{array}\right)$$

We have obtained that *it is sufficient to retain only the first m columns from U (and similarly, the first m rows from V^*)*, since the other colums (rows) do not appear in the above formulation. If the 'nearly zero' singular values are replaced by zeros, then, in a lot of cases, the quality of the image is decreased by a small amount only. However, the memory requirement may be significantly decreased. Thus, the SVD can be applied as an image compression method.

Remark: The above algorithm can be generalized to nonsquare matrices as well as colored images in a straightforward way.

6.5.1 Example 1

To illustrate the above outlined phenomenon, consider the following 512×512 image together with the graph of the first 100 singular values as can be seen in Figure 1:

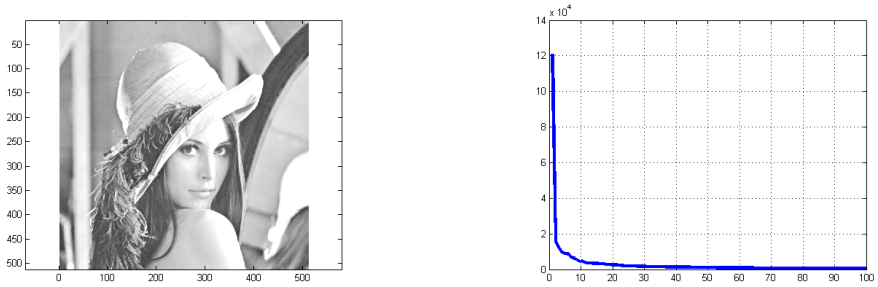

Figure 1: An 512×512 and the first 100 singular values

Let us truncate the singular values of the image keeping the first m singular values and omitting the other ones. m is called the number of *modes*. The reconstructed images can be seen in Figure 2.

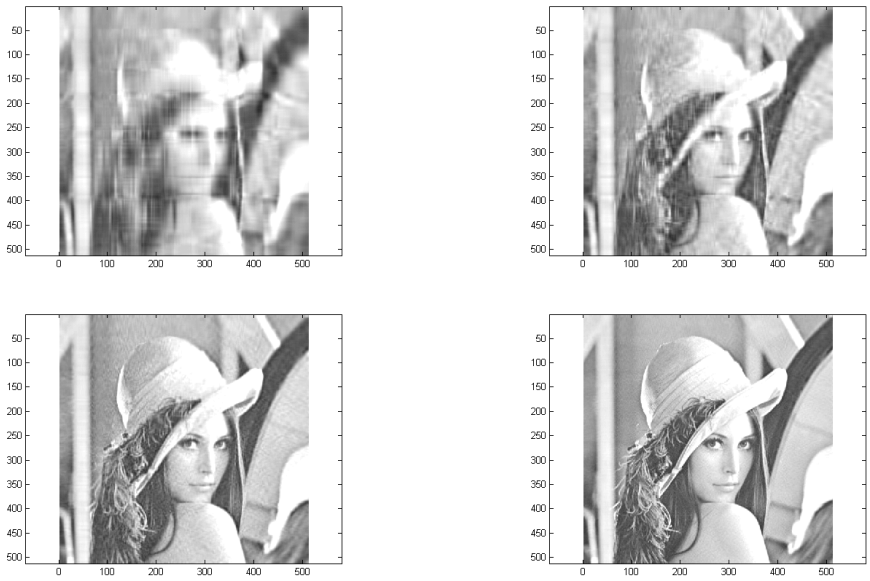

Figure 2: Reconstructed images from truncated SVD. Number of modes: 10 (uppert left), 20 (upper right), 50 (lower left) and 100 (lower right)

6.5.2 Example 2

Now consider the following 512×512 image and also its graph of the first 100 singular values (see Figure 3). Figure 4 shows some reconstructions with different number of modes.

