NUMERICAL LINEAR ALGEBRA

Lecture notes for MA 660A/B

Rudi Weikard

Contents

CHAPTER 1

Numerical Linear Algebra

These notes follow closely the book *Numerical Linear Algebra* by L.N. Trefethen and D. Bau, III (SIAM, Philadelphia, 1997).

1.1. Fundamentals

1.1.1. Matrix multiplication. The set of $m \times n$ matrices (m rows, n columns) with entries in a field $\mathbb{K}$ is denoted by $\mathbb{K}^{m\times n}$. For any matrix M we denote its entry in row j and column k by $M_{j,k}$. However, if $M \in \mathbb{K}^{m\times 1} = \mathbb{K}^m$ we will mostly use M_j instead of $M_{j,1}$. We denote the identity matrix by I, its entries, however, are denoted by $\delta_{k,j}$, which is called the Kronecker symbol. We will use the notation $M_{j:k,j':k'}$ for the submatrix of M whose upper left corner is the element $M_{j,j'}$ and whose lower right corner is the element $M_{k,k'}$. If $j = k$ or $j' = k'$ we will also use $M_{j,j':k'}$ and $M_{j:k,j'}$, respectively to refer to the resulting row or column.

The set $\mathbb{K}^{m\times n}$ is equipped in the usual way with an addition and a scalar multiplication by elements of K and hence it can be viewed as a vector space over $\mathbb{K}$. If $A \in \mathbb{K}^{\ell\times m}$ and $B \in \mathbb{K}^{m\times n}$ then a "product" $C = AB \in \mathbb{K}^{\ell\times n}$ can be defined by letting

$$C_{j,k} = \sum_{r=1}^{m} A_{j,r}B_{r,k}.$$

The set $\mathbb{K}^{m\times m}$ of $m \times m$ square matrices is an associative algebra over $\mathbb{K}$.

$m \times n$ matrices are used to represent linear transformations from $\mathbb{K}^n$ to $\mathbb{K}^m$. In this respect it is important to point out that it is customary to view the elements of $\mathbb{K}^m$ or $\mathbb{K}^n$ as columns rather than as rows.

In an equation $Ax = b$ where $A \in \mathbb{K}^{m\times n}$, $x \in \mathbb{K}^n$, and $b \in \mathbb{K}^m$ the vector b may be seen as the image of x under the transformation A. Another, equally important, point of view is to recognize that b is expressed as a linear combination of the columns of A whose coefficients are the entries (or components) of x. In particular, if A is invertible, the vector $x = A^{-1}b$ gives the coefficients of b when expanded with respect to the basis which is given by the columns of A.

1.1.2. Triangular and diagonal matrices. A matrix U is called *upper triangular* if $U_{j,k} = 0$ whenever $j > k$. Similarly a matrix L is called *lower triangular* if $L_{j,k} = 0$ whenever $j < k$. A matrix which is both upper and lower triangular is called *diagonal*. Equivalently, A is diagonal if and only if $A_{j,k} = 0$ whenever $j \neq k$.

1.1.3. Adjoint matrices. From now on we will agree that $\mathbb{K}$ is the field of complex numbers or one of its subfields. The *adjoint* or *hermitian conjugate* of a matrix $A \in \mathbb{C}^{m\times n}$, denoted by A^*, is the matrix whose entries satisfy $A^*_{j,k} = \overline{A_{k,j}}$. In particular, if the entries of A are real then $A^* = A^t$, the transpose of A. If A is an $m \times n$ matrix, then A^* is an $n \times m$ matrix. The following rules hold:

$(A+B)^* = A^* + B^*$,
$(\alpha A)^* = \overline{\alpha} A^*$,
$(AB)^* = B^* A^*$,
$A^{**} = A$.

The space $\mathbb{C}^m$ is an inner product space, the inner product is given by $(x, y) = x^* y$ whenever $x, y \in \mathbb{C}^m$. Note that it is convenient to have linearity in the second argument and antilinearity in the first argument[1]. The mapping $x \mapsto \|x\| = \sqrt{x^* x}$ is a norm in $\mathbb{C}^m$. With the help of the inner product the adjoint A^* of $A \in \mathbb{C}^{m\times n}$ can be characterized as the unique matrix in $\mathbb{C}^{n\times m}$ satisfying

$$(Ax, y) = (x, A^* y) \quad \text{whenever } x \in \mathbb{C}^n \text{ and } y \in \mathbb{C}^m.$$

A square matrix A is called *normal* if $AA^* = A^* A$. A is called *self-adjoint* or *hermitian* if $A = A^*$. If A is real and $A = A^t$, it is called *symmetric*. A square matrix A is called unitary if it is invertible and $A^* = A^{-1}$ (if, in this case, A is real it is also called orthogonal).

1.1.4. Orthogonal vectors and sets. Two vectors x, y in $\mathbb{C}^m$ are called orthogonal if $x^* y = 0$. Two sets $R, S \subset \mathbb{C}^m$ are called orthogonal if for all $x \in R$ and all $y \in S$ we have that $x^* y = 0$. A set is called orthonormal if its elements are pairwise orthogonal and each have norm 1. An orthonormal set is linearly independent. Let $\{q_1, ..., q_k\}$ be an orthonormal set in $\mathbb{C}^m$ and Q the matrix whose columns are the vectors $q_1, ..., q_k$. Let b be any vector in $\mathbb{C}^m$. Then there exists a unique r, which is orthogonal to each of the q_j, and which satisfies

$$b = r + (q_1^* b)q_1 + ... + (q_k^* b)q_k = r + Q(Q^* b).$$

In particular, if $k = m$ then $QQ^* = I$. Hence $r = 0$ and

$$b = (q_1^* b)q_1 + ... + (q_m^* b)q_m = Q(Q^* b),$$

i.e., the components of the vector $Q^* b$ are the coefficients of b when expanded with respect to the basis given by the columns of Q. The numbers $q_j^* b$, $j = 1, ..., m$ are often called the *Fourier coefficients* of b. Note that

$$\sum_{j=1}^{k} |q_j^* b|^2 \le b^* b = \|b\|^2.$$

This inequality is called *Bessel's inequality*. If $k = m$ we have in fact equality.

