Numerical Linear Algebra

Nikolai Chernov

1 Review of Linear Algebra

1.1 Matrices and vectors

The set of $m \times n$ matrices (m rows, n columns) with entries in a field $\mathbb{F}$ is denoted by $\mathbb{F}^{m\times n}$. We will only consider two fields: complex ($\mathbb{F} = \mathbb{C}$) and real ($\mathbb{F} = \mathbb{R}$). For any matrix $A \in \mathbb{F}^{m\times n}$, we denote its entries by

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}.$$

The vector space $\mathbb{F}^n$ consists of column vectors with n components:

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \in \mathbb{F}^n.$$

The product $y = Ax$ is a vector in $\mathbb{F}^m$:

$$y = Ax = \left[\begin{array}{c|c|c|c} a_1 & a_2 & \cdots & a_n \end{array}\right] \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = x_1 \begin{bmatrix} \\ a_1 \\ \\ \end{bmatrix} + x_2 \begin{bmatrix} \\ a_2 \\ \\ \end{bmatrix} + \cdots + x_n \begin{bmatrix} \\ a_n \\ \\ \end{bmatrix}$$

where $a_1, \ldots, a_n$ denote the columns of the matrix A. Note that Ax is a linear combination of the columns of A. Every matrix $A \in \mathbb{F}^{m\times n}$ defines a linear transformation

$$A\colon \mathbb{F}^n \to \mathbb{F}^m \qquad \text{by} \qquad x \mapsto Ax.$$

The *range* of A is a vector subspace of $\mathbb{F}^m$ defined by

$$\operatorname{Range} A = \{Ax\colon x \in \mathbb{F}^n\} \subset \mathbb{F}^m.$$

The range of A is a subspace of $\mathbb{F}^m$ spanned by the columns of A:

$$\operatorname{Range} A = \operatorname{span}\{a_1, \ldots, a_n\}.$$

The *rank* of A is defined by

$$\operatorname{rank} A = \dim(\operatorname{Range} A).$$

The *kernel* (also called the *nullspace*) of A is a vector subspace of $\mathbb{F}^n$:

$$\operatorname{Ker} A = \{x \colon Ax = 0\} \subset \mathbb{F}^n.$$

Note that

$$\dim(\operatorname{Range} A) + \dim(\operatorname{Ker} A) = n.$$

∘ The transformation A is *surjective* iff $\operatorname{Range} A = \mathbb{F}^m$.
∘ The transformation A is *injective* iff $\operatorname{Ker} A = \{0\}$.
∘ If A is *bijective*, then $m = n$ and we call A an *isomorphism*.

1.2 Square matrices

Every square matrix $A \in \mathbb{F}^{n\times n}$ defines a linear transformation $\mathbb{F}^n \to \mathbb{F}^n$, called an *operator* on $\mathbb{F}^n$. The *inverse* of a square matrix $A \in \mathbb{F}^{n\times n}$ is a square matrix $A^{-1} \in \mathbb{F}^{n\times n}$ uniquely defined by

$$A^{-1}A = AA^{-1} = I \qquad \text{(identity matrix)}.$$

A matrix $A \in \mathbb{F}^{n\times n}$ is said to be *invertible* (*nonsingular*) iff A^{-1} exists, otherwise the matrix is *noninvertible* (*singular*). The following are equivalent:

(a) A is invertible
(b) $\operatorname{rank} A = n$
(c) $\operatorname{Range} A = \mathbb{F}^n$
(d) $\operatorname{Ker} A = \{0\}$
(e) 0 is not an eigenvalue of A
(f) $\det A \neq 0$

Note that

$$(AB)^{-1} = B^{-1}A^{-1}$$

A matrix is A *upper triangular* if $a_{ij} = 0$ for all $i > j$. A matrix A is *lower triangular* if $a_{ij} = 0$ for all $i < j$. A matrix D is *diagonal* if $d_{ij} = 0$ for all $i \neq j$, and in that case we write $D = \operatorname{diag}\{d_{11}, \ldots, d_{nn}\}$. Note: if A and B are upper (lower) triangular, then so are AB and A^{-1} and B^{-1} (if the inverses exist).

We say that $\lambda \in \mathbb{F}$ is an *eigenvalue* for a matrix $A \in \mathbb{F}^{n\times n}$ with an *eigenvector* $x \in \mathbb{F}^n$ if

$$Ax = \lambda x \qquad \text{and} \qquad x \neq 0.$$

Eigenvalues are the roots of the characteristic polynomial

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

By the fundamental theorem of algebra, every polynomial of degree n with complex coefficients has exactly n complex roots $\lambda_1, \ldots, \lambda_n$ (counting multiplicities). This implies

$$C_A(\lambda) = \prod_{i=1}^{n} (\lambda - \lambda_i).$$

A square matrix $A \in \mathbb{F}^{n\times n}$ is *diagonalizable* (over $\mathbb{F}$) iff

$$A = X\Lambda X^{-1},$$

where $\Lambda = \operatorname{diag}\{\lambda_1, \ldots, \lambda_n\}$ is a diagonal matrix and $X \in \mathbb{F}^{n\times n}$. In this case $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of A and the columns $x_1, \ldots, x_n$ of the matrix X are the corresponding eigenvectors. To see the latter, rewrite the above equation in the form $AX = X\Lambda$ and note that

$$AX = \left[\begin{array}{c} \\ A \\ \\ \end{array}\right] \left[\begin{array}{c|c|c|c} x_1 & x_2 & \cdots & x_n \end{array}\right] = \left[\begin{array}{c|c|c|c} Ax_1 & Ax_2 & \cdots & Ax_n \end{array}\right]$$

and

$$X\Lambda = \left[\begin{array}{c|c|c|c} x_1 & x_2 & \cdots & x_n \end{array}\right] \left[\begin{array}{ccc} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{array}\right] = \left[\begin{array}{c|c|c|c} \lambda_1 x_1 & \lambda_2 x_2 & \cdots & \lambda_n x_n \end{array}\right],$$

therefore $Ax_1 = \lambda_1 x_1$, $Ax_2 = \lambda_2 x_2$, …, $Ax_n = \lambda_n x_n$.

Note that a real matrix $A \in \mathbb{R}^{n\times n}$ may be diagonalizable over $\mathbb{C}$ but not over $\mathbb{R}$. This happens, for example, when A has distinct complex eigenvalues.

The *trace* of a matrix $A \in \mathbb{F}^{n\times n}$ is defined by

$$\operatorname{tr} A = \sum_{i=1}^{n} a_{ii}$$

Trace has the following properties:

(a) $\operatorname{tr} AB = \operatorname{tr} BA$;

(b) if $A = X^{-1}BX$, then $\operatorname{tr} A = \operatorname{tr} B$;

(c) $\operatorname{tr} A = \lambda_1 + \cdots + \lambda_n$ (the sum of all complex eigenvalues).

1.3 Transposed and adjoint matrices

For any matrix $A = (a_{ij}) \in \mathbb{F}^{m\times n}$ we denote by $A^T = (a_{ji}) \in \mathbb{F}^{n\times m}$ the *transpose* of A. Note that

$$(AB)^T = B^T A^T, \qquad \left(A^T\right)^T = A.$$

If A is a square matrix, then

$$\det A^T = \det A, \qquad \left(A^T\right)^{-1} = \left(A^{-1}\right)^T.$$

For any matrix $A = (a_{ij}) \in \mathbb{C}^{m\times n}$ we denote by $A^* = (\bar{a}_{ji}) \in \mathbb{F}^{n\times m}$ the *adjoint* of A. Note that

$$(AB)^* = B^* A^*, \qquad \left(A^*\right)^* = A.$$

If A is a square matrix, then

$$\det A^* = \overline{\det A}, \qquad \left(A^*\right)^{-1} = \left(A^{-1}\right)^*.$$

For $A \in \mathbb{R}^{m\times n}$, we have $A^* = A^T$.

1.4 Norms

A *norm* on a vector space V over $\mathbb{C}$ is a real valued function $\|\cdot\|$ on V satisfying three axioms:

1. $\|v\| \geq 0$ for all $v \in V$ and $\|v\| = 0$ if and only if $v = 0$.
2. $\|cv\| = |c|\,\|v\|$ for all $c \in \mathbb{C}$ and $v \in V$.
3. $\|u + v\| \leq \|u\| + \|v\|$ for all $u, v \in V$ (**triangle inequality**).

If V is a vector space over $\mathbb{R}$, then an inner product is a function $V \to \mathbb{R}$ satisfying the same axioms, except $c \in \mathbb{R}$.

Some common norms on $\mathbb{C}^n$ and $\mathbb{R}^n$ are:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i| \qquad \text{(1-norm)}$$

$$\|x\|_2 = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{1/2} \qquad \text{(2-norm)}$$

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \qquad (p\text{-norm, } p \geq 1)$$

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i| \qquad (\infty\text{-norm})$$

The 2-norm in $\mathbb{R}^n$ corresponds to the Euclidean distance.

Some norms on $C[a, b]$, the space of continuous functions on $[a, b]$:

$$\|f\|_1 = \int_a^b |f(x)|\, dx \qquad \text{(1-norm)}$$

$$\|f\|_2 = \left(\int_a^b |f(x)|^2\, dx \right)^{1/2} \qquad \text{(2-norm)}$$

$$\|f\|_\infty = \max_{a \leq x \leq b} |f(x)| \qquad (\infty\text{-norm})$$

We define

$$\mathbb{S}_1 = \{v \in V \colon \|v\| = 1\} \qquad \textbf{(unit sphere)}$$

The vectors $v \in \mathbb{S}_1$ are called *unit vectors*. For any $v \neq 0$, the vector $u = v/\|v\|$ is a unit vector.

The space $\mathbb{C}^{m \times n}$ of matrices is isomorphic to $\mathbb{C}^{mn}$, hence we can define norms on it in a similar way. In particular, an analogue of the 2-norm

$$\|A\|_{\mathrm{F}} = \left(\sum_i \sum_j |a_{ij}|^2 \right)^{1/2}$$

is known as *Frobenius norm* of a matrix. Note that

$$\|A\|_F^2 = \operatorname{tr}(A^* A) = \operatorname{tr}(A A^*).$$

However, most important are matrix norms induced by vector norms:

1.5 Induced matrix norms

Let $A\colon \mathbb{F}^n \to \mathbb{F}^m$, and let the spaces $\mathbb{F}^n$ and $\mathbb{F}^m$ be equipped with certain norms, $\|\cdot\|$. Then

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

defines the *induced norm* (also called *operator norm*) of A. Respectively, we obtain $\|A\|_2$, $\|A\|_1$, and $\|A\|_\infty$, if the spaces $\mathbb{F}^n$ and $\mathbb{F}^m$ are equipped with $\|\cdot\|_2$, $\|\cdot\|_1$, and $\|\cdot\|_\infty$.

The supremum here is always attained and can be replaced by maximum. This follows from the compactness of $\mathbb{S}_1$ and the continuity of $\|\cdot\|$. For the 2-norm, this can be also proved by an algebraic argument, see Chapter 4.

There are norms on $\mathbb{C}^{n\times n}$ that are not induced by any norm on $\mathbb{C}^n$, for example $\|A\| = \max_{i,j} |a_{ij}|$ (see Exercise 1.1).

We have simple rules for computing $\|A\|_1$ and $\|A\|_\infty$:

$$\|A\|_1 = \max_{1\le j\le n} \sum_{i=1}^{m} |a_{ij}| \qquad \text{(maximum column sum)}$$

$$\|A\|_\infty = \max_{1\le i\le m} \sum_{j=1}^{n} |a_{ij}| \qquad \text{(maximum row sum)}$$

but there is no explicit formulas for $\|A\|_2$ in terms of the a_{ij}'s.

Any induced matrix norm satisfies

$$\|Ax\| \le \|A\|\,\|x\| \quad \text{and} \quad \|AB\| \le \|A\|\,\|B\|.$$

1.6 Inner products

Let V be a vector space over $\mathbb{C}$. An *inner product* on V is a function on $V \times V \to \mathbb{C}$, denoted by $\langle\cdot,\cdot\rangle$, satisfying four axioms:

1. $\langle u, v\rangle = \overline{\langle v, u\rangle}$ for all $u, v \in V$.
2. $\langle cu, v\rangle = c\langle u, v\rangle$ for all $c \in \mathbb{C}$ and $u, v \in V$.
3. $\langle u + v, w\rangle = \langle u, w\rangle + \langle v, w\rangle$ for all $u, v, w \in V$.
4. $\langle u, u\rangle \ge 0$ for all $u \in V$, and $\langle u, u\rangle = 0$ iff $u = 0$.

Note that $\langle u, v+w\rangle = \langle u, v\rangle + \langle u, w\rangle$ and $\langle u, cv\rangle = \bar{c}\langle u, v\rangle$.

If V is a vector space over $\mathbb{R}$, then an inner product is a function $V \times V \to \mathbb{R}$ satisfying the same axioms (except $c \in \mathbb{R}$, and there is no need to take a conjugate). A standard inner product in $\mathbb{C}^n$ is

$$\langle x, y\rangle = \sum_{i=1}^{n} x_i \bar{y}_i = x^T \bar{y} = y^* x.$$

A standard inner product in $\mathbb{R}^n$ is

$$\langle x, y\rangle = \sum_{i=1}^{n} x_i y_i = x^T y = y^T x.$$

A standard inner product in $C([a, b])$, the space of continuous functions, is:

$$\langle f, g\rangle = \int_a^b f(x)\, \bar{g}(x)\, dx$$

1.7 Cauchy-Schwarz inequality

Let V be an inner product space. Then

$$|\langle u, v\rangle| \leq \langle u, u\rangle^{1/2} \langle v, v\rangle^{1/2}$$

for all $u, v \in V$.

The equality holds if and only if u and v are linearly dependent.

Proof. If $v = 0$, then $\langle u, v\rangle = 0$ and $\langle v, v\rangle = 0$, so the claim is trivial. Assume that $v \neq 0$. Consider the function

$$\begin{aligned} f(z) &= \langle u - zv, u - zv\rangle \\ &= \langle u, u\rangle - z\langle v, u\rangle - \bar{z}\langle u, v\rangle + |z|^2 \langle v, v\rangle \end{aligned}$$

of a complex variable z. Let $z = re^{i\theta}$ and $\langle u, v\rangle = se^{i\varphi}$ be the polar forms of the complex numbers z and $\langle u, v\rangle$, respectively. Set $\theta = \varphi$ and assume that r varies from $-\infty$ to ∞, then

$$0 \leq f(z) = \langle u, u\rangle - 2sr + r^2 \langle v, v\rangle$$

Since this holds for all $r \in \mathbb{R}$, the discriminant has to be ≤ 0, i.e. $s^2 - \langle u, u\rangle\langle v, v\rangle \leq 0$. The equality case in the theorem corresponds to the zero discriminant, hence the above polynomial assumes a zero value, and hence $u = zv$ for some $z \in \mathbb{C}$. $\square$

1.8 Induced norms

If V is an inner product vector space, then

$$\|v\| = \langle v, v\rangle^{1/2} \qquad \textbf{(induced norm)}$$

defines a norm on V (to verify the triangle inequality, one can use Theorem 1.7). In vector spaces over $\mathbb{R}$, a norm is induced by an inner product if and only if the function

$$\langle u, v\rangle := \tfrac{1}{4}\left(\|u+v\|^2 - \|u-v\|^2\right) \qquad \textbf{(polarization identity)}$$

satisfies the axioms of inner products. (A similar but more complicated polarization identity holds in vector spaces over $\mathbb{C}$.)

1.9 Orthogonal vectors

Two vectors $u, v \in V$ are said to be *orthogonal* if $\langle u, v\rangle = 0$. In this case

$$\|u+v\|^2 = \|u\|^2 + \|v\|^2 \qquad \textbf{(Pythagorean theorem)}$$

Inductively, if $u_1, \ldots, u_k$ are mutually orthogonal, then

$$\|u_1 + \cdots + u_k\|^2 = \|u_1\|^2 + \cdots + \|u_k\|^2$$

If nonzero vectors $u_1, \ldots, u_k$ are mutually orthogonal, then they are linearly independent.

1.10 Orthonormal basis (ONB)

The set $\{u_1, \ldots, u_n\}$ is an *orthonormal basis* (ONB) in V, if it is a basis and all the vectors u_i are mutually orthogonal and have unit length (i.e., $\|u_i\| = 1$ for all i). Note: $\langle u_i, u_j\rangle = \delta_{ij}$ (this is the *Kronecker delta symbol* defined as follows: $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$).

If $\{u_1, \ldots, u_n\}$ is an ONB, then for any vector v

$$v = \sum_{i=1}^{n} \langle v, u_i\rangle\, u_i \qquad \textbf{(Fourier expansion)}$$

In other words, the numbers $\langle v, u_i\rangle$ (*Fourier coefficients*) are the coordinates of the vector v in this basis.

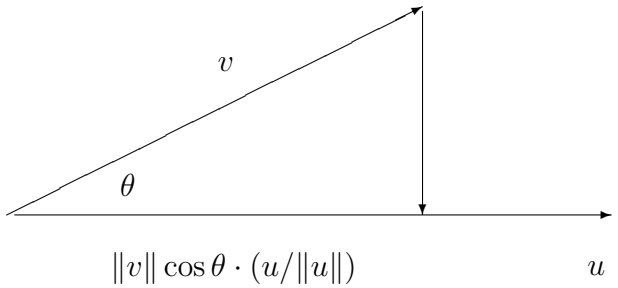
1.11 Orthogonal projection

Let $u, v \in V$, and $u \neq 0$. The *orthogonal projection* of v onto u is

$$\mathrm{Pr}_u v = \frac{\langle v, u \rangle}{\|u\|^2}\, u$$

Note that the vector $w \colon = v - \mathrm{Pr}_u v$ is orthogonal to u. Therefore, v is the sum of two vectors, $\mathrm{Pr}_u v$ (parallel to u), and w (orthogonal to u), see the diagram below.

Figure 1: Orthogonal projection of u onto v



In the real case, for any nonzero vectors $u, v \in V$ let

$$\cos\theta = \frac{\langle v, u \rangle}{\|u\|\,\|v\|} \qquad \textbf{(angle)}$$

By Section 1.7, we have $\cos\theta \in [-1, 1]$. Hence, there is a unique angle $\theta \in [0, \pi]$ with this value of cosine. It is called the *angle between* u and v.

Note that $\cos\theta = 0$ (i.e., $\theta = \pi/2$) if and only if u and v are orthogonal. Also, $\cos\theta = \pm 1$ if and only if u, v are proportional, i.e. $v = cu$. In that case the sign of c coincides with the sign of $\cos\theta$.

1.12 Gram-Schmidt orthogonalization

Let $u_1, \ldots, u_k$ be unit mutually orthogonal vectors. For $v \in V$, set

$$w = v - \sum_{i=1}^{k} \langle v, u_i \rangle u_i.$$

Then the vectors $u_1, \ldots, u_k, w$ are mutually orthogonal, and

$$\operatorname{span}\{u_1, \ldots, u_k, v\} = \operatorname{span}\{u_1, \ldots, u_k, w\}.$$

In particular, $w = 0$ if and only if $v \in \operatorname{span}\{u_1, \ldots, u_k\}$.

Next, let $\{v_1, \ldots, v_n\}$ be a basis in V. Define

$$w_1 = v_1 \qquad \text{and} \qquad u_1 = w_1/\|w_1\|,$$

and then inductively, for $k \geq 1$,

$$w_k = v_k - \sum_{i=1}^{k-1} \langle v_k, u_i \rangle \, u_i, \quad \text{and} \quad u_k = w_k/\|w_k\| \tag{GS}$$

This gives an orthogonal basis $\{u_1, \ldots, u_n\}$, which 'agrees' with the basis $\{v_1, \ldots, v_n\}$ in the following sense:

$$\operatorname{span}\{v_1, \ldots, v_k\} = \operatorname{span}\{u_1, \ldots, u_k\} \quad \forall\ 1 \leq k \leq n$$

As a corollary, we conclude that every finite dimensional vector space with an inner product has an ONB. Furthermore, every set of orthonormal vectors $\{u_1, \ldots, u_k\}$ can be extended to an ONB.

1.13 Legendre polynomials

Let $V = \mathbb{P}_n(\mathbb{R})$, the space of real polynomials of degree $\leq n$, with the inner product given by $\langle f, g \rangle = \int_0^1 f(x)g(x)\,dx$. Applying Gram-Schmidt orthogonalization to the basis $\{1, x, \ldots, x^n\}$ gives the first $n+1$ of the so called *Legendre polynomials*.

1.14 Orthogonal complement

Let $S \subset V$ be a subset (not necessarily a subspace). Then

$$S^\perp \colon = \{v \in V \colon \langle v, w \rangle = 0 \ \text{ for all } \ w \in S\}$$

is called the *orthogonal complement* to S. Note that $S^\perp$ is a vector subspace of V. Also, $(W^\perp)^\perp \subset W$, and if V is finite dimensional, then $(W^\perp)^\perp = W$ (see Exercise 1.4).

If W is a finite dimensional subspace of V, then $V = W \oplus W^\perp$. To prove this, choose an ONB $\{u_1, \ldots, u_n\}$ of W (by 1.12), and then for any $v \in V$

$$v - \sum_{i=1}^{n} \langle v, u_i \rangle \, u_i \in W^\perp.$$

Note: the condition $\dim W < \infty$ is essential. Let $V = C[a, b]$ with the standard inner product and $W \subset V$ be the set of real polynomials restricted to the interval $[a, b]$. Then $W^\perp = \{0\}$, and at the same time $V \neq W$.

Note: if $\{u_1, \dots, u_n\}$ is an orthonormal subset of V, then

$$\|v\|^2 \geq \sum_{i=1}^{n} |\langle v, u_i \rangle|^2 \qquad \textbf{(Bessel's inequality)}$$

If $\{u_1, \dots, u_n\}$ is an ONB, then

$$\|v\|^2 = \sum_{i=1}^{n} |\langle v, u_i \rangle|^2$$

and, more generally,

$$\langle v, w \rangle = \sum_{i=1}^{n} \langle v, u_i \rangle \overline{\langle w, u_i \rangle} \qquad \textbf{(Parceval's identity)}$$

This follows from the Fourier expansion, see Section 1.10.

Parceval's identity can be written as follows. Suppose

$$v = \sum_{i=1}^{n} a_i u_i \qquad \text{and} \qquad w = \sum_{i=1}^{n} b_i u_i,$$

so that $(a_1, \dots, a_n)$ and $(b_1, \dots, b_n)$ are the coordinates of the vectors v and w, respectively, in the ONB $\{u_1, \dots, u_n\}$. Then

$$\langle v, w \rangle = \sum_{i=1}^{n} a_i \bar{b}_i.$$

In particular,

$$\|v\|^2 = \langle v, v \rangle = \sum_{i=1}^{n} a_i \bar{a}_i = \sum_{i=1}^{n} |a_i|^2.$$

Exercise 1.1. Show that the norm $\|A\| = \max_{i,j} |a_{ij}|$ on the space of $n \times n$ real matrices is not induced by any vector norm in $\mathbb{R}^n$. Hint: use inequalities from Section 1.5.

Exercise 1.2. Prove the *Neumann lemma*: if $\|A\| < 1$, then $I - A$ is invertible. Here $\|\cdot\|$ is a norm on the space of $n \times n$ matrices induced by a vector norm.

Exercise 1.3. Let V be an inner product space, and $\|\cdot\|$ denote the norm induced by the inner product. Prove the *parallelogram law*

$$\|u+v\|^2 + \|u-v\|^2 = 2\|u\|^2 + 2\|v\|^2.$$

Based on this, show that the norms $\|\cdot\|_1$ and $\|\cdot\|_\infty$ in $\mathbb{C}^2$ are not induced by any inner products.

Exercise 1.4. Let $W \subset V$ be a subspace of an inner product space V.

(i) Prove that $W \subset (W^\perp)^\perp$.

(ii) If, in addition, V is finite dimensional, prove that $W = (W^\perp)^\perp$.

Exercise 1.5. Let $\{u_1, \ldots, u_n\}$ be an ONB in $\mathbb{C}^n$. Assuming that n is even, compute

$$\|u_1 - u_2 + u_3 - \cdots + u_{n-1} - u_n\|$$

2 Unitary matrices

2.1 Isometry (definition)

Let V and W be two inner product spaces (both real or both complex). An isomorphism $T\colon V \to W$ is called an *isometry* if it preserves the inner product, i.e.

$$\langle Tv, Tw\rangle = \langle v, w\rangle$$

for all $v, w \in V$. In this case V and W are said to be *isometric*.

2.2 Characterization of isometries - I

T is an isometry iff T preserves the induced norm, i.e. $\|Tv\| = \|v\|$ for all vectors $v \in V$.

(This follows from Polarization Identity; see Section 1.8.)

Moreover, T is an isometry iff $\|Tu\| = \|u\|$ for all *unit* vectors $u \in V$.

(Because every non-zero vector v can be normalized by $u = v/\|v\|$.)

2.3 Characterization of isometries - II

Let $\dim V < \infty$. A linear transformation $T\colon V \to W$ is an isometry if and only if there exists an ONB $\{u_1, \ldots, u_n\}$ in V such that $\{Tu_1, \ldots, Tu_n\}$ is an ONB in W.

(This follows from Parceval's Identity; see Section 1.14.)

2.4 Identification of finite-dimensional inner product spaces

Finite dimensional inner product spaces V and W (over the same field) are *isometric* if $\dim V = \dim W$.

(This follows from Section 2.3.)

As a result, we can make the following useful identifications:

- © All complex n-dimensional inner product spaces can be identified with $\mathbb{C}^n$ equipped with the standard inner product $\langle x, y\rangle = y^* x$.
- ® All real n-dimensional inner product spaces can be identified with $\mathbb{R}^n$ equipped with the standard inner product $\langle x, y\rangle = y^T x$.

These identifications allow us to focus on the study of the standard spaces $\mathbb{C}^n$ and $\mathbb{R}^n$ equipped with the standard inner product. Isometries $\mathbb{C}^n \to \mathbb{C}^n$ and $\mathbb{R}^n \to \mathbb{R}^n$ are operators that, in a standard basis $\{e_1, \ldots, e_n\}$, are given by matrices of a special type, as defined below.

2.5 Unitary and orthogonal matrices (definition)

A matrix $Q \in \mathbb{C}^{n\times n}$ is said to be *unitary* if $Q^*Q = I$, i.e., $Q^* = Q^{-1}$.

A matrix $Q \in \mathbb{R}^{n\times n}$ is said to be *orthogonal* if $Q^TQ = I$, i.e., $Q^T = Q^{-1}$.

Note:

$$Q \text{ is unitary} \Leftrightarrow Q^* \text{ is unitary} \Leftrightarrow Q^T \text{ is unitary} \Leftrightarrow \bar{Q} \text{ is unitary}$$

In the real case: Q is orthogonal $\Leftrightarrow Q^T$ is orthogonal.

2.6 Theorem

Ⓒ The linear transformation of $\mathbb{C}^n$ defined by a matrix $Q \in \mathbb{C}^{n\times n}$ is an isometry (preserves the standard inner product) iff Q is unitary.

Ⓡ The linear transformation of $\mathbb{R}^n$ defined by a matrix $Q \in \mathbb{R}^{n\times n}$ is an isometry (preserves the standard inner product) iff Q is orthogonal.

Proof. $\langle Qx, Qy\rangle = (Qy)^*Qx = y^*Q^*Qx = y^*x = \langle x, y\rangle$. □

2.7 Group property

If $Q_1, Q_2 \in \mathbb{C}^{n\times n}$ are unitary matrices, then so is Q_1Q_2 (as well as Q_1^{-1} and Q_2^{-1}). Thus, unitary $n \times n$ matrices make a group, denoted by $U(n)$. Similarly, orthogonal $n \times n$ matrices make a group, denoted by $O(n)$.

2.8 Examples of orthogonal matrices

$Q = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ defines a reflection across the diagonal line $y = x$;

$Q = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ is a counterclockwise rotation by angle θ.

2.9 Characterizations of unitary and orthogonal matrices

A matrix $Q \in \mathbb{C}^{n\times n}$ is unitary iff its columns make an ONB in $\mathbb{C}^n$.

A matrix $Q \in \mathbb{C}^{n\times n}$ is unitary iff its rows make an ONB in $\mathbb{C}^n$.

A matrix $Q \in \mathbb{R}^{n\times n}$ is orthogonal iff its columns make an ONB in $\mathbb{R}^n$.

A matrix $Q \in \mathbb{R}^{n\times n}$ is orthogonal iff its rows make an ONB in $\mathbb{R}^n$.

2.10 Theorem
If Q is unitary/orthogonal, then $|\det Q| = 1$.

Proof. $1 = \det I = \det Q^*Q = \det Q^* \cdot \det Q = |\det Q|^2$. □

Orthogonal matrices have determinant 1 or -1. Orthogonal $n \times n$ matrices with determinant 1 make a subgroup of $O(n)$, denoted by $SO(n)$.

2.11 Theorem
If λ is an eigenvalue of a unitary/orthogonal matrix, then $|\lambda| = 1$.

Proof. If $Qx = \lambda x$ for some $x \neq 0$, then $\langle x, x\rangle = \langle Qx, Qx\rangle = \langle \lambda x, \lambda x\rangle = \lambda\bar{\lambda}\langle x, x\rangle = |\lambda|^2\langle x, x\rangle$, so that $|\lambda|^2 = 1$.

Note: orthogonal matrices $Q \in \mathbb{R}^{n\times n}$ may not have any real eigenvalues; see the rotation matrix in Example 2.8.

2.12 Theorem
Let $T\colon V \to V$ be an isometry and $\dim V < \infty$. If a subspace $W \subset V$ is invariant under T, i.e. $TW \subset W$, then so is its orthogonal complement $W^\perp$, i.e. $TW^\perp \subset W^\perp$.

Proof. Since $\operatorname{Ker} T = \{0\}$, we have $TW = W$, hence $\forall w \in W\ \exists w' \in W\colon\ Tw' = w$. Now, if $v \in W^\perp$, then $\forall w \in W$ we have $\langle w, Tv\rangle = \langle Tw', Tv\rangle = \langle w', v\rangle = 0$, hence $Tv \in W^\perp$.

2.13 Corollary
For any isometry T of a finite dimensional <u>complex</u> space V there is an ONB of V consisting of eigenvectors of T.