Remark: An interesting online image compression demo based on the SVD can be seen here:

http://timbaumann.info/svd-image-compression-demo/

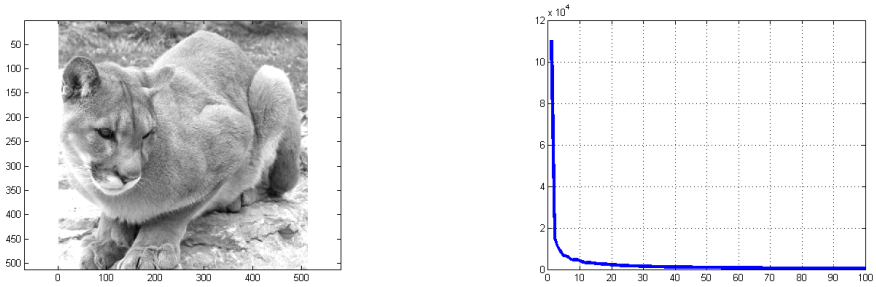

Figure 3: An 512×512 image and the first 100 singular values

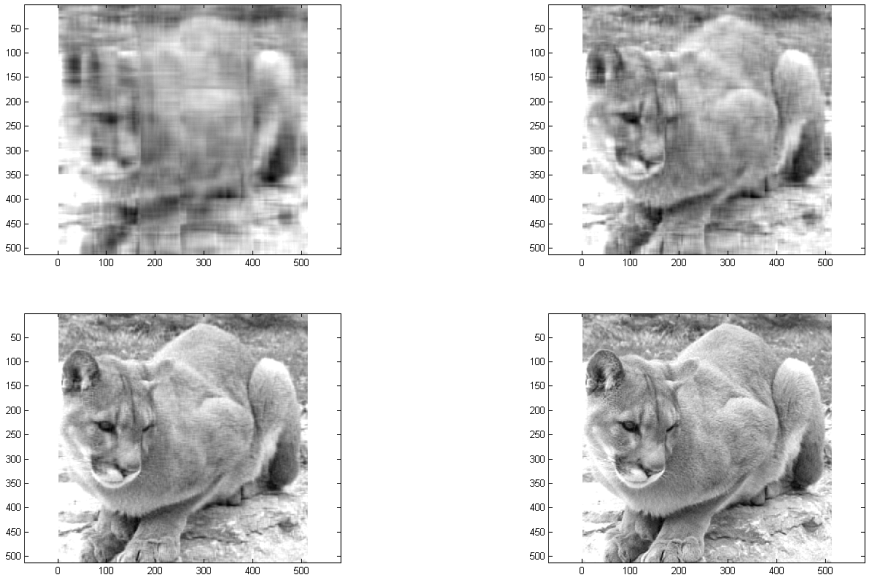

Figure 4: Reconstructed images from truncated SVD. Number of modes: 10 (uppert left), 20 (upper right), 50 (lower left) and 100 (lower right)

7 The Discrete and the Fast Fourier Transform

7.1 Trigonometric Fourier series

An arbitrary real function $f \in L_2(0, 2\pi)$ can be expressed as a trigonometric Fourier series which is convergent with respect to the $L_2(0, 2\pi)$-norm:

$$f(x) = a_0 + \sum_{k=1}^{\infty} a_k \cos kx + \sum_{k=1}^{\infty} b_k \sin kx$$

where the coefficients can be calculated as:

$$a_0 = \frac{1}{2\pi} \int_0^{2\pi} f(x)\, dx,$$

$$a_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \cos kx\, dx, \quad b_k = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin kx\, dx$$

Complex exponential function. For any $z \in \mathbf{C}$, define the *exponential series* as:

$$e^z := \sum_{k=0}^{\infty} \frac{z^k}{k!} = 1 + \frac{z}{1!} + \frac{z^2}{2!} + \frac{z^3}{3!} + ...$$

Theorem (Euler's formula): For every $t \in \mathbf{R}$:

$$e^{it} = \cos t + i \cdot \sin t$$

Proof: Utilizing the well-known Taylor series of the sine and cosine functions:

$$\cos t = 1 + \frac{t^2}{2!} + \frac{t^4}{4!} + ...$$

$$\sin t = \frac{t}{1!} + \frac{t^3}{3!} + \frac{t^5}{5!} + ...$$

which implies that:

$$e^{it} = 1 + \frac{it}{1!} + \frac{i^2t^2}{2!} + \frac{i^3t^3}{3!} + \frac{i^4t^4}{4!} + ... =$$

$$= 1 + \frac{it}{1!} - \frac{t^2}{2!} - \frac{it^3}{3!} + \frac{t^4}{4!} + ...$$

Separating the real and imaginary parts, we have the theorem.