THEOREM. The columns (also the rows) of a unitary matrix form an orthonormal set. Also $(Qx)^*(Qy) = x^* y$ and $\|Qx\| = \|x\|$, i.e., a unitary matrix preserves angles between vectors and lengths of vectors.

Sketch of proof: Since $Q^* Q = I$ we have $\delta_{j,k} = q_j^* q_k$. This proves the first claim. For the second claim note that $(Qx)^* = x^* Q^*$. □

1.1.5. Norms. For every $x \in \mathbb{C}^m$ define

$$\|x\|_p = \left(\sum_{j=1}^{m} |x_j|^p \right)^{1/p}, \quad 1 \le p < \infty$$

and

$$\|x\|_\infty = \max\{|x_j| : 1 \le j \le m\}.$$

[1]The opposite convention was made in the Algebra notes.

THEOREM. If $1 \le p, q \le \infty$ satisfy $1/p + 1/q = 1$ and if $x, y \in \mathbb{C}^m$ then

$$|x^* y| \le \|x\|_p \|y\|_q \tag{1}$$

and

$$\|x + y\|_p \le \|x\|_p + \|y\|_p. \tag{2}$$

Inequality (1) is called Hölder's inequality. When $p = q = 2$ it becomes Schwarz's inequality. Inequality (2) is called Minkowski's inequality.

Sketch of proof: We first prove Hölder's inequality. If either p or q is infinity or if either x or y is zero the statement becomes trivial. Hence assume that $1 < p, q < \infty$ and that x and y are different from zero. It is then enough to consider the case when $\|x\|_p = \|y\|_q = 1$. If $x_j y_j \neq 0$ let $s_j = p \ln(|x_j|)$ and $t_j = q \ln(|y_j|)$. Since the exponential is convex we obtain

$$|x_j||y_j| = \exp(\frac{s_j}{p} + \frac{t_j}{q}) \le \frac{\exp(s_j)}{p} + \frac{\exp(t_j)}{q}.$$

Now sum over j and observe that

$$\sum_{\substack{1 \le j \le m \\ x_j y_j \neq 0}} \exp(s_j) \le \sum_{j=1}^{m} |x_j|^p = 1 \quad \text{and} \quad \sum_{\substack{1 \le j \le m \\ x_j y_j \neq 0}} \exp(t_j) \le \sum_{j=1}^{m} |y_j|^q = 1.$$

Next we consider Minkowski's inequality. We assume that $1 < p < \infty$ since the cases $p = 1$ and $p = \infty$ are trivial. From Hölder's inequality we obtain

$$\sum_{j=1}^{m} |x_j||x_j + y_j|^{p-1} \le \|x\|_p \ \|x + y\|_p^{p/q}$$

where $q = p/(p-1)$. Similarly

$$\sum_{j=1}^{m} |y_j||x_j + y_j|^{p-1} \le \|y\|_p \ \|x + y\|_p^{p/q}$$

Adding these two inequalities and using the triangle inequality we get

$$\|x + y\|_p^p \le (\|x\|_p + \|y\|_p) \, \|x + y\|_p^{p-1}.$$

which is the desired result. □

Minkowski's inequality is the hard part in showing that $x \mapsto \|x\|_p$ are norms when $1 \le p \le \infty$. These norms are called p-norms. If $p = \infty$ one calls it the sup-norm. Note that the 2-norm is the norm induced by the inner product. One may also introduce weighted p-norms: let W be an invertible $m \times m$ matrix. Then

$$\|x\|_{W,p} = \|Wx\|_p$$

is a norm.

As a vector space $\mathbb{C}^{m \times n}$ is isomorphic to $\mathbb{C}^{mn}$ and as such it can be normed as described above. For square matrices the most important of these is the 2-norm, usually called the Hilbert-Schmidt or Frobenius norm

$$\|A\|_2 = \left(\sum_{k=1}^{m} \sum_{j=1}^{m} |A_{j,k}|^2 \right)^{1/2}.$$

Note that

$$\|A\|_2^2 = \sum_{k=1}^{m} (A^*)_{k,1:m} A_{1:m,k} = \operatorname{tr}(A^* A) = \operatorname{tr}(A A^*). \tag{3}$$

1.1.6. Operator norms. Also important are induced norms for matrices which are called operator norms. Let A be an $m \times n$ matrix and assume that $\mathbb{C}^m$ and $\mathbb{C}^n$ are equipped with arbitrary norms. Then define

$$\|A\| = \sup\{\frac{\|Ax\|}{\|x\|} : 0 \neq x \in \mathbb{C}^n\}.$$

$A \mapsto \|A\|$ is a norm. Note that

$$\|A\| = \sup\{\|Ax\| : x \in \mathbb{C}^n, \|x\| = 1\} = \max\{\|Ax\| : x \in \mathbb{C}^n, \|x\| = 1\}$$

and

$$\|A\| = \inf\{C : \forall x \in \mathbb{C}^n : \|Ax\| \leq C\|x\|\}.$$

THEOREM. $\|AB\| \leq \|A\|\|B\|$ when the norms indicate induced norms.

Sketch of proof: $\|ABx\| \leq \|A\|\ \|Bx\| \leq \|A\|\ \|B\|\ \|x\|$. □

We will denote the operator norm of $A : \mathbb{C}^n \to \mathbb{C}^m$ by $\|A\|_{r,s}$ when $\mathbb{C}^n$ is equipped with the r-norm and $\mathbb{C}^m$ is equipped with the s-norm.