Proof. Use induction on the dimension of the space and Theorem 2.12.

Note: the above corollary is **not true** for <u>real</u> vector spaces.

2.14 Lemma
Every operator $T\colon V \to V$ on a finite dimensional <u>real</u> space V has either a one-dimensional or a two-dimensional invariant subspace $W \subset V$.

Proof. Let T be represented by a matrix $A \in \mathbb{R}^{n\times n}$ in some basis. If A has a real eigenvalue, then $Ax = \lambda x$ with some $x \neq 0$, and we get a one-dimensional

invariant subspace span$\{x\}$. If A has no real eigenvalues, then the matrix A, *considered as a complex matrix*, has a complex eigenvalue $\lambda = a + \mathbf{i}b$, with $a, b \in \mathbb{R}$ and $\mathbf{i} = \sqrt{-1}$, and a complex eigenvector $x + \mathbf{i}y$, with $x, y \in \mathbb{R}^n$. The equation

$$A(x + \mathbf{i}y) = (a + \mathbf{i}b)(x + \mathbf{i}y) = (ax - by) + (bx + ay)\mathbf{i}$$

can be written as

$$Ax = ax - by$$
$$Ay = bx + ay$$

Thus, the two-dimensional space span$\{x, y\}$ is invariant. □

Note: x and y are linearly dependent iff $b = 0$, i.e. $\lambda \in \mathbb{R}$.

2.15 Theorem

Let $T\colon V \to V$ be an isometry of a finite dimensional real space V. Then $V = V_1 \oplus \cdots \oplus V_m$, where V_i are mutually orthogonal subspaces, each V_i is T-invariant, and either $\dim V_i = 1$ or $\dim V_i = 2$.

Proof. Use induction on $\dim V$ and apply Sections 2.12 and 2.14.

Note: the restriction of the operator T to each of the two-dimensional invariant subspaces is simply a rotation by some angle (as in Example 2.8); this follows from Exercises 2.3 and 2.4.

Recall that two $n \times n$ matrices A and B are *similar* (usually denoted by $A \sim B$) if there exists an invertible matrix C such that $B = C^{-1}AC$. Two matrices are similar if they represent the same linear operator on an n-dimensional space, but under two different bases. In that case C is the change of basis matrix.

2.16 Unitary and orthogonal equivalence

© Two complex matrices $A, B \in \mathbb{C}^{n\times n}$ are said to be *unitary equivalent* if $B = P^{-1}AP$ for some unitary matrix P. This can be also written as $B = P^*AP$.

® Two real matrices $A, B \in \mathbb{R}^{n\times n}$ are said to be *orthogonally equivalent* if $B = P^{-1}AP$ for some orthogonal matrix P. This can be also written as $B = P^TAP$.

Two complex/real matrices are unitary/orthogonally equivalent if they represent the same linear operator on a complex/real n-dimensional inner product space, but under two different orthonormal bases (ONBs). Then P is the change of basis matrix, which must be unitary/orthogonal, because it changes an ONB to another ONB.

In this course we will mostly deal with ONBs, thus unitary/orthogonal equivalence will play the same major role as similarity in Linear Algebra. In particular, for any type of matrices we will try to find simplest matrices which are unitary/orthogonal equivalent to matrices of the given type.

2.17 Unitary matrices in their simples form

Any unitary matrix $Q \in \mathbb{C}^{n\times n}$ is unitary equivalent to a diagonal matrix $D = \operatorname{diag}\{d_1, \ldots, d_n\}$, whose diagonal entries belong to the unit circle, i.e. $|d_i| = 1$ for $1 \le i \le n$.

(This follows from Theorem 2.13.)

2.18 Orthogonal matrices in their simples form

Any orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ is orthogonally equivalent to a block-diagonal matrix

$$B = \begin{bmatrix} R_{11} & 0 & \cdots & 0 \\ 0 & R_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{mm} \end{bmatrix}$$

where R_{ii} are 1×1 and 2×2 diagonal blocks. Furthermore, all 1×1 blocks are either $R_{ii} = +1$ or $R_{ii} = -1$, and the 2×2 blocks are rotation matrices

$$R_{ii} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix}$$

(This follows from Theorem 2.15.)

Exercise 2.1. Let $A \in \mathbb{C}^{m\times n}$. Show that

$$\|UA\|_2 = \|AV\|_2 = \|A\|_2$$

for any unitary matrices $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$.

Exercise 2.2. Let $A \in \mathbb{C}^{m\times n}$. Show that

$$\|UA\|_F = \|AV\|_F = \|A\|_F$$

for any unitary matrices $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$. Here $\|\cdot\|_F$ stands for the Frobenius norm.

Exercise 2.3. Let Q be a real orthogonal 2×2 matrix and $\det Q = 1$.
Show that

$$Q = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

for some $\theta \in [0, 2\pi)$.

In geometric terms, Q represents a rotation of $\mathbb{R}^2$ by angle θ.

Exercise 2.4. Let Q be a real orthogonal 2×2 matrix and $\det Q = -1$.
Show that

$$Q = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix}$$

for some $\theta \in [0, 2\pi)$.
Also prove that $\lambda_1 = 1$ and $\lambda_2 = -1$ are the eigenvalues of Q.

In geometric terms, Q represents a reflection of $\mathbb{R}^2$ across the line spanned by the eigenvector corresponding to $\lambda_1 = 1$.

3 Adjoint and self-adjoint matrices

Beginning with this chapter, we will always deal with finite dimensional inner product spaces (unless stated otherwise). Recall that all such spaces can be identified with $\mathbb{C}^n$ or $\mathbb{R}^n$ equipped with the standard inner product (Section 2.4). Thus we will mostly deal with these standard spaces.

3.1 Adjoint matrices

Recall that every complex matrix $A \in \mathbb{C}^{m\times n}$ defines a linear transformation $\mathbb{C}^n \to \mathbb{C}^m$. The adjoint matrix $A^* \in \mathbb{C}^{n\times m}$ defines a linear transformation $\mathbb{C}^m \to \mathbb{C}^n$. Furthermore, for any $x \in \mathbb{C}^n$ and $y \in \mathbb{C}^m$ we have

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

Likewise, $\langle y, Ax\rangle = \langle A^*y, x\rangle$. In plain words, A can be moved from one argument of the inner product to the other, but must be changed to A^*.

Every real matrix $A \in \mathbb{R}^{m\times n}$ defines a linear transformation $\mathbb{R}^n \to \mathbb{R}^m$. The transposed matrix $A^T \in \mathbb{R}^{n\times m}$ defines a linear transformation $\mathbb{R}^m \to \mathbb{R}^n$. Furthermore, for any $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ we have

$$\langle Ax, y\rangle = y^TAx = (A^Ty)^Tx = \langle x, A^Ty\rangle$$

Thus in the real case, too, A can be moved from one argument of the inner product to the other, but must be changed to $A^* = A^T$.

3.2 Adjoint transformations

More generally, for any linear transformation $T\colon V \to W$ of two inner product vector spaces V and W (both real or both complex) there exists a unique linear transformation $T^*\colon W \to V$ such that $\forall v \in V,\ \forall w \in W$

$$\langle Tv, w\rangle = \langle v, T^*w\rangle.$$

T^* is called the *adjoint* of T.

The existence and uniqueness of T^* can be proved by a general argument, avoiding identification of the spaces V and W with $\mathbb{C}^n$ or $\mathbb{R}^n$. The argument is outlined below, in Sections 3.3 to 3.6. (This part of the course can be skipped.)

3.3 Riesz representation theorem

Let $f \in V^*$, i.e. f is a linear functional on V. Then there is a unique vector $u \in V$ such that

$$f(v) = \langle v, u \rangle \qquad \forall v \in V$$

Proof. Let $B = \{u_1, \ldots, u_n\}$ be an ONB in V. Then for any $v = \sum c_i u_i$ we have $f(v) = \sum c_i f(u_i)$ by linearity. Also, for any $u = \sum d_i u_i$ we have $\langle v, u \rangle = \sum c_i \bar{d}_i$. Hence, the vector $u = \sum \overline{f(u_i)}\, u_i$ will suffice. To prove the uniqueness of u, assume $\langle v, u \rangle = \langle v, u' \rangle$ for all $v \in V$. Setting $v = u - u'$ gives $\langle u - u', u - u' \rangle = 0$, hence $u = u'$.

3.4 Corollary

The identity $f \leftrightarrow u$ established in the previous theorem is "quasi-linear" in the following sense: $f_1 + f_2 \leftrightarrow u_1 + u_2$ and $cf \leftrightarrow \bar{c}u$. In the real case, it is perfectly linear, though, and hence it is an isomorphism between V^* and V.

3.5 Remark

If $\dim V = \infty$, then Theorem 3.3 fails. Consider, for example, $V = C[0,1]$ (real functions) with the inner product $\langle F, G \rangle = \int_0^1 F(x)G(x)\,dx$. Pick a point $t \in [0,1]$. Let $f \in V^*$ be a linear functional defined by $f(F) = F(t)$. It does not correspond to any $G \in V$ so that $f(F) = \langle F, G \rangle$. In fact, the lack of such functions G has led mathematicians to the concept of *delta-functions*: a delta-function $\delta_t(x)$ is "defined" by three requirements: $\delta_t(x) \equiv 0$ for all $x \neq t$, $\delta_t(t) = \infty$ and $\int_0^1 F(x)\delta_t(x)\,dx = F(t)$ for every $F \in C[0,1]$.

3.6 Adjoint transformation

Let $T\colon V \to W$ be a linear transformation. Then there is a unique linear transformation $T^*\colon W \to V$ such that $\forall v \in V$ and $\forall w \in W$

$$\langle Tv, w \rangle = \langle v, T^* w \rangle.$$

Proof. Let $w \in W$. Then $f(v)\colon = \langle Tv, w \rangle$ defines a linear functional $f \in V^*$. By the Riesz representation theorem, there is a unique $v' \in V$ such that $f(v) = \langle v, v' \rangle$. Then we define T^* by setting $T^* w = v'$. The linearity of T^* is a routine check. Note that in the complex case the conjugating bar appears twice and thus cancels out. The uniqueness of T^* is obvious. □

We return to our main course.

3.7 Theorem

Let $T\colon V \to W$ be a linear transformation. Then

$$\operatorname{Ker} T^* = (\operatorname{Range} T)^\perp$$

In particular, $W = \operatorname{Range} T \oplus \operatorname{Ker} T^*$.

Proof is a routine check.

Next we will focus on the case $V = W$, in which T is an operator.

3.8 Selfadjoint operators and matrices

A linear operator $T\colon V \to V$ is said to be *selfadjoint* if $T^* = T$.

A square matrix A is said to be *selfadjoint* if $A^* = A$.

In the real case, this is equivalent to $A^T = A$, i.e. A is a symmetric matrix. In the complex case, selfadjoint matrices are called *Hermitian matrices*.

3.9 Examples

The matrix $\left[\begin{smallmatrix} 1 & 3 \\ 3 & 2 \end{smallmatrix}\right]$ is symmetric. The matrix $\left[\begin{smallmatrix} 1 & 3+i \\ 3-i & 2 \end{smallmatrix}\right]$ is Hermitian.

The matrices $\left[\begin{smallmatrix} 1 & 3+i \\ 3+i & 2 \end{smallmatrix}\right]$ and $\left[\begin{smallmatrix} 1+i & 3+i \\ 3-i & 2 \end{smallmatrix}\right]$ are **not** Hermitian (why?).

Note: the diagonal components of a Hermitian matrix must be real numbers!

3.10 Unitary/orthogonal equivalence for self-adjoint matrices

© If A is a complex Hermitian matrix unitary equivalent to B, then B is also a complex Hermitian matrix.

® If A is a real symmetric matrix orthogonally equivalent to B, then B is also a real symmetric matrix.

3.11 Theorem

Let T be a selfadjoint operator and a subspace W be T-invariant, i.e. $TW \subset W$. Then $W^\perp$ is also T-invariant, i.e. $TW^\perp \subset W^\perp$.

Proof. If $v \in W^\perp$, then for any $w \in W$ we have $\langle Tv, w\rangle = \langle v, Tw\rangle = 0$, so $Tv \in W^\perp$.

3.12 Spectral theorem

Let $T\colon V \to V$ be a selfadjoint operator. Then there is an ONB consisting of eigenvectors of T, and all the eigenvalues of T are real numbers.

Proof. First let V be a complex space. Then we use induction on $n = \dim V$ and apply Theorem 3.11 to construct an ONB of eigenvectors, as we did in Corollary 10.11.

Now T is represented in the canonical basis $\{e_1, \dots, e_n\}$ by a Hermitian matrix A. In the (just constructed) ONB of eigenvectors, T is represented by a diagonal matrix $D = \operatorname{diag}\{d_1, \dots, d_n\}$, and $d_1, \dots, d_n$ are the eigenvalues of T. Since A and D are unitary equivalent, D must be Hermitian, too (by Section 3.10), hence its components $d_1, \dots, d_n$ are real numbers. This completes the proof of Spectral Theorem for complex spaces.

Before we proceed to real spaces, we need to record a useful fact about complex Hermitian matrices. Every Hermitian matrix $A \in \mathbb{C}^{n\times n}$ defines a selfadjoint operator on $\mathbb{C}^n$. As we just proved, there exists an ONB of eigenvectors. Hence A is unitary equivalent to a diagonal matrix $D = \operatorname{diag}\{d_1, \ldots, d_n\}$. Since D is Hermitian, its diagonal entries $d_1, \ldots, d_n$ are real numbers. Therefore

Every Hermitian matrix has only real eigenvalues

Now let V be a real space. Now in the canonical basis $\{e_1, \ldots, e_n\}$ the self-adjoint operator $T\colon V \to V$ is represented by a real symmetric matrix A. The latter, *considered as a complex matrix*, is Hermitian, thus it has only real eigenvalues (see above). Hence the construction of an ONB of eigenvectors, as done in Corollary 10.11, works again. The proof is now complete. □

Note: For every real symmetric matrix A, the characteristic polynomial is $C_A(x) = \prod_i (x - \lambda_i)$, where all λ_i's are real numbers.

3.13 Corollary

© A complex matrix is Hermitian iff it is unitary equivalent to a diagonal matrix with real diagonal entries.

® A real matrix is symmetric iff it is orthogonally equivalent to a diagonal matrix (whose entries are automatically real).

More generally: if an operator $T\colon V \to V$ has an ONB of eigenvectors, and all its eigenvalues are real numbers, then T is self-adjoint.

3.14 Remark

Let $A = QDQ^*$, where Q is a unitary matrix and D is a diagonal matrix. Denote by q_i the ith column of Q and by d_i the ith diagonal entry of D. Then

$$Aq_i = d_i q_i, \qquad 1 \le i \le n$$

i.e. the columns of Q are the eigenvectors of A, whose eigenvalues are the corresponding diagonal components of D (see Section 1.2).

3.15 Theorem

If an operator T is selfadjoint and invertible, then so is T^{-1}. If a matrix A is selfadjoint and nonsingular, then so is A^{-1}.

Proof. By Spectral Theorem 3.12, there is an ONB consisting of eigenvectors of T, and the eigenvalues of T are real numbers. Now T^{-1} has the same eigenvectors, and its eigenvalues are the reciprocals of those of T, hence they are real numbers, too. Therefore, T^{-1} is selfadjoint.

3.16 Projections

Let $V = W_1 \oplus W_2$, i.e., a vector space is a direct sum of two subspaces. Recall that for each $v \in V$ there is a unique decomposition $v = w_1 + w_2$ with $w_1 \in W_1$ and $w_2 \in W_2$.

The operator $P\colon V \to V$ defined by

$$Pv = P(w_1 + w_2) = w_2$$

is called *projection* (or *projector*) of V *on* W_2 *along* W_1.

Note that $\operatorname{Ker} P = W_1$ and $\operatorname{Range} P = W_2$. Also note that $P^2 = P$.

3.17 Projections (alternative definition)

An operator $P\colon V \to V$ is a projection iff $P^2 = P$.

Proof. $P^2 = P$ implies that for every $v \in V$ we have $P(v - Pv) = 0$, so

$$w_1 \colon = v - Pv \in \operatorname{Ker} P$$

Denoting $w_2 = Pv$ we get $v = w_1 + w_2$ with $w_1 \in \operatorname{Ker} P$ and $w_2 \in \operatorname{Range} P$. Furthermore,

$$Pv = Pw_1 + Pw_2 = 0 + P(Pv) = P^2 v = Pv = w_2$$

We also note that $\operatorname{Ker} P \cap \operatorname{Range} P = \{0\}$. Indeed, for any $v \in \operatorname{Range} P$ we have $Pv = v$ (as above) and for any $v \in \operatorname{Ker} P$ we have $Pv = 0$. Thus $v \in \operatorname{Ker} P \cap \operatorname{Range} P$ implies $v = 0$. Hence $V = \operatorname{Ker} P \oplus \operatorname{Range} P$. □

An extra note: if $V = W_1 \oplus W_2$, then there is a unique projection P_1 on W_1 along W_2 and a unique projection P_2 on W_2 along W_1, and we have $P_1 + P_2 = I$.

3.18 Orthogonal projections

Let V be an inner product vector space (not necessarily finite dimensional) and $W \subset V$ a finite dimensional subspace. Then the projection on W along $W^\perp$ is called the *orthogonal projection* on W.

Note: the assumption $\dim W < \infty$ is made to ensure that $V = W \oplus W^\perp$, recall Section 1.14.

3.19 Theorem

Let P be a projection. Then P is an orthogonal projection if and only if P is selfadjoint.

Proof. By definition, P be a projection on W_2 along W_1, and $V = W_1 \oplus W_2$. For any vectors $v, w \in V$ we have $v = v_1 + v_2$ and $w = w_1 + w_2$ with some $v_i, w_i \in W_i$, $i = 1, 2$. Now, if P is orthogonal, then $\langle Pv, w\rangle = \langle v_2, w\rangle = \langle v_2, w_2\rangle = \langle v, w_2\rangle = \langle v, Pw\rangle$. If P is not orthogonal, then there are $v_1 \in W_1$, $w_2 \in W_2$ so that $\langle v_1, w_2\rangle \neq 0$. Then $\langle v_1, Pw_2\rangle \neq 0 = \langle Pv_1, w_2\rangle$.

Exercise 3.1. Let V be an inner product space and $W \subset V$ a finite dimensional subspace with ONB $\{u_1, \ldots, u_n\}$. For every $x \in V$ define

$$P(x) = \sum_{i=1}^{n} \langle x, u_i\rangle u_i$$

(i) Prove that $x - P(x) \in W^\perp$, hence P is the orthogonal projection onto W.
(ii) Prove that $\|x - P(x)\| \leq \|x - z\|$ for every $z \in W$, and that if $\|x - P(x)\| = \|x - z\|$ for some $z \in W$, then $z = P(x)$.

Exercise 3.2. (JPE, May 1999) Let $P \in \mathbb{C}^{n\times n}$ be a projector. Show that $\|P\|_2 \geq 1$ with equality if and only if P is an orthogonal projector.

4 Positive definite matrices

4.1 Bilinear forms

A *bilinear form* on a complex vector space V is a mapping $f\colon V \times V \to \mathbb{C}$ such that

$$\begin{aligned} f(u_1 + u_2, v) &= f(u_1, v) + f(u_2, v) \\ f(cu, v) &= cf(u, v) \\ f(u, v_1 + v_2) &= f(u, v_1) + f(u, v_2) \\ f(u, cv) &= \bar{c}f(u, v) \end{aligned}$$

for all vectors $u, v, u_i, v_i \in V$ and scalars $c \in \mathbb{C}$. In other words, f is linear in the first argument and conjugate linear in the second.

A bilinear form on a real vector space is a mapping $f\colon V \times V \to \mathbb{R}$ that satisfies the same properties, except c is a real scalar and so $\bar{c} = c$.

4.2 Theorem

Let V be a finite dimensional inner product space. Then for every bilinear form f on V, then there is a unique linear operator $T\colon V \to V$ such that

$$f(u, v) = \langle Tu, v\rangle \qquad \forall u, v \in V$$

Proof. For every $v \in V$ the function $g(u) = f(u, v)$ is linear in u, so by the Riesz representation theorem 3.3 there is a vector $w \in V$ such that $f(u, v) = \langle u, w\rangle$. Define a map $S\colon V \to V$ by $Sv = w$. It is then a routine check that S is linear. Setting $T = S^*$ proves the existence. The uniqueness is obvious. □

4.3 Corollary

© Every bilinear form on $\mathbb{C}^n$ can be represented by $f(x, y) = \langle Ax, y\rangle$ with $A \in \mathbb{C}^{n\times n}$.

® Every bilinear form on $\mathbb{R}^n$ can be represented by $f(x, y) = \langle Ax, y\rangle$ with $A \in \mathbb{R}^{n\times n}$.

Bilinear forms generalize the notion of an inner product. In order for a bilinear form to become an inner product, though, it needs two additional properties: conjugate symmetry $f(x, y) = \overline{f(y, x)}$ and the positivity $f(x, x) > 0$ for all $x \neq 0$. We will see next what this means in terms of the matrix A that defines the bilinear form (in $\mathbb{C}^n$ or $\mathbb{R}^n$).

4.4 Hermitian/symmetric forms

A bilinear form f on a complex (real) vector space V is called Hermitian (resp., symmetric) if

$$f(u,v) = \overline{f(v,u)} \qquad \forall u, v \in V$$

In the real case, the bar can be dropped.

4.5 Quadratic forms

For a Hermitian bilinear form f, the function $q\colon V \to \mathbb{R}$ defined by $q(u) \colon = f(u,u)$ is called the *quadratic form* associated with f. Note that $q(u) \in \mathbb{R}$ even in the complex case, because $f(u,u) = \overline{f(u,u)}$.

4.6 Theorem

A linear operator $T\colon V \to V$ is selfadjoint if and only if the bilinear form $f(u,v) = \langle Tu, v\rangle$ is Hermitian (in the real case, symmetric).

Proof. If T is selfadjoint, then $f(u,v) = \langle Tu, v\rangle = \langle u, Tv\rangle = \overline{\langle Tv, u\rangle} = \overline{f(v,u)}$. If f is Hermitian, then $\langle u, Tv\rangle = \overline{\langle Tv, u\rangle} = \overline{f(v,u)} = f(u,v) = \langle Tu, v\rangle = \langle u, T^*v\rangle$, hence $T = T^*$. □

Therefore, Hermitian bilinear forms on $\mathbb{C}^n$ are defined by Hermitian matrices.

4.7 Positive definite forms and matrices

A Hermitian (symmetric) bilinear form f on a vector space V is said to be *positive definite* if $f(u,u) > 0$ for all $u \neq 0$.

A selfadjoint operator $T\colon V \to V$ is said to be *positive definite* if $\langle Tu, u\rangle > 0$ for all $u \neq 0$.

A selfadjoint matrix A is said to be *positive definite* if $\langle Ax, x\rangle > 0$ for all $x \neq 0$ (equivalently, $x^*Ax > 0$ for all $x \neq 0$).

By replacing "> 0" with "≥ 0", one gets *positive semi-definite* bilinear forms, operators, and matrices.

4.8 Theorem

The following are equivalent:

(a) a bilinear form $f(u,v)$ is an inner product.

(b) $f(u,v) = \langle Tu, v\rangle$, where T is a positive definite operator.

(c) in $\mathbb{C}^n$ and $\mathbb{R}^n$, $f(x,y) = \langle Ax, y\rangle$ with a positive definite matrix A.

4.9 Lemma

Let $A \in \mathbb{C}^{n\times n}$ be a Hermitian matrix with eigenvalues $\lambda_1, \dots, \lambda_n$. Then

$$\|x\|_2^2 \min_{1\le i\le n} \lambda_i \le \langle Ax, x\rangle \le \|x\|_2^2 \max_{1\le i\le n} \lambda_i$$

for any $x \in \mathbb{C}^n$. Furthermore, the left/right inequality turns into an equality if and only if x is an eigenvector corresponding to the smallest/largest eigenvalue, respectively.

Proof. Let us denote

$$\lambda_{\min} = \min_{1\le i\le n} \lambda_i \qquad \text{and} \qquad \lambda_{\max} = \max_{1\le i\le n} \lambda_i.$$

By Spectral Theorem 3.12, there is an ONB $\{u_1, \dots, u_n\}$ consisting of eigenvectors of A. Then for any vector $x = \sum c_i u_i$ we have $Ax = \sum \lambda_i c_i u_i$ and

$$\langle Ax, x\rangle = \lambda_1 |c_1|^2 + \cdots + \lambda_n |c_n|^2$$

Therefore

$$\langle Ax, x\rangle \ge \lambda_{\min}\big(|c_1|^2 + \cdots + |c_n|^2\big) = \lambda_{\min}\|x\|_2^2$$

where we used the formula $\|x\|_2^2 = \sum_{i=1}^n |c_i|^2$ derived in the end of Section 1.14. We also see that

$$\langle Ax, x\rangle - \lambda_{\min}\|x\|_2^2 = \sum_{i=1}^n |c_i|^2(\lambda_i - \lambda_{\min})$$

where all the terms are nonnegative because $\lambda_i \ge \lambda_{\min}$. Thus, the left inequality in the lemma turns into an equality if and only if $c_i = 0$ for all i's such that $\lambda_i > \lambda_{\min}$, which means that x is an eigenvector corresponding to the smallest eigenvalue $\lambda_{\min}$.

Similarly,

$$\langle Ax, x\rangle \le \lambda_{\max}\big(|c_1|^2 + \cdots + |c_n|^2\big) = \lambda_{\max}\|x\|_2^2$$

and

$$\lambda_{\max}\|x\|_2^2 - \langle Ax, x\rangle = \sum_{i=1}^n |c_i|^2(\lambda_{\max} - \lambda_i)$$

Thus, the right inequality in the lemma turns into an equality if and only if $c_i = 0$ for all i's such that $\lambda_{\max} > \lambda_i$, which means that x is an eigenvector corresponding to the largest eigenvalue $\lambda_{\max}$. □

4.10 Lemma

Let $A \in \mathbb{C}^{n\times n}$ be a Hermitian matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$. Then

$$\|A\|_2 = \max_{1\le i\le n} |\lambda_i|.$$

Proof. In the notation of the previous proof

$$\|Ax\|_2^2 = \langle Ax, Ax\rangle = \lambda_1^2|c_1|^2 + \cdots + \lambda_n^2|c_n|^2$$

(remember that $\lambda_i \in \mathbb{R}$, but $c_i \in \mathbb{C}$). Denote

$$\bar{\lambda} = \max_{1\le i\le n} |\lambda_i|$$

the largest absolute value of the eigenvalues of A. Then

$$\|Ax\|_2^2 \le \big[\max_{1\le i\le n} \lambda_i^2\big] \sum_{j=1}^{n} |c_j|^2 = \bar{\lambda}^2 \|x\|_2^2.$$

Taking the square root and assuming $x \neq 0$ we get

$$\|Ax\|_2/\|x\|_2 \le \bar{\lambda}$$

Now there exists $k \in [1, n]$ such that $|\lambda_k| = \bar{\lambda}$. If x is an eigenvector corresponding to λ_k, then $c_i = 0$ for all $i \neq k$, hence

$$\|Ax\|_2^2 = \lambda_k^2 |c_k|^2 = \bar{\lambda}^2 \|x\|_2^2$$

Thus

$$\|A\|_2 = \max_{x\neq 0} \|Ax\|_2/\|x\|_2 = \bar{\lambda}$$

4.11 Theorem

A Hermitian matrix $A \in \mathbb{C}^{n\times n}$ is positive definite iff all its eigenvalues are positive.

Proof. This follows from Lemma 4.9. □

Similarly, A is positive-semidefinite iff it is Hermitian and all its eigenvalues are nonnegative.

4.12 Corollary

If a matrix A is positive definite, then so is A^{-1}. If an operator T is positive definite, then so is T^{-1}.

Interestingly, the condition $\langle Ax, x\rangle > 0$ for all $x \in \mathbb{C}^n$ implies that the matrix A is Hermitian! We will show this in a few steps.

4.13 Lemma

Let A, B be complex matrices. If $\langle Ax, x\rangle = \langle Bx, x\rangle$ for all $x \in \mathbb{C}^n$, then $A = B$. (Proof: see exercises.)

Note: This lemma holds in complex spaces, but it fails in real spaces.

4.14 Corollary

If A is a complex matrix such that $\langle Ax, x\rangle \in \mathbb{R}$ for all $x \in \mathbb{C}^n$, then A is Hermitian. In particular, if $\langle Ax, x\rangle > 0$ for all $x \in \mathbb{C}^n$, then A is positive definite.

Proof. $\langle Ax, x\rangle = \overline{\langle Ax, x\rangle} = \langle x, Ax\rangle = \langle A^*x, x\rangle$, hence $A = A^*$ by Lemma 4.13. Now the condition $\langle Ax, x\rangle > 0$ implies that all the eigenvalues of A are positive, thus A is positive definite by Theorem 4.11. □

4.15 Theorem

A matrix $A \in \mathbb{C}^{n\times n}$ is positive definite iff there is a nonsingular matrix B such that $A = B^*B$.

Proof. "⇐" If $A = B^*B$, then $A^* = B^*(B^*)^* = A$ and $\langle Ax, x\rangle = \langle Bx, Bx\rangle > 0$ for any $x \neq 0$, because B is nonsingular.
"⇒" By Sections 3.13 and 4.11, $A = P^{-1}DP$, where D is a diagonal matrix with positive diagonal entries and P a unitary matrix. If $D = \text{ diag }\{d_1, \ldots, d_n\}$, then denote $D^{1/2} = \text{ diag }\{\sqrt{d_1}, \ldots, \sqrt{d_n}\}$. Now $A = B^2$ where $B = P^{-1}D^{1/2}P$, and B is selfadjoint by Section 3.13. □

Remark. A matrix $A \in \mathbb{C}^{n\times n}$ is positive semi-definite if and only if there is a matrix B (not necessarily nonsingular) such that $A = B^*B$.

4.16 Definition

A matrix $A \in \mathbb{C}^{m\times n}$ is said to have *full rank* if

$$\text{rank } A = \min\{m, n\}$$

Otherwise, A is said to be *rank deficient*.

4.17 Theorem

Let A be a rectangular $m \times n$ matrix. Then the matrices A^*A and AA^* are Hermitian and positive semi-definite. If $m \neq n$ and A has full rank, then the smaller of the two matrices A^*A and AA^* is positive definite.