7.2 The Discrete Fourier Transform

If $f_0, f_1, ..., f_{N-1} \in \mathbf{C}$ is a finite sequence, then define its **discrete Fourier transform** as:

$$\hat{f}_k := \sum_{j=0}^{N-1} f_j \cdot e^{\frac{2\pi i k j}{N}} \qquad (k = 0, 1, ..., N-1)$$

Relationship with the Fourier series: Let f be a continuous function defined on the interval $[0, 2\pi)$, and denote by $f_j := f(\frac{2\pi j}{N})$. Then the sum

$$\frac{2\pi}{N} \cdot \sum_{j=0}^{N-1} f_j \cdot e^{\frac{2\pi i k j}{N}}$$

is a Riemann sum of the integral

$$\int_0^{2\pi} f(x) e^{ikx}\, dx.$$

Utilizing Euler's formula, we have:

$$\frac{1}{N}\hat{f}_k = \frac{1}{N} \sum_{j=0}^{N-1} f_j \cdot e^{\frac{2\pi i k j}{N}} \approx \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{ikx}\, dx =$$

$$= \frac{1}{2\pi} \int_0^{2\pi} f(x) \cos kx\, dx + i \cdot \frac{1}{2\pi} \int_0^{2\pi} f(x) \sin kx\, dx = a_k + i b_k,$$

where a_k, b_k are the trigonometric Fourier coefficients.

The inverse Discrete Fourier Transform (iDFT): Every finite sequence can uniquely be reconstructed from its DFT, namely:

$$f_k := \frac{1}{N} \cdot \sum_{j=0}^{N-1} \hat{f}_j \cdot e^{-\frac{2\pi i k j}{N}} \qquad (k = 0, 1, ..., N-1)$$

Proof:

$$\frac{1}{N} \sum_{j=0}^{N-1} \hat{f}_j e^{-\frac{2\pi i k j}{N}} = \frac{1}{N} \sum_{j=0}^{N-1} \sum_{r=0}^{N-1} f_r e^{\frac{2\pi i r j}{N}} e^{-\frac{2\pi i k j}{N}} = \sum_{r=0}^{N-1} f_r \cdot \frac{1}{N} \sum_{j=0}^{N-1} e^{\frac{2\pi i (r-k) j}{N}}$$

If $r = k$, then $z = 1$, therefore the inner sum equals to 1. If $r \neq k$, then the inner sum is the sum of a finite geometric sequence, which equals to 0. This completes the proof.

The Discrete Fourier Transform can be performed by multiplying the column vector $f = \begin{pmatrix} f_0 \\ f_1 \\ ... \\ f_{N-1} \end{pmatrix} \in \mathbf{C}^N$, by the matrix

$$A = \begin{pmatrix} 1 & 1 & 1 & ... & 1 \\ 1 & e^{\frac{2\pi i}{N}} & e^{\frac{4\pi i}{N}} & ... & e^{\frac{2(N-1)\pi i}{N}} \\ 1 & e^{\frac{4\pi i}{N}} & e^{\frac{8\pi i}{N}} & ... & e^{\frac{4(N-1)\pi i}{N}} \\ ... & ... & ... & ... & ... \\ 1 & e^{\frac{2(N-1)\pi i}{N}} & e^{\frac{4(N-1)\pi i}{N}} & ... & e^{\frac{2(N-1)(N-1)\pi i}{N}} \end{pmatrix} \in \mathbf{M}_{N\times N}.$$

This requires $\mathcal{O}(N^2)$ arithmetic operations which may be expensive from computational point of view especially when N is great and/or the DFT should be performed very quickly (e.g. in the case when the data come from real-time measurements). Thus, the above computational cost should be significantly reduced.

7.3 The Fast Fourier Transform

Denote by $F_N : \mathbf{C} \to \mathbf{C}$ the linear operator of the DFT:

$$(F_N f)_k = \sum_{j=0}^{N-1} f_j \cdot e^{\frac{2\pi i k j}{N}} \qquad (k = 0, 1, ..., N-1)$$