The operator norms are not easily calculated except when $r = s = 1$ or $r = s = \infty$. In these cases we have

$$\|A\|_{1,1} = \max\{\sum_{j=1}^{m} |A_{j,k}| : 1 \leq k \leq m\} \quad \text{and} \quad \|A\|_{\infty,\infty} = \max\{\sum_{k=1}^{m} |A_{j,k}| : 1 \leq j \leq m\}.$$

1.1.7. The singular value decomposition. A *singular value decomposition* (SVD) of a matrix $A \in \mathbb{C}^{m\times n}$ is a factorization

$$A = U\Sigma V^*$$

where $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ are unitary, Σ is diagonal, and, letting $p = \min\{m, n\}$,

$$\Sigma_{1,1} \geq \Sigma_{2,2} \geq ... \geq \Sigma_{p,p} \geq 0.$$

The nonnegative real numbers $\sigma_j = \Sigma_{j,j}$ are called the singular values of A. The columns of U are called the *left singular vectors* and the columns of V (not those of V^*) are called right singular vectors.

Note that $\Sigma\Sigma^* = U^{-1}AA^*U$ and $\Sigma^*\Sigma = V^{-1}A^*AV$. Hence, if $m \leq n$, the eigenvalues of AA^* are precisely the squares of the singular values of A, counting multiplicities, and the left singular vectors of A are the eigenvectors of AA^*. Similarly, if $n \leq m$, the eigenvalues of A^*A are precisely the squares of the singular values of A and the right singular vectors of A are the eigenvectors of A^*A.

THEOREM. Every matrix $A \in \mathbb{C}^{m\times n}$ has a singular value decomposition. The singular values are uniquely determined. Furthermore, if the singular values are pairwise distinct, then the following statements hold: if $m \leq n$ the left singular vectors are uniquely determined up to unimodular factors (complex numbers of modulus one) while, if $n \leq m$, the right singular vectors are uniquely determined up to unimodular factors.

Sketch of proof: Let $\|\cdot\|$ denote the matrix norm induced by equipping both $\mathbb{C}^n$ and $\mathbb{C}^m$ with the 2-norm. Obviously A^* has a singular value decomposition if and only if A does. It is therefore sufficient to let $\ell = n - m$ be fixed (but arbitrary in $\mathbb{N}_0$) and prove the theorem by induction over m. Let

$$M = \{m \in \mathbb{N} : \forall A \in \mathbb{C}^{m\times(\ell+m)} : A \text{ has an SVD}\}.$$

Let $m = 1$ and $\sigma_1 = \|A\|$. There exists a vector $v_1 \in \mathbb{C}^{\ell+1}$ and a number u_1 such that $\|v_1\| = 1 = |u_1|$, and $Av_1 = \sigma_1 u_1$. Extend $\{v_1\}$ to an orthonormal basis $\{v_1, ..., v_{\ell+1}\}$ of $\mathbb{C}^{\ell+1}$, let V denote the matrix whose k-th column is v_k, and let U be the 1×1 matrix with

entry u_1. Then we have $U^*AV = (\sigma_1, w^*)$ for some vector $w \in \mathbb{C}^\ell$. We will show $w = 0$ so that $\Sigma = (\sigma_1, 0, ..., 0)$ and hence $1 \in M$. To prove $w = 0$ note that

$$\sqrt{\sigma_1^2 + w^*w} \left\| \begin{pmatrix} \sigma_1 \\ w \end{pmatrix} \right\| = |(U^*AV) \begin{pmatrix} \sigma_1 \\ w \end{pmatrix}| \leq \|U\| \, \|A\| \, \|V\| \, \left\| \begin{pmatrix} \sigma_1 \\ w \end{pmatrix} \right\| = \sigma_1 \left\| \begin{pmatrix} \sigma_1 \\ w \end{pmatrix} \right\|.$$

Next suppose that $m-1 \in M$ and that $A \in \mathbb{C}^{m\times(\ell+m)}$. Again let $\sigma_1 = \|A\|$. Now there is a $v_1 \in \mathbb{C}^{\ell+m}$ and $u_1 \in \mathbb{C}^m$ such that $\|v_1\| = 1 = \|u_1\|$ and $Av_1 = \sigma_1 u_1$. Again extend $\{v_1\}$ to an orthonormal basis $\{v_1, ..., v_{\ell+m}\}$ of $\mathbb{C}^{\ell+m}$ and also $\{u_1\}$ to an orthonormal basis $\{u_1, ..., u_m\}$ of $\mathbb{C}^m$. Collect these vectors in matrices V_1 and U_1 with v_1 and u_1 as first columns, respectively. Then we have

$$U_1^*AV_1 = \begin{pmatrix} \sigma_1 & w^* \\ 0 & B \end{pmatrix}$$

where B is some matrix in $\mathbb{C}^{(m-1)\times(\ell+m-1)}$ and w is some vector in $\mathbb{C}^{\ell+m-1}$. As before

$$\sqrt{\sigma_1^2 + w^*w} \left\| \begin{pmatrix} \sigma_1 \\ w \end{pmatrix} \right\| \leq \left\| U_1^*AV_1 \begin{pmatrix} \sigma_1 \\ w \end{pmatrix} \right\| \leq \|U_1\| \, \|A\| \, \|V_1\| \, \left\| \begin{pmatrix} \sigma_1 \\ w \end{pmatrix} \right\| \leq \sigma_1 \left\| \begin{pmatrix} \sigma_1 \\ w \end{pmatrix} \right\|$$

shows that $w = 0$. Since $m-1 \in M$ the matrix B has a singular value decomposition given by $B = U_2\Sigma_2V_2^*$, where the largest element in Σ_2 is $\|B\| \leq \sigma_1$. Now let

$$U = U_1 \begin{pmatrix} 1 & 0 \\ 0 & U_2 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix} \quad \text{and} \quad V = V_1 \begin{pmatrix} 1 & 0 \\ 0 & V_2 \end{pmatrix}.$$

Then $U\Sigma V^*$ is a singular value decomposition of A.