Proof. First we verify the Hermitian property:

$$(A^*A)^* = A^*(A^*)^* = A^*A$$

and similarly $(AA^*)^* = AA^*$. Next we verify positive semidefiniteness:

$$\langle A^*Ax, x\rangle = \langle Ax, Ax\rangle \geq 0$$

for any $x \in \mathbb{C}^n$ and similarly

$$\langle AA^*y, y\rangle = \langle A^*y, A^*y\rangle \geq 0$$

for any $y \in \mathbb{C}^m$.

Now let A have full rank. If $m \geq n$, then $\mathrm{Ker}(A) = \{0\}$, hence $Ax \neq 0$ for any $0 \neq x \in \mathbb{C}^n$, so that $\langle A^*Ax, x\rangle = \langle Ax, Ax\rangle > 0$, which implies that A^*A is positive definite. If $m < n$, then $\mathrm{Range}(A) = \mathbb{C}^m$, hence for any $0 \neq y \in \mathbb{C}^m$ there is $0 \neq x \in \mathbb{C}^n$ such that $y = Ax$. Therefore

$$\langle A^*y, x\rangle = \langle y, Ax\rangle = \langle y, y\rangle > 0$$

which implies $A^*y \neq 0$ (i.e., $\mathrm{Ker}(A^*) = \{0\}$). Therefore

$$\langle AA^*y, y\rangle = \langle A^*y, A^*y\rangle > 0$$

which implies that AA^* is positive definite. □

4.18 Theorem

For any $A \in \mathbb{C}^{m\times n}$ we have

$$\|A\|_2^2 = \|A^*\|_2^2 = \|A^*A\|_2 = \|AA^*\|_2 = \lambda_{\max}$$

where $\lambda_{\max}$ is the largest eigenvalue of both A^*A and AA^*.

Remark: this theorem gives a practical method to compute $\|A\|_2$ for small matrices (when $m = 2$ or $n = 2$), since one of the two matrices A^*A and AA^* is 2×2, thus its eigenvalues are easily computable.

Proof of Theorem 4.18 is long and will be done step by step. In the proof, $\|\cdot\|$ will always denote the 2-norm.

Lemma. For every vector $z \in \mathbb{C}^n$ we have $\|z\| = \max_{\|y\|=1} |\langle y, z\rangle|$.

Proof. Indeed, by the Cauchy-Schwarz inequality

$$|\langle y, z\rangle| \le \langle y, y\rangle^{1/2} \langle z, z\rangle^{1/2} = \|z\|$$

and the equality is attained whenever y is parallel to z. So we can set $y = \pm\frac{z}{\|z\|}$ and achieve the maximum. □

Step 1. To prove that $\|A\| = \|A^*\|$ we write

$$\|A\| = \sup_{\|x\|=1} \|Ax\| = \sup_{\|x\|=1} \sup_{\|y\|=1} |\langle y, Ax\rangle| = \sup_{\|x\|=1} \sup_{\|y\|=1} |\langle A^*y, x\rangle|$$

$$= \sup_{\|y\|=1} \sup_{\|x\|=1} |\langle x, A^*y\rangle| = \sup_{\|y\|=1} \|A^*y\| = \|A^*\|$$

Step 2. To prove that $\|A\|^2 = \|A^*A\|$ we write

$$\|A^*A\| = \sup_{\|x\|=1} \|A^*Ax\| = \sup_{\|x\|=1} \sup_{\|y\|=1} |\langle y, A^*Ax\rangle| = \sup_{\|x\|=1} \sup_{\|y\|=1} |\langle Ay, Ax\rangle|$$

Then again by the Cauchy-Schwarz inequality

$$|\langle Ay, Ax\rangle| \le \|Ax\|\,\|Ay\| \le \|A\|\,\|A\| = \|A\|^2$$

hence $\|A^*A\| \le \|A\|^2$. On the other hand,

$$\|A^*A\| = \sup_{\|x\|=1} \sup_{\|y\|=1} |\langle Ay, Ax\rangle| \ge \sup_{\|x\|=1} |\langle Ax, Ax\rangle| = \|A\|^2.$$

Therefore, $\|A^*A\| = \|A\|^2$.

Step 3. Using an obvious symmetry we conclude that $\|A^*\|^2 = \|AA^*\|$

Step 4. By Lemma 4.10, we have

$$\|A^*A\|_2 = \max |\lambda_i(A^*A)|$$

Recall that A^*A is a positive-semidefinite matrix, so its eigenvalues $\lambda_i(A^*A)$ are real and ≥ 0, hence $\max |\lambda_i(A^*A)| = \lambda_{\max}(A^*A)$, the largest eigenvalue of A^*A. The same argument applies to AA^*. In particular, we see that

$$\lambda_{\max}(A^*A) = \lambda_{\max}(AA^*).$$

This completes the proof of Theorem 4.18. □

4.19 Example

Let $A = \begin{bmatrix} 3 \\ 4 \end{bmatrix} \in \mathbb{C}^{2\times 1}$. For any unit vector $x \in \mathbb{C}^1$ we can write $x = [e^{\mathbf{i}\theta}]$, therefore $Ax = \begin{bmatrix} 3e^{\mathbf{i}\theta} \\ 4e^{\mathbf{i}\theta} \end{bmatrix}$ and

$$\|Ax\| = \sqrt{|3e^{\mathbf{i}\theta}|^2 + |4e^{\mathbf{i}\theta}|^2} = \sqrt{3^2 + 4^2} = 5$$

which implies $\|A\| = 5$. Now let us find the norm of $A^* = \begin{bmatrix} 3 & 4 \end{bmatrix} \in \mathbb{C}^{1\times 2}$, i.e., $\|A^*\| = \sup_{\|y\|=1} \|A^* y\|$. For simplicity, we will only use *real* unit vectors $y \in \mathbb{C}^2$, which can be described by $y = \begin{bmatrix} \cos\varphi \\ \sin\varphi \end{bmatrix}$ for $\varphi \in [0, 2\pi]$. We have $A^*y = 3\cos\varphi + 4\sin\varphi$, thus $\|A^*y\| = |3\cos\varphi + 4\sin\varphi|$. Finding the maximum of this function (over the interval $0 \le \varphi \le 2\pi$) is a Calculus-I problem: the maximum is achieved at $\cos\varphi = \pm 3/5$ and $\sin\varphi = \pm 4/5$, and we get

$$\|A^*\| = \max_{\|y\|=1} \|A^* y\| = \frac{9}{5} + \frac{16}{5} = \frac{25}{5} = 5$$

We see that, indeed, $\|A\| = \|A^*\|$. Note that $A^*A = [25] \in \mathbb{C}^{1\times 1}$, so obviously $\|A^*A\| = 25$, in full agreement with Theorem 4.18.

4.20 Corollary

If $\lambda_{\max}$ again denotes the largest eigenvalue of A^*A, then

$$\|Ax\|_2 = \|A\|_2 \|x\|_2 \qquad \Longleftrightarrow \qquad A^*Ax = \lambda_{\max} x.$$

Hence, the supremum in Section 1.5 is attained (on the eigenvectors of A^*A corresponding to $\lambda_{\max}$) and can be replaced by maximum. Moreover, this implies that the 2-norm of a real matrix is the same, whether it is computed in the complex space or in the real space.

Proof. On the one hand

$$\|Ax\|_2^2 = \langle Ax, Ax\rangle = \langle A^*Ax, x\rangle$$

and on the other hand

$$\|A\|^2 = \lambda_{\max},$$

so for any vector x with $\|x\| = 1$ we have

$$\|Ax\|_2^2 = \|A\|_2^2 \qquad \Longleftrightarrow \qquad \langle A^*Ax, x\rangle = \lambda_{\max}.$$

Then we use Lemma 4.9. □

Exercise 4.1. Let $A \in \mathbb{C}^{n\times n}$ satisfy $A^* = -A$. Show that the matrix $I - A$ is invertible. Then show that the matrix $(I - A)^{-1}(I + A)$ is unitary.

Exercise 4.2. Let $A = (a_{ij})$ be a complex $n \times n$ matrix. Assume that $\langle Ax, x\rangle = 0$ for all $x \in \mathbb{C}^n$. Prove that
(a) $a_{ii} = 0$ for $1 \leq i \leq n$ by substituting $x = e_i$
(b) $a_{ij} = 0$ for $i \neq j$ by substituting $x = pe_i + qe_j$ then using (a) and putting $p, q = \pm 1, \pm\mathbf{i}$ (here $\mathbf{i} = \sqrt{-1}$) in various combinations.
Conclude that $A = 0$.

Exercise 4.3. Let A, B be complex $n \times n$ matrices such that $\langle Ax, x\rangle = \langle Bx, x\rangle$ for all $x \in \mathbb{C}^n$. Prove that $A = B$.

Exercise 4.4. Find a real 2×2 matrix $A \neq 0$ such that $\langle Ax, x\rangle = 0$ for all $x \in \mathbb{R}^2$. Thus find two real 2×2 matrices A and B such that $\langle Ax, x\rangle = \langle Bx, x\rangle$ for all $x \in \mathbb{R}^2$, but $A \neq B$.

Exercise 4.5. Find a real 2×2 matrix A such that $\langle Ax, x\rangle > 0$ for all $x \in \mathbb{R}^2$, but A is not positive definite.

5 Singular value decomposition (SVD)

A matrix $A \in \mathbb{C}^{m\times n}$ defines a linear transformation $T\colon \mathbb{C}^n \to \mathbb{C}^m$. If B is an ONB in $\mathbb{C}^n$ and B' an ONB in $\mathbb{C}^m$, then T is represented in the bases B and B' by the matrix U^*AV, where $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ are unitary matrices. The following theorem shows that one can always find bases B and B' so that the matrix U^*AV will be diagonal.

Note: $D \in \mathbb{C}^{m\times n}$ is said to be diagonal if $D_{ij} = 0$ for $i \neq j$. It has exactly $p = \min\{m, n\}$ diagonal entries and can be denoted by $D = \operatorname{diag}\{d_1, \ldots, d_p\}$.

5.1 Singular value decomposition (SVD)

Let $A \in \mathbb{C}^{m\times n}$ have rank r and let $p = \min\{m, n\}$. Then there are unitary matrices $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ and a real diagonal matrix $D = \operatorname{diag}\{\sigma_1, \ldots, \sigma_p\}$ such that

$$A = UDV^* \qquad \textbf{(SVD)}$$

On the diagonal of D, exactly r elements are positive and the other $p - r$ elements are zero. If we require, additionally, that $\sigma_1 \geq \cdots \geq \sigma_r > 0$ and $\sigma_{r+1} = \cdots = \sigma_p = 0$, then the matrix D is unique.

Proof. Observe that (SVD) is equivalent to $A^* = VD^TU^*$, hence A has an SVD if and only if A^* does. Without loss of generality, we assume that $m \geq n$, then fix $l = m - n$ and use the induction on n (taking $m = n + l$).

Let $\sigma_1 = \|A\|_2$. There is a unit vector $v_1 \in \mathbb{C}^n$ such that $\|Av_1\|_2 = \|A\|_2$, see 4.20, and a unit vector $u_1 \in \mathbb{C}^m$ such that $Av_1 = \sigma_1 u_1$. Extend v_1 to an ONB $\{v_1, \ldots, v_n\}$ in $\mathbb{C}^n$ and extend u_1 to an ONB $\{u_1, \ldots, u_m\}$ in $\mathbb{C}^m$. Let V_1 denote the matrix whose columns are $v_1, \ldots, v_n$ and U_1 denote the matrix whose columns are $u_1, \ldots, u_m$. Then we have

$$U_1^*AV_1 = S = \begin{bmatrix} \sigma_1 & w^* \\ 0 & B \end{bmatrix} \qquad (*)$$

If $n = 1$, then $S = \left[\begin{smallmatrix} \sigma_1 \\ 0 \end{smallmatrix}\right]$ is diagonal yielding an SVD. For $n > 1$, observe that

$$\|S\|_2 = \|S^*\|_2 \geq \left\| \begin{bmatrix} \sigma_1 & 0 \\ w & B^* \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right\|_2 = \left\| \begin{bmatrix} \sigma_1 \\ w \end{bmatrix} \right\|_2 = \sqrt{\sigma_1^2 + w^*w}.$$

On the other hand, the matrices U_1 and V_1 are unitary, hence $\|S\|_2 = \|A\|_2 = \sigma_1$, thus $w = 0$. By our inductive assumption the matrix B has an SVD $B = \hat{U}\hat{D}\hat{V}^*$. Now it is easily verified that

$$A = U_1 \begin{bmatrix} 1 & 0 \\ 0 & \hat{U} \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \hat{D} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \hat{V}^* \end{bmatrix} V_1^*$$

is an SVD of A.

To prove the uniqueness, observe that

$$A^*A = VD^TDV^* \qquad \text{and} \qquad AA^* = UDD^TU^*$$

hence $\sigma_1^2, \ldots, \sigma_p^2$ are the eigenvalues of both A^*A and AA^* (hence, these matrices have common non-zero eigenvalues). Note also that the columns of U are the eigenvectors of AA^* and the columns of V are the eigenvectors of A^*A, see 3.14. □

Note: if $A \in \mathbb{R}^{m\times n}$, then

$$A = UDV^T \qquad \textbf{(Real SVD)}$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices.

5.2 Singular values and vectors

The positive numbers $\sigma_1, \ldots, \sigma_r$ are called the *singular values* of A. The columns $v_1, \ldots, v_n$ of the matrix V (not those of V^*) are called the *right singular vectors* for A, and the columns $u_1, \ldots, u_m$ of the matrix U are called the *left singular vectors* for A.

5.3 Corollary

For $1 \le i \le r$ we have

$$Av_i = \sigma_i u_i, \qquad A^*u_i = \sigma_i v_i$$

We also have

$$\begin{aligned} \operatorname{Ker} A &= \operatorname{span}\{v_{r+1}, \ldots, v_n\}, & \operatorname{Ker} A^* &= \operatorname{span}\{u_{r+1}, \ldots, u_m\} \\ \operatorname{Range} A &= \operatorname{span}\{u_1, \ldots, u_r\}, & \operatorname{Range} A^* &= \operatorname{span}\{v_1, \ldots, v_r\} \end{aligned}$$

and

$$\operatorname{rank} A = \operatorname{rank} A^* = r.$$

Here is a diagram illustrating the previous relations:

$$\begin{array}{cccccc} & & A & & A^* & \\ v_1 & \xrightarrow{\sigma_1} & u_1 & \xrightarrow{\sigma_1} & v_1 \\ v_2 & \xrightarrow{\sigma_2} & u_2 & \xrightarrow{\sigma_2} & v_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ v_r & \xrightarrow{\sigma_r} & u_r & \xrightarrow{\sigma_r} & v_r \\ v_{r+1} & \to 0 & u_{r+1} & \to 0 & \\ \vdots & \vdots & \vdots & \vdots & \\ v_n & \to 0 & u_m & \to 0 & \end{array}$$

5.4 Remarks

For any matrix $A \in \mathbb{C}^{m\times n}$

$$\|A\|_2 = \|A^*\|_2 = \|D\|_2 = \sigma_1.$$

If A is a square invertible matrix, then

$$A^{-1} = VD^{-1}U^*$$

and

$$\|A^{-1}\|_2 = \|D^{-1}\|_2 = \sigma_n^{-1}.$$

If $A \in \mathbb{C}^{n\times n}$ is Hermitian with eigenvalues $\lambda_1, \ldots, \lambda_n$, then its singular values are $|\lambda_1|, \ldots, |\lambda_n|$ (this follows from Corollary 3.13).

5.5 Computing SVD

In practice, to compute an SVD of a $m \times 2$ matrix, find the eigenvalues $\lambda_1 \geq \lambda_2$ and the corresponding unit eigenvectors v_1, v_2 of the matrix A^*A, then compute $\sigma_1 = \sqrt{\lambda_1}$, $\sigma_2 = \sqrt{\lambda_2}$ and $u_1 = \sigma_1^{-1}Av_1$, $u_2 = \sigma_2^{-1}Av_2$, and lastly extend $\{u_1, u_2\}$ to an ONB in $\mathbb{C}^m$ arbitrarily.

5.6 Reduced SVD

Let $A \in \mathbb{C}^{m\times n}$ with $m > n$ and $\operatorname{rank} A = r$. Then there is a matrix $\hat{U} \in \mathbb{C}^{m\times n}$ with orthonormal columns, a unitary matrix $V \in \mathbb{C}^{n\times n}$ and a square diagonal matrix $\hat{D} = \operatorname{diag}\{\sigma_1, \ldots, \sigma_n\}$ such that

$$A = \hat{U}\hat{D}V^* \qquad \text{(reduced SVD)}$$

For $m < n$, the reduced SVD is similar.

Proof. Use the (full) SVD $A = UDV^*$ given by Section 5.1 and then erase the last $m - n$ columns of U and the bottom $m - n$ rows of D.

5.7 Rank-one expansion

We have the following:

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^* \qquad \text{(SVD expansion)}$$

Proof. It is enough to observe that for every v_j

$$\left(\sum_{i=1}^{r} \sigma_i u_i v_i^* \right) v_j = \sigma_j u_j = A v_j$$

because $v_i^* v_j = \delta_{ij}$.

5.8 Remarks

Recall the Frobenius norm of a matrix, cf. Section 1.4. We have

$$\|A\|_F^2 = \text{ (Exercise 2.2) } = \|D\|_F^2 = \sigma_1^2 + \cdots + \sigma_r^2.$$

In particular we see that $\|A\|_2 \leq \|A\|_F$, cf. Section 5.4.

The value of $\|A\|_F^2$ can be interpreted as the *energy* of the matrix A. The energy is conserved under multiplication by unitary matrices, and the SVD pulls all the energy of a matrix onto its diagonal.

Now, for any unit vectors u and v we have

$$\|uv^*\|_F = 1$$

hence the SVD expansion presents A as a sum of rank-one matrices so that each partial sum captures as much energy of A as possible.

5.9 Low-rank approximation

For any $1 \leq k \leq r$, define

$$A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^*$$

Then

$$\sigma_{k+1} = \|A - A_k\|_2 = \inf_{\substack{B \in \mathbb{C}^{m\times n} \\ \operatorname{rank} B \le k}} \|A - B\|_2$$

(with the convention $\sigma_{r+1} = 0$). Thus, A_k gives the best approximation to A by rank k matrices.

Proof. Suppose there is some matrix B with $\operatorname{rank} B \le k$ such that $\|A-B\|_2 < \sigma_{k+1}$. Then $\dim(\operatorname{Ker} B) \ge n - k$, and for any nonzero vector $v \in \operatorname{Ker} B$

$$\|Av\|_2 = \|(A - B)v\|_2 \le \|A - B\|_2\|v\|_2 < \sigma_{k+1}\|v\|_2$$

On the other hand, there is a $(k+1)$-dimensional subspace $\operatorname{span}\{v_1, \ldots, v_{k+1}\}$ on which $\|Av\|_2 \ge \sigma_{k+1}\|v\|_2$. Since $(n - k) + (k + 1) > n$, these two spaces have a common nonzero vector, a contradiction.

5.10 Corollary

Let $A \in \mathbb{C}^{n\times n}$ be a nonsingular matrix. Then

$$\min\{\|A - A_s\|_2 \colon \ A_s \text{ is singular}\} = \sigma_n$$

Proof. Recall: A_s is singular iff $\operatorname{rank} A_s \le n - 1$, then use the previous fact.

5.11 Rank with tolerance ε

The *rank* of $A \in \mathbb{C}^{m\times n}$ with *tolerance* $\varepsilon > 0$ (also called *numerical rank*) is defined by

$$\operatorname{rank}(A, \varepsilon) = \min_{\|E\|_2 \le \varepsilon} \operatorname{rank}(A + E)$$

Note: $\operatorname{rank}(A, \varepsilon) \le \operatorname{rank} A$. The rank with tolerance ε gives the minimum rank of A under perturbations by small matrices having 2-norm $\le \varepsilon$. If A has full rank, but $\operatorname{rank}(A, \varepsilon) < p = \min\{m, n\}$ for a small ε, then A is 'nearly rank deficient'.

5.12 Corollary

$\operatorname{rank}(A, \varepsilon)$ equals the number of singular values of A (counted with multiplicity) that are greater than ε.

Note: $\operatorname{rank}(A^*, \varepsilon) = \operatorname{rank}(A, \varepsilon)$, since A and A^* have the same singular values.

5.13 Definition

The vector space $\mathbb{C}^{m\times n}$ with the distance between matrices defined by

$$\operatorname{dist}(A,B) = \|A-B\|_2$$

is a *metric space*. Then topological notions, like open sets, dense sets, etc., apply. Note that if a matrix $E \in \mathbb{C}^{m\times n}$ has small components, say $|e_{ij}| < \varepsilon$ for all $1 \le i \le m$ and $1 \le j \le n$, then its 2-norm is bounded by

$$\|E\|_2 \le \|E\|_F \le \varepsilon\sqrt{mn}.$$

Thus if we perturb every component of a matrix $A \in \mathbb{C}^{m\times n}$ by less than ε, then the perturbed matrix will be within distance $< \varepsilon\sqrt{mn}$ from A.

5.14 Theorem

Full rank matrices make an open and dense subset of $\mathbb{C}^{m\times n}$.

Openness means that for any full rank matrix A there is an $\varepsilon > 0$ such that $A+E$ has full rank whenever $\|E\|_2 \le \varepsilon$. Denseness means that if A is rank deficient, then for any $\varepsilon > 0$ there is E, $\|E\|_2 \le \varepsilon$, such that $A+E$ has full rank.

Proof. To prove denseness, let A be a rank deficient matrix and $A = UDV^*$ its SVD. For $\varepsilon > 0$, put $D_\varepsilon = \varepsilon I$ and $E = UD_\varepsilon V^*$. Then $\|E\|_2 = \|D_\varepsilon\|_2 = \varepsilon$ and

$$\operatorname{rank}(A+E) = \operatorname{rank}(U(D+D_\varepsilon)V^*) = \operatorname{rank}(D+D_\varepsilon) = \min\{m,n\}$$

Openness follows from Section 5.12. Indeed, if $\sigma_{\min} > 0$ is the smallest singular value of a full rank matrix A, then every matrix B such that

$$\|A-B\|_2 < \sigma_{\min}$$

also has full rank.

5.15 Theorem

Diagonalizable matrices make a dense subset of $\mathbb{C}^{n\times n}$.

Proof. See Exercise 6.3.

Exercise 5.1. Let $x \in \mathbb{C}^n$ and $y \in \mathbb{C}^m$. Consider the $m \times n$ matrix defined by $A = yx^*$.

(a) Show that $\operatorname{rank} A = 1$.

(b) Show that $\|A\|_2 = \|x\|_2\|y\|_2$.

(c) Show that $\|A\|_F = \|x\|_2\|y\|_2$.

Exercise 5.2. (JPE, September 1996) Compute the singular values of

$$A = \begin{pmatrix} 0 & -1.6 & 0.6 \\ 0 & 1.2 & 0.8 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Exercise 5.3. (JPE, May 2003) Determine the singular value decomposition for the matrix

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

Exercise 5.4. Find the numerical rank with tolerance 0.9 of the matrix

$$A = \begin{pmatrix} 3 & 2 \\ -4 & -5 \end{pmatrix}$$

Exercise 5.5. Let $Q \in \mathbb{C}^{n\times n}$ be unitary. Find all singular values of Q.

Exercise 5.6. Show that if two matrices $A, B \in \mathbb{C}^{n\times n}$ are unitary equivalent, then they have the same singular values. Is the converse true? (Prove or give a counterexample.)

6 Schur decomposition

Recall that every complex matrix $A \in \mathbb{C}^{n\times n}$ is equivalent to a Jordan matrix. In other words, for any linear operator $T\colon \mathbb{C}^n \to \mathbb{C}^n$ there exists a basis in which T is represented by a Jordan matrix. What if we restrict our interests to orthonormal bases in $\mathbb{C}^n$? In other words, to what extend one can simplify a complex matrix by using unitary equivalence?

6.1 Schur decomposition

Any matrix $A \in \mathbb{C}^{n\times n}$ is unitary equivalent to an upper triangular matrix T. Moreover, one can find T is such a way that the eigenvalues of A appear in any given order on the diagonal of T.

Note: the unitary equivalence means that $Q^*AQ = T$ for some unitary matrix Q. The columns of the matrix Q are called *Schur vectors*.

Proof. We use induction on n. The theorem is obvious for $n = 1$. Assume that it holds for matrices of order less than n. Let λ be an eigenvalue of A and let x be a unit eigenvector for λ. Let Q_1 be a unitary matrix whose first column is x (note: such a matrix exists, because there is an ONB in $\mathbb{C}^n$ whose first vector is x, by Section 1.12, and then Q_1 can be constructed so that the vectors of that ONB are the columns of Q_1). Note that $Q_1e_1 = x$, and hence $Q_1^*x = e_1$, since $Q_1^{-1} = Q_1^*$. Hence, we have

$$Q_1^*AQ_1e_1 = Q_1^*Ax = \lambda Q_1^*x = \lambda e_1$$

so e_1 is an eigenvector of the matrix $Q_1^*AQ_1$ for the eigenvalue λ. Thus,

$$Q_1^*AQ_1 = \begin{bmatrix} \lambda & w^* \\ 0 & B \end{bmatrix}$$

with some $w \in \mathbb{C}^{n-1}$ and $B \in \mathbb{C}^{(n-1)\times(n-1)}$. By the inductive assumption, there is a unitary matrix $\hat{Q} \in \mathbb{C}^{(n-1)\times(n-1)}$ such that $\hat{Q}^*B\hat{Q} = \hat{T}$, where $\hat{T}$ is upper triangular. Let

$$Q = Q_1 \begin{bmatrix} 1 & 0 \\ 0 & \hat{Q} \end{bmatrix}$$

which is a unitary matrix. Next,

$$Q^*AQ = \begin{bmatrix} 1 & 0 \\ 0 & \hat{Q}^* \end{bmatrix} \begin{bmatrix} \lambda & w^* \\ 0 & B \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \hat{Q} \end{bmatrix}$$
$$= \begin{bmatrix} \lambda & w^*\hat{Q} \\ 0 & \hat{Q}^*B\hat{Q} \end{bmatrix} = \begin{bmatrix} \lambda & w^*\hat{Q} \\ 0 & \hat{T} \end{bmatrix}$$

which is upper triangular, as required.

6.2 Normal matrices

A matrix $A \in \mathbb{C}^{n\times n}$ is said to be *normal* if $AA^* = A^*A$.

Note: unitary and Hermitian matrices are normal.

6.3 Lemma

If A is normal and Q unitary, then $B = Q^*AQ$ is also normal (i.e., the class of normal matrices is closed under unitary equivalence).

Proof. Note that $B^* = Q^*A^*Q$ and $BB^* = Q^*AA^*Q = Q^*A^*AQ = B^*B$.

6.4 Lemma

If A is normal and upper triangular, then A is diagonal.

Proof. We use induction on n. For $n = 1$ the theorem is trivial. Assume that it holds for matrices of order less than n. Compute the top left element of the matrix $AA^* = A^*A$. On the one hand, it is

$$\sum_{i=1}^{n} a_{1i}\bar{a}_{1i} = \sum_{i=1}^{n} |a_{1i}|^2$$

On the other hand, it is just $|a_{11}|^2$. Hence, $a_{12} = \cdots = a_{1n} = 0$, and

$$A = \begin{bmatrix} a_{11} & 0 \\ 0 & B \end{bmatrix}$$

One can easily check that $AA^* = A^*A$ implies $BB^* = B^*B$. By the inductive assumption, B is diagonal. □

Note: Any diagonal matrix is obviously normal.

6.5 Theorem

A matrix $A \in \mathbb{C}^{n\times n}$ is normal if and only if it is unitary equivalent to a diagonal matrix. In that case the Schur decomposition takes form

$$Q^* A Q = D$$

where D is a diagonal matrix, and the columns of Q (Schur vectors) become eigenvectors of A

Proof. It follows from Sections 6.1–6.4 and 3.14. □

6.6 Remark

Three classes of complex matrices have the same property: they are unitary equivalent to a diagonal matrix (i.e., admit an ONB consisting of eigenvectors). The difference between those classes lies in restrictions on the eigenvalues: unitary matrices have eigenvalues on the unit circle ($|\lambda| = 1$), Hermitian matrices have real eigenvalues ($\lambda \in \mathbb{R}$), and now normal matrices have arbitrary complex eigenvalues.

6.7 Real Schur decomposition

If $A \in \mathbb{R}^{n\times n}$, then there exists an orthogonal matrix Q such that

$$Q^T A Q = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1m} \\ 0 & R_{22} & \cdots & R_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R_{mm} \end{bmatrix}$$

where each diagonal block R_{ii} is either a 1×1 matrix or a 2×2 matrix.

Proof. We use the induction on n. If the matrix A has a real eigenvalue, the we can reduce the dimension and use induction just like in the proof of Schur Theorem 6.1. If A has no real eigenvalues, then by Lemma 2.14 there is a two-dimensional subspace $W \subset \mathbb{R}^2$ invariant under A. Let $\{x_1, x_2\}$ be an ONB of W. The invariance under A implies

$$\begin{aligned} Ax_1 &= r_{11}\, x_1 + r_{21}\, x_2 \\ Ax_2 &= r_{12}\, x_1 + r_{22}\, x_2 \end{aligned}$$

with some $r_{ij} \in \mathbb{R}$. We now extend $\{x_1, x_2\}$ to an ONB $\{x_1, \ldots, x_n\}$ in $\mathbb{R}^n$ and denote by $\tilde{Q}$ the orthogonal matrix with columns $x_1, \ldots, x_n$. Observe that $\tilde{Q}e_1 = x_1$ and $\tilde{Q}e_2 = x_2$, hence $\tilde{Q}^T x_1 = e_1$ and $\tilde{Q}^T x_2 = e_2$. Therefore,

$$\tilde{Q}^T A \tilde{Q} e_1 = \tilde{Q}^T A x_1 = \tilde{Q}^T (r_{11}\, x_1 + r_{21}\, x_2) = r_{11}\, e_1 + r_{21}\, e_2$$

and similarly

$$\tilde{Q}^T A \tilde{Q} e_2 = \tilde{Q}^T A x_2 = \tilde{Q}^T (r_{12}\, x_1 + r_{22}\, x_2) = r_{12}\, e_1 + r_{22}\, e_2$$

Thus,

$$\tilde{Q}^T A \tilde{Q} = \begin{bmatrix} R_{11} & \tilde{R}_{12} \\ 0 & \tilde{R}_{22} \end{bmatrix}$$

where

$$R_{11} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix}$$

$\tilde{R}_{12}$ is some $2 \times (n-2)$ matrix and $\tilde{R}_{22}$ is some $(n-2) \times (n-2)$ matrix. Now we can apply our inductive assumption to the $(n-2) \times (n-2)$ matrix $\tilde{R}_{22}$.