Assume that N is even: $N = 2N_1$. Let us separate the terms with even and odd indices in the expression of $(F_N f)_k$. First, let k be a 'small' index: $k = 0, 1, ..., N_1 - 1$:

$$(F_N f)_k = \sum_{\ell=0}^{N_1-1} f_{2\ell} \cdot e^{\frac{2\pi i k \cdot 2\ell}{N}} + \sum_{\ell=0}^{N_1-1} f_{2\ell+1} \cdot e^{\frac{2\pi i k \cdot (2\ell+1)}{N}} =$$

$$= \sum_{\ell=0}^{N_1-1} f_{2\ell} \cdot e^{\frac{2\pi i k \cdot 2\ell}{N}} + e^{\frac{2\pi i k}{N}} \cdot \sum_{\ell=0}^{N_1-1} f_{2\ell+1} \cdot e^{\frac{2\pi i k \cdot 2\ell}{N}} =$$

$$= \sum_{\ell=0}^{N_1-1} f_{2\ell} \cdot e^{\frac{2\pi i k\ell}{N_1}} + e^{\frac{2\pi i k}{N}} \cdot \sum_{\ell=0}^{N_1-1} f_{2\ell+1} \cdot e^{\frac{2\pi i k\ell}{N_1}}$$

Now consider the 'great' indices which have the form $N_1 + k$, where $k = 0, 1, ..., N_1 - 1$:

$$(F_N f)_{N_1+k} = \sum_{\ell=0}^{N_1-1} f_{2\ell} \cdot e^{\frac{2\pi i (N_1+k)\cdot 2\ell}{N}} + \sum_{\ell=0}^{N_1-1} f_{2\ell+1} \cdot e^{\frac{2\pi i (N_1+k)\cdot (2\ell+1)}{N}} =$$

$$= \sum_{\ell=0}^{N_1-1} f_{2\ell} \cdot e^{\frac{2\pi i (N_1+k)\cdot 2\ell}{N}} + e^{\frac{2\pi i (N_1+k)}{N}} \cdot \sum_{\ell=0}^{N_1-1} f_{2\ell+1} \cdot e^{\frac{2\pi i (N_1+k)\cdot 2\ell}{N}} =$$

$$= \sum_{\ell=0}^{N_1-1} f_{2\ell} \cdot e^{\frac{2\pi i k\ell}{N_1}} - e^{\frac{2\pi i k}{N}} \cdot \sum_{\ell=0}^{N_1-1} f_{2\ell+1} \cdot e^{\frac{2\pi i k\ell}{N_1}}$$

For both the 'small' and 'great' indices, both sums on the right-hand sides are discrete Fourier transforms with smaller vectors. The procedure can recursively be continued, if N is a power of two.

With recursive invocations:

- $N_1 := N/2$

$$f^{even} := (f_0, f_2, ..., f_{2N_1-2}), \quad f^{odd} := (f_1, f_3, ..., f_{2N_1-1})$$

- $\hat{f}^{even} := F_{N_1} f^{even}, \quad \hat{f}^{odd} := F_{N_1} f^{odd}$

- $$\begin{aligned} (F_N f)_k &:= \hat{f}_k^{even} + e^{\frac{2\pi i k}{N}} \cdot \hat{f}_k^{odd} \qquad (k = 0, 1, ..., N_1 - 1) \\ (F_N f)_{N_1+k} &:= \hat{f}_k^{even} - e^{\frac{2\pi i k}{N}} \cdot \hat{f}_k^{odd} \qquad (k = 0, 1, ..., N_1 - 1) \end{aligned}$$

where for $N = 1$, $F_1 f := f$ (f has one component only)

This means that at each level of subdivision, only $\mathcal{O}(N)$ arithmetic operations have to be performed. If N is a power of two, then the number of levels is $\log N$, so that the total computational cost: $\mathcal{O}(N \cdot \log N)$, which is much smaller than the computational cost of the original DFT ($\mathcal{O}(N^2)$

operations). As pointed out earlier this reduction is of primary importance in pracatical applications. This is the reason why the algorithm of FFT has been selected among the 'top ten algorithms' of the 20th century.

The technique is illustrated through the following example. Here the original vector is chosen from the eight-dimensional vector space $\mathbf{C}^8$. The number of levels of subdivision is 3. The vectors the discrete Fourier transform of which have to be computed are eight-dimensional at level 0; four-dimensional at level 1; two-dimensional at level 2, and one-dimensional at level 3. Here the discrete Fourier transform of a vector equals to the single component which forms the vector.