The uniqueness statement follows from the relationship of the singular values and singular vectors of A with the eigenvalues and eigenvectors of AA^* and A^*A. □

1.1.8. Further properties of the SVD. The importance of the SVD becomes apparent from the following properties. Let A be an $m \times n$-matrix of rank r.

(1) The rank of A equals the number of nonzero singular values of A.
(2) $\mathrm{im}(A) = \langle u_1, ..., u_r \rangle$ and $\ker(A) = \langle v_{r+1}, ..., v_n \rangle$.
(3) $\|A\|_{2,2} = \sigma_1$ and $\|A\|_2 = \sqrt{\sigma_1^2 + ... + \sigma_r^2}$.
(4) If $m = n$ then $|\det(A)| = \prod_{j=1}^r \sigma_j$.
(5) For any $k \in \{1, ..., r\}$ define $A_k = \sigma_1 u_1 v_1^* + ... + \sigma_k u_k v_k^*$. Then $A = A_r$ and

$$\|A - A_k\|_{2,2} = \inf\{\|A - B\|_{2,2} : B \in \mathbb{C}^{m\times n}, \mathrm{rank}(B) \leq k\} = \sigma_{k+1}$$

where we define $\sigma_k = 0$, if $k > \min\{m, n\}$.

Sketch of proof: Only the second statement in (3) and statement (5) need closer examination. The former follows from $\|A\|_2 = \|U^*AV\|_2$ which, in turn, follows from equation (3) in 1.1.5. For the latter, note firstly that $\|A - A_k\|_{2,2} = \sigma_{k+1}$ by Bessel's inequality. Then assume that there is a B with $\mathrm{rank}(B) \leq k$ such that $\|A - B\|_{2,2} < \|A - A_k\|_{2,2}$. Note that $W = \ker(B)$ has dimension at least $n - k$. Hence, if $0 \neq w \in W$, then

$$\|Aw\|_2 = \|(A - B)w\|_2 \leq \|A - B\|_{2,2}\|w\|_2 < \sigma_{k+1}\|w\|_2.$$

On the other hand the first $k + 1$ right singular vectors of A span a space X such that $\|Ax\|_2 \geq \sigma_{k+1}\|x\|_2$. This gives that $W \cap X$ is nontrivial, which is impossible. □

1.1.9. Flops and operation count.

DEFINITION. A *flop* is any of the operations of an addition, subtraction, multiplication, division, or extraction of a square root.

For example, to compute an inner product in $\mathbb{C}^m$ one has to perform m multiplications and $m-1$ sums. Hence this computation requires $2m-1$ flops. The computation of the norm of a vector in $\mathbb{C}^m$ requires $2m$ flops.

This definition of flop gives only a very simplified model of the actual computation cost on a real computer. For example we ignore the difference between real and complex arithmetic, the moving of data between the CPU and the memory. Taking these and other aspects into account is far beyond the scope of this course. Instead we are just trying to sensitize ourselves to the importance of considerations of this kind.

1.2. Error Analysis

1.2.1. Axioms of idealized computer arithmetic. In a number system with base β (where β is an integer not smaller than two) a positive integer may be represented as a sequence of digits, i.e.,

$$d_1 d_2 ... d_k = d_1 \beta^{k-1} + ... + d_k \beta^0.$$

Similarly a positive real number is represented as (a possible semi-infinite) sequence of digits. By introducing an exponential notation we can represent every positive real number as

$$d_0.d_1 d_2 ... \times \beta^e = \sum_{k=0}^{\infty} d_k \beta^{e-k}$$

where e is chosen such that $d_0 \neq 0$. This digital representation is called the normalized or floating point representation of the number. The fractional part $d_0.d_1 d_2 ...$ is called the *mantissa*, and the integer e is called the *exponent* of the number. When $\beta = 10$ we obtain our familiar decimal system. Because of their digital structure computers use $\beta = 2$. In our considerations below we will always use $\beta = 2$.

To represent real numbers in computer languages like Fortran one faces the difficulty of having only finite storage space. This implies firstly, that the exponent can assume only finitely many values preventing the representation of numbers which are too large or too small in absolute value (leading to overflow and underflow problems). Secondly the mantissa may have only finitely many digits preventing that numbers can be arbitrarily closely approximated. Therefore computer arithmetic is somewhat different from our familiar arithmetic. In this lecture, following Trefethen and Bau, we will use a model for computer arithmetic which still does not capture reality but with which the basic issues can be studied. In doing so we disregard the fact that on a computer the exponent can only come from a finite interval of integers and hence we disregard the possibility of overflow and underflow.

Fix a positive integer p and define

$$F = \{\pm \left(1 + \frac{m}{2^p}\right) 2^e : m \in \{0, 1, 2, ..., 2^p - 1\}, e \in \mathbb{Z}\} \cup \{0\}.$$

F is called the set of floating point numbers. Note that $1 + m2^{-p} \in [1, 2)$. The number p represents the (relative) precision with which real numbers can be approximated.

Sometimes, when we discuss algorithms, we will use the following axiom which characterizes idealized computer arithmetic:

AXIOM. There exists a positive real number $\varepsilon_M < 1/2$ and a function $\mathrm{fl} : \mathbb{R} \to F$ such that

(1) for all $x \in \mathbb{R}$ we have that $|\operatorname{fl}(x) - x| \leq \varepsilon_M |x|$ and
(2) for all $x, y \in F$ and every $* \in \{+, -, \times, \div\}$ we have that $|\operatorname{fl}(x*y) - x*y| \leq \varepsilon_M |x*y|$.

The number ε_M is typically a small multiple of the machine precision 2^{-p}.