Exercise 6.1. (combined from JPE, October 1990 and May 1997) Let $A \in \mathbb{C}^{n \times n}$ be a normal matrix.

(a) Prove that $A - \lambda I$ is normal for any $\lambda \in \mathbb{C}$.

(b) Prove that $\|Ax\| = \|A^* x\|$ for all x.

(c) Prove that (λ, x) is an eigenpair of A if and only if $(\bar{\lambda}, x)$ is an eigenpair of A^*. (Hence, A and A^* have the same eigenvectors.)

Exercise 6.2. (JPE, September 2002) A matrix $A \in \mathbb{C}^{n \times n}$ is said to be *skew Hermitian* if $A^* = -A$.

(a) Prove that if A is skew Hermitian and B is unitary equivalent to A, then B is also skew Hermitian.

(b) Prove that the eigenvalues of a skew Hermitian matrix are purely imaginary, i.e. they satisfy $\bar{\lambda} = -\lambda$.

(c) What special form does the Schur decomposition take for a skew Hermitian matrix A?

Exercise 6.3. (JPE, September 1998). Show that diagonalizable complex matrices make a dense subset of $\mathbb{C}^{n \times n}$. That is, for any $A \in \mathbb{C}^{n \times n}$ and $\varepsilon > 0$ there is a diagonalizable $B \in \mathbb{C}^{n \times n}$ such that $\|A - B\|_2 < \varepsilon$.

Exercise 6.4 (Bonus)**.** (JPE, May 1996). Let T be a linear operator on a finite dimensional complex inner product space V, and let T^* be the adjoint of T. Prove that $T = T^*$ if and only if $T^*T = T^2$.

7 Gaussian elimination and LU decomposition

7.1 Gaussian Elimination

Let $A \in \mathbb{C}^{n\times n}$ be a matrix with $a_{11} \neq 0$. Denote $A^{(1)} = A$ and $a_{ij}^{(1)} = a_{ij}$. We define *multipliers*

$$m_{i1} = a_{i1}^{(1)}/a_{11}^{(1)} \qquad \text{for } \; i = 2, \ldots, n$$

and replace the i-th row a_i' of the matrix A with $a_i' - m_{i1}a_1'$ for all $i = 2, \ldots, n$. This creates zeros in the first column of $A^{(1)}$, which then takes the form

$$A^{(2)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix}$$

where

$$a_{ij}^{(2)} = a_{ij}^{(1)} - m_{i1}a_{1j}^{(1)} \qquad \text{for } \; 2 \leq i, j \leq n$$

Next, assume that $a_{22}^{(2)} \neq 0$. Then we can continue this process and define multipliers

$$m_{i2} = a_{i2}^{(2)}/a_{22}^{(2)} \qquad \text{for } \; i = 3, \ldots, n$$

and replace the i-th row a_i' of the matrix $A^{(2)}$ with $a_i' - m_{i2}a_2'$ for all $i = 3, \ldots, n$. This creates a matrix, $A^{(3)}$, with zeros in the second column below the main diagonal, and so on. If all the numbers $a_{ii}^{(i)}$, $1 \leq i \leq n-1$, are different from zero, then one ultimately obtains an upper triangular matrix

$$A^{(n)} = U = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(3)} & \cdots & a_{3n}^{(3)} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & a_{nn}^{(n)} \end{bmatrix}$$

The elements $a_{ii}^{(i)}$, $1 \leq i \leq n$, are called *pivots*.

The above procedure has to stop prematurely if (and only if) one of the pivots $a_{ii}^{(i)}$, $1 \leq i \leq n-1$, happens to be zero. In that case we say that the Gaussian elimination fails.

7.2 Principal minors

Let $A \in \mathbb{C}^{n\times n}$. For $1 \le k \le n$, the k-th principal minor of A is the $k \times k$ matrix formed by the entries in the first k rows and the first k columns of A (i.e., the top left $k \times k$ block of A). We denote the k-th principal minor of A by A_k.

7.3 Theorem (Criterion of failure)

Gaussian elimination fails if and only if $\det A_k = 0$ for some $k = 1, \dots, n-1$. This is because for each $k = 1, \dots, n$

$$\det A_k = a_{11}^{(1)} \cdots a_{kk}^{(k)}$$

7.4 Gauss matrices

Assume that $A \in \mathbb{C}^{n\times n}$ has non-singular principal minors up to the order $n-1$, so that the Gaussian elimination works. For each $j = 1, \dots, n-1$ the Gauss matrix G_j is defined by

$$G_j = \begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ddots & 1 & \cdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & -m_{j+1,j} & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & 1 & 0 \\ 0 & 0 & \cdots & -m_{n,j} & \cdots & 0 & 1 \end{bmatrix}$$

Note that $G_j = I - m^{(j)} e_j^*$ where

$$m^{(j)} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ m_{j+1,j} \\ \vdots \\ m_{n,j} \end{bmatrix}$$

7.5 Lemma

For each $j = 1, \dots, n-1$ we have $G_j A^{(j)} = A^{(j+1)}$, and therefore

$$U = A^{(n)} = G_{n-1} \cdots G_2 G_1 A$$

7.6 Lemma

For each $j = 1, \ldots, n-1$ we have

$$L_j := G_j^{-1} = I + m^{(j)} e_j^*$$

so that

$$L_j = \begin{bmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ddots & 1 & \cdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & m_{j+1,j} & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & 1 & 0 \\ 0 & 0 & \cdots & m_{n,j} & \cdots & 0 & 1 \end{bmatrix}$$

and

$$L := (I + m^{(1)} e_1^*)(I + m^{(2)} e_2^*) \cdots (I + m^{(n-1)} e_{n-1}^*) = I + \sum_{k=1}^{n-1} m^{(k)} e_k^*$$

so that

$$L = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ m_{21} & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ m_{n-1,1} & m_{n-1,2} & \cdots & 1 & 0 \\ m_{n1} & m_{n2} & \cdots & m_{n,n-1} & 1 \end{bmatrix}$$

is a *unit* lower triangular matrix (see below).

7.7 Remarks

A matrix L is said to be unit lower triangular if it is lower triangular and has ones on its diagonal. Note that $\det L = 1$. Also, L^{-1} is also a unit lower triangular matrix (this follows from Cramer's rule). If L_1 and L_2 are both unit lower triangular matrices, then so it $L_1 L_2$. In a similar way we define unit upper triangular matrices.

7.8 Theorem (LU decomposition)

Let $A \in \mathbb{C}^{n \times n}$ have non-singular principal minors up to the order $n-1$. Then there is a decomposition

$$A = LU$$

where L is a unit lower triangular matrix and U is an upper triangular matrix. In that case

$$\det A = \det U = u_{11} \cdots u_{nn}$$

If in addition, A is non-singular, then the LU decomposition is unique.

Proof. The existence is the result of Gaussian elimination. To prove uniqueness, we argue by way of contradiction. Let $A = \tilde{L}\tilde{U} = LU$. As A is non-singular, it follows that $\tilde{U}$ is also non-singular, and hence $L^{-1}\tilde{L} = U\tilde{U}^{-1}$. Now, L^{-1} is unit lower triangular, so that $L^{-1}\tilde{L}$ is also unit lower triangular. On the other hand, $U\tilde{U}^{-1}$ is upper triangular. The only matrix that is both unit lower triangular and upper triangular is the identity matrix I, hence $L^{-1}\tilde{L} = U\tilde{U}^{-1} = I$. This implies $\tilde{L} = L$ and $\tilde{U} = U$.

7.9 Forward and backward substitutions

Assume that $A \in \mathbb{C}^{n\times n}$ is nonsingular and is decomposed as $A = LU$, where L is lower triangular and U upper triangular. To solve a system of equations $Ax = b$, one writes it as $LUx = b$ and then solves it in two steps:

Step 1. Denote $Ux = y$ and solve the lower triangular system $Ly = b$ for y via "forward substitution" (finding $y_1, \ldots, y_n$ subsequently).

Step 2. Solve the system $Ux = y$ for x via "backward substitution" (finding $x_n, \ldots, x_1$ subsequently).

7.10 Cost of computation

The cost of computation is measured in "flops", where a flop (*fl*oating point *op*eration) is an arithmetic operation (addition, subtraction, multiplication, division, or a root extraction). Let us estimate the cost of the LU decomposition. The cost of computation of $A^{(2)}$ is $n-1$ divisions to compute the multipliers and $2n(n-1)$ flops ($n-1$ rows with $2n$ flops per row) to make the zeros in the first column, i.e. total of approximately $2n^2$ flops. The computation of $A^{(3)}$ then takes $2(n-1)^2$ flops, and so on. Thus the total computational cost for the LU factorization is

$$2\big(n^2 + (n-1)^2 + \cdots + 1^2\big) = \frac{2n(n+1)(2n+1)}{6} \approx \frac{2n^3}{3}$$

flops.

If one solves a system $Ax = b$, then the LU decomposition is followed by solving two triangular systems (Section 7.9). The cost to solve one triangular

system is about n^2 flops. So there is an additional cost of $2n^2$ flops, which is negligible compared to $2n^3/3$. Hence, the total cost is still $\approx 2n^3/3$.

Note that the LU decomposition takes most of the computations required for solving a system $Ax = b$. Thus, this method is particularly well suited to very common situations in which one is solving systems $Ax = b$ for more than one vector b, but with the same matrix A. In that case the LU decomposition is done just once, and then each additional b will require $\approx 2n^2$ flops.

7.11 Computation of A^{-1}

Assume that $A \in \mathbb{C}^{n\times n}$ is non-singular and has non-singular principal minors up to the order $n-1$, so that the Gaussian elimination works. One can find the matrix $X = A^{-1}$ by solving the system $AX = I$ for $X \in \mathbb{C}^{n\times n}$. This amounts to solving n systems of linear equations $Ax_k = e_k$, for $k = 1, \ldots, n$, where x_k stands for the k-th column of the matrix X. The computational cost of this procedure is $2n^3/3 + n \times 2n^2 = 8n^3/3$. As a matter of fact, this is the fastest way of computing the inverse A^{-1}.

7.12 Diagonally dominant matrices

A matrix $A \in \mathbb{C}^{n\times n}$ such that

$$|a_{ii}| > \sum_{j\neq i} |a_{ij}|$$

for all i is said to be *strictly row diagonally dominant*. If

$$|a_{jj}| > \sum_{i\neq j} |a_{ij}|$$

for all j the matrix is said to be *strictly column diagonally dominant*.

7.13 Theorem

If a matrix A is strictly row (or column) diagonally dominant, then $\det A_k \neq 0$ for all $1 \leq k \leq n$. Hence, no zero pivots will be encountered during Gaussian elimination.

Note that if A is strictly column diagonally dominant, then all the multipliers (i.e., the elements of L_j) have absolute value less than one.

7.14 Remarks

It is easy to find matrices for which Gaussian elimination does *not* work. For example, if $A = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, then the method fails instantly. (Furthermore, for this A there is no decomposition $A = LU$ with L lower triangular and U upper triangular; see Exercise 7.2.)

In practice, when a pivot is close to zero, then multipliers become very large, and numerical computations tend to become very inaccurate. One tries to avoid this problem by using an appropriate *pivoting strategy*.

7.15 Partial pivoting

The idea is to avoid small (in absolute value) pivots by interchanging rows, if necessary. At any step of Gaussian elimination, one looks for the largest (in absolute value) element in the pivot column (at or below the main diagonal). For a non-singular matrix, it cannot happen that all of those elements in that column are zero. Then the row containing the largest element is interchanged with the current row. Now the largest element is on the main diagonal. After that the usual elimination step is performed.

7.16 Remarks

Partial pivoting ensures that all the multipliers (i.e., the elements of L_j) have absolute value less than or equal to one.

If a matrix A is strictly column diagonally dominant, then the Gaussian elimination with no pivoting is equivalent to Gaussian elimination with partial pivoting (i.e., no row interchanges are necessary).

7.17 Complete pivoting

The method of *complete pivoting* involves both row and column interchanges to make use of the largest pivot available. This method provides additional insurance against buildup of computational errors.

Exercise 7.1. Find a nonzero matrix $A \in \mathbb{R}^{2\times 2}$ that admits at least two LU decomposition, i.e. $A = L_1U_1 = L_2U_2$, where L_1 and L_2 are two *distinct* unit lower triangular matrices and U_1 and U_2 are two *distinct* upper triangular matrices.

Exercise 7.2. Show that the matrix $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ admits no LU decomposition, even if we only require that L be lower triangular (not necessarily unit lower triangular).

Exercise 7.3. The spectral radius of a matrix $A \in \mathbb{C}^{n\times n}$ is defined by

$$\rho(A) = \max\{|\lambda| : \ \lambda \text{ eigenvalue of } A\}.$$

(a) Show that $\rho(A) \leq \|A\|_2$.

(b) Give an example of a 2×2 matrix A such that $\rho(A) < 1$ but $\|A\|_2 > 100$.

(c) Show that if

$$\lim_{n\to\infty} \|A^n\|_2 = 0,$$

then $\rho(A) < 1$.

Exercise 7.4 (Bonus)**.** In the notation of the previous problem, show that if $\rho(A) < 1$, then

$$\lim_{n\to\infty} \|A^n\|_2 = 0.$$

Hint: use Jordan decomposition.

8 Cholesky factorization

8.1 LDM* Decomposition

Assume that $A \in \mathbb{C}^{n\times n}$ has non-singular principal minors up to the order n. Then there are unique matrices L, D, M such that L, M are unit lower triangular and D is diagonal, and

$$A = LDM^*$$

Proof. Let $A = LU$ be the LU decomposition of A and $u_{11}, \ldots, u_{nn}$ denote the diagonal entries of U. Set $D = \operatorname{diag}\{u_{11}, \ldots, u_{nn}\}$. Then the matrix $M^* := D^{-1}U$ is unit upper triangular, and $A = LDM^*$.

To establish uniqueness, let $A = LDM^* = L_1D_1M_1^*$. By the uniqueness of the LU decomposition, we have $L = L_1$. Hence, $(D_1^{-1}D)M^* = M_1^*$. Since both M^* and M_1^* are unit upper triangular, the diagonal matrix $D_1^{-1}D$ must be the identity matrix. Hence, $D = D_1$, and then $M = M_1$.

8.2 Corollary

If, in addition, A is Hermitian, then there exist a unique unit lower triangular matrix L and a unique diagonal matrix D such that

$$A = LDL^*$$

Moreover, the matrix D has real diagonal entries.

Proof. By the previous theorem $A = LDM^*$. Then $A = A^* = MD^*L^*$, and by the uniqueness of the LDM* decomposition we have $L = M$ and $D = D^*$.

8.3 Sylvester's Theorem

Let $A \in \mathbb{C}^{n\times n}$ be a Hermitian matrix. Then A is positive definite if and only if $\det A_k > 0$ for all $k = 1, \ldots, n$.

Proof. Let A be positive definite. By Section 4.15, $A = B^*B$, hence

$$\det A = \det B \times \det B^* = \det B \times \overline{\det B} = |\det B|^2 > 0.$$

Any principal minor A_k is also a Hermitian positive definite matrix, therefore by the same argument $\det A_k > 0$. Conversely, let $\det A_k > 0$. By Corollary 8.2 we have $A = LDL^*$. Denote by L_k and D_k the k-th principal minors of L and D, respectively. Then $A_k = L_kD_kL_k^*$. Note that

$\det D_k = \det A_k > 0$ for all $k = 1, \ldots, n$, therefore all the diagonal entries of D are real and positive. Lastly, $\langle Ax, x\rangle = \langle DL^*x, L^*x\rangle = \langle Dy, y\rangle > 0$ because $y = L^*x \neq 0$ whenever $x \neq 0$. □

8.4 Corollary

Let A be a positive definite matrix. Then $a_{ii} > 0$ for all $i = 1, \ldots, n$. Furthermore, let $1 \leq i_1 < i_2 < \cdots < i_k \leq n$, and let A' be the $k \times k$ matrix formed by the intersections of the rows and columns of A with numbers $i_1, \ldots, i_k$. Then $\det A' > 0$.

Proof. Just reorder the coordinates in $\mathbb{C}^n$ so that A' becomes a principal minor.

8.5 Cholesky Factorization

Let $A \in \mathbb{C}^{n\times n}$ be Hermitian and positive definite. Then there exists a unique lower triangular matrix G with real positive diagonal entries such that

$$A = GG^*$$

Proof. By Corollary 8.2 we have $A = LDL^*$. Let $D = \operatorname{diag}\{d_1, \ldots, d_n\}$. As it was shown in the proof of 8.3, all d_i are real and positive. Let $D^{1/2} = \operatorname{diag}\{\sqrt{d_1}, \ldots, \sqrt{d_n}\}$. Then $D = D^{1/2}D^{1/2}$ and setting $G = LD^{1/2}$ gives $A = GG^*$. The diagonal entries of G are $\sqrt{d_1}, \ldots, \sqrt{d_n}$, so they are positive.

To establish uniqueness, let $A = GG^* = \tilde{G}\tilde{G}^*$. Then $\tilde{G}^{-1}G = \tilde{G}^*(G^*)^{-1}$. Since this is the equality of a lower triangular matrix and an upper triangular one, then both matrices are diagonal:

$$\tilde{G}^{-1}G = \tilde{G}^*(G^*)^{-1} = D' = \operatorname{diag}\{d'_1, \ldots, d'_n\}.$$

Hence, $\tilde{G} = G(D')^{-1}$ and $\tilde{G}^* = D'G^* \Rightarrow \tilde{G} = G(D')^*$. Thus the diagonal components of $\tilde{G}$ are $\tilde{g}_{ii} = g_{ii}/d'_i = g_{ii}\bar{d}'_i$. This gives us $d'_i\bar{d}'_i = |d_i|^2 = 1$, and on the other hand $d'_i = g_{ii}/\tilde{g}_{ii}$ must be a real positive number. Therefore $d'_i = 1$ for each $i = 1, \ldots, n$, hence $D' = I$, and so $\tilde{G} = G$. □

8.6 Algorithm for Cholesky factorization

Here we outline the algorithm for computing the matrix $G = (g_{ij})$ from the matrix $A = (a_{ij})$, in the **real case**, $A \in \mathbb{R}^{n\times n}$. Note that G is lower

triangular, so $g_{ij} = 0$ for $i < j$. Hence,

$$a_{ij} = \sum_{k=1}^{\min\{i,j\}} g_{ik} g_{jk}$$

Setting $i = j = 1$ gives $a_{11} = g_{11}^2$, so $g_{11} = \sqrt{a_{11}}$. Next, for $2 \le i \le n$ we have $a_{i1} = g_{i1} g_{11}$, hence

$$g_{i1} = a_{i1}/g_{11} \qquad i = 2, \ldots, n$$

This gives the first column of G. Now, inductively, assume that we already have the first $j-1$ columns of G. Then $a_{jj} = \sum_{k=1}^{j} g_{jk}^2$, hence

$$g_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} g_{jk}^2}$$

Next, for $j+1 \le i \le n$ we have $a_{ij} = \sum_{k=1}^{j} g_{ik} g_{jk}$, hence

$$g_{ij} = \frac{1}{g_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} g_{ik} g_{jk} \right)$$

8.7 Cost of computation

The computation of g_{ij} takes $\approx 2j$ flops for each $i = j, \ldots, n$, so the total is

$$\sum_{j=1}^{n} 2j(n-j) \approx 2n\frac{n^2}{2} - 2\,\frac{n^3}{3} = \frac{n^3}{3}$$

Recall that the LU decomposition takes about $2n^3/3$ flops, so the Cholesky factorization is nearly twice as fast. It is also more stable than the LU decomposition, see Chapter 12.

8.8 Remark

The above algorithm can be used to verify that a given real symmetric matrix, A, is positive definite. Whenever all the square root extractions in Section 8.6 are possible and give non zero numbers, i.e. whenever

$$a_{11} > 0 \quad \text{and} \quad a_{jj} - \sum_{k=1}^{j-1} g_{jk}^2 > 0 \quad \forall j \ge 2$$

the matrix A is positive definite.

Exercise 8.1. (JPE, May 1994) Let $A \in \mathbb{R}^{n \times n}$ be given, symmetric and positive definite. Define $A_0 = A$, and consider the sequence of matrices defined by

$$A_k = G_k G_k^t \qquad \text{and} \qquad A_{k+1} = G_k^t G_k$$

where $A_k = G_k G_k^t$ is the Cholesky factorization for A_k. Prove that the A_k all have the same eigenvalues.

9 QR decomposition

9.1 Gram-Schmidt orthogonalization (revisited)

Let $v_1, \dots, v_n \in \mathbb{C}^n$ be a basis. The Gram-Schmidt orthogonalization, see Section 1.12, gives an ONB $\{u_1, \dots, u_n\}$ in $\mathbb{C}^n$ such that

$$\begin{aligned} v_1 &= r_{11}u_1, \\ v_2 &= r_{12}u_1 + r_{22}u_2, \\ v_3 &= r_{13}u_1 + r_{23}u_2 + r_{33}u_3, \\ &\dots \\ v_n &= r_{1n}u_1 + r_{2n}u_2 + \dots + r_{n-1,n}u_{n-1} + r_{nn}u_n \end{aligned}$$

where $r_{ik} = \langle v_k, u_i \rangle$ for $i < k \le n$ and $r_{kk} = \|w_k\|$. Note that $r_{kk} > 0$.

9.2 QR decomposition

For any $A \in \mathbb{C}^{m\times n}$ with $m \ge n$ there exist a unitary matrix $Q \in \mathbb{C}^{m\times m}$ and an upper triangular matrix $R \in \mathbb{C}^{m\times n}$ such that

$$A = QR$$

In addition, if A has full rank (i.e., $\operatorname{rank} A = n$), then Q can be chosen so that the diagonal entries of R are real and positive.

Proof. First, assume that $m = n$ and A is nonsingular. Let $v_1, \dots, v_n$ be the columns of A. Since they make a basis in $\mathbb{C}^n$, we can apply Gram-Schmidt orthogonalization and obtain a system of equations 9.1. Let Q be the matrix whose columns are $u_1, \dots, u_n$ and

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \end{bmatrix}$$

Then the above equations can be rewritten in a matrix form

$$A = QR$$

Since the columns of Q make an ONB, it is a unitary matrix.

Next, assume that $m = n$ but the matrix A is singular, so that its columns $v_1, \ldots, v_n$ are linearly dependent. Then the Gram-Schmidt algorithm 1.12 can be adjusted accordingly. In that case, for some k we may have $v_k \in \text{span}\{v_1, \ldots, v_{k-1}\}$, and then $w_k = 0$, in the notation of Section 1.12. Now the vector $u_k = w_k/\|w_k\|$ cannot be determined. Instead, we need to set u_k to an arbitrary unit vector orthogonal to $u_1, \ldots, u_{k-1}$, and then continue the calculations as described in Section 1.12. In that case, in the equations of Section 1.12 we get $r_{k,k} = 0$, but the procedure goes through.

Now let $m > n$. We extend the matrix A on the right by adding $m - n$ columns consisting of zeroes. Denote by A' the resulting $m \times m$ matrix with columns

$$v_1, v_2, \ldots, v_{n-1}, v_n,\ v_{n+1} = 0, \ldots, v_m = 0$$

This is a square matrix, so it has a QR decomposition

$$A' = Q'R'$$

The last $m - n$ columns of the matrix A are zeroes, so we obtain

$$0 = v_k = \sum_{i=1}^{k} r'_{1k} u'_k$$

for all $k = n + 1, \ldots, m$. Since $u'_1, \ldots, u'_m$ are basis vectors, $r'_{ik} = 0$ for all $k > n$ and $1 \le i \le m$, hence the last $m - n$ columns of R' are zeroes.

Now we erase the last $m - n$ columns of the matrices A' and R' to obtain the desired QR decomposition

$$A = QR$$

Theorem 9.2 is proved. □

Note: if A is a real matrix, then the matrices Q and R are also real, and so Q is orthogonal.

9.3 Reduced ("skinny") QR decomposition

Let $A \in \mathbb{C}^{m \times n}$ with $m \ge n$. Then there is a matrix $\hat{Q} \in \mathbb{C}^{m \times n}$ with orthonormal columns and an upper triangular matrix $\hat{R} \in \mathbb{C}^{n \times n}$ such that

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

Proof. By Theorem 9.2, $A = QR$. Let $\hat{Q}$ be the left $m \times n$ rectangular block of Q (the first n columns of Q). Let $\hat{R}$ be the top $n \times n$ square block of R (note that the remainder of R is zero). Then $A = \hat{Q}\hat{R}$. □

9.4 Corollary

If, in addition, A has full rank ($\operatorname{rank} A = n$), then

(a) The columns of $\hat{Q}$ make an ONB in the column space of A (this is the subspace in $\mathbb{C}^n$ spanned by the columns of A).

(b) One can find $\hat{Q}$ so that the diagonal entries of $\hat{R}$ will be real and positive ($r_{ii} > 0$).

(c) The $\hat{Q}$ and $\hat{R}$ described in part (b) are unique.

Proof. (a) and (b) follow from Section 9.2. To prove (c), let $A = \hat{Q}\hat{R} = \hat{Q}_1\hat{R}_1$. Then $A^*A = \hat{R}^*\hat{R} = \hat{R}_1^*\hat{R}_1$. Since A^*A is positive definite, we can use the uniqueness of Cholesky factorization and obtain $\hat{R} = \hat{R}_1$. Then also $\hat{Q}_1 = \hat{Q}\hat{R}\hat{R}_1^{-1} = \hat{Q}$. □

9.5 Cost of computation

In order to compute the reduced QR decomposition 9.3, one needs to apply Gram-Schmidt orthogonalization to the n columns of the matrix A and compute the n columns $u_1, \dots, u_n$ of the matrix $\hat{Q}$. The vector u_k is found by

$$w_k = v_k - \sum_{i=1}^{k-1} \langle v_k, u_i \rangle \, u_i, \quad \text{and} \quad u_k = \frac{w_k}{\|w_k\|},$$

see Section 1.12. Here each scalar product $\langle v_k, u_i \rangle$ requires m multiplications and m additions, and then subtracting every term $\langle v_k, u_i \rangle \, u_i$ from v_k requires m multiplication and m subtractions, for each $i = 1, \dots, k$. The total is $4mk$ flops. The subsequent computation of $\|w_{k+1}\|$ and then u_{k+1} requires $3m$ flops, which is a relatively small number, and we ignore it. The total flop count is

$$\sum_{k=1}^{n} 4mk \approx 2mn^2$$

9.6 Modified Gram-Schmidt orthogonalization

The algorithm 1.12 is often called *classical* Gram-Schmidt orthogonalization, as opposed to the *modified* Gram-Schmidt orthogonalization we present next. Given a basis $\{v_1, \dots, v_n\}$ in V, we denote $v_i^{(1)} = v_i$ for $i = 1, \dots, n$, then compute

$$u_1 = v_1^{(1)} / \|v_1^{(1)}\|,$$

and then modify all the remaining vectors by the rule

$$v_i^{(2)} = v_i^{(1)} - \langle v_i^{(1)}, u_1 \rangle \, u_1 \qquad \text{for} \;\; 1 < i \leq n.$$

After that, inductively, for each $k \geq 2$ we compute

$$u_k = v_k^{(k)} / \| v_k^{(k)} \|,$$

and then modify all the remaining vectors by the rule

$$v_i^{(k+1)} = v_i^{(k)} - \langle v_i^{(k)}, u_k \rangle \, u_k \qquad \text{for} \;\; k < i \leq n.$$

The modified and classical Gram-Schmidt methods produce the same orthonormal basis $\{u_1, \ldots, u_n\}$ (i.e., these two methods are mathematically equivalent). They are based on a different logic, though.

The classical Gram-Schmidt computes u_k by making the current vector v_k orthogonal to all the previously constructed vectors $u_1, \ldots, u_{k-1}$, without touching the remaining vectors $v_{k+1}, \ldots, v_n$. The amount of work increases as k grows from 1 to n. This is a "lazy man schedule" - do as little as possible and leave the rest of the work "for later".

The modified Gram-Schmidt computes u_k and makes all the remaining vectors $v_{k+1}, \ldots, v_n$ orthogonal to it. Once this is done, the remaining vectors will be in the orthogonal complement to the subspace $\text{span}\{u_1, \ldots, u_k\}$ and there is no need to involve the previously constructed vectors anymore. The amount of work decreases as k grows from 1 to n. This is an "industrious man schedule" - do as much as possible *now* and reduce the workload.

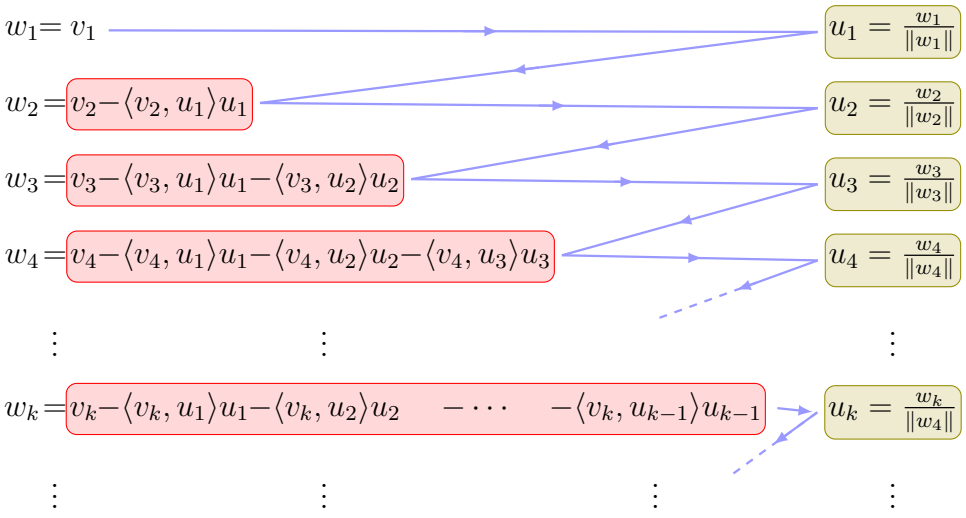
Overall, both methods require the same amount of flops. But the modified Gram-Scmidt has an important advantage that it gives more accurate numerical results in computer calculations; see programming assignment.