Level 0 :	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7
Level 1 :	f_0	f_2	f_4	f_6	f_1	f_3	f_5	f_7
Level 2 :	f_0	f_4	f_2	f_6	f_1	f_5	f_3	f_7
Level 3 :	f_0	f_4	f_2	f_6	f_1	f_5	f_3	f_7

7.4 The 2D Discrete Fourier Transform

The DFT of a matrix $f \in \mathbf{M}_{N\times N}$ is the matrix $\hat{f} \in \mathbf{M}_{N\times N}$ with the following entries:

$$\hat{f}_{kj} := \sum_{r=0}^{N-1}\sum_{s=0}^{N-1} f_{rs} \cdot e^{\frac{2\pi ikr}{N}} e^{\frac{2\pi ijs}{N}}$$

The algorithm of the computation of the DFT of a matrix:

- For every row of the matrix f, substitute the 1D DFT of the corresponding row.
- For every column of this matrix, substitute the 1D DFT of the corresponding column.

This results in the 2D DFT of the original matrix f. Of course, the 1D Discrete Fourier Transforms are recommended to be performed by the Fast Fourier Transform algorithm.

7.5 An application – image compression using FFT

An image can be considered a matrix of pixels. For simplicity, we restrict ourselves to black-and-white (greyscale) images. Every pixel has an intensity value which varies between zero (black) and a maximal value (white). Thus, a greyscale image can be considered a (not necessarily square) matrix.

Computing the DFT of an N-by-N image, the distribution of the absolute values of the DFT of the image is generally far from uniform. A typical phenomenon is that the components with larger absolute value are concentrated in the vicinity of the corners of the discrete Fourier transformed image matrix. These Fourier components are responsible for the low-frequency part of the original image, while the remaining Fourier components are related to the high-frequency part such as abrupt changes. If these components are replaced by zeros, then, in a lot of cases, the quality of the image is decreased by a small amount only; however, the memory requirement may be significantly decreased. Thus, the FFT can be applied as an image compression method.

7.5.1 Example 1

To illustrate the above outlined phenomenon, consider the following 512×512 image and also the 2D Fourier transform of it (blue pixels indicate small absolute values) as can be seen in Figure 5:

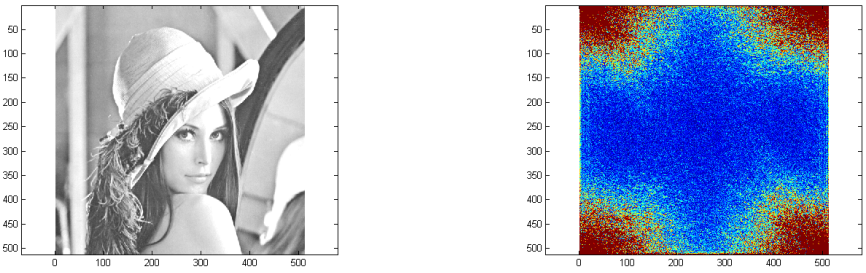

Figure 5: An 512×512 image and its 2D Discrete Fourier Transform

Let us truncate the DFT matrix by keeping all matrix entries that are located from the corner points closer than 100 pixels (with respect to the Euclidean distance) and by replacing the remaining matrix elements by zeros. Figure 6 shows the truncated DFT matrix and also the reconstructed image from the truncated DFT matrix (i.e. the inverse DFT of the truncated matrix). It can be seen that though the reconstructed image differs from the original one, the difference is visually very small. However, the total area of the quarters of circle at the corners (in which the information of the reconstructed image is contained) is much smaller than the area of the image:

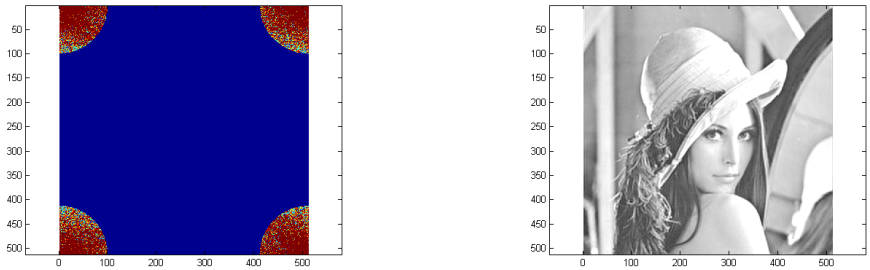

Figure 6: 2D Truncated Discrete Fourier Transform and the reconstructed image

Of course the smaller the retained area in the Fourier transform matrix, the bigger the difference between the original and the reconstructed images. In other words, the rate of compression and the quality of the reconstructed image change just in the opposite direction, as expected. This phenomenon can be seen in Figure 7. The 'radius of truncation' means always the maximal distance (measured in pixels with respect to the 2D Euclidean norm) from the corners, the closer matrix entries of which are kept.