1.2.2. Implementation on actual computers. We will now describe how numbers are stored according to the widely used IEEE[2] standard 754-1985. A *bit* (binary digit) is a unit of storage which may have either value one or value zero. A number is stored in $1+q+p$ bits. The first bit contains the sign of a number (0 for positive and 1 for negative). Next come q bits to encode the exponent, which represent an integer e such that $0 \leq e \leq 2^q - 1$. In order to get negative exponents one subtracts always a fixed number b from e called the *bias*. The last p bits are used to encode the mantissa. They represent an integer m such that $0 \leq m \leq 2^p - 1$. If $e \neq 0$ the mantissa is given by $1 + m2^{-p}$. If $e = 0$ and $m = 0$ one has the number zero. If $e = 0$ and $m \neq 0$ the number represented is smaller than 2^{-b} and cannot be represented as a normalized number without choosing the exponent to be out of bounds (i.e., smaller than $-b$). These numbers are called denormal and will be disregarded in what follows. In summary, the number represented by $e \neq 0$ and m is

$$x = \pm(1 + \frac{m}{2^p})2^{e-b}.$$

For the IEEE single-precision standard we have $q = 8$ and $p = 23$ so that a number uses 32 bits. We also have $b = 127$. For the IEEE double-precision standard we have $q = 11$ and $p = 52$ so that a number uses 64 bits. In this case $b = 1023$. If all exponent bits are equal to 1 the number represents infinity. The largest exponents are therefore $2^8 - 2$ and $2^{11} - 2$, respectively. The largest numbers represented are close to $2^{128} \approx 3.4 \times 10^{38}$ and $2^{1024} \approx 1.8 \times 10^{308}$, respectively. The smallest positive normalized numbers are $2^{-126} \approx 1.2 \times 10^{-38}$ and $2^{-1022} \approx 2.2 \times 10^{-308}$, respectively. The smallest number bigger than 1 is $1 + 2^{-p}$ and 2^{-p} equals 1.2×10^{-7} or 2.2×10^{-16} in single- and double-precision, respectively.

The function fl is typically implemented by either chopping off or rounding off unrepresentable digits. If overflow or underflow occurs programs usually stop with an appropriate error message.

1.2.3. Relative errors. There are typically two sources of errors when one tries to compute quantities in applications: one is that data are often not precisely known (for instance measured data) and the others are errors introduced by the algorithms used. An example of the latter are, most notably, the round-off errors one has to deal with in numeric computations. Needless to say, it is the goal to keep errors small and, in any case, to keep track of the size of errors.

Let f be a continuous (but not necessarily linear) function from a normed vector space X to a normed vector space Y. The function $\tilde{f} : X \to Y$ is called an *algorithm* for f if $\tilde{f}$ approximates in some sense the function f at least in some subset S of X. For instance, if f is the identity on $\mathbb{R}$ the $\tilde{f}$ could be the function fl. Or, in order to describe inaccurate data we could have $\tilde{f}(x) = x(1 + \varepsilon(x))$ where it is known that $\|\varepsilon(x)\|$ is bounded by some fixed small number. Not surprisingly, this shows that inaccurate data and round-off errors are treated in the same way.

We are interested in the relative errors (rather than absolute errors) introduced by using $\tilde{f}$ instead of f. The relative error at $x \in X$ is defined by

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|}$$

[2]IEEE stands for Institute of Electrical and Electronics Engineers, Inc.

and it is our goal to study this quantity.

The key idea of estimating the relative error is due to Wilkinson: one tries to prove the existence of a point $\tilde{x}$ close to x such that $\|f(\tilde{x}) - \tilde{f}(x)\|$ can be easily controlled. If then the function f is insensitive to small perturbations, i.e., if $\|f(\tilde{x}) - f(x)\|$ can be shown to be relatively small one can establish an error bound. In other words one tries to make use of the inequality

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} \leq \frac{\|\tilde{f}(x) - f(\tilde{x})\|}{\|f(x)\|} + \frac{\|f(\tilde{x}) - f(x)\|}{\|f(x)\|}.$$

The first term on the right describes then the properties of the algorithm (and can often be shown to be zero). It will be discussed in 1.2.4. The second term on the right describes then the sensitivity (or condition) of the problem. It will be discussed in 1.2.8. In 1.2.13 we shall come back to combine these two ingredients to obtain a bound for the relative error.

1.2.4. Stability. Let f be a continuous (but not necessarily linear) function from a normed vector space X to a normed vector space Y. Let S be a subset of X. The function $\tilde{f} : X \to Y$ is called a *stable algorithm* for f on S if for all $x \in S$ there exist nonnegative real numbers C_1 and C_2 and an $\tilde{x} \in X$ with the following properties:

$$\|x - \tilde{x}\| \leq C_1 \varepsilon_M \|x\| \quad \text{and} \quad \|\tilde{f}(x) - f(\tilde{x})\| \leq C_2 \varepsilon_M \|f(\tilde{x})\|.$$

The numbers C_1 and C_2 characterize the algorithm. Too large values for either of these numbers render an algorithm useless.

If $C_2 = 0$ then $\tilde{f}$ is called a *backward stable algorithm*.

1.2.5. Backward stability of idealized computer arithmetic. Let $A_* : \mathbb{R}^2 \to \mathbb{R}$ represent one of the four basic arithmetic operations, i.e., $A_*(x, y) = x * y$ where $* \in \{+, -, \times, \div\}$. A computer can not directly apply A_*. Instead it has to use the function

$$\tilde{A}_* : \mathbb{R}^2 \to F : (x, y) \mapsto \mathrm{fl}(\mathrm{fl}(x) * \mathrm{fl}(y)).$$

THEOREM. Each of the operations $\tilde{A}_*$ is backward stable. Using the sup-norm in $\mathbb{R}^2$ we have that $C_1 \leq 3$.