9.7 Computation of SVD

We note that there is no finite algorithms for the computation of SVD decomposition 5.1 or its reduced version 5.6 (except for small matrices, see Remark 5.5). The reason will be discussed in Chapter 16. In practice, SVD is computed by special iterative algorithms. The computation of reduced SVD requires approximately

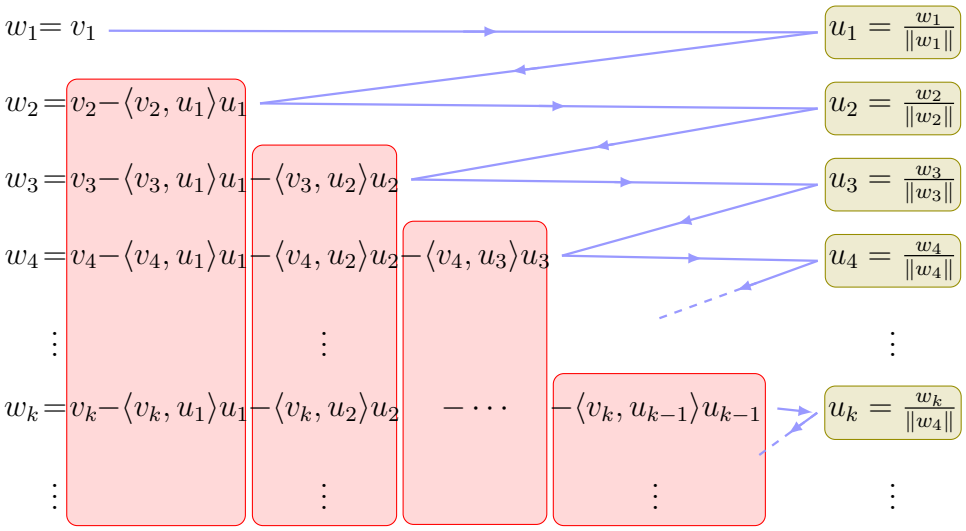
$$2mn^2 + 11n^3 \qquad \text{flops.}$$

Classical Gram-Schmidt:



The amount of work increases at each step (the red rows grow longer)

Modified Gram-Schmidt:



The amount of work decreases at each step (the red columns get shorter)

Exercise 9.1. (JPE, September 2002) Consider three vectors

$$v_1 = \begin{pmatrix} 1 \\ \epsilon \\ 0 \\ 0 \end{pmatrix}, \quad v_2 = \begin{pmatrix} 1 \\ 0 \\ \epsilon \\ 0 \end{pmatrix}, \quad v_3 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \epsilon \end{pmatrix}.$$

where $\epsilon \ll 1$.

(a) Use the classical Gram-Schmidt method to compute 3 **orthonormal** vectors q_1, q_2, q_3, making the approximation that $1 + \epsilon^2 \approx 1$ (that is, replace any term containing ϵ^2 or smaller with zero, **but** retain terms containing ϵ). Are q_i $(i = 1, 2, 3)$ pairwise orthogonal? If not, why not?

(b) Repeat (a) using the modified Gram-Schmidt orthogonalization process. Are the $q_i (i = 1, 2, 3)$ pairwise orthogonal? If not, why not?

10 Overdetermined linear systems

10.1 Definition

A system of linear equations $Ax = b$ with $A \in \mathbb{C}^{m\times n}$, $x \in \mathbb{C}^n$ and $b \in \mathbb{C}^m$, is said to be *overdetermined* if $m > n$. Since there are more equations than unknowns, the system usually has no solutions; so we will write it as $Ax \approx b$.

Recall that the matrix A defines a linear transformation $\mathbb{C}^n \to \mathbb{C}^m$. It is clear that $Ax = b$ has a solution if and only if $b \in \operatorname{Range} A$. In the latter case the solution is unique if and only if $\operatorname{Ker} A = \{0\}$, i.e. $\operatorname{rank} A = n$.

10.2 Least squares solution

Let $Ax \approx b$ be an overdetermined linear system. A vector $x \in \mathbb{C}^n$ that minimizes the function

$$E(x) = \|b - Ax\|_2$$

is called a *least squares solution* of $Ax \approx b$. The vector $r = b - Ax$ is called the *residual vector* and $\|r\|_2$ the *residual norm*.

10.3 Normal equations

Let $Ax \approx b$ be an overdetermined linear system. Then the linear system

$$A^*Ax = A^*b$$

is called the *system of normal equations* associated with $Ax \approx b$.

10.4 Theorem

Let $Ax \approx b$ be an overdetermined linear system. Then

(a) A vector x minimizes $E(x) = \|b - Ax\|_2$ if and only if it is an exact solution of the system $Ax = \hat{b}$, where $\hat{b}$ is the orthogonal projection of b onto $\operatorname{Range} A$.

(b) A vector x minimizing $E(x)$ always exists. It is unique if and only if A has full rank, i.e., if and only if $\operatorname{Ker} A = \{0\}$.

(c) A vector x minimizes $E(x)$ if and only if it is a solution of the system of normal equations $A^*Ax = A^*b$.

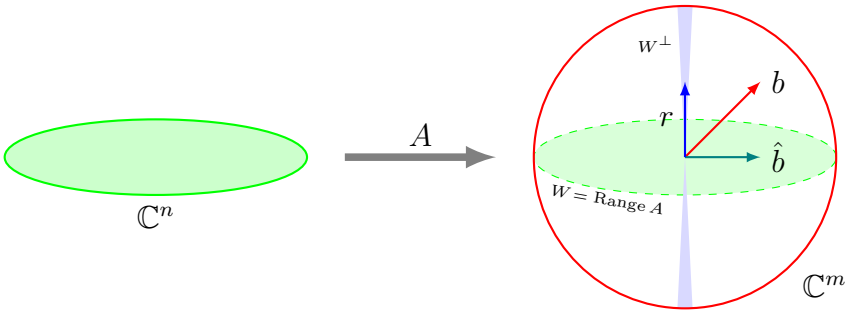
Proof. Denote $W = \operatorname{Range} A$. We have an orthogonal decomposition $\mathbb{C}^m = W \oplus W^\perp$, in particular $b = \hat{b} + r$, where $\hat{b} \in W$ and $r \in W^\perp$ are uniquely determined by b. Now Pythagorean Theorem gives

$$\begin{aligned}[E(x)]^2 = \|b - Ax\|_2^2 &= \| \underbrace{b - \hat{b}}_{r \in W^\perp} + \underbrace{\hat{b} - Ax}_{\in W} \|^2 \\ &= \|r\|_2^2 + \|\hat{b} - Ax\|_2^2 \geq \|r\|_2^2\end{aligned}$$

Hence, $\min_x E(x) = \|r\|_2$ is attained whenever $Ax = \hat{b}$. Since $\hat{b} \in \operatorname{Range} A$, there is always an $x \in \mathbb{C}^n$ such that $Ax = \hat{b}$. The vector x is unique whenever the map $A\colon \mathbb{C}^n \to \mathbb{C}^m$ is injective, i.e., whenever $\operatorname{Ker} A = \{0\}$, i.e., whenever A has full rank. This proves (a) and (b).
To prove (c), recall that $(\operatorname{Range} A)^\perp = \operatorname{Ker} A^*$, by Section 3.7, therefore $r = b - \hat{b} \in \operatorname{Ker} A^*$. Moreover, $b - Ax \in \operatorname{Ker} A^*$ if and only if $Ax = \hat{b}$, because $\hat{b}$ and r are uniquely determined by b. Now

$$x \text{ minimizes } E(x) \quad \Leftrightarrow \quad Ax = \hat{b} \quad \Leftrightarrow \quad Ax - b \in \operatorname{Ker} A^* \quad \Leftrightarrow \quad A^*Ax = A^*b$$

The proof is complete. □



Next we give examples that lead to overdetermined systems and least squares problems.

10.5 Linear least squares fit

Let (x_i, y_i), $1 \le i \le m$, be points in the xy plane. For any straight line $y = a_0 + a_1 x$ one defines the "combined distance" of that line from the given points by

$$E(a_0, a_1) = \left[\sum_{i=1}^{m} (a_0 + a_1 x_i - y_i)^2 \right]^{1/2}$$

The line $y = a_0 + a_1 x$ that minimizes the function $E(a_0, a_1)$ is called the *least squares fit* to the points (x_i, y_i). This is a basic tool in statistics. Let

$$A = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Then

$$E(a_0, a_1) = \|\mathbf{b} - A\mathbf{x}\|_2$$

Hence the least squares fit is equivalent to the least squares problem $A\mathbf{x} = \mathbf{b}$.

10.6 Polynomial least squares fit

Generalizing 10.5, one can fit a set of data points (x_i, y_i), $1 \le i \le m$, by a polynomial $y = p(x) = a_0 + a_1 x + \cdots + a_n x^n$ with $n + 1 \le m$. The least squares fit is based on minimizing the function

$$E(a_0, \ldots, a_n) = \left[\sum_{i=1}^{m} \Big(a_0 + a_1 x_i + \cdots + a_n x_i^n - y_i \Big)^2 \right]^{1/2}$$

This is equivalent to the least squares problem for an overdetermined linear system

$$a_0 + a_1 x_i + \cdots + a_n x_i^n = y_i \qquad 1 \le i \le m$$

in which $a_0, \ldots, a_n$ are unknowns.

10.7 Continuous least squares fit

Instead of fitting a discrete data set (x_i, y_i) one can fit a continuous function $y = f(x)$ on $[0, 1]$ by a polynomial $y = p(x) \in P_n(\mathbb{R})$. The least squares fit is based on minimization of

$$E(a_0, \ldots, a_n) = \left[\int_0^1 |f(x) - p(x)|^2 \, dx \right]^{1/2}$$

The solution of this problem is the orthogonal projection of $f(x)$ onto $P_n(\mathbb{R})$. To find the solution, consider a basis $\{1, x, \ldots, x^n\}$ in $P_n(\mathbb{R})$. Then $a_0, \ldots, a_n$ can be found by solving the system of equations (analogous to normal equations)

$$\sum_{j=0}^{n} a_j \langle x^j, x^i \rangle = \langle f, x^i \rangle \qquad 1 \le i \le n$$

The matrix of coefficients here is

$$\langle x^j, x^i \rangle = \int_0^1 x^{i+j} \, dx = \frac{1}{1+i+j}$$

for $0 \le i, j \le n$.

Next we present three methods for solving the least square problem.

10.8 Algorithm 1, based on normal equations

This is the simplest one:

1. Form the matrix A^*A and the vector A^*b.
2. Compute the Cholesky factorization $A^*A = GG^*$.
3. Solve the lower-triangular system $Gz = A^*b$ for z.
4. Solve the upper-triangular system $G^*x = z$ for x.

The cost of this algorithm is dominated by steps 1 and 2. Because of symmetry, the computation of A^*A requires $mn(n+1) \approx mn^2$ flops. The computation of A^*b requires only $2mn$ flops, a relatively small amount which we ignore. The Cholesky factorization takes $n^3/3$ flops (see 8.7), a total of

$$\boxed{\approx mn^2 + \tfrac{1}{3}n^3 \qquad \text{flops}}$$

10.9 Algorithm 2, based on QR decomposition

Using the reduced QR decomposition (Section 9.3) allows us to rewrite the system of normal equations as

$$\hat{R}^*\hat{Q}^*\hat{Q}\hat{R}x = \hat{R}^*\hat{Q}^*b$$

If A has full rank, the matrix $\hat{R}^*$ is nonsingular and we cancel it out. Also, since the columns of $\hat{Q}$ are orthonormal vectors, $\hat{Q}^*\hat{Q} = I$. Hence

$$\hat{R}x = \hat{Q}^*b$$

This suggests the following algorithm:

1. Compute the reduced QR decomposition $A = \hat{Q}\hat{R}$
2. Compute the vector $\hat{Q}^*b$.
3. Solve the upper-triangular system $\hat{R}x = \hat{Q}^*b$ for x.

The cost of this algorithm is dominated by step 1, the reduced QR factorization, which requires

$$\boxed{\approx 2mn^2 \qquad \text{flops}}$$

see 9.5. This is approximately twice as much as Algorithm 1 requires.

10.10 Algorithm 3, based on SVD decomposition

Using the reduced SVD decomposition 5.6 allows us to rewrite the system of normal equations as

$$V\hat{D}\hat{U}^*\hat{U}\hat{D}V^*x = V\hat{D}\hat{U}^*b$$

The matrix V is unitary and we cancel it out. If A has full rank, the matrix $\hat{D}$ is nonsingular and we cancel it out, too. Since the columns of $\hat{U}$ are orthonormal vectors, $\hat{U}^*\hat{U} = I$. Hence

$$\hat{D}V^*x = \hat{U}^*b$$

This suggests the following algorithm:

1. Compute the reduced SVD decomposition $A = \hat{U}\hat{D}V^*$
2. Compute the vector $\hat{U}^*b$.
3. Solve the diagonal system $\hat{D}z = \hat{U}^*b$ for z.
4. Set $x = Vz$.

The cost of this algorithm is dominated by step 1, the reduced SVD decomposition, which requires

$$\boxed{\approx 2mn^2 + 11n^3 \qquad \text{flops}}$$

see Section 9.7. This is approximately the same amount as in Algorithm 2 for $m \gg n$, but for $n \approx m$ this algorithm is much more expensive.

Algorithm 1 is the simplest and the cheapest, but it often gives inaccurate results in numerical calculations. Algorithms 2 and 3 are more complicated and expensive (in terms of flops), but usually give more accurate numerical results, for the reasons we learn in the next chapters.

10.11 The case of a rank deficient A

If $\operatorname{rank} A < n$, then the least squares solution is not unique: the set of solutions is $\{x \in \mathbb{C}^n \colon Ax = \hat{b}\}$, which is a line or a plane parallel to $\operatorname{Ker} A$. In this case the "best" solution is the one of minimal norm:

$$Ax_{\text{best}} = \hat{b} \quad \text{and} \quad \|x_{\text{best}}\|_2 \le \|x\|_2 \quad \forall x \colon Ax = \hat{b}$$

Algorithms 1 and 2 fail to find any solution for a rank deficient matrix A. On the contrary, Algorithm 3 easily finds the minimal norm solution as follows. When solving the diagonal system $\hat{D}z = \hat{U}^* b$ for z, we just set $z_i = 0$ whenever $d_{ii} = 0$ (in that case $(\hat{U}^* b)_i = 0$ automatically). This obviously gives a minimal norm vector z. Since $\|x\|_2 = \|Vz\|_2 = \|z\|_2$, we get a minimal norm vector x as well.

Exercise 10.1. (JPE, September 1997) Let

$$A = \begin{bmatrix} 3 & 3 \\ 0 & 4 \\ 4 & -1 \end{bmatrix}, \quad \text{and} \quad b = \begin{bmatrix} 2 \\ -2 \\ 1 \end{bmatrix}$$

Use the Gram-Schmidt process to find an orthonormal basis for the column space of A. Factor A into a product QR where $Q \in \mathbb{R}^{3\times 2}$ has an orthonormal set of column vectors and $R \in \mathbb{R}^{2\times 2}$ is upper triangular. Solve the least squares problem $Ax = b$. Compute the norm of the residual vector, $\|r\|$.

Exercise 10.2. (JPE, May 1998) Given the data (0,1), (3,4) and (6,5), use a QR factorization technique to find the best least squares fit by a linear function. Also, solve the problem via the system of normal equations.

11 Machine arithmetic

11.1 Decimal number system

In our decimal system, natural numbers are represented by a sequence of digits

$$N = (a_n \cdots a_1 a_0)_{10} = 10^n a_n + \cdots + 10 a_1 + a_0$$

where $a_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ are digits. Fractional numbers require an additional *fractional part*:

$$f = .d_1 d_2 \ldots d_t \ldots = 10^{-1} d_1 + 10^{-2} d_2 + \cdots + 10^{-t} d_t + \cdots$$

which may be finite or infinite.

11.2 Floating point representation

Alternatively, any real number can be written as a product of a fractional part with a sign and a power of ten:

$$r = \big(\pm .d_1 d_2 \ldots d_t \ldots\big) \times 10^e$$

where d_i are decimal digits and $e \in \mathbb{Z}$ is an integer. For example, $18.2 = 0.182 \times 10^2 = 0.0182 \times 10^3$, etc. This is called *floating point* representation of decimal numbers. The part $.d_1 \ldots d_t$ is called *mantissa* and e is called *exponent*. By changing the exponent e with a fixed mantissa $.d_1 \ldots d_t$ we can move ("float") the decimal point, for example $0.182 \times 10^2 = 18.2$ and $0.182 \times 10^1 = 1.82$.

11.3 Normalized floating point representation

To avoid unnecessary multiple representations of the same number (as that of 18.2 by 0.182×10^2 and 0.0182×10^3 above), we require that $d_1 \neq 0$. We say the floating point representation is *normalized* if $d_1 \neq 0$. Then 0.182×10^2 is the only normalized representation of the number 18.2.

For every positive real $r > 0$ there is a unique integer $e \in \mathbb{Z}$ such that $f := 10^{-e} r \in [0.1, 1)$. Then $r = f \times 10^e$ is the normalized representation of r. For most of real numbers, the normalized representation is unique, however, there are exceptions, such as

$$\big(.9999\ldots\big) \times 10^0 = \big(.1000\ldots\big) \times 10^1,$$

which admit dual normalized representations. In such cases one of the two representation has a finite fractional part, and the other – infinite.

11.4 Binary number system

In the binary number system, the base is 2 (instead of 10), and there are only two digits: 0 and 1. Any natural number N can be written, in the binary system, as a sequence of binary digits:

$$N = (a_n \cdots a_1 a_0)_2 = 2^n a_n + \cdots + 2a_1 + a_0$$

where $a_i \in \{0, 1\}$. For example, $5 = 101_2$, $11 = 1011_2$, $64 = 1000000_2$, etc. Binary system, due to its simplicity, is used by all computers. In the modern computer world, a *bit* means a *b*inary dig*it*.

11.5 Other number systems

Now suppose we are working in a number system with base $\beta \geq 2$. By analogy with Sections 11.1 and 11.4, any natural number N can be written, in that system, as a sequence of digits:

$$N = (a_n \cdots a_1 a_0)_\beta = \beta^n a_n + \cdots + \beta a_1 + a_0$$

where $a_i \in \{0, 1, \ldots, \beta - 1\}$ are digits. Fractional numbers require an additional *fractional part*:

$$f = .d_1 d_2 \ldots d_t \ldots = \beta^{-1} d_1 + \beta^{-2} d_2 + \cdots + \beta^{-t} d_t + \cdots$$

which may be finite or infinite. The floating point representation of real numbers in the system with base β is given by

$$r = \big(\pm .d_1 d_2 \ldots d_t \ldots\big) \times \beta^e$$

where $.d_1 \ldots d_t$ is called *mantissa* and $e \in \mathbb{Z}$ is called *exponent*. Again, we say that the above representation is normalized if $d_1 \neq 0$, this ensures uniqueness for almost all real numbers.

11.6 Machine floating point numbers

Every computer can only handle a certain number of bits in its memory or in its processor. Hence, the number of digits d_i's in the mantissa must be fixed, and possible values of the exponent e are limited to a certain fixed interval. Assume that the length of the mantissa is set to t digits and the exponent is bounded by $L \leq e \leq U$. Then the four integers β, t, L, U completely characterize the set of real numbers

$$r = \big(\pm .d_1 d_2 \ldots d_t\big) \times \beta^e, \quad L \leq e \leq U$$

that a given machine system can handle. Note that zero cannot be represented in the above format, since $d_1 \neq 0$. Every real machine systems includes a few special numbers, like zero, that require a different form of representation. We also note that any real computer can usually adopt many possible machine systems, with different values of t, L, U, see Section 11.8.

11.7 Remark

The maximal (in absolute value) number that a machine system can handle is $M = \beta^U(1 - \beta^{-t})$. The minimal positive number is $m = \beta^{L-1}$.

11.8 Examples

Most modern computers conform to the IEEE floating-point standard (ANSI/IEEE Standard 754-1985), which provides two machine systems:

Ⓘ *Single precision* is characterized by $t = 24$, $L = -125$ and $U = 128$.

Ⓘⓘ *Double precision* is characterized by $t = 53$, $L = -1021$ and $U = 1024$.

11.9 Relative errors

How can real numbers be represented in a machine system characterized by β, t, L, U? Let $x \neq 0$ be a real number with a normalized floating point representation

$$x = \big(\pm 0.d_1 d_2 \ldots\big) \times \beta^e$$

where the number of digits may be finite or infinite. If $e > U$ or $e < L$, then x cannot be correctly represented in the machine system (it is either "too large" or "too small"). If $e \in [L, U]$ is within the right range, then the mantissa has to be reduced to t digits (if it is longer or infinite). There are two standard ways to do such a reduction:

(a) keep the first t digits and chop off the rest;

(b) round off to the nearest available, i.e. use the rules

$$\begin{cases} .d_1 \ldots d_t & \text{if } \ d_{t+1} < \beta/2 \\ .d_1 \ldots d_t + .0 \ldots 01 & \text{if } \ d_{t+1} \geq \beta/2 \end{cases}$$

Denote the obtained number by x_c (the computer representation of x). The relative error in this representation can be estimated as

$$\frac{x_c - x}{x} = \varepsilon \quad \text{or} \quad x_c = x(1 + \varepsilon)$$

where the maximal possible value of ε is

$$\mathbf{u} = \begin{cases} \beta^{1-t} & \text{for chopped arithmetic (a)} \\ \frac{1}{2}\beta^{1-t} & \text{for rounded arithmetic (b)} \end{cases}$$

The number $\mathbf{u}$ is called the *unit round off* or the *machine precision.*

11.10 Examples

a) For the IEEE floating-point single precision standard with chopped arithmetic $\mathbf{u} = 2^{-23} \approx 1.2 \times 10^{-7}$. In other words, approximately 7 decimal digits are accurate.

b) For the IEEE floating-point double precision standard with chopped arithmetic $\mathbf{u} = 2^{-52} \approx 2.2 \times 10^{-16}$. In other words, approximately 16 decimal digits are accurate.

11.11 Example

Consider the system of equations

$$\begin{bmatrix} 0.01 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

The exact solution is $x = \frac{200}{197} \approx 1.015$ and $y = \frac{196}{197} \approx 0.995$.

Let us solve this system by using chopped arithmetic with base $\beta = 10$ and $t = 2$ (i.e. working with a two digit mantissa). If we use the Gaussian elimination without pivoting, then the computed solution will be $x_c = 0.0$ and $y_c = 1.0$. The value of x is 100% off! Increasing the length of the mantissa to $t = 3$ gives $x_c = 2.0$ and $y_c = 0.994$, not much of improvement, since x_c is still about 100% off. We postpone the explanation until Section 13.7.

On the other hand, if we apply the partial pivoting (interchanging rows), then the computed solution with $t = 2$ will be $x_c = 1.0$, $y_c = 1.0$, and with $t = 3$ it will be $x_c = 1.02$, $y_c = 0.994$, which is good. The table below shows that the relative error of the numerical solutions is proportional to minimal round-off error 10^{-t}, with a factor of about 2 to 5.

	relative error	min. error	factor
$t = 2$	1.5×10^{-2}	10^{-2}	1.5
$t = 3$	4.8×10^{-3}	10^{-3}	4.8
$t = 4$	2.2×10^{-4}	10^{-4}	2.2

Conclusion: Gaussian elimination without pivoting may lead to catastrophic errors and unreliable numerical solutions. Pivoting is more reliable... but see the next example:

11.12 Example

Consider another system of equations:

$$\begin{bmatrix} 3 & 1 \\ 1 & 0.35 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 5 \\ 1.7 \end{bmatrix}$$

The exact solution here is $x = 1$ and $y = 2$. The largest coefficient is at the top left corner already, so there is no need for pivoting.

Solving this system in chopped arithmetic with $\beta = 10$ and $t = 2$ gives $x_c = 0$ and $y_c = 5$, which is 150% off. Increasing the length of the mantissa to $t = 3$ gives $x_c = 0.883$ and $y_c = 2.35$, so the relative error is 17%. With $t = 4$, we obtain $x_c = 0.987$ and $y_c = 2.039$, now the relative error is 2%. The table below shows that the relative error of the numerical solutions is proportional to the minimal round-off error 10^{-t}, with a factor of about 150 to 200.

	relative error	min. error	factor
$t = 2$	1.5×10^{-0}	10^{-2}	150
$t = 3$	1.7×10^{-1}	10^{-3}	170
$t = 4$	2.0×10^{-2}	10^{-4}	200

We postpone a complete analysis of these two examples until Section 13.9.

11.13 Computational errors

Let x, y be two real numbers represented in a machine system by x_c, y_c. An arithmetic operation $x * y$, where $*$ is one of $+, -, \times, \div$, is performed by a computer in the following way. The computer finds $x_c * y_c$ exactly and then represents that number by the machine system. The result is $z = (x_c * y_c)_c$. Note that, generally, z is different from $(x*y)_c$, which is the machine representation of the exact result $x * y$. Hence, z is not necessarily the best representation for $x*y$. In other words, the computer makes additional round off errors during at each computation. Assuming that $x_c = x(1 + \varepsilon_1)$ and $y_c = y(1 + \varepsilon_2)$ we have

$$(x_c * y_c)_c = (x_c * y_c)\,(1 + \varepsilon_3) = [\,x(1 + \varepsilon_1)\,] * [\,y(1 + \varepsilon_2)\,]\,(1 + \varepsilon_3)$$

where $|\varepsilon_1|, |\varepsilon_2|, |\varepsilon_3| \le \mathbf{u}$.

11.14 Multiplication and division

For multiplication, we have

$$z = xy(1+\varepsilon_1)(1+\varepsilon_2)(1+\varepsilon_3) \approx xy(1+\varepsilon_1+\varepsilon_2+\varepsilon_3)$$

(here we ignore higher order terms), so the relative error is (approximately) bounded by $3\mathbf{u}$. A similar estimate can be made in the case of division:

$$z = \frac{x(1+\varepsilon_1)(1+\varepsilon_3)}{y(1+\varepsilon_2)} \approx \frac{x}{y}\,(1+\varepsilon_1-\varepsilon_2+\varepsilon_3)$$

(here we use Taylor expansion for $(1+\varepsilon_2)^{-1}$ and ignore higher order terms). Hence, machine multiplication and machine division increase relative errors by a factor of three, at most.

11.15 Addition and subtraction

For addition, we have

$$z = (x+y+x\varepsilon_1+y\varepsilon_2)(1+\varepsilon_3) = (x+y)\left(1+\frac{x\varepsilon_1+y\varepsilon_2}{x+y}\right)(1+\varepsilon_3)$$

The relative error is now small if $|x|$ and $|y|$ are not much bigger than $|x+y|$. Again ignoring higher order terms, we can bound the relative error of z by

$$\frac{|x|+|y|}{|x+y|}\,\mathbf{u}+\mathbf{u}$$

Thus, the operation of addition increases relative errors by a factor of

$$\frac{|x|+|y|}{|x+y|}+1$$

Similar estimates can be made in the case of subtraction $x-y$: it increases relative errors by a factor

$$\frac{|x|+|y|}{|x-y|}+1$$

We see that the addition and subtraction increase relative errors by a variable factor which depends on x and y. This factor may be arbitrarily large if $x+y \approx 0$ for addition or $x-y \approx 0$ for subtraction. This phenomenon is known as *catastrophic cancelation*. It occurred in our Example 11.11, where we attempted to solve the system without pivoting.

Exercise 11.1. (JPE, September 1993). Solve the system

$$\begin{pmatrix} 0.001 & 1.00 \\ 1.00 & 2.00 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1.00 \\ 3.00 \end{pmatrix}$$

using the LU decomposition with and without partial pivoting and chopped arithmetic with base $\beta = 10$ and $t = 3$ (i.e., work with a three digit mantissa). Obtain computed solutions (x_c, y_c) in both cases. Find the exact solution, compare, make comments.

Exercise 11.2. (JPE, May 2003). Consider the system

$$\begin{pmatrix} \varepsilon & 1 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Assume that $|\varepsilon| \ll 1$. Solve the system by using the LU decomposition with and without partial pivoting and adopting the following rounding off models (at all stages of the computation!):

$$\begin{aligned} a + b\varepsilon &= a \qquad (\text{for } a \neq 0), \\ a + b/\varepsilon &= b/\varepsilon \qquad (\text{for } b \neq 0). \end{aligned}$$

Find the exact solution, compare, make comments.

12 Conditioning

12.1 Condition number of a function

Let V and W be two normed vector spaces, and $f\colon V \to W$ a function. The *condition number* κ of f at a point $x \in V$ is defined by

$$\kappa = \kappa(f, x) = \lim_{\delta \to 0} \sup_{\|\Delta x\| \le \delta} \left(\frac{\|\Delta f\|}{\|f\|} \Big/ \frac{\|\Delta x\|}{\|x\|} \right)$$

where $\Delta f = f(x + \Delta x) - f(x)$. This is the maximal factor by which relative errors are magnified by f in the vicinity of the point x. The condition number characterizes the *sensitivity* of $f(x)$ to small perturbations of x.

12.2 Lemma

Let $\mathbb{C}^n \to \mathbb{C}^n$ be a linear operator defined by a nonsingular matrix $A \in \mathbb{C}^{n \times n}$, and let $\|\cdot\|$ be a norm on $\mathbb{C}^n$. Then for every $x \in \mathbb{C}^n$

$$\kappa(A, x) \le \|A\| \, \|A^{-1}\| \qquad \text{and} \qquad \sup_x \kappa(A, x) = \|A\| \, \|A^{-1}\|$$

where $\|A\|$ denotes the induced matrix norm.