7.5.2 Example 2

Now consider the following 512×512 image and also its 2D Fourier transform (see Figure 8). Figure 9 shows some reconstructions with different radii of truncation.

Remarks:

- In case of colored images, the same algorithm can be applied to each color component.
- The widely used JPEG image compression is based on the same idea. However, this algorithm applies the 2D Fourier transform to small parts of the original image and not to the whole image. Details are omitted.

Figure 7: Reconstructed images from truncated Discrete Fourier Transform. Radius of truncation: 10 (uppermost left), 20 (uppermost right), 40 (middle left), 50 (middle right), 60 (lowermost left) and 80 (lowermost right)

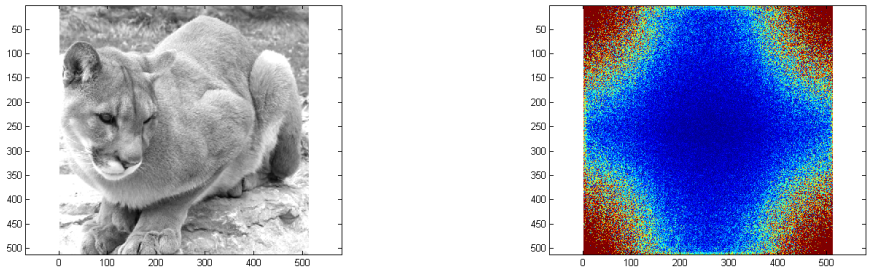

Figure 8: An 512×512 image and its 2D Discrete Fourier Transform

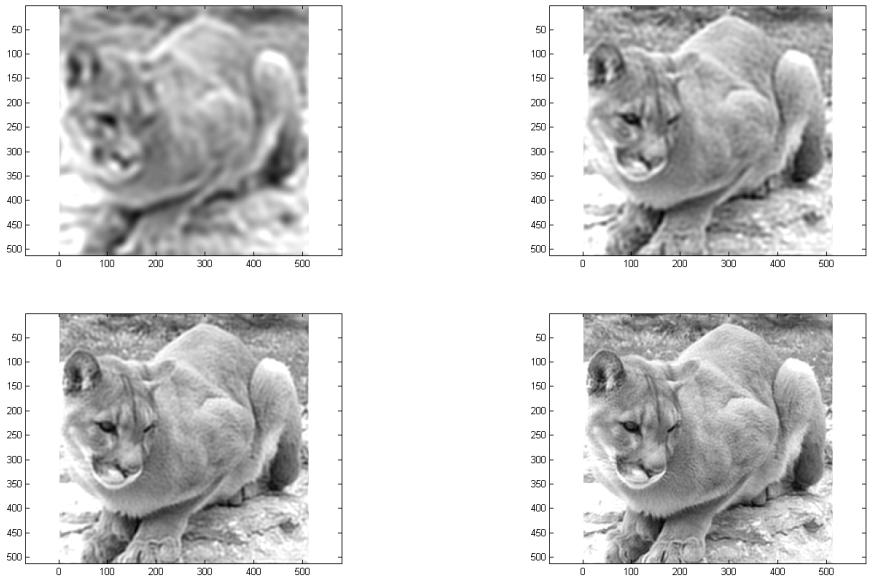

Figure 9: Reconstructed images from truncated Discrete Fourier Transform. Radius of truncation: 20 (upper left), 40 (upper right), 60 (lower left) and 80 (lower right)

8 Scattered Data Interpolation

8.1 The interpolation problem

Let $x_1, x_2, ..., x_N \in \mathbf{R}^2$ be given locations on the place (*interpolation points*), and let $f_1, f_2, ..., f_N \in \mathbf{R}$ be some predefined values associated to the interpolation points. Find a function $f : \mathbf{R}^2 \to \mathbf{R}$ *interpolation function* (which is as smooth as possible), which satisfies the *interpolation conditions*:

$$f(x_k) = f_k \qquad\qquad (k = 1, 2, ..., N)$$

No special structure (grid of mesh) of the interpolation points is assumed.