Sketch of proof: We will perform only the proof for addition. From Axiom 1.2.1 we have that $\mathrm{fl}(x) = x(1+\varepsilon_1)$, $\mathrm{fl}(y) = y(1+\varepsilon_2)$, and that $\mathrm{fl}(\mathrm{fl}(x)+\mathrm{fl}(y)) = (\mathrm{fl}(x)+\mathrm{fl}(y))(1+\varepsilon_3)$ where $|\varepsilon_1|, |\varepsilon_2|, |\varepsilon_3| \leq \varepsilon_M$. Hence

$$\tilde{A}_+(x, y) = \mathrm{fl}(\mathrm{fl}(x)+\mathrm{fl}(y)) = x(1+\varepsilon_1+\varepsilon_3+\varepsilon_1\varepsilon_3)+y(1+\varepsilon_2+\varepsilon_3+\varepsilon_2\varepsilon_3) = \tilde{x}+\tilde{y} = A_+(\tilde{x}, \tilde{y})$$

defining $\tilde{x} = x(1 + \varepsilon_1 + \varepsilon_3 + \varepsilon_1\varepsilon_3)$ and $\tilde{y} = y(1 + \varepsilon_2 + \varepsilon_3 + \varepsilon_2\varepsilon_3)$. Finally note that

$$\|(\tilde{x}, \tilde{y}) - (x, y)\|_\infty \leq 3\varepsilon_M \|(x, y)\|_\infty.$$

□

For simplicity we introduce the following notation: $a_1 \oplus a_2 = \tilde{A}_+(a_1, a_2)$ and $a_1 \otimes a_2 = \tilde{A}_\times(a_1, a_2)$. (It should be remarked that in Trefethen and Bau $\oplus$ and $\otimes$ are only defined for elements of F.)

1.2.6. Backward stability of adding many terms. In computer arithmetic addition and multiplication are not associative anymore, which prompts us to define

$$\bigoplus_{j=1}^{n} a_j = ((a_1 \oplus a_2) \oplus a_3) \oplus \ldots \oplus a_n.$$

Let $\mathrm{fl}(a_j) = a_j(1+\delta_j)$ and hence $|\delta_j| \le \varepsilon_M$. A repeated application Axiom 1.2.1 gives then

$$\bigoplus_{j=1}^{n} a_j = a_1(1+\delta_1)\prod_{k=1}^{n-1}(1+\varepsilon_j) + \sum_{j=2}^{n} a_j(1+\delta_j)\prod_{k=j-1}^{n-1}(1+\varepsilon_j)$$

where the ε_j are suitable numbers satisfying $|\varepsilon_j| \le \varepsilon_M$.

One may prove by induction that

$$\left|\prod_{j=1}^{n}(1+\varepsilon_j) - 1\right| \le n\varepsilon_M + n^2\varepsilon_M^2 \tag{4}$$

if $|\varepsilon_j| \le \varepsilon_M \le 1/n$.

Hence we proved the following theorem.

THEOREM. If $a \in \mathbb{R}^n$ then there exists $\tilde{a} \in \mathbb{R}^n$ such that

$$\bigoplus_{j=1}^{n} a_j = \sum_{j=1}^{n} \tilde{a}_j$$

and

$$\|\tilde{a} - a\|_\infty \le (n\varepsilon_M + n^2\varepsilon_M^2)\|a\|_\infty.$$

This shows that round-off errors may accumulate.

1.2.7. Backward stability of back substitution. One of the easiest and most important problems in linear algebra is the solution of $Rx = b$, where R is upper triangular and nonsingular. Beginning with the last variable on can successively compute all the unknowns, a process called back substitution. Consider the system

$$\begin{pmatrix} R_{1,1} & R_{1,2} & \cdots & R_{1,m} \\ & R_{2,2} & & R_{2,m} \\ & & \ddots & \vdots \\ & & & R_{m,m} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}.$$

To solve this system we will employ the following algorithm:

ALGORITHM.

for $j = 1$ **to** m
  for $k = m + 2 - j$ **to** m
    $b_{m+1-j} = b_{m+1-j} - R_{m+1-j,k}x_k$
  $x_{m+1-j} = b_{m+1-j}/R_{m+1-j,m+1-j}$
end

To work through this algorithm takes $\sum_{j=1}^{m}(2(j-1)+1) = m^2$ flops.

THEOREM. Let $R \in \mathbb{C}^{m\times m}$ be upper triangular and non-singular and let $b \in \mathbb{C}^M$. Assuming the validity of Axiom 1.2.1 Algorithm 1.2.7 is backward stable with a value of $C_1 = (m+3) + (m+3)^2\varepsilon_M$. Specifically, let $f(R,b) = R^{-1}b$ and $\tilde{f}(R,b)$ the corresponding value computed by the algorithm. Then, for every pair (R,b), there exists a pair $(\tilde{R},\tilde{b})$ such that

$$\tilde{f}(R,b) = f(\tilde{R},\tilde{b}),$$

$$|\tilde{b}_k - b_k| \le ((m+2)\varepsilon_M + (m+2)^2\varepsilon_M)|b_k|,$$

and

$$|\tilde{R}_{j,k} - R_{j,k}| \le ((m+3)\varepsilon_M + (m+3)^2\varepsilon_M)|R_{j,k}|.$$

Sketch of proof: We will prove the theorem by induction. Let S be the set of all $s \leq m$ which satisfy the following three requirements:

(1) There are numbers $\tilde{R}_{m+1-j,k}$ such that

$$|\tilde{R}_{m+1-j,k} - R_{m+1-j,k}| \leq ((m+4-k)\varepsilon_M + (m+4-k)^2\varepsilon_M^2)|R_{m+1-j,k}|$$

for $k = 1, ..., m$ and $j = 1, ..., s$.

(2) There are numbers $\tilde{b}_{m+1-j}$ such that

$$|\tilde{b}_{m+1-j} - b_{m+1-j}| \leq ((j+1)\varepsilon_M + (j+1)^2\varepsilon_M^2)|b_{m+1-j}|$$

for $j = 1, ..., s$.

(3) The numbers $\tilde{x}_k$ computed by Algorithm 1.2.7 satisfy

$$\sum_{k=1}^{m} \tilde{R}_{m-j+1,k}\tilde{x}_k = \tilde{b}_{m-j+1}$$

for $j = 1, ..., s$.