Proof. Let $y = Ax$. Since $\Delta y = A(\Delta x)$, then

$$\kappa(A, x) = \sup_{\Delta x \ne 0} \frac{\|A(\Delta x)\|}{\|\Delta x\|} \frac{\|x\|}{\|Ax\|} = \|A\| \frac{\|x\|}{\|Ax\|}$$

and $\sup_x \|x\|/\|Ax\| = \sup_y \|A^{-1}y\|/\|y\| = \|A^{-1}\|$. □

12.3 Condition number of a matrix

For a nonsingular matrix $A \in \mathbb{C}^{n \times n}$, the condition number with respect to a given matrix norm $\|\cdot\|$ is defined by

$$\kappa(A) = \|A\| \, \|A^{-1}\|$$

We denote by $\kappa_1(A)$, $\kappa_2(A)$, $\kappa_\infty(A)$ the condition numbers with respect to the 1-norm, 2-norm, and ∞-norm, respectively.

12.4 Theorem

Let $A \in \mathbb{C}^{n\times n}$ be a nonsingular matrix and

$$Ax = b \tag{1}$$
$$(A + \Delta A)(x + \Delta x) = b + \Delta b \tag{2}$$

Assume that $\|\Delta A\|$ is small so that $\|\Delta A\|\,\|A^{-1}\| < 1$. Then

$$\frac{\|\Delta x\|}{\|x\|} \le \frac{\kappa(A)}{1 - \kappa(A)\frac{\|\Delta A\|}{\|A\|}} \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|}\right)$$

Proof. Expanding out the second equation (2), subtracting the first equation (1), and multiplying by A^{-1} gives

$$\Delta x = -A^{-1}\Delta A(x + \Delta x) + A^{-1}\Delta b$$

Taking norms and using the triangle inequality gives

$$\|\Delta x\| \le \|A^{-1}\|\,\|\Delta A\|\Big(\|x\| + \|\Delta x\|\Big) + \|A^{-1}\|\,\|\Delta b\|$$

Using $\|b\| \le \|A\|\,\|x\|$, the above inequality rearranges to

$$\Big(1 - \|A^{-1}\|\,\|\Delta A\|\Big)\|\Delta x\| \le \left(\|A^{-1}\|\,\|\Delta A\| + \|A^{-1}\|\,\|A\|\frac{\|\Delta b\|}{\|b\|}\right)\|x\|$$

Recall that $\|\Delta A\|\,\|A^{-1}\| < 1$, so the first factor above is positive. The theorem now follows immediately. □

Note: The smaller the condition number $\kappa(A)$, the tighter (better) estimate on $\|\Delta x\|/\|x\|$ we get. The value of $\kappa(A)$ thus characterizes the *sensitivity* of the solution of the linear system $Ax = b$ to small perturbations of A and b.

Interpretation. Let $Ax = b$ be a system of linear equations to be solved numerically. A computer represents A by $A_c = A + \Delta A$ and b by $b_c = b + \Delta b$. Assume that the computer finds the exact solution x_c of the perturbed system, i.e., x_c satisfies $A_c x_c = b_c$. Denote by $\Delta x = x_c - x$ the resulting error, where x denotes the exact solution of the true system $Ax = b$. Then the relative error $\|\Delta x\|/\|x\|$ can be estimated by Theorem 12.4.

12.5 Corollary

Consider the problem of solving a system of linear equations $Ax = b$ with a nonsingular matrix A. Then:

(a) If we fix A and vary b, we get a map $f_A\colon b \mapsto x$. The condition number of f_A satisfies

$$\kappa(f_A, b) \le \kappa(A) \quad \text{and} \quad \sup_{b\in\mathbb{C}^n} \kappa(A, b) = \kappa(A)$$

(b) If we fix b and vary A, we get a map $f_b\colon A \mapsto x$. The condition number of f_b satisfies

$$\kappa(f_b, A) \le \kappa(A)$$

Proof. Both inequalities immediately follow from Theorem 12.4. To prove the equality in (a), note that $x = A^{-1}b$ and apply Lemma 12.2. □

Remark. In the part (b), we actually have equality $\kappa(f_b, A) = \kappa(A)$, but the proof is beyond the scope of our course (it can be found in the textbook).

12.6 Corollary

Assume that in Theorem 12.4 we have $\|\Delta A\| \le \mathbf{u}\|A\|$ and $\|\Delta b\| \le \mathbf{u}\|b\|$, i.e. the matrix A and the vector b are represented with the best possible machine accuracy. Then

$$\frac{\|\Delta x\|}{\|x\|} \le \frac{2\mathbf{u}\kappa(A)}{1 - \mathbf{u}\kappa(A)}$$

12.7 Remark

Assume that $\mathbf{u} \approx 10^{-l}$, i.e. the machine system provides l accurate digits. Then if $\kappa(A) \approx 10^k$ with $k < l$, then $\|\Delta x\|/\|x\| \le 10^{-(l-k)}$, i.e. the numerical solution provides $l - k$ accurate digits.

In most practical considerations (as above) only the order of magnitude of $\kappa(A)$ matters, not its exact value. For instance, there is little difference between $\kappa(A) = 100$ and $\kappa(A) = 200$, it is still about 10^2.

Linear systems $Ax = b$ with small $\kappa(A)$ ($\sim$ 1, 10, 10^2) are often called *well-conditioned*. Those with large $\kappa(A)$ ($\sim$ 10^3, 10^4, etc.) are called *ill-conditioned*. Their numerical solutions are unreliable and should be avoided.

12.8 Proposition

1. We have
$$\kappa(A) = \frac{\sup_{\|x\|=1} \|Ax\|}{\inf_{\|x\|=1} \|Ax\|}$$
In other words, $\kappa(A)$ shows how much the linear map $A\colon \mathbb{C}^n \to \mathbb{C}^n$ distorts the unit sphere.

2. If a_j denotes the j-th column of A, then $\kappa(A) \geq \|a_j\|/\|a_i\|$

3. $\kappa(A) \geq 1$ and $\kappa(I) = 1$

4. $\kappa_2(A) = 1$ if and only if A is a multiple of a unitary matrix.

5. For any unitary matrix Q,
$$\kappa_2(QA) = \kappa_2(AQ) = \kappa_2(A)$$

6. If $D = \operatorname{diag}\{d_1, \ldots, d_n\}$ then
$$\kappa_2(D) = \kappa_1(D) = \kappa_\infty(D) = \frac{\max_{1\leq i\leq n} |d_i|}{\min_{1\leq i\leq n} |d_i|}$$

7. If A is Hermitian with eigenvalues $\lambda_1, \ldots, \lambda_n$, then
$$\kappa_2(A) = \frac{\max_i |\lambda_i|}{\min_i |\lambda_i|}$$

8. If $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ denote the singular values of A, then
$$\kappa_2(A) = \sigma_1/\sigma_n$$

9. We have
$$[\kappa_2(A)]^2 = \kappa_2(A^*A) = \kappa_2(AA^*) = \frac{\lambda_{\max}(A^*A)}{\lambda_{\min}(A^*A)}$$

10. We have $\kappa_2(A) = \kappa_2(A^*)$.

12.9 Remark

Another way to look at the condition number $\kappa(A)$ is the following:

$$\min\left\{\frac{\|A-A_s\|_2}{\|A\|_2}\colon\ A_s \text{ is singular}\right\}=\frac{1}{\kappa_2(A)}$$

hence $1/\kappa_2(A)$ is the relative distance from A to the nearest singular matrix. This follows from Section 5.10.

12.10 Remark

Here is yet another way to look at the condition number $\kappa(A)$. Since in practice the exact solution x of the system $Ax=b$ is rarely known, one can find its numerical solution x_c and compute the *residual vector* $r=b-Ax_c$. If it is small, the numerical solution x_c is good. But how small should it be?

Since $Ax_c=b+r$, Theorem 12.4 with $\Delta A=0$ implies that

$$\frac{\|x_c-x\|}{\|x\|}\le\kappa(A)\frac{\|r\|}{\|b\|}$$

If A is well conditioned, the smallness of $\|r\|/\|b\|$ ensures the smallness of the relative error $\|x_c-x\|/\|x\|$. If A is ill-conditioned, such a conclusion cannot be made: the smallness of r does not guarantee a good accuracy of x_c.

12.11 Definition

The condition number of a rectangular $m\times n$ matrix A with $m>n$ is defined by

$$\kappa(A)\colon\ =\frac{\sup_{\|x\|=1}\|Ax\|}{\inf_{\|x\|=1}\|Ax\|},$$

motivated by Proposition 12.8 (1). Under this definition, the property 8 of 12.8 still holds, and the property 9 must be shortened to

$$[\kappa_2(A)]^2=\kappa_2(A^*A)=\frac{\lambda_{\max}(A^*A)}{\lambda_{\min}(A^*A)}.$$

The above definition emphasizes the geometric interpretation of $\kappa_2(A)$: it is the maximum distortion factor of the map $A\colon\mathbb{C}^n\to\mathbb{C}^m$.

Exercise 12.1. (JPE, September 1997). Show that, given a matrix $A\in\mathbb{R}^{n\times n}$, one can choose vectors b and Δb so that if

$$Ax=b$$

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

then

$$\frac{||\Delta x||_2}{||x||_2} = \kappa_2(A)\frac{||\Delta b||_2}{||b||_2}$$

Explain the significance of this result for the 'optimal' role of condition numbers in the sensitivity analysis of linear systems.
(Hint: use SVD to show that it is enough to consider the case where A is a diagonal matrix.)

Exercise 12.2. (JPE, combined May 1997 and May 2008)

(a) Compute the condition numbers κ_1, κ_2 and κ_∞ for the matrix

$$A = \begin{pmatrix} 1 & 2 \\ 1.01 & 2 \end{pmatrix}$$

(b) Show that for every non-singular 2×2 matrix A we have $\kappa_1(A) = \kappa_\infty(A)$.

Exercise 12.3. (JPE, September 2002). Consider a linear system $Ax = b$. Let x^* be the exact solution, and let x_c be some computed approximate solution. Let $e = x^* - x_c$ be the error and $r = b - Ax_c$ the residual for x_c. Show that

$$\frac{1}{\kappa(A)}\frac{\|r\|}{\|b\|} \leq \frac{\|e\|}{\|x^*\|} \leq \kappa(A)\frac{\|r\|}{\|b\|}$$

Interpret the above inequality for $\kappa(A)$ close to 1 and for $\kappa(A)$ large.

Exercise 12.4. Prove properties 7 and 8 of condition numbers listed in Proposition 12.8.

Exercise 12.5. Suppose the condition number of a rectangular matrix $A \in \mathbb{C}^{m\times n}$ with $m > n$ is defined by

$$\kappa(A) := \frac{\sup_{\|x\|=1}\|Ax\|}{\inf_{\|x\|=1}\|Ax\|}.$$

Prove that

$$[\kappa_2(A)]^2 = \kappa_2(A^*A) = \frac{\lambda_{\max}(A^*A)}{\lambda_{\min}(A^*A)}$$

13 Stability

13.1 Round-off error analysis

When a system of linear equations $Ax = b$ is solved numerically, the computer only knows machine representations A_c and b_c of A and b. Then, at best, it can find $x_c = A_c^{-1}b_c$, instead of $x = A^{-1}b$. We know that the resulting error will be

$$E_1 \colon = \frac{\|x_c - x\|}{\|x\|} \lesssim 2\mathbf{u}\kappa(A)$$

by 12.6 (we ignored the small term $\mathbf{u}\kappa(A)$ in the denominator). This may be bad enough already when the matrix A is ill-conditioned. However, in reality things appear to be even worse, since the computer does *not* evaluate $x_c = A_c^{-1}b_c$ precisely, apart from trivial cases. The computer executes a certain sequence of arithmetic operations (a program) designed to solve the given system $Ax = b$. As the program runs, more and more round-off errors are made at each step and the errors compound toward the end. As a result, the computer finds a vector $\hat{x}_c$ different from x_c, i.e. $A_c\hat{x}_c \neq b_c$. The actual output $\hat{x}_c$ depends not only on the machine system but even more on the algorithm that is used to solve the system $Ax = b$. See the diagram nearby.

Of course, we do not expect the final error

$$E_2 \colon = \frac{\|\hat{x}_c - x\|}{\|x\|}$$

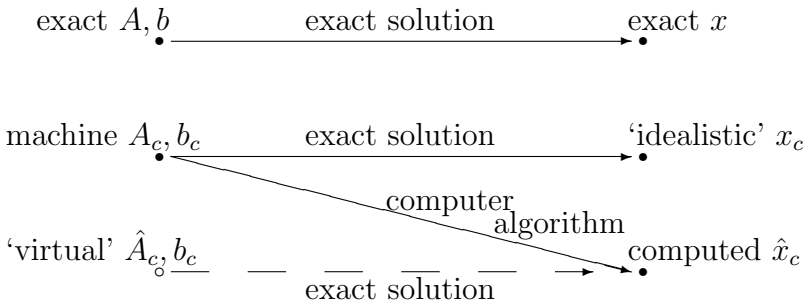
to be smaller than E_1, but we hope that it will not be much larger either. In other words, a good algorithm should not magnify the errors caused already by conditioning. If this is the case, the algorithm is said to be *stable*.

13.2 Stable algorithms (definition)

An algorithm for solving a system of linear equations $Ax = b$ is said to be *stable* (or *numerically stable*) if

$$\frac{\|\hat{x}_c - x\|}{\|x\|} \leq C\mathbf{u}\kappa(A)$$

where $C > 0$ is a constant. More precisely, C must be independent of A, b and the machine system, but it may depend on the size of the matrix, n.



13.3 Backward error analysis

In order to estimate the final error E_2, a typical approach is to "trace the errors backwards" and find another matrix, $\hat{A} = A + \delta A$ that satisfies $(A + \delta A)\hat{x}_c = b_c$. We call $A + \delta A$ a *virtual* matrix, since it is neither given nor computed numerically. Moreover, it is far from being unique. One wants to find a virtual matrix as close to A as possible, to make δA small, for the reasons made clear below.

13.4 Backward stable algorithms (definition)

An algorithm for solving a system of linear equations $Ax = b$ is said to be *backward stable* if there exists a virtual matrix $A + \delta A$ such that

$$\frac{\|\delta A\|}{\|A\|} \leq C\mathbf{u}$$

where $C > 0$ is a constant. More precisely, C must be independent of A, b and the machine system, but it may depend on the size of the matrix, n.

13.5 Theorem

Every backward stable algorithm is stable.

Proof. By Theorem 12.4,

$$\frac{\|\hat{x}_c - x\|}{\|x\|} \leq \frac{\kappa(A)}{1 - \kappa(A)\frac{\|\delta A\|}{\|A\|}} \frac{\|\delta A\|}{\|A\|} \lesssim C\mathbf{u}\kappa(A). \qquad \square$$

The proofs of stability (or instability) of algorithms of linear algebra are quite involved. We only present relevant facts here, without proofs.

13.6 Theorem (without proof)

If one uses the LU decomposition $A = LU$ for solving a system $Ax = b$, then there is a virtual matrix $A + \delta A$ such that

$$\|\delta A\| \leq C\|L\|\,\|U\|\mathbf{u},$$

where $C > 0$ is a constant independent of A and the machine system (but it may depend on the size of the matrix, n). Thus, the LU algorithm is unstable, its accuracy deteriorates when $\|L\|\,\|U\|$ is large.

13.7 Example

In Example 11.11, the LU decomposition (without pivoting) is

$$\begin{bmatrix} 0.01 & 2 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 100 & 1 \end{bmatrix} \begin{bmatrix} 0.01 & 2 \\ 0 & -197 \end{bmatrix}$$

hence $\|L\|\,\|U\| \sim 10^4$. This explains the huge errors of the corresponding numerical solutions that we observed.

13.8 Remarks

(a) Applying partial pivoting ensures that the entries of L are uniformly bounded: $|L_{ij}| \leq 1$. Also, it is observed in practice that in most cases $\|U\| \leq C\|A\|$, hence the partial pivoting algorithm is usually stable.

(b) The LU decomposition with complete pivoting is always stable, in this case one can prove that $\|U\| \leq C\|A\|$.

(c) The Cholesky factorization $A = GG^*$ of a positive definite matrix A is a particular form of the LU decomposition, so the above analysis applies. In this case, we know that

$$a_{ii} = \sum_{j=1}^{i} g_{ij}^2$$

see Section 8.6. Thus, one can easily prove that $\|G\| \leq C\|A\|^{1/2}$, hence the Cholesky factorization is always stable.

13.9 Example

In Example 11.11, the matrix

$$\begin{bmatrix} 0.01 & 2 \\ 1 & 3 \end{bmatrix}$$

has singular values $\sigma_1 = 3.7037$ and $\sigma_2 = 0.5319$, hence its condition number is $\kappa(A) = \sigma_1/\sigma_2 = 6.96$. This explains a moderate factor (≤ 5) by which relative errors of the numerical solutions are related to the minimal error 10^{-t} in Example 11.11.

In Example 11.12, the matrix

$$\begin{bmatrix} 3 & 1 \\ 1 & 0.35 \end{bmatrix}$$

has singular values $\sigma_1 = 3.33$ and $\sigma_2 = 0.0150$, hence its condition number is $\kappa(A) = 222$. This explains a large factor (up to 200) by which relative errors of the numerical solutions are related to the minimal error 10^{-t} in Example 11.12, even though we used a stable algorithm (the LU with complete pivoting). Remember that a stable algorithm *should not increase* errors already caused by conditioning, but it cannot cancel them out.

Exercise 13.1. (JPE, September 2004) Compute the LU decomposition $A = LU$ for the matrix

$$A = \begin{bmatrix} 0.01 & 2 \\ 1 & 3 \end{bmatrix}$$

Compute $\|L\|_\infty \|U\|_\infty$. What does this imply about the numerical stability of solving a system of linear equations $Ax = y$ by LU decomposition without pivoting?

14 Numerical solution of overdetermined systems

In Chapter 10 we discussed the least squares solution of overdetermined systems of equations

$$Ax = b$$

where $A \in \mathbb{C}^{m \times n}$ with $m \geq n$ and presented three algorithms: (i) based on normal equations, (ii) based on the QR decomposition, and (iii) based on the SVD decomposition.

14.1 Comparison of algorithms

(a) Algorithm 10.8 via normal equations has many advantages. It is the most compact and elegant one, and it is twice as cheap as the other two. It should be used whenever the computations are precise. However, if computations involve round-off errors, other considerations come into play. If the matrix A is ill-conditioned (i.e. its condition number is large, $\kappa_2(A) \gg 1$), then by Section 12.11

$$\kappa_2(A^*A) = [\kappa_2(A)]^2$$

hence the condition of the matrix A^*A will be much worse than that of A, and solving the normal equations can be disastrous. For example:

$$A = \begin{bmatrix} 1 & 1 \\ \varepsilon & 0 \\ 0 & \varepsilon \end{bmatrix}$$

then

$$A^*A = \begin{bmatrix} 1+\varepsilon^2 & 1 \\ 1 & 1+\varepsilon^2 \end{bmatrix}$$

If ε is so small that $\varepsilon^2 < \mathbf{u}$ (for example, $\varepsilon = 10^{-4}$ in single precision), then the matrix A^*A will be stored in computer memory as $\left[\begin{smallmatrix} 1 & 1 \\ 1 & 1 \end{smallmatrix}\right]$, which is a singular matrix, so the algorithm via normal equations will fail. Still, it is possible to find a good numerical solution to the original system $Ax = b$ if one uses more elaborate methods.

(b) Due to Proposition 12.8, $\kappa_2(A) = \kappa_2(QA)$ for any unitary matrix $Q \in \mathbb{C}^{m \times m}$, hence $\kappa_2(A) = \kappa_2(R)$ in the QR-based algorithm 10.9 and $\kappa_2(A) = \kappa_2(D)$ in the SVD-based algorithm 10.10. Hence, these

methods are safe, regarding the conditioning of the problem. The other aspect of numerical algorithms is stability. It turns out that the classical Gram-Schmidt orthogonalization (Section 1.12) is *unstable*, hence it leads to unpredictable round-off errors in numerical computations. On the other hand, the modified Gram-Schmidt algorithm (Section 9.6) is *stable*; see programming assignment. Even better algorithms for constructing the QR decomposition are based on reflection and rotation matrices, which we will learn in this chapter.

(c) The SVD-based algorithm 10.10 requires the computation of the SVD of the matrix A, for which no simple algorithm exists (the computation of SVD is beyond the scope of this course). Standard software packages (like MATLAB) use only stable algorithms for the SVD computation. Practically, the SVD-based method 10.10 is as good as the QR-based method 10.9 (see programming assignment). It is observed that the SVD-based method is more reliable when the matrix A is nearly singular or just singular, cf. Section 10.11.

14.2 Hyperplanes and reflections

Let V be a finite dimensional vector space. A subspace $W \subset V$ is called a *hyperplane* if $\dim W = \dim V - 1$. Note that in this case $\dim W^\perp = 1$.

Let $W \subset V$ be a hyperplane. For any vector $v \in V$ we have a unique decomposition $v = w + w'$, where $w \in W$ and $w' \in W^\perp$. The linear operator P on V defined by $Pv = w - w'$ is called a *reflection* (or *reflector*) across the hyperplane W. It is identity on W and negates vectors orthogonal to W.

14.3 Householder reflector matrices

Let $x \neq 0$ be a vector in $\mathbb{R}^n$ or $\mathbb{C}^n$. The $n \times n$ matrix

$$P = I - 2\,\frac{xx^*}{x^*x} = I - 2\,\frac{xx^*}{\|x\|^2}$$

is called the *Householder reflector matrix* corresponding to x. Obviously, P is unchanged if x is replaced by cx for any $c \neq 0$.

14.4 Theorem

Let P be the reflector matrix corresponding to a vector $x \neq 0$. Then

(a) $Px = -x$.

(b) $Py = y$ whenever $\langle y, x\rangle = 0$.

(c) P is Hermitian (in the real case it is symmetric).

(d) P is unitary (in the real case it is orthogonal).

(e) P is involution, i.e. $P^2 = I$.

Proof. Direct calculation.

14.5 Theorem

Let y be a vector in $\mathbb{R}^n$ or $\mathbb{C}^n$. Choose a scalar σ so that $|\sigma| = \|y\|$ and $\sigma \cdot \langle e_1, y\rangle \in \mathbb{R}$. Suppose that $x = y + \sigma e_1 \neq 0$. Let $P = I - 2xx^*/\|x\|^2$ be the reflector matrix defined in 14.3. Then $Py = -\sigma e_1$.

Proof. First, $\langle y - \sigma e_1, y + \sigma e_1\rangle = \|y\|^2 - \sigma\langle e_1, y\rangle + \bar{\sigma}\langle y, e_1\rangle - |\sigma|^2 = 0$. Now

14.4(a) implies:	$P(y + \sigma e_1) = -y - \sigma e_1$
14.4(b) implies:	$P(y - \sigma e_1) = \quad y - \sigma e_1$

Adding these two equations proves the theorem. □

14.6 Remarks

(a) To choose σ in Theorem 14.5, write a polar representation for $\langle e_1, y\rangle = \bar{y}_1 = re^{i\theta}$ and then set $\sigma = \pm\|y\|e^{-i\theta}$.

(b) In the real case, we have $y_1 \in \mathbb{R}$, and one can just set $\sigma = \pm\|y\|$.

(c) It is geometrically obvious that for any two unit vectors $x, y \in \mathbb{R}^n$ there is a reflector P that takes x to y. In the complex space $\mathbb{C}^n$, this is *not true*: for generic unit vectors $x, y \in \mathbb{C}^n$ there is no reflector that takes x to y. But according to Theorem 14.5, one can always find a reflector that takes x to cy with some scalar $c \in \mathbb{C}$.

14.7 Corollary

For any vector y in $\mathbb{R}^n$ or $\mathbb{C}^n$ there is a scalar σ (which was defined in 14.5 and specified in 14.6) and a matrix P, which is either a reflector or the identity ($P = I$), such that $Py = -\sigma e_1$.

Proof. Apply Theorem 14.5 in the case $y + \sigma e_1 \neq 0$ and set $P = I$ otherwise.

14.8 QR Decomposition via Householder reflectors

For any $A \in \mathbb{C}^{m\times n}$ with $m \geq n$ there is a QR decomposition with a unitary matrix $Q \in \mathbb{C}^{m\times m}$ that is a product of at most n reflector matrices.

Proof. We use induction on n. Let $n = 1$, so that A is a column m-vector. By Corollary 14.7 there is a matrix P (a reflection or identity) such that $PA = -\sigma e_1$ for a scalar σ. Hence, $A = PR$ where $R = -\sigma e_1$ is upper triangular. Now, let $n \geq 1$ and a_1 the first column of A. Again, by 14.7 there is a (reflection or identity) matrix P such that $Pa_1 = -\sigma e_1$. Hence,

$$PA = \begin{bmatrix} -\sigma & w^* \\ 0 & B \end{bmatrix}$$

where $w \in \mathbb{C}^{n-1}$ and $B \in \mathbb{C}^{(m-1)\times(n-1)}$. By the inductive assumption, there is a unitary matrix $Q_1 \in \mathbb{C}^{(m-1)\times(m-1)}$ and an upper triangular matrix $R_1 \in \mathbb{C}^{(m-1)\times(n-1)}$ such that $B = Q_1R_1$. Consider the unitary $m \times m$ matrix

$$Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & Q_1 \end{bmatrix}$$

By Section 2.9, the matrix Q_2 is unitary whenever Q_1 is. Furthermore, if Q_1 is a product of $\leq n-1$ reflectors, then the same is true for Q_2. Now one can easily check that $PA = Q_2R$ where

$$R = \begin{bmatrix} -\sigma & w^* \\ 0 & R_1 \end{bmatrix}$$

is an upper triangular matrix. Hence, $A = QR$ with $Q = PQ_2$.

14.9 Remark

In the real case, there are two choices for the scalar σ, that is $\sigma = \pm\|y\|$; see Remark 14.6 (b). The better one is

$$\sigma = \operatorname{sgn}(y_1)\,\|y\|$$

i.e. the sign of σ is determined by the sign of y_1. Then computing the vector $x = y + \sigma e_1$ is always stable, there is no danger of catastrophic cancellation.

14.10 Givens rotation matrices

Let $1 \leq p < q \leq m$ and $\theta \in [0, 2\pi)$. The matrix $G = G_{p,q,\theta} = (g_{ij})$ defined by $g_{pp} = \cos\theta$, $g_{pq} = \sin\theta$, $g_{qp} = -\sin\theta$, $g_{qq} = \cos\theta$ and $g_{ij} = \delta_{ij}$ otherwise is called a *Givens rotation matrix* (or a *Givens rotator*). It defines a rotation through the angle θ of the x_px_q coordinate plane in $\mathbb{R}^m$ with all the other coordinates fixed. Obviously, G is an orthogonal matrix.

14.11 QR decomposition via Givens rotators

For any $A \in \mathbb{R}^{m\times n}$ with $m \geq n$ there is a QR decomposition with an orthogonal matrix $Q \in \mathbb{R}^{m\times m}$ that is a product of Givens rotators.

Proof. Let a_j be the leftmost column of A that contains a nonzero entry below the main diagonal, $a_{ij} \neq 0$ with some $i > j$. Consider the matrix $A' = GA$ where $G = G_{j,i,\theta}$ is the Givens rotator. One easily checks that
(a) the first $j-1$ columns of A' are zero below the main diagonal;
(b) in the j-th column, only the elements a'_{jj} and a'_{ij} will be different from the corresponding elements of A, and moreover

$$a'_{ij} = -a_{jj}\sin\theta + a_{ij}\cos\theta$$

Now we want to find $\sin\theta$ and $\cos\theta$ to make $a'_{ij} = 0$. For example,

$$\cos\theta = \frac{a_{jj}}{\sqrt{a_{jj}^2 + a_{ij}^2}} \qquad \text{and} \qquad \sin\theta = \frac{a_{ij}}{\sqrt{a_{jj}^2 + a_{ij}^2}}$$

will do. Note that one never actually evaluates the angle θ, since $G_{j,i,\theta}$ only contains $\cos\theta$ and $\sin\theta$, and these are given by the above formulas.

In this way we eliminate one nonzero element a_{ij} below the main diagonal. Working from left to right, one can convert A into an upper triangular matrix $\tilde{G}A = R$ where $\tilde{G}$ is a product of Givens rotators. Each nonzero element of A below the main diagonal requires one multiplication by a rotator. Then we get $A = QR$ with an orthogonal matrix $Q = \tilde{G}^T$. □

14.12 Cost of QR via Givens rotators

The evaluation of $\cos\theta$ and $\sin\theta$ takes 6 flops (the square root extraction is counted here as one flop), then the subsequent multiplication of A by $G_{j,i,\theta}$ takes $6n$ flops. Thus, if A originally had p nonzero subdiagonal entries, then the QR decomposition via Givens rotators takes $6pn$ flops.

When p is close to its maximal value, $mn - n^2/2$, then the total cost $\sim 6mn^2 - n^3/2$ greatly exceeds the cost of QR via Householder reflectors or Gram-Schmidt decomposition. Hence Givens rotators are very inefficient for generic matrices. But they work well if the matrix A is *sparse*, i.e. contains just a few nonzero elements below the main diagonal. Then Givens rotators can give the quickest result. We will see such instances later.

Exercise 14.1. Let $x, y \in \mathbb{C}^n$ be such that $x \neq y$ and $\|x\|_2 = \|y\|_2 \neq 0$. Show that there is a reflector matrix P such that $Px = y$ if and only if $\langle x, y \rangle \in \mathbb{R}$. For an extra credit: show that if the above reflector exists, then it is unique.

Exercise 14.2. (simplified of JPE, May 2011) Prove that any Givens rotator matrix in $\mathbb{R}^2$ is a product of two Householder reflector matrices. Can a Householder reflector matrix be a product of Givens rotator matrices?

Exercise 14.3 (Bonus)**.** (JPE May, 2010) Let

$$A = \begin{bmatrix} 3 & -3 \\ 0 & 4 \\ 4 & 1 \end{bmatrix}$$

(a) Find the QR factorization of A by Householder reflectors.

(b) Use the results in (a) to find the least squares solution of $Ax = b$, where

$$b = [16 \;\; 11 \;\; 17]^T$$

(*Note: there is a typo in the original JPE exam, it is corrected here.*)

15 Computation of eigenvalues: theory

15.1 Preface

Eigenvalues of a matrix $A \in \mathbb{C}^{n\times n}$ are the roots of its characteristic polynomial, $C_A(x)$. It is a consequence of the famous Galois group theory (Abel's theorem) that there is no closed formula for the roots of a generic polynomial of degree > 4. Hence, there are no finite algorithms for computation of the roots of polynomials (or eigenvalues, for that matter).