8.2 Shepard's method

The oldest and simplest scattered data interpolation method is as follows. Let $x_1, x_2, ..., x_N \in \mathbf{R}^2$ be given locations on the place (interpolation points), and let $f_1, f_2, ..., f_N \in \mathbf{R}$ be some predefined values associated to the interpolation points. If x is not an interpolation points, then:

$$f(x) := \frac{\sum_{j=1}^{N} f_j \cdot \frac{1}{||x - x_j||^p}}{\sum_{j=1}^{N} \frac{1}{||x - x_j||^p}}$$

The exponent p is a predefined positive constant: its usual value is 2 or 4.

The interpolation function f is not defined at the interpolation points but has limit values, which satisfy the interpolation conditions:

Theorem: $f(x) \to f_k$, whenever $x \to x_k$, for arbitrary interpolation point x_k.

Proof: Without loss of generality, one can assume that $k = 1$. Assume that $x \to x_1$. Then:

$$f(x) = \frac{f_1 \cdot \frac{1}{||x - x_1||^p} + f_2 \cdot \frac{1}{||x - x_2||^p} + ... + f_N \cdot \frac{1}{||x - x_N||^p}}{\frac{1}{||x - x_1||^p} + \frac{1}{||x - x_2||^p} + ... + \frac{1}{||x - x_N||^p}} =$$

$$= \frac{f_1 + f_2 \cdot \dfrac{||x - x_1||^p}{||x - x_2||^p} + ... + f_N \cdot \dfrac{||x - x_1||^p}{||x - x_N||^p}}{1 + \dfrac{||x - x_1||^p}{||x - x_2||^p} + ... + \dfrac{||x - x_1||^p}{||x - x_N||^p}} \to \frac{f_1}{1} = f_1$$

Moreover, after a bit longer calculations, it can be proved that:

Theorem: $\partial_1 f(x), \partial_2 f(x) \to 0$, whenever $x \to x_k$, for arbitrary interpolation point x_k:

This means that the Shepard method results in an interpolation surfaces that exhibits 'flat' regions in the vicinity of the interpolation points.

The method is simple, easily programmable and numerically stable. If $||x|| \to +\infty$, then the interpolated values $f(x)$ remain bounded, moreover, in this case $f(x) \to \dfrac{1}{N} \cdot \sum_{j=1}^{N} f_j$ (why?). However, if a given function is to be approximated by its Shepard interpolant belonging to a predefined set of interpolation points, the accuracy of the method is low.

To illustrate the method, consider the test function defined by the formula

$$f(x, y) := \sin \pi x \cdot \sin \pi y$$

defined on the unit square $\Omega := \{(x, y) \in \mathbf{R}^2 : \ 0 \leq x, y \leq 1\}$ (where we used the more familiar notations x and y for the spatial variables). Define 30 interpolation points in the unit square in a random way, and compute the Shepard interpolation function with the choice $p := 2$. The figure below shows the original function f and the Shepard interpolation function. Observe that the interpolation function poorly approximates the original test function. In the next figure, we used also 30 randomly selected interpolation points, but the parameter p was set to the value 4.

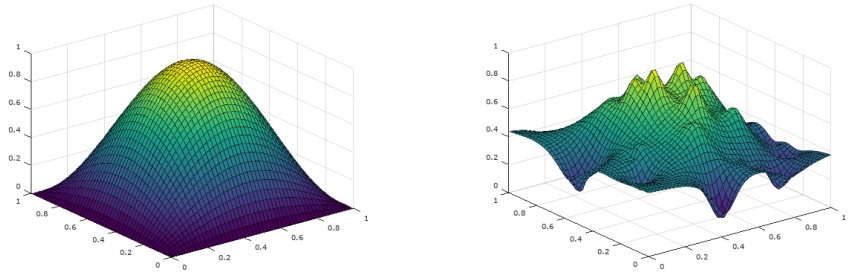

Figure 10: Scattered data interpolation with 30 randomly chosen interpolation points. Test surface (left) and the Shepard interpolant (right) with $p = 2$