We first prove that $1 \in S$. Define $\tilde{R}_{m,k} = 0$ for $k = 1, ..., m-1$. Note that, by Axiom 1.2.1 there are numbers ρ_m, $\sigma_{m,m}$, and ε_m not larger than ε_M in absolute value such that

$$\tilde{x}_m = \frac{\mathrm{fl}(b_m)(1+\varepsilon_m)}{\mathrm{fl}(R_{m,m})} = \frac{b_m(1+\rho_m)(1+\varepsilon_m)}{R_{m,m}(1+\sigma_{m,m})}.$$

Hence we define

$$\tilde{R}_{m,m} = R_{m,m}(1+\sigma_{m,m})$$

and

$$\tilde{b}_m = b_m(1+\rho_m)(1+\varepsilon_m).$$

Recalling inequality 4 this shows that $1 \in S$.

Next assume that $1, ..., s \in S$ and consider $s+1$. Recall that

$$x_{m-s} = \frac{b_{m-s} - \sum_{k=m-s+1}^{m} R_{m-s,k}x_k}{R_{m-s,m-s}}.$$

We let $\tilde{R}_{m-s,k} = 0$ for $k = 1, ..., m-s-1$. We further define for $k = m-s+1, ..., m$

$$T_{m-s,k} = \tilde{x}_k \otimes \mathrm{fl}(R_{m-s,k}) = \tilde{x}_k \otimes R_{m-s,k}(1+\sigma_{m-s,k}) = \tilde{x}_k R_{m-s,k}(1+\sigma_{m-s,k})(1+\delta_{m-s,k})$$

and

$$T_{m-s,m-s} = -\mathrm{fl}(b_{m-s}) = -b_{m-s}(1+\rho_{m-s}).$$

Note that $\sigma_{m-s,k}$, $\delta_{m-s,k}$, and ρ_{m-s} are not larger than ε_M in absolute value.

Once we have computed $\bigoplus_{k=m-s}^{m} T_{m-s,k}$ we get

$$\tilde{x}_{m-s} = -\frac{\bigoplus_{k=m-s}^{m} T_{m-s,k}}{\mathrm{fl}(R_{m-s,m-s})}(1+\epsilon_{m-s}) = -\frac{(1+\epsilon_{m-s})\bigoplus_{k=m-s}^{m} T_{m-s,k}}{\tilde{R}_{m-s,m-s}}$$

where $\tilde{R}_{m-s,m-s} = \mathrm{fl}(R_{m-s,m-s})$. As in 1.2.6 we have

$$(1+\epsilon_{m-s})\bigoplus_{k=m-s}^{m} T_{m-s,k} = -\tilde{b}_{m-s} + \sum_{k=m-s+1}^{m} \tilde{x}_k\tilde{R}_{m-s,k}$$

where

$$\tilde{b}_{m-s} = b_{m-s}(1+\rho_{m-s,k})(1+\epsilon_{m-s})\prod_{r=0}^{s-1}(1+\zeta_{m-s,r})$$

and, for $k = m - s + 1, ..., m$,

$$\tilde{R}_{m-s,k} = R_{m-s,k}(1 + \sigma_{m-s,k})(1 + \delta_{m-s,k})(1 + \epsilon_{m-s}) \prod_{r=k+s-m-1}^{s-1} (1 + \zeta_{m-s,r})$$

for suitable values of $\zeta_{m-s,r}$ which, in absolute value, do not exceed ε_M. Employing now inequality 4 shows that $s + 1 \in S$. □

1.2.8. Condition numbers. Let f be a continuous (but not necessarily linear) function from a normed vector space X to a normed vector space Y and let x be an element of X. If $x \neq 0$ define

$$\hat{\kappa}(f, x, \varepsilon) = \inf\{k \geq 0 : \forall h \in X : \|h\| > \varepsilon\|x\| \ \vee \ \|f(x + h) - f(x)\| \leq k\|h\|\}.$$

Note that the infimum is actually a minimum. The number $\hat{\kappa}(f, x, \varepsilon)$ is called the *absolute condition number* associated with f, x, and ε.

THEOREM. The function $\hat{\kappa}(f, x, \cdot)$ is monotonically increasing and, if f is differentiable at x, then

$$\lim_{\varepsilon \to 0} \hat{\kappa}(f, x, \varepsilon) = \|f'(x)\|.$$

Sketch of proof: If ε becomes larger we have to check the inequality $\|f(x+h) - f(x)\| \leq k\|h\|$ for a larger set of vectors h and hence fewer values of k may be suitable. The second statement follows from this and the definition of the derivative. □

If $x \neq 0$ and $f(x) \neq 0$ we call

$$\begin{aligned}\kappa(f, x, \varepsilon) &= \frac{\|x\|}{\|f(x)\|}\hat{\kappa}(f, x, \varepsilon) \\ &= \inf\{k \geq 0 : \forall h \in X : \|h\| > \varepsilon\|x\| \ \vee \ \frac{\|f(x + h) - f(x)\|}{\|f(x)\|} \leq k\frac{\|h\|}{\|x\|}\}\end{aligned}$$

the *relative condition number* associated with f, x, and ε. Relative condition numbers are scale-invariant and are therefore the objects we are really interested in.

If Ω is a subset of $X - \{0\}$ and if f does not assume the value zero on Ω define the relative condition number associated with f, x, and Ω by

$$\kappa(f, \Omega, \varepsilon) = \sup\{\kappa(f, x, \varepsilon) : x \in \Omega\}.$$

The equality $\kappa(f, x, \varepsilon) = \kappa(f, \{x\}, \varepsilon)$ justifies this abuse of notation. The number $\kappa(f, \Omega, \varepsilon)$ is the relative condition number of f and ε uniformly over Ω.