Thus, all the methods for computing eigenvalues of matrices of size $n \geq 5$ are necessarily iterative, they provide successive approximations to the eigenvalues, but never exact results. Furthermore, even though for $n = 3$ and $n = 4$ exact formulas exist, they are rather impractical and often numerically unstable, so even in these cases iterative methods should be used instead.

For this reason, matrix decompositions that involve eigenvalues (Schur and SVD) cannot be implemented by finite algorithms. On the other hand, decompositions that do *not* involve eigenvalues (e.g, QR, or LU, or Cholesky) *can* be implemented by finite algorithms (and we learned some of those).

Now we learn iterative algorithms for computing eigenvalues and eigenvectors. Note that the most important matrix decomposition, SVD, requires the eigenvalues of a Hermitian positive semidefinite matrix A^*A, hence it is particularly important to develop algorithms for this class of matrices.

If an eigenvalue λ of a matrix A is known, an eigenvector x can be found by solving the linear system $(A - \lambda I)x = 0$ (say, by LU decomposition). Conversely, if an eigenvector x is known, the corresponding eigenvalue λ can be immediately found by $(Ax)_i/x_i$ whenever $x_i \neq 0$. Hence, eigenvalues and eigenvectors are often computed 'in parallel'. In this chapter, we develop a theoretical basis for computation of eigenvalues and eigenvectors, while in the next chapter we turn to practical algorithms.

15.2 Rayleigh quotient

Let $A \in \mathbb{C}^{n\times n}$. We call

$$r(x) = \frac{x^*Ax}{x^*x} = \frac{\langle Ax, x\rangle}{\langle x, x\rangle}, \qquad x \neq 0$$

the *Rayleigh quotient* of A. It is a function on $\mathbb{C}^n \setminus \{0\}$, with values in $\mathbb{C}$.

Note: $r(cx) = r(x)$ for $c \neq 0$, hence $r(x)$ is constant on the line span$\{x\}$ (with the zero vector removed). Since any nonzero vector is a scalar multiple

of a unit vector, $r(x)$ is completely defined by its values on the unit sphere $\mathbb{S}_1$, on which

$$r(x) = x^*Ax = \langle Ax, x\rangle, \qquad \text{because } \langle x, x\rangle = 1 \text{ on } \mathbb{S}^1$$

Thus $r(x)$ is a quadratic function of the coordinates of x, on the sphere $\mathbb{S}_1$.

If A is Hermitian, then $r(x) \in \mathbb{R}$ for any nonzero $x \in \mathbb{C}^n$.

If x is a unit eigenvector with an eigenvalue λ, then $Ax = \lambda x$ and so

$$r(x) = \lambda$$

If x is an arbitrary unit vector, then $r(x)\,x = (x^*Ax)\,x$ is the orthogonal projection of the vector Ax on the line spanned by x. Hence

$$\|Ax - r(x)\,x\|_2 = \min_{\mu\in\mathbb{C}} \|Ax - \mu x\|_2$$

If one regards x as an 'approximate' eigenvector, then the Rayleigh quotient $r(x)$ is the best choice that one could make for the associated 'approximate' eigenvalue in the sense that the value $\mu = r(x)$ comes closest (in the 2-norm) to achieving the desired relation $Ax - \mu x = 0$.

15.3 Theorem

Let $A \in \mathbb{C}^{n\times n}$ and x a unit eigenvector of A corresponding to eigenvalue λ. Let y be another unit vector and $r = y^*Ay$. Then

$$|\lambda - r| \le 2\,\|A\|_2\,\|x - y\|_2$$

Moreover, if A is a Hermitian matrix, then there is a constant $C = C(A) > 0$ such that

$$|\lambda - r| \le C\,\|x - y\|_2^2$$

Proof. To prove the first part, put

$$\lambda - r = x^*A(x - y) + (x - y)^*Ay$$

and then use the triangle inequality and Cauchy-Schwarz inequality. Now, assume that A is Hermitian. Then there is an ONB of eigenvectors, and we can assume that x is one of them. Denote that ONB by $\{x, x_2, \ldots, x_n\}$ and

the corresponding eigenvalues by $\lambda, \lambda_2, \ldots, \lambda_n$. Let $y = cx + c_2x_2 + \cdots + c_nx_n$. Then

$$\|y - x\|^2 = |c - 1|^2 + \sum_{i=2}^{n} |c_i|^2 \geq \sum_{i=2}^{n} |c_i|^2$$

On the other hand, $\|y\| = 1$, so

$$\lambda = \lambda|c|^2 + \sum_{i=2}^{n} \lambda|c_i|^2$$

Now, $Ay = c\lambda x + \sum_{i=2}^{n} c_i\lambda_i x_i$, so

$$r = \langle Ay, y\rangle = \lambda|c|^2 + \sum_{i=2}^{n} \lambda_i|c_i|^2$$

Therefore,

$$\lambda - r = \sum_{i=2}^{n} (\lambda - \lambda_i)|c_i|^2$$

The result now follows with

$$C = \max_{2\leq i\leq n} |\lambda - \lambda_i|$$

The theorem is proved. □

15.4 Lemma

Let $A \in \mathbb{C}^{n\times n}$ be a Hermitian matrix with eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$ (they are all real, hence we can order them). Then for any $\|x\| = 1$

$$\lambda_1 \leq x^*Ax \leq \lambda_n$$

Proof. Let $\{x_1, \ldots, x_n\}$ be an ONB of eigenvectors of A and $x = c_1x_1 + \cdots + c_nx_n$. Then $x^*Ax = \lambda_1|c_1|^2 + \cdots + \lambda_n|c_n|^2$. The result now follows easily. □

15.5 Lemma

Let L and G be subspaces of $\mathbb{C}^n$ and $\dim G > \dim L$. Then there is a nonzero vector in G orthogonal to L.

Proof. By way of contradiction, if $G \cap L^\perp = \{0\}$, then $G \oplus L^\perp$ is a subspace of $\mathbb{C}^n$ with dimension $\dim G + n - \dim L$. Now our assumptions imply that this value exceeds n, a contradiction. □

15.6 Courant-Fisher Minimax Theorem

Let $A \in \mathbb{C}^{n\times n}$ be a Hermitian matrix with eigenvalues $\lambda_1 \le \cdots \le \lambda_n$ (they are all real, hence we can order them). Then for every $i = 1, \ldots, n$

$$\lambda_i = \min_{L\colon \dim L = i} \max_{x\in L\setminus\{0\}} \frac{x^*Ax}{x^*x}$$

where L stands for a vector subspace of $\mathbb{C}^n$.

Proof. Let $\{u_1, \ldots, u_n\}$ be an ONB of eigenvectors of A corresponding to the eigenvalues $\lambda_1, \ldots, \lambda_n$. If $\dim L = i$, then by Lemma 15.5 there is a nonzero vector $x \in L$ orthogonal to the space $\operatorname{span}\{u_1, \ldots, u_{i-1}\}$. Hence, the first $i-1$ coordinates of x are zero, i.e. $x = \sum_{j=i}^n c_j u_j$. Thus

$$\frac{x^*Ax}{x^*x} = \frac{\sum_{j=i}^n |c_j|^2\lambda_j}{\sum_{j=i}^n |c_j|^2} \ge \lambda_i$$

Therefore,

$$\max_{x\in L\setminus\{0\}} \frac{x^*Ax}{x^*x} \ge \lambda_i$$

Now, take the subspace $L = \operatorname{span}\{u_1, \ldots, u_i\}$. Obviously, $\dim L = i$ and for every nonzero vector $x \in L$ we have $x = \sum_{j=1}^i c_j u_j$, so

$$\frac{x^*Ax}{x^*x} = \frac{\sum_{j=1}^i |c_j|^2\lambda_j}{\sum_{j=1}^i |c_j|^2} \le \lambda_i$$

The theorem is proved. □

15.7 Theorem

Let A and ΔA be Hermitian matrices. Let $\alpha_1 \le \cdots \le \alpha_n$ be the eigenvalues of A. Let $\delta_{\min}$ and $\delta_{\max}$ the smallest and the largest eigenvalues of ΔA. Denote the eigenvalues of the matrix $B = A + \Delta A$ by $\beta_1 \le \cdots \le \beta_n$. Then for each $i = 1, \ldots, n$

$$\alpha_i + \delta_{\min} \le \beta_i \le \alpha_i + \delta_{\max}$$

Proof. Let $\{u_1, \ldots, u_n\}$ be an ONB of eigenvectors of A corresponding to the

eigenvalues $\alpha_1, \ldots, \alpha_n$. Let $L = \text{span}\{u_1, \ldots, u_i\}$. Then, by Theorem 15.6

$$\begin{aligned}
\beta_i &\leq \max_{x \in L \setminus \{0\}} \frac{x^* B x}{x^* x} \\
&\leq \max_{x \in L \setminus \{0\}} \frac{x^* A x}{x^* x} + \max_{x \in L \setminus \{0\}} \frac{x^* \Delta A x}{x^* x} \\
&\leq \alpha_i + \max_{x \in C^n \setminus \{0\}} \frac{x^* \Delta A x}{x^* x} \\
&= \alpha_i + \delta_{\max}
\end{aligned}$$

which is the right inequality. Now apply this theorem to the matrices B, $-\Delta A$ and $A = B + (-\Delta A)$. Then its right inequality, just proved, will read $\alpha_i \leq \beta_i - \delta_{\min}$ (note that the largest eigenvalue of $-\Delta A$ is $-\delta_{\min}$). The theorem is completely proved. □

15.8 Corollary

Since $\|\Delta A\|_2 = \max\{|\delta_{\min}|, |\delta_{\max}|\}$, we have

$$\alpha_i - \|\Delta A\|_2 \leq \beta_i \leq \alpha_i + \|\Delta A\|_2 \qquad \forall i = 1, \ldots, n$$

15.9 Remark

Suppose one knows an approximate eigenvalue λ and an approximate unit eigenvector x of a matrix A. To estimate the closeness of λ to the actual but unknown eigenvalue of A, one can compute the residual $r = Ax - \lambda x$. Assume that r is small and define the matrix $\Delta A = -rx^*$. Then $\|\Delta A\|_2 = \|r\|_2$ and

$$(A + \Delta A)x = Ax - rx^*x = \lambda x$$

Therefore, (λ, x) are an exact eigenpair of perturbed matrix $A + \Delta A$, and the norm $\|\Delta A\|_2$ is known. One could then apply Corollary 15.8 to estimate the closeness of λ to the actual eigenvalue of A, if the matrices A and ΔA were Hermitian. Since this is not always the case, we need to study how eigenvalues of a generic matrix change under small perturbations of the matrix. This is the issue of *eigenvalue sensitivity*.

15.10 Bauer-Fike theorem

Let $A \in \mathbb{C}^{n \times n}$ be a diagonalizable matrix, so that

$$X^{-1} A X = D = \text{diag}\{\lambda_1, \ldots, \lambda_n\}$$

If μ is an eigenvalue of a perturbed matrix $A + \Delta A$, then

$$\min_{1 \le i \le n} |\lambda_i - \mu| \le \kappa_p(X) \, \|\Delta A\|_p$$

where $\| \cdot \|_p$ stands for any p-norm ($1 \le p \le \infty$).

Proof. If μ is an eigenvalue of A, the claim is trivial. If not, the matrix $D - \mu I$ is invertible. Observe that

$$\begin{aligned} X^{-1}(A + \Delta A - \mu I)X &= D + X^{-1}\Delta A X - \mu I \\ &= (D - \mu I)\big[I + (D - \mu I)^{-1}(X^{-1}\Delta A X)\big] \end{aligned}$$

Since the matrix $A + \Delta A - \mu I$ is singular, so is the matrix $I + (D - \mu I)^{-1}(X^{-1}\Delta A X)$. Then the Neumann lemma (Exercise 1.2) implies

$$1 \le \|(D - \mu I)^{-1}(X^{-1}\Delta A X)\|_p \le \|(D - \mu I)^{-1}\|_p \|X^{-1}\|_p \|\Delta A\|_p \|X\|_p$$

Lastly, observe that $(D - \mu I)^{-1}$ is diagonal, so

$$\|(D - \mu I)^{-1}\|_p = \max_{1 \le i \le n} \frac{1}{|\lambda_i - \mu|} = \frac{1}{\min_{1 \le i \le n} |\lambda_i - \mu|}$$

The theorem now follows. □

15.11 Corollary

If A is a normal matrix, then in the above theorem

$$\min_{1 \le i \le n} |\lambda_i - \mu| \le \|\Delta A\|_2$$

because X is a unitary matrix and so $\kappa_2(X) = 1$.

Theorem 15.10 answers the question raised in Section 15.9, it gives an estimate on the error in the eigenvalue in terms of $\|\Delta A\|$ and $\kappa(X)$. However, this answer is not good enough – it gives one estimate for all eigenvalues. In practice, some eigenvalues can be estimated better than others. It is important then to develop finer estimates for individual eigenvalues.

15.12 Left eigenvectors (definition)

Let $A \in \mathbb{C}^{n \times n}$. A nonzero vector $x \in \mathbb{C}^n$ is called a *left eigenvector* of A corresponding to an eigenvalue λ if

$$x^* A = \lambda x^*$$

Note that this is equivalent to $A^* x = \bar{\lambda} x$, i.e. x being an ordinary (right) eigenvector of A^* corresponding to the eigenvalue $\bar{\lambda}$.

15.13 Lemma

A matrix A has a left eigenvector corresponding to λ if and only if λ is an eigenvalue of A (a root of the characteristic polynomial of A).

Proof. $x^*A = \lambda x^*$ for an $x \neq 0$ is equivalent to $(A^* - \bar{\lambda}I)x = 0$, which means that $\det(A^* - \bar{\lambda}I) = 0$, or equivalently, $\det(A - \lambda I) = 0$, i.e. $C_A(\lambda) = 0$. □

This explains why we do not introduce a notion of a *left* eigenvalue: the set of eigenvalues for left eigenvectors is just the same as the set of eigenvalues for ordinary (right) eigenvectors.

15.14 Lemma

For any eigenvalue λ of A the dimension of the ordinary (right) eigenspace equals the dimension of the left eigenspace (i.e., the *geometric multiplicity* of λ is the same, in the left and right senses).

Proof. $\dim \operatorname{Ker}(A - \lambda I) = n - \operatorname{rank}(A - \lambda I) = n - \operatorname{rank}(A^* - \bar{\lambda}I) = \dim \operatorname{Ker}(A^* - \bar{\lambda}I)$. □

15.15 Lemma

Let $A \in \mathbb{C}^{n\times n}$. Then we have:

(a) If λ is an eigenvalue with a right eigenvector x, and $\mu \neq \lambda$ is another eigenvalue with a left eigenvector y, then $y^*x = 0$, i.e. $x \perp y$.

(b) If λ is a simple eigenvalue (this means its algebraic multiplicity is one) with right and left eigenvectors x and y, respectively, then $y^*x \neq 0$.

Proof. To prove (a), observe that $\langle Ax, y\rangle = \lambda\langle x, y\rangle$ and, by a remark after Section 15.12, $\langle x, A^*y\rangle = \langle x, \bar{\mu}y\rangle = \mu\langle x, y\rangle$. Hence, $\lambda\langle x, y\rangle = \mu\langle x, y\rangle$, which proves (a), since $\lambda \neq \mu$.

To prove (b), assume that $\|x\| = 1$. By the Schur decomposition theorem, there is a unitary matrix R with first column x such that

$$R^*AR = \begin{bmatrix} \lambda & h^* \\ 0 & B \end{bmatrix}$$

with some $h \in \mathbb{C}^{n-1}$ and $B \in \mathbb{C}^{(n-1)\times(n-1)}$. Note also that $Re_1 = x$. Since λ is a simple eigenvalue of A, it is not an eigenvalue of B. Thus the matrix $\lambda I - B$ is invertible, hence so is $\bar{\lambda}I - B^*$. Let $z = (\bar{\lambda}I - B^*)^{-1}h$. Then

$$\bar{\lambda}z - B^*z = h \quad \Longrightarrow \quad h^* + z^*B = \lambda z^*$$

Now one can readily verify that

$$[\,1\,z^*\,]\,R^*AR = \lambda\,[\,1\,z^*\,] \quad \Longrightarrow \quad [\,1\,z^*\,]\,R^*A = \lambda\,[\,1\,z^*\,]\,R^*$$

Denote $w^* = [\,1\,z^*\,]\,R^*$. The above equation now takes a short form

$$w^*A = \lambda w^*$$

Hence w is a left eigenvector of A. By the simplicity of λ, the vector w is a nonzero multiple of y. However, observe that

$$w^*x = [\,1\,z^*\,]\,R^*Re_1 = 1 \ (\neq 0)$$

which proves the lemma. □

15.16 Theorem

Let $A \in \mathbb{C}^{n\times n}$ have a simple eigenvalue λ with right and left unit eigenvectors x and y, respectively. Let $E \in \mathbb{C}^{n\times n}$ such that $\|E\|_2 = 1$. For small ε, denote by $\lambda(\varepsilon)$ and $x(\varepsilon)$, $y(\varepsilon)$ the eigenvalue and right and left unit eigenvectors of the matrix $A + \varepsilon E$ obtained from λ and x, y. Then

$$|\lambda'(0)| \leq \frac{1}{|y^*x|}$$

Proof. It follows from the inverse function theorem that $\lambda(\varepsilon)$ and $x(\varepsilon)$ are differentiable for sufficiently small ε. Write the equation

$$(A + \varepsilon E)\,x(\varepsilon) = \lambda(\varepsilon)\,x(\varepsilon)$$

and differentiate it in ε, set $\varepsilon = 0$, and get

$$Ax'(0) + Ex = \lambda'(0)x + \lambda\,x'(0)$$

Then multiply this equation through on the left by the vector y^*, use the fact that $y^*A = \lambda y^*$ and get

$$y^*Ex = \lambda'(0)\,y^*x$$

Now the result follows since

$$|y^*Ex| = |\langle Ex, y\rangle| \leq \|Ex\|_2\,\|y\|_2 \leq \|E\|_2\,\|x\|_2\,\|y\|_2 = 1$$

This proves the theorem. Note that $y^*x \neq 0$ by Lemma 15.15. $\square$

Note: The matrix $A + \varepsilon E$ is the perturbation of A "in the direction" of E. If the perturbation matrix E is known, one has exactly

$$\lambda'(0) = \frac{y^*Ex}{y^*x}$$

and so

$$\lambda(\varepsilon) = \lambda + \frac{y^*Ex}{y^*x}\varepsilon + O(\varepsilon^2)$$

by Taylor expansion, a fairly precise estimate on $\lambda(\varepsilon)$. In practice, however, the matrix E is absolutely unknown, so one has to use the bound of Theorem 15.16 to estimate the sensitivity of λ to small perturbations of A.

15.17 Condition Number of An Eigenvalue

Let λ be a simple eigenvalue (this means that its algebraic multiplicity is one) of a matrix $A \in \mathbb{C}^{n\times n}$ and x, y the corresponding right and left unit eigenvectors. The *condition number* of λ is

$$K(\lambda) = \frac{1}{|y^*x|}$$

Note that $|y^*x|$ does not depend on the particular choice of x and y.

The condition number $K(\lambda)$ describes the sensitivity of a (simple) eigenvalue to small perturbations of the matrix. Large $K(\lambda)$ signifies an *ill-conditioned* eigenvalue.

15.18 Simple properties of $K(\lambda)$

(a) First, $K(\lambda) \geq 1$, because $|y^*x| \leq \|x\|_2\|y\|_2 = 1$.

(b) If a matrix A is normal, then $K(\lambda) = 1$ for all its simple eigenvalues.

(c) Conversely, if a matrix A has all simple eigenvalues with $K(\lambda) = 1$ for each of them, then it is normal.

Proof. See Exercises 15.3 and 15.4.

Normal matrices are characterized by the fact that the Schur decomposition $Q^*AQ = T$ results in a diagonal matrix T. One can expect that if the matrix T is nearly diagonal (i.e., its off-diagonal elements are small), then the eigenvalues of A are well-conditioned. On the contrary, if some off-diagonal elements of T are large, then at least some eigenvalues are ill-conditioned.

15.19 Remark

It remains to discuss the case of multiple eigenvalues (of algebraic multiplicity ≥ 2). If λ is a multiple eigenvalue, the left and right eigenvectors may be orthogonal even if the geometric multiplicity of λ equals one. Example: $A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, the right and left eigenvectors are $x = e_1$ and $y = e_2$, respectively. Moreover, if the geometric multiplicity is ≥ 2, then for any right eigenvector x there is a left eigenvector y such that $y^*x = 0$ (Exercise 15.5). Hence, the definition 15.17 gives an infinite value of $K(\lambda)$.

This does not necessarily mean that a multiple eigenvalue is always ill-conditioned. It does mean, however, that an ill-conditioned simple eigenvalue is 'nearly multiple'. Precisely, if λ is a simple eigenvalue of A with $K(\lambda) > 1$, then there is a matrix E such that

$$\frac{\|E\|_2}{\|A\|_2} \leq \frac{1}{\sqrt{K(\lambda)^2 - 1}}$$

and λ is a multiple eigenvalue of $A + E$. We leave out the proof. We will not further discuss the sensitivity of multiple eigenvalues.

15.20 Theorem (1st Gershgorin)

Let $A \in \mathbb{C}^{n \times n}$ be 'almost diagonal'. Precisely, let $A = D + E$, where $D = \operatorname{diag}\{d_1, \ldots, d_n\}$ and $E = (e_{ij})$ is small. Then every eigenvalue of A lies in at least one of the circular disks

$$D_i = \left\{ z \in \mathbb{C} \colon\ |z - d_i| \leq \sum_{j=1}^{n} |e_{ij}| \right\}$$

Note: D_i are called *Gershgorin disks*.

Proof. Let λ be an eigenvalue of A with eigenvector x. Let

$$|x_r| = \max_i \{|x_1|, \ldots, |x_n|\}$$

be the maximal (in absolute value) component of x. We can normalize x so that $x_r = 1$ and $|x_i| \leq 1$ for all i. On equating the r-th components in $Ax = \lambda x$ we obtain

$$(Ax)_r = d_r x_r + \sum_{j=1}^{n} e_{rj} x_j = d_r + \sum_{j=1}^{n} e_{rj} x_j = \lambda x_r = \lambda$$

Hence

$$|\lambda - d_r| \le \sum_{j=1}^{n} |e_{rj}|\,|x_j| \le \sum_{j=1}^{n} |e_{rj}|$$

The theorem is proved. □

15.21 Theorem (2nd Gershgorin)

Suppose k of the Gershgorin disks D_i make a *cluster* in the following sense: the union of their interiors (open disks) is a connected domain in $\mathbb{C}$ that is disjoint from the other $n-k$ Gershgorin disks. Then there are precisely k eigenvalues of A (counting multiplicity) in that cluster.

Proof. For brevity, denote

$$h_i = \sum_{j=1}^{n} |e_{ij}|$$

Consider a family of matrices $A(s) = D + sE$ for $0 \le s \le 1$. It is a standard fact in complex analysis that the roots of a complex polynomial change continuously with its coefficients. Hence the eigenvalues of $A(s)$ depend continuously on s. The Gershgorin disks $D_i(s)$ for the matrix $A(s)$ are centered at d_i and have radii sh_i. As s increases, each disk $D_i(s)$ grows concentrically, until it reaches the size of the Gershgorin disk D_i of Theorem 15.20 at $s = 1$. When $s = 0$, each Gershgorin disk $D_i(0)$ is just a point, d_i, which is an eigenvalue of the matrix $A(0) = D$. So, if $d_{i_1} = \cdots = d_{i_m}$ is an eigenvalue of multiplicity $m \ge 1$, then m degenerate disks $D_{i_1}(0), \dots, D_{i_m}(0)$ will coincide. For small $s > 0$, the corresponding m disks $D_{i_1}(s), \dots, D_{i_m}(s)$ will have a common center and make a cluster containing m eigenvalues of $A(s)$. As the disks grow with s, the eigenvalues cannot jump from one cluster to another (by continuity), unless the two clusters overlap and then make one cluster. Once two clusters overlap (merge) for some $s > 0$, all their disks will belong in one cluster for all larger values of s, including $s = 1$. This proves the theorem. □

Note: If the Gershgorin disks $D_1, \dots, D_n$ are all disjoint, then each contains exactly one eigenvalue of A.

Exercise 15.1. (JPE May, 1994). Let $X^{-1}AX = D$, where D is a diagonal matrix.

(i) Show that the columns of X are right eigenvectors and the conjugate rows of X^{-1} are left eigenvectors of A.

(ii) Let $\lambda_1 \ldots, \lambda_n$ be the eigenvalues of A. Show that there are right eigenvectors $x_1, \ldots, x_n$ and left eigenvectors $y_1, \ldots, y_n$ such that

$$A = \sum_{i=1}^{n} \lambda_i x_i y_i^*$$

Exercise 15.2. Let $A \in \mathbb{C}^{n\times n}$ be Hermitean with eigenvalues $\lambda_1 \leq \cdots \leq \lambda_n$. Let $\mu_1 \leq \cdots \leq \mu_{n-1}$ be all the eigenvalues of the $(n-1)$-st principal minor A_{n-1} of A. Use the Minimax theorem to prove the *interlacing property*

$$\lambda_1 \leq \mu_1 \leq \lambda_2 \leq \cdots \leq \lambda_{n-1} \leq \mu_{n-1} \leq \lambda_n$$

Exercise 15.3. Let $A \in \mathbb{C}^{n\times n}$. Show that

(i) λ is an eigenvalue of A iff $\bar{\lambda}$ is an eigenvalue of A^*.

(ii) if A is normal, then for each eigenvalue the left and right eigenspaces coincide;

(iii) if A is normal, then for any simple eigenvalue λ of A we have $K(\lambda) = 1$.

Exercise 15.4. Let $A \in \mathbb{C}^{n\times n}$ and $B = Q^*AQ$, where Q is a unitary matrix. Show that if the left and right eigenspaces of A are equal, then B enjoys the same property. After that show that A is normal. Finally, prove that if A has all simple eigenvalues with $K(\lambda) = 1$, then A is normal.

Exercise 15.5. Suppose λ is an eigenvalue of geometric multiplicity ≥ 2 for a matrix A. Show that for each right eigenvector x there is a left eigenvector y such that $y^*x = 0$.

Exercise 15.6. Use the Gershgorin theorems to show that a symmetric, strictly row diagonally dominant real matrix with positive diagonal elements is positive definite.

16 Computation of eigenvalues: power method

To simplify the matter, we always assume that the matrix A is diagonalizable, i.e., it has a complete set of eigenvectors $x_1, \dots, x_n$ with eigenvalues $\lambda_1, \dots, \lambda_n$. The latter are assumed to be ordered in absolute value:

$$|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$$

16.1 Dominant eigenvalue and eigenvector

Assume that $|\lambda_1| > |\lambda_2|$, i.e. the largest eigenvalue is simple. We call λ_1 the *dominant eigenvalue* and x_1 a *dominant eigenvector*.

16.2 Power method: the idea

Let λ_1 be the dominant eigenvalue of A and

$$q = c_1 x_1 + \cdots + c_n x_n$$

an arbitrary vector such that $c_1 \neq 0$. Then

$$\begin{aligned} A^k q &= c_1 \lambda_1^k x_1 + \cdots + c_n \lambda_n^k x_n \\ &= \lambda_1^k [c_1 x_1 + c_2 (\lambda_2/\lambda_1)^k x_2 + \cdots + c_n (\lambda_n/\lambda_1)^k x_n] \end{aligned}$$

Denote

$$q^{(k)} = A^k q / \lambda_1^k = c_1 x_1 + \underbrace{c_2 (\lambda_2/\lambda_1)^k x_2 + \cdots + c_n (\lambda_n/\lambda_1)^k x_n}_{\Delta_k}$$

16.3 Lemma

The vector $q^{(k)}$ converges to $c_1 x_1$. Moreover,

$$\|\Delta_k\| = \|q^{(k)} - c_1 x_1\| \leq \text{ const} \cdot r^k$$

where $r = |\lambda_2/\lambda_1| < 1$.

Therefore, the vectors $A^k q$ (obtained by the powers of A) will align in the direction of the dominant eigenvector x_1 as $k \to \infty$. The number r characterizes the speed of alignment, i.e. the speed of convergence $\|\Delta_k\| \to 0$ (the smaller r the faster convergence). Note that if $c_2 \neq 0$, then

$$\|q^{(k+1)} - c_1 x_1\| / \|q^{(k)} - c_1 x_1\| \to r$$

The number r is called the *convergence ratio* or the *contraction number*.

16.4 Linear, quadratic and cubic convergence

We say that the convergence $x_k \to x$ is *linear* if

$$|x_{k+1} - x| \le r|x_k - x|$$

for some $r < 1$ and all sufficiently large k. If

$$|x_{k+1} - x| \le C|x_k - x|^a$$

with some $C > 0$ and $a > 1$, then the convergence is said to be *superlinear* (it is faster than linear). For $a = 2$ the convergence is *quadratic*, and for $a = 3$ *cubic*.

In order to come close to within ε of the limit, the linear convergence takes $k \sim \mathcal{O}(|\log \varepsilon|)$ iterations, while the superlinear convergence takes $k \sim \mathcal{O}\big(|\log(|\log \varepsilon|)|\big)$ iterations.

16.5 Remarks on convergence

If the convergence is linear, then by induction we have $|a_k - a| \le Cr^k$, where $C = |a_0 - a|$. The sequence Cr^k decreases to zero exponentially fast, which is very fast by calculus standards. However, in numerical calculations standards are different.

The linear convergence practically means that each iteration adds a fixed number of accurate digits to the result. For example, if $r = 0.1$, then each iteration adds one correct decimal digit. This is not superb, since it will take 6–7 iterations to reach the maximum accuracy in single precision arithmetic and 15–16 iterations in double precision. If $r = 0.5$, then each iteration adds one binary digit (a bit), hence one needs 22–23 iterations in single precision and 52–53 in double precision. Now imagine how long it might take when $r = 0.9$ or $r = 0.99$.

On the other hand, the quadratic convergence means that each iteration doubles (!) the number of digits of accuracy. Starting with just one accurate binary digit, one needs 4–5 iterations in single precision arithmetic and only one more iteration in double precision. The cubic convergence means that each iteration triples (!!!) the number of digits of accuracy. See Example 16.14 for an illustration.