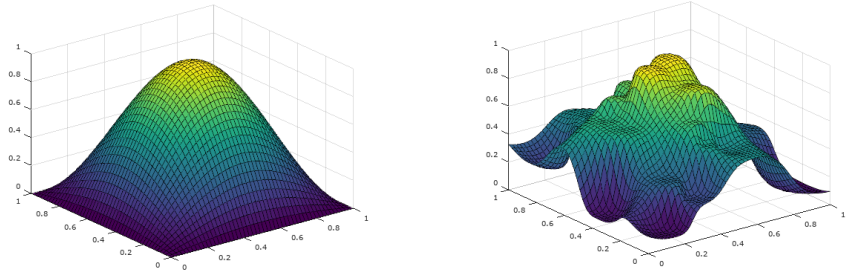

Figure 11: Scattered data interpolation with 30 randomly chosen interpolation points. Test surface (left) and the Shepard interpolant (right) with $p = 4$

8.3 The method of radial basis functions

Let $x_1, x_2, ..., x_N \in \mathbf{R}^2$ be given it interpolation points on the plane, and let $f_1, f_2, ..., f_N \in \mathbf{R}$ be given values associated to the interpolation points. Let Φ be a predefined radial (i.e. circularly symmetric) function, i.e. suppose that $\Phi(x)$ depends only on $||x||$. Seek the interpolation function is the following form:

$$f(x) := \sum_{j=1}^{N} \alpha_j \cdot \Phi(x - x_j)$$

The a priori unknown coefficients α_j can be computed by solving the system

of interpolation equations:

$$\sum_{j=1}^{N} \alpha_j \cdot \Phi(x_k - x_j) = f_k \qquad (k = 1, 2, ..., N)$$

Some usual choices of Φ:

- $\Phi(x) := \sqrt{||x||^2 + c^2}$ (Method of multiquadrics, MQ)
  where c is a predefined scaling constant;
- $\Phi(x) := \frac{1}{\sqrt{||x||^2 + c^2}}$ (Inverse multiquadrics, iMQ)
  where c is a predefined scaling constant;
- $\Phi(x) := ||x||^2 \cdot \log ||x||$ (Thin plate splines, TPS)
  This is a 'self-controlled' method, which contains no scaling parameter. At the origin, the function is defined as a limit value here: $\Phi(\mathbf{0}) := 0$.
- $\Phi(x) := e^{-c^2||x||^2}$ (Gauss functions)
  where c is a predefined scaling factor. The value of c controls the size of the 'essential support' of Φ. The larger the value c, the narrower the essential support (where the function takes essentially nonzero values).

Numerical features:

- Very good accuracy;
- At each point of evaluation, the number of necessary arithmetic operations is $\mathcal{O}(N)$, but the computational cost of the calculation of the coefficients is $\mathcal{O}(N^3)$;
- In general, the calculation of the coefficients lead to a system of equations with fully populated and extremely ill-conditioned matrix.

For comparison, for the previous test function $f(x, y) := \sin \pi x \cdot \sin \pi y$ (using 30 randomly chosen interpolation points again), we computed also the interpolation function using the MQ interpolation with the choice of the parameter $c := 1$. It can be clearly seen that the accuracy of the interpolation is much better than that of the Shepard interpolation. The price of

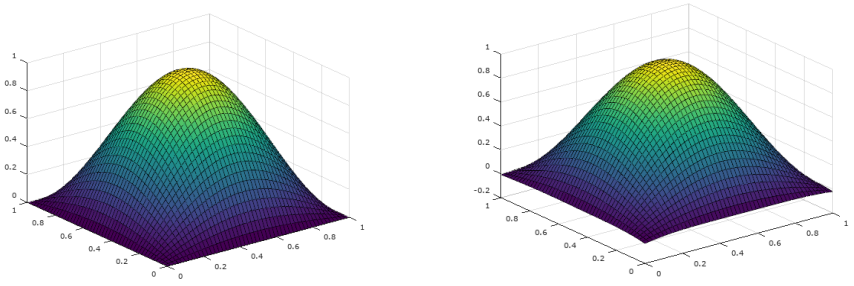

Figure 12: Scattered data interpolation with 30 randomly chosen interpolation points. Test surface (left) and the MQ-interpolant (right) with $c = 1$

the higher accuracy is the necessity of definition and solution of the interpolation equations, which may cause serious numerical problems when the number of interpolation points is great.

Remark: The method of thin plate splines has an interesting physical interpretation. It can be shown that if a thin plate made of an elastic material e.g. steel is fixed at the interpolation points at given heights, otherwise it is allowed to have a shape which is predicted by its elasticity, then the shape of the plate equals (with high precision) to the surface of the interpolation function defined by the TPS method.