16.6 Scaling problem in the power method

In practice, the vector $q^{(k)} = A^k q/\lambda_1^k$ is inaccessible because we do not know λ_1 in advance. But we cannot just drop the factor λ_1^k, because then

$\|A^k q\| \to \infty$ if $|\lambda_1| > 1$ and $\|A^k q\| \to 0$ if $|\lambda_1| < 1$, possibly causing overflow or underflow in numerical computations. Thus we must somehow normalize, or scale, the vector $A^k q$.

16.7 Power method: two choices for the scaling factor

Pick an initial vector q_0. For $k \geq 1$, define

$$q_k = Aq_{k-1}/\sigma_k$$

where σ_k is a properly chosen scaling factor. One common choice is

$$\sigma_k = \|Aq_{k-1}\|, \qquad \text{then} \qquad \|q_k\| = 1$$

In this case one can approximate the eigenvalue λ_1 by the Rayleigh quotient

$$\lambda_1^{(k)} = q_k^* A q_k.$$

Another popular choice for σ_k is the largest (in absolute value) component of the vector Aq_{k-1}. This ensures that the largest (in absolute value) component of q_k equals one, in particular $\|q_k\|_\infty = 1$. Assume that the vector x has one component with the largest absolute value. In that case σ_k itself is a good approximation for λ_1, and we set

$$\lambda_1^{(k)} = \sigma_k.$$

To estimate how close the unit vector q_k is to the one-dimensional eigenspace $\text{span}\{x_1\}$, denote by p_k the orthogonal projection of q_k on $\text{span}\{x_1\}$ and by $d_k = q_k - p_k$ the orthogonal component. Then $\|d_k\|$ measures the distance from q_k to $\text{span}\{x_1\}$.

16.8 Theorem (convergence of the power method)

Assume that λ_1 is the dominant eigenvalue, and $q_0 = \sum c_i x_i$ is chosen so that $c_1 \neq 0$. Then the distance from q_k to the eigenspace $\text{span}\{x_1\}$ converges to zero and $\lambda_1^{(k)}$ converges to λ_1. Furthermore,

$$\|d_k\| \leq \text{const} \cdot r^k \qquad |\lambda_1^{(k)} - \lambda_1| \leq \text{const} \cdot r^k$$

Note: The sequence of vectors q_k need not have a limit, see examples.

Proof. It is a direct calculation, based on the representation $A^k q_0 = \lambda_1^k (c_1 x_1 + \Delta_k)$ of Section 16.2 and Lemma 16.3.

16.9 Examples

(a) Let $A = \begin{bmatrix} 3 & 2 \\ 1 & 1 \end{bmatrix}$. Pick $q_0 = (1,1)^T$ and use the second choice of σ_k in 16.7. Then $\sigma_1 = 5$ and $q_1 = (1, 0.4)^T$, $\sigma_2 = 3.8$ and $q_2 = (1, 0.368)^T$, $\sigma_3 = 3.736$ etc. Here σ_k converges to the dominant eigenvalue $\lambda_1 = 2 + \sqrt{3} = 3.732$ and q_k converges to a dominant eigenvector $(1, \sqrt{3}/2 - 1/2)^T = (1, 0.366)^T$.

(b) Let $A = \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix}$. Pick $q_0 = (1,1)^T$ and use the first choice of σ_k in 16.7. Then $q_k = ((-1)^k, 0)$ does not have a limit, it keeps flipping. With the second choice of σ_k in 16.7, we have $q_k = (1,0)$ and $\sigma_k = -1 = \lambda_1$ for all $k \geq 1$.

16.10 Initial choice

The choice of the initial vector q_0 only has to fulfill the requirement $c_1 \neq 0$. Since the vectors with $c_1 = 0$ form a hyperplane in $\mathbb{C}^n$, one hopes that a vector q_0 picked "at random" will not lie in that hyperplane. Furthermore, even if $c_1 = 0$, round-off errors will most likely pull the numerical vectors q_k away from that hyperplane. If that does not seem to be enough, one can carry out the power method for n different initial vectors that make a basis, say $e_1, \ldots, e_n$. One of these vectors surely lies away from that hyperplane.

16.11 Inverse power method

Assume that A is invertible. Then $\lambda_1^{-1}, \ldots, \lambda_n^{-1}$ are the eigenvalues of A^{-1}, with the same eigenvectors $x_1, \ldots, x_n$. Note that $|\lambda_1^{-1}| \leq \cdots \leq |\lambda_n^{-1}|$.

Assume that $|\lambda_n^{-1}| > |\lambda_{n-1}^{-1}|$. Then λ_n^{-1} is the dominant eigenvalue of A^{-1} and x_n a dominant eigenvector. One can apply the power method to A^{-1} and find λ_n^{-1} and x_n. The rate of convergence of iterations will be characterized by the ratio $r = |\lambda_n/\lambda_{n-1}| < 1$. This is called the *inverse power method*.

Note: in practice, there is no need to compute the inverse matrix A^{-1} explicitly. To find $A^{-1}b$ for any given vector b, one can just solve the system $Ax = b$, e.g., by the LU decomposition of the matrix A.

Now we know how to compute the largest and the smallest eigenvalues. The following trick allows us to compute any simple eigenvalue.

16.12 Power method with shift

Recall that if λ is an eigenvalue of A with eigenvector x, then $\lambda - \rho$ is an eigenvalue of $A - \rho I$ with the same eigenvector x.

Assume that ρ is a good approximation to a simple eigenvalue λ_i of A, so that $|\lambda_i - \rho| < |\lambda_j - \rho|$ for all $j \neq i$. Then the matrix $A - \rho I$ will have the smallest eigenvalue $\lambda_i - \rho$ with the eigenvector x_i.

The inverse power method can now be applied to $A - \rho I$ to find $\lambda_i - \rho$ and x_i. The convergence of iterations will be linear with ratio

$$r = \frac{|\lambda_i - \rho|}{\min_{j \neq i} |\lambda_j - \rho|} < 1$$

Hence, the better ρ approximates λ_i, the faster convergence is guaranteed.

By subtracting ρ from all the eigenvalues of A we shift the entire spectrum of A by ρ. The number ρ is called the *shift*. The above algorithm for computing λ_i and x_i is called the *(inverse) power method with shift*.

The power method with shift allows us to compute all the simple eigenvalues and eigenvectors of a matrix, but the convergence is slow (just linear).

16.13 Power method with Rayleigh quotient shift

This is an improvement of the algorithm 16.12. Since at each iteration of the inverse power method we obtain a better approximation to the eigenvalue λ_i, we can use it as the shift ρ for the next iteration. So, the shift ρ will be updated at each iteration. This will ensure a faster convergence.

One chooses an initial vector q_0 and an initial approximation ρ_0, and for $k \geq 1$ computes

$$q_k = \frac{(A - \rho_{k-1} I)^{-1} q_{k-1}}{\sigma_k}$$

and

$$\rho_k = \frac{q_k^* (A - \rho_{k-1} I)^{-1} q_k}{q_k^* q_k}$$

where σ_k a convenient scaling factor, for example, $\sigma_k = \|(A - \rho_{k-1} I)^{-1} q_{k-1}\|$.

The convergence of the Rayleigh quotient iterations is, generally, quadratic (better than linear). If the matrix A is Hermitian, the convergence is even faster – it is cubic!!!

16.14 Example

Consider the symmetric matrix

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

and let $q_0 = (1, 1, 1)^T$ be the initial vector and $q_0 = 5$ the initial shift. When Rayleigh quotient iteration is applied to A, the following values ρ_k are computed by the first two iterations:

$$\rho_1 = 5.2131, \qquad \rho_2 = 5.214319743184$$

The actual value is $\lambda = 5.214319743377$. After only two iterations, Rayleigh quotient method produced 10 accurate digits. The next iteration would bring about 30 accurate digits – more than enough in double precision.

16.15 Power method: pros and cons

(a) The power method is classic. It is very simple and generally good.

(b) An obvious concern is the numerical stability of the method. The matrices used in the inverse power method with shift tend to be exceedingly ill-conditioned. As a result, the numerical solution of the system $(A-\rho I)x = b$, call it x_c, will deviate significantly from its exact solution x. However, for some peculiar reason (we do not elaborate) the difference $x_c - x$ tends to align with the vector x. Therefore, the normalized vectors $x_c/\|x_c\|$ and $x/\|x\|$ are close to each other. Hence, ill conditioning of the matrices does not cause trouble.

(c) On the other hand, the power method is slow. Each iteration requires solving a linear system of equations $(A - \rho_{k-1}I)x = b$, every time with a new matrix, so the LU decomposition must be repeated, which takes $2n^3/3$ fops. If we want to compute all n eigenpairs, and make p iterations per eigenpair, then the total cost is $2pn^4/3$ flops.

(d) Lastly, when the matrix A is real, then in order to compute its complex eigenvalues one has to deal with complex matrices $(A - \rho_{k-1}I)$, which is inconvenient and expensive. It would be nice to stick to real matrices for as long as possible and obtain pairs of complex conjugate eigenvalues only at the final step. The QR algorithm, to be discussed in the next chapter, provides such a luxury.

Exercise 16.1. (JPE, May 2003) Let A be a symmetric matrix with eigenvalues such that $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_{n-1}| > |\lambda_n|$. Suppose $z \in \mathbb{R}^n$ with $z^T x_1 \neq 0$, where $Ax_1 = \lambda_1 x_1$. Prove that, for some constant C,

$$\lim_{k\to\infty} \frac{A^k z}{\lambda_1^k} = Cx_1$$

and use this result to devise a reliable algorithm for computing λ_1 and x_1. Explain how the calculation should be modified to obtain (a) λ_n and (b) the eigenvalue closest to 2.

Exercise 16.2. (JPE, September 1996) The matrix

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

has eigenpairs

$$(\lambda, x) = \left(2, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}\right), \left(-1, \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}\right), \left(3, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}\right),$$

Suppose the power method is applied with starting vector

$$z_0 = [1, 1, -1]^t/\sqrt{3}$$

(a) Determine whether or not the iteration will converge to an eigenpair of A, and if so, which one. Assume exact arithmetic.

(b) Repeat (a), except now use the inverse iteration with the same starting vector z_0 and the Rayleigh quotient of z_0 as approximation for the eigenvalue.

(c) Now answer both (a) and (b) again, except this time use standard fixed precision floating point arithmetic, i.e. computer arithmetic.

17 Computation of eigenvalues: QR algorithm

The QR algorithm (not to be confused with QR decomposition!) dates back to the early 1960s, and in the recent decades it became the most widely used method for calculating the complete set of eigenvalues and eigenvectors.

17.1 Pure QR algorithm

Let $A \in \mathbb{C}^{n\times n}$. The algorithm starts with $A_0 = A$ and generates a sequence of matrices A_k defined as follows:

$$A_{k-1} = Q_k R_k, \qquad A_k = R_k Q_k.$$

That is, a QR decomposition of A_{k-1} is computed and then its factors are recombined in reverse order to produce A_k. One iteration of the QR algorithm is called *QR step*.

17.2 Lemma

(a) All matrices A_k in the QR algorithm are unitary equivalent, in particular, they have the same eigenvalues.

(b) If A is a Hermitian matrix, then all A_k's are Hermitian matrices as well.

Proof. To prove (a), we note that $A_k = Q_k^* A_{k-1} Q_k$. Now (b) follows by induction from $A_k^* = Q_k^* A_{k-1}^* Q_k = Q_k^* A_{k-1} Q_k = A_k$. □

17.3 Theorem (convergence of the QR algorithm)

Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of A satisfying

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0$$

Under one technical condition, see below, the matrix $A_k = (a_{ij}^{(k)})$ is guaranteed to converge to an upper triangular form, so that

(a) $a_{ij}^{(k)} \to 0$ as $k \to \infty$ for all $i > j$.

(b) $a_{ii}^{(k)} \to \lambda_i$ as $k \to \infty$ for all i.

The convergence is linear, with the ratio

$$r = \max_k |\lambda_{k+1}/\lambda_k| < 1.$$

This theorem is given without proof. The technical condition mentioned above is that the matrix Y whose i-th row is a left eigenvector of A corresponding to λ_i for all i, must have an LU decomposition (i.e. all its principal minors must be nonsingular).

17.4 Remarks

(a) If A is Hermitian, then A_k will converge to a diagonal matrix.

(b) All the matrices A_k (and R_k) involved in the QR algorithm have the same 2-condition number (by Section 12.8), thus the QR algorithm is numerically superior to the power method. This is also true for all the variations of the QR method discussed below.

(c) On the other hand, the pure QR algorithm is quite expensive. Especially, each iteration is very costly: the QR decomposition takes $2n^3$ flops and the multiplication of two matrices R_k and Q_k (even if we take advantage of the triangular structure of R_k!) takes n^3 flops, a total of $3n^3$ flops. Thus if we make p iterations then the total cost is $3pn^3$. This seems to be an improvement over $2pn^4/3$ flops of the power method, but we also need to remember that the convergence is slow (linear), thus it may require many more iterations, according to Section 16.4. Fortunately, the computational cost can be reduced with the help of Hessenberg matrices, see below.

(d) The pure QR algorithm fails on matrices with multiple eigenvalues (see Example 17.16 below) and on real matrices with complex eigenvalues. Indeed, complex eigenvalues of a real matrix come in conjugate pairs $a \pm ib$, which have equal absolute values $|a+bi| = |a-bi|$, violating the main assumption of the theorem. Furthermore, if A is real, then all Q_k, R_k and A_k are real matrices as well, and thus we cannot even expect the real diagonal elements of A_k to converge to the complex eigenvalues of A. In this case A_k may not converge to anything.

17.5 Hessenberg matrix

$A \in \mathbb{C}^{n\times n}$ is called an *(upper) Hessenberg matrix* if $a_{ij} = 0$ for all $i > j+1$, i.e. A has the form

$$A = \begin{bmatrix} \times & \times & \cdots & \cdots & \times \\ \times & \times & \ddots & \ddots & \vdots \\ 0 & \times & \times & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \times & \times \end{bmatrix}$$

17.6 Lemma

If an invertible matrix A_0 is Hessenberg, then all the matrices A_k generated by the QR algorithm will be Hessenberg as well.

Proof. By induction, let A_{k-1} be Hessenberg. Then $A_{k-1} = Q_k R_k$ and so $Q_k = A_{k-1} R_k^{-1}$. Since this is a product of a Hessenberg matrix and an upper triangular matrix, it is verified by direct inspection that Q_k is Hessenberg. Then, similarly, $A_k = R_k Q_k$ is a product of an upper triangular and Hessenberg matrices, so it is Hessenberg. □

For noninvertible Hessenberg matrices, one can show that Q_k constructed by Gram-Schmidt orthogonalization, see Chapter 9, is also Hessenberg, thus A_k will again be a Hessenberg matrix.

17.7 Cost of a QR step for Hessenberg matrices

For Hessenberg matrices, the QR algorithm can be implemented with a substantial reduction of computational cost. First, the QR decomposition via Givens rotators takes $6n^2$ flops, according to Section 14.12. Second, Q_k is a product of $n-1$ rotators, hence the multiplication of R_k by Q_k takes $\leq 6n^2$ flops. The total for the QR step is then $\leq 12n^2$ flops, a dramatic drop from $3n^3$ flops as required for a non-Hessenberg matrix A.

17.8 Theorem

Every matrix $A \in \mathbb{C}^{n\times n}$ is unitary equivalent to a Hessenberg matrix, i.e.

$$A = Q^* H Q$$

where H is a Hessenberg matrix and Q is a unitary matrix. There is an exact finite algorithm for computing H and Q.

Note: the existence of H immediately follows from Schur decomposition $A = Q^*TQ$, so we only need to show that H and Q here can be constructed by a finite algorithm. Remember that there is no finite algorithm for Schur decomposition, since it involves the eigenvalues of the matrix A (Section 15.1).

Proof. An explicit algorithm for computing Q and H is known as *Arnoldi algorithm*. The matrix equation $A = Q^*HQ$ can be rewritten as $AQ^* = Q^*H$. Denote by q_i, $1 \le i \le n$, the columns of the unitary matrix Q^* and by h_{ij} the entries of H. Equating the columns of the matrices AQ^* and Q^*H (and remembering that $h_{ij} = 0$ for $i > j + 1$) we obtain a system of equations

$$\begin{aligned}
Aq_1 &= h_{11}q_1 + h_{2,1}q_2 \\
Aq_2 &= h_{12}q_1 + h_{22}q_2 + h_{3,2}q_3 \\
&\cdots \\
Aq_i &= h_{1i}q_1 + \cdots + h_{ii}q_i + h_{i+1,i}q_{i+1} \\
&\cdots \\
Aq_n &= h_{1n}q_1 + \cdots + h_{n-1,n}q_{n-1} + h_{n,n}q_n
\end{aligned}$$

(Note that the last equation is slightly different from the others since it terminates on a diagonal entry of H.)

Now the Arnoldi algorithm goes along the lines similar to the classical Gram-Schmidt orthogonalization. We pick an arbitrary unit vector q_1, compute $v_1 = Aq_1$, and represent

$$v_1 = \operatorname{Pr}_{q_1} v_1 + w_2 = \langle v_1, q_1 \rangle \, q_1 + w_2$$

where w_2 is orthogonal to q_1. Then we set $h_{11} = \langle v_1, q_1 \rangle$ and $h_{21} = \|w_2\|$ and define $q_2 = w_2/\|w_2\|$. This enforces the first equation in the above system.

Generally, for every $i = 1, \ldots, n-1$ we make four steps:

Step 1: compute $v_i = Aq_i$.
Step 2: for all $j = 1, \ldots, i$ compute $h_{ji} = \langle v_i, q_j \rangle$.
Step 3: $w_i = v_i - \sum_{j=1}^{i} h_{ji}q_j$ (note: this vector is orthogonal to $q_1, \ldots, q_i$).
Step 4: $h_{i+1,i} = \|w_i\|$ and $q_{i+1} = w_i/h_{i+1,i}$, unless $h_{i+1,i} = 0$, see below.

If $h_{i+1,i} = 0$ in Step 4, we pick an arbitrary unit vector q_{i+1} orthogonal to $q_1, \ldots, q_i$. Finally, for $i = n$ we execute steps 1 and 2 only. $\square$

Theorem 17.8 shows that one can first transform A to a Hessenberg matrix $A_0 = H$, which has the same eigenvalues as A does (by similarity) and then start the QR algorithm with A_0.

17.9 Remarks on terminology

The exceptional case $h_{i+1,i} = 0$ in Step 4 is referred to as the *breakdown* of the Arnoldi algorithm. This term is quite misleading, since the method does not really break down. In fact, the resulting Hessenberg matrix H will have a simpler structure (an extra zero on its subdiagonal), and then the matrix H has form

$$H = \begin{bmatrix} H_1 & C \\ 0 & H_2 \end{bmatrix}$$

where $H_1 \in \mathbb{C}^{i\times i}$ and $H_2 \in \mathbb{C}^{(n-i)\times(n-i)}$. Clearly, the set of eigenvalues of H is the union of the eigenvalues of H_1 and H_2, and those can be found by the QR algorithm applied to the two smaller matrices H_1 and H_2 separately. This leads to a reduction of the problem.

The Arnoldi algorithm is often applied to very large (or even infinite) matrices, where the complete construction of H and Q is out of the question. Then one can run the Arnoldi algorithm partway to obtain approximations to H and Q. Such a method is referred to as *Arnoldi iterations*.

17.10 Cost of Arnoldi algorithm

The cost of Step 1 is $2n^2$ flops, the cost of Step 2 is $2ni$ flops, the same for Step 3, and lastly Step 4 takes $3n$ flops. The total cost is then

$$\sum_{i=1}^{n}(2n^2 + 4ni + 3n) \sim 4n^3$$

(actually, by choosing $q_1 = e_1$ one can save some work and compute H and Q in $\frac{10}{3}n^3$ flops; see the textbook for more details). This cost is comparable to the cost of one QR step for a generic (non-Hessenberg) matrix. The Arnoldi algorithm can be regarded as a *pre-processing* of the matrix A; it only needs to be done once, and then the QR algorithm is applied to the resulting Hessenberg matrix.

17.11 The case of Hermitian matrices

If A is a Hermitian matrix, then H will be both Hermitian and Hessenberg. Hence H will be a *tridiagonal matrix* ($h_{ij} = 0$ for all $|i - j| > 1$). Its construction by the Arnoldi algorithm takes only $\frac{4}{3}n^3$ flops (see the textbook).

17.12 Theorem

Assume that A_0, and hence A_k for all $k \geq 1$, are Hessenberg matrices. Then the convergence $a_{i,i-1}^{(k)} \to 0$ as $k \to \infty$ in Theorem 17.3 is linear with ratio $r_i = |\lambda_i/\lambda_{i-1}|$. In addition, the convergence $a_{nn}^{(k)} \to \lambda_n$ is linear with ratio $r_n = |\lambda_n/\lambda_{n-1}|$.

This theorem is given without proof.

Note that $|\lambda_{i+1}/\lambda_i| < 1$ for all i. The smaller this ratio, the faster the convergence. It is also important to note that each subdiagonal entry $a_{i,i-1}^{(k)}$ has its own rate of convergence. This allows us to accelerate the convergence of some selected entries, see below.

17.13 QR algorithm with shift - 1

One can modify the matrix A to decrease the ratio $|\lambda_n/\lambda_{n-1}|$ and thus make the convergence

$$a_{n,n-1}^{(k)} \to 0 \qquad \text{and} \qquad a_{nn}^{(k)} \to \lambda_n \tag{B}$$

of the two bottom entries faster with the help of shifting, as in Section 16.12. To achieve this, one applies the QR steps to the matrix $A - \rho I$ where ρ is a properly chosen approximation to λ_n. Then the convergence (B) will be linear with ratio $r = |\lambda_n - \rho|/|\lambda_{n-1} - \rho|$. The better ρ approximates λ_n the faster the convergence.

17.14 QR algorithm with shift - 2

The approximation ρ can be updated at every iteration, as in Section 16.13, by using Rayleigh quotient

$$\rho = \rho_k = u_k^* A_k u_k$$

where u_k is an approximate unit eigenvector of the matrix A_k corresponding to the smallest eigenvalue λ_n. In practice, a simple and convenient choice for u_k is $u_k = e_n$, which gives $\rho_k = a_{nn}^{(k)}$. Then the *QR algorithm with shift* goes as follows:

$$\begin{aligned} A_{k-1} - \rho_{k-1} I &= Q_k R_k && \text{(QR decomposition of } A_{k-1} - \rho_{k-1} I\text{)} \\ R_k Q_k + \rho_{k-1} I &= A_k && \text{(computation of the next matrix } A_k\text{)} \\ \rho_k &= a_{nn}^{(k)} && \text{(setting } \rho_k \text{ to the trailing element of } A_k\text{)} \end{aligned}$$

This is called the *Rayleigh quotient shift*. The convergence (B) is now quadratic for the same reasons as in Section 16.13.

17.15 QR algorithm with shift - 3

However, the other subdiagonal entries, $a^{(k)}_{i+1,i}$, $1 \le i \le n-2$, move to zero slowly (linearly, with variable ratios). To speed them up, one uses the following trick. After making $a^{(k)}_{n,n-1}$ practically zero, one ensures that $a^{(k)}_{nn}$ is practically equal to λ_n. Then one can partition the matrix A_k as

$$A_k = \begin{bmatrix} \hat{A}_k & b_k \\ 0 & \lambda_n \end{bmatrix}$$

where $\hat{A}_k$ is an $(n-1) \times (n-1)$ Hessenberg matrix, whose eigenvalues are (obviously) $\lambda_1, \ldots, \lambda_{n-1}$. Then one can apply further steps of the QR algorithm (with shift) to the matrix $\hat{A}_k$, instead of A_k. This quickly produces its smallest eigenvalue, λ_{n-1}, which can be split off in the same manner. This procedure is called the *deflation* of the matrix A.

In practice, each eigenvalue of A requires just 3-5 iterations (QR steps), on the average. For Hermitian matrices, it is even faster – just 2-3 iterations per eigenvalue. Thus the QR algorithm with shift and deflation achieves a top speed.

It remains to discuss the problem of computing multiple real eigenvalues and complex eigenvalues for real matrices described in Remark 17.4 (d).

17.16 Example

Let $A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. Then the pure QR algorithm gives

$$A_0 = Q_1 R_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

hence

$$A_1 = R_1 Q_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = A$$

so the process goes nowhere. The Rayleigh quotient shift $\rho = a_{22}$ has no effect either, since $a_{22} = 0$. The reason of this failure is that the eigenvalues of A, which are $+1$ and -1, have equal absolute values – this is a symmetry which confuses the QR algorithm, it "cannot decide" which eigenvalue to approach. To break this symmetry, one needs to choose the shift ρ differently.

17.17 Wilkinson iteration

At step k of the QR algorithm with shift, consider the trailing 2×2 matrix at the bottom right of A_k:

$$B_k = \begin{bmatrix} a^{(k)}_{n-1,n-1} & a^{(k)}_{n-1,n} \\ a^{(k)}_{n,n-1} & a^{(k)}_{n,n} \end{bmatrix}$$

Now set ρ_k to the eigenvalue of B_k that is closer to $a^{(k)}_{n,n}$ (in case of a tie, either one can be taken). This is called the *Wilkinson shift*.

The eigenvalues of a 2×2 matrix can be easily (and precisely) computed by the quadratic formula, whether they are real or complex. If they are real, then the Wilkinson shift will help to break the symmetry in the case of multiple eigenvalues of A.

17.18 Example 17.16 continued

The Wilkinson shift here is either $\rho = 1$ or $\rho = -1$. Let us choose $\rho = -1$. Then

$$A_0 - \rho I = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = Q_1 R_1 = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} \sqrt{2} & \sqrt{2} \\ 0 & 0 \end{bmatrix}$$

and then

$$A_1 = R_1 Q_1 + \rho I = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

so the QR algorithm converges in one step.

17.19 Wilkinson iteration for complex eigenvalues

If the eigenvalues of B_k are complex, then the shift ρ_k will be complex, and the matrix $A - \rho_k I$ will be complex, too. The QR step will then produce a complex matrix A_{k+1}, which is inconvenient. In this case one can use the following trick to avoid further working with complex matrices: set $\rho_{k+1} = \bar{\rho}_k$ (which is the other complex eigenvalue of B_k) for the next QR step. Then the resulting matrix A_{k+2} will be real again, see bellow.

Furthermore, the values ρ_k and $\bar{\rho}_k$ will approximate two complex eigenvalues of the matrix A. Therefore, the QR algorithm with Wilkinson shift is able to compute conjugate pairs of complex eigenvalues of a real matrix A, thus resolving the concern raised in Section 16.15 (d).

Actually, there is no need to compute the complex matrix A_{k+1} mentioned above. One can just combine the two QR steps together and construct A_{k+2} directly from A_k (bypassing A_{k+1}). This can be carried out entirely in real arithmetic. The resulting all-real procedure is called the *double-step QR algorithm with Wilkinson shift*.

17.20 Lemma

Let $A_0 \in \mathbb{R}^{n\times n}$ and ρ, $\bar{\rho}$ not eigenvalues of A_0. Consider a pair of QR steps with complex conjugate shifts:

$$\begin{aligned} A_0 - \rho I &= Q_1 R_1 \qquad & R_1 Q_1 + \rho I &= A_1 \\ A_1 - \bar{\rho} I &= Q_2 R_2 \qquad & R_2 Q_2 + \bar{\rho} I &= A_2 \end{aligned}$$

Since the matrices $A_0 - \rho I$ and $A_1 - \bar{\rho} I$ are nonsingular, the above QR decompositions may be constructed so that R_1 and R_2 have positive real entries, cf. Section 9.4. In this case A_2 will be real.

Proof. First, note that $A_1 = Q_1^* A_0 Q_1$ and $A_2 = Q_2^* A_1 Q_2$. Now

$$\begin{aligned} (A_0 - \bar{\rho} I)(A_0 - \rho I) &= (A_0 - \bar{\rho} I) Q_1 R_1 \\ &= Q_1 Q_1^* (A_0 - \bar{\rho} I) Q_1 R_1 \\ &= Q_1 (A_1 - \bar{\rho} I) R_1 \\ &= Q_1 Q_2 R_2 R_1, \end{aligned}$$

This is actually a QR decomposition of the matrix $(A_0 - \bar{\rho} I)(A_0 - \rho I)$, with $Q = Q_1 Q_2$ and $R = R_2 R_1$. Note that the upper triangular matrix $R = R_2 R_1$ has real positive diagonal entries. Since the matrix

$$\begin{aligned} (A_0 - \bar{\rho} I)(A_0 - \rho I) &= A_0^2 - (\rho + \bar{\rho}) A_0 + \rho \bar{\rho} I \\ &= A_0^2 - 2(\operatorname{Re} \rho) A_0 + |\rho|^2 I \end{aligned}$$

is obviously real, the above QR decomposition is unique by Corollary 9.4, thus it must be real, hence the matrices $Q_1 Q_2$ and $R_2 R_1$ are entirely real. Thus

$$A_2 = (Q_1 Q_2)^* A_0 (Q_1 Q_2) = (Q_1 Q_2)^T A_0 (Q_1 Q_2)$$

is a real matrix as well. □

Exercise 17.1. (JPE, September 2009) Let $A \in \mathbb{C}^{n\times n}$ be nonsingular. Let $A = Q_1 R_1$ be a QR decomposition of A, and for $k \geq 1$ define inductively $AQ_k = Q_{k+1} R_{k+1}$, a QR decomposition of AQ_k.

(a) Prove that there exists an upper triangular matrix U_k such that $Q_k = A^k U_k$ and a lower triangular matrix L_k such that $Q_k = (A^*)^{-k} L_k$.

(b) Suppose $\lim_{k\to\infty} R_k = R_\infty$ and $\lim_{k\to\infty} Q_k = Q_\infty$ exist. Determine the eigenvalues of A in terms of R_∞.

Exercise 17.2. (JPE, May 2006) Let $A \in \mathbb{C}^{n\times n}$ be tri-diagonal and Hermitian, with all its super-diagonal entries nonzero. Prove that the eigenvalues of A are distinct.

(Hint: show that for any scalar λ, the matrix $A - \lambda I$ has rank at least $n - 1$.)