1.2.9. The condition number of a linear function. Let $f(x) = Ax$ where $A \in \mathbb{C}^{m \times n}$. Since $f(x + h) - f(x) = Ah$ and since $\|Ah\| \leq \|A\|\|h\|$ we obtain for all $x \neq 0$ and all positive ε

$$\hat{\kappa}(f, x, \varepsilon) = \inf\{k \geq 0 : \forall h \in X : \|Ah\| \leq k\|h\|\} = \|A\|.$$

Now suppose that $m \geq n$ and that A has full rank n. Then A^*A is invertible. To see this suppose $A^*Ax = 0$. Then $0 = x^*(A^*Ax) = \|Ax\|^2$, i.e., $Ax = 0$. Since A has full rank it has trivial kernel and hence $x = 0$. The matrix

$$A^+ = (A^*A)^{-1}A^*$$

is called the *pseudo-inverse* of A, since $A^+A = I$. (Note that $A^+ = A^{-1}$ if A itself is invertible.) Now

$$\|x\| = \|A^+Ax\| \leq \|A^+\| \ \|Ax\|.$$

Hence

$$\kappa(f, x, \varepsilon) \leq \|A\| \ \|A^+\|$$

and, since $\ker(A) = \{0\}$,

$$\kappa(f, \mathbb{C}^n - \{0\}, \varepsilon) = \sup\{\|A\| \frac{\|A^+ Ax\|}{\|Ax\|} : Ax \neq 0\} = \|A\| \ \|A^+\|$$

regardless of ε. Hence $\|A\| \ \|A^+\|$ is the uniform relative condition number of $f : x \mapsto Ax$ regardless of ε. It is simply denoted by $\kappa(A)$.

1.2.10. The condition number of addition and catastrophic cancellations. The operation of adding two complex numbers is a linear transformation from $\mathbb{C}^2$ to $\mathbb{C}$. The associated matrix is $A = (1, 1)$. The relative condition number is therefore

$$\kappa_p = 2^{(p-1)/p} \frac{(|x_1|^p + |x_2|^p)^{1/p}}{|x_1 + x_2|}.$$

If $x_1 + x_2$ is nearly zero then the condition number becomes very large. This phenomenon is called catastrophic cancellation.

1.2.11. The condition of solving a system of equations. For invertible $A \in \mathbb{C}^{m\times m}$ and $b \in \mathbb{C}^m$ define $f : (A, b) \mapsto x = A^{-1}b$. Suppose $h = (\delta A, \delta b)$ where $\|\delta A\| \ \|A^{-1}\| \leq 1/2$. Then $\delta x = f((A, b) + h) - f(A, b)$ satisfies

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

and this implies

$$\begin{aligned} \frac{1}{2}\|\delta x\| &\leq (1 - \|\delta A\| \ \|A^{-1}\|)\|\delta x\| \leq \|A^{-1}\|(\|\delta b\| + \|\delta A\| \ \|x\|) \\ &\leq \|A^{-1}\| \ \|A\| \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right) \|x\|. \end{aligned}$$

Hence

$$\frac{\|\delta x\|}{\|x\|} \leq 2\kappa(A) \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right). \tag{5}$$

From this inequality one can easily compute an estimate on the condition number κ such that

$$\frac{\|\delta x\|}{\|x\|} \leq \kappa(f, (A, b), \varepsilon_M) \frac{\|(\delta A, \delta b)\|}{\|(A, b)\|}$$

but it is even more useful in the form (5).

1.2.12. The condition number of root extraction. We will show here that the problem of finding roots of polynomials may have a large condition number. Let

$$p_a(z) = \prod_{j=1}^{20} (z - j) = \sum_{j=0}^{20} a_j z^j.$$

Assume that the coefficients a_j are perturbed by (small) quantities h_j. Let $a = (a_0, ..., a_{20})$, $h = (h_0, ..., h_{20})$, and $f(a + h)$ the fifteenth root of p_{a+h} (where the roots are ordered by magnitude). We then have $f(a) = 15$. By the inverse function theorem we have

$$f'(a) = -\frac{(\frac{\partial p_{a+h}}{\partial h_0}(15), ..., \frac{\partial p_{a+h}}{\partial h_{20}}(15))}{p'_a(15)}$$

and hence

$$\lim_{\varepsilon \to 0} \kappa(f, a, \varepsilon) = \|f'(a)\| \frac{\|a\|}{15}.$$

Assuming for simplicity that only the coefficient of z^{13} changes, i.e., that f depends only on h_{13} we obtain

$$\lim_{\varepsilon\to 0}\kappa(f,a,\varepsilon)=\frac{15^{13}}{5!14!}\frac{756111184500}{15}\approx 9.38\times 10^{12}.$$

Please recall that the value of κ for a positive ε can not be smaller than this.

1.2.13. Backward Error Analysis. We are now ready to estimate relative errors suppose that $\tilde f : X \to Y$ is a stable algorithm for $f : X \to Y$, that $\tilde f$ is characterized by the constants C_1 and C_2, and that f has condition number κ with respect to x and ε_M. Then we have that there is a point $\tilde x$ such that $\|\tilde x - x\| \le C_1\varepsilon_M\|x\|$ and

$$\begin{aligned}\frac{\|\tilde f(x)-f(x)\|}{\|f(x)\|} &\le \frac{\|\tilde f(x)-f(\tilde x)\|}{\|f(x)\|}+\frac{\|f(\tilde x)-f(x)\|}{\|f(x)\|}\\ &\le C_2\varepsilon_M\frac{\|f(\tilde x)\|}{\|f(x)\|}+\frac{\hat\kappa(f,x,C_1\varepsilon_M)\|\tilde x-x\|}{\|f(x)\|}\\ &\le C_2\varepsilon_M\left(1+\kappa(f,x,C_1\varepsilon_M)\frac{\|\tilde x-x\|}{\|x\|}\right)+\kappa(f,x,C_1\varepsilon_M)\frac{\|\tilde x-x\|}{\|x\|}\\ &\le \left(C_2(1+\kappa(f,x,C_1\varepsilon_M)C_1\varepsilon_M)+\kappa(f,x,C_1\varepsilon_M)C_1\right)\varepsilon_M.\end{aligned}$$

In particular, in the case of backward stability ($C_2 = 0$) we have

$$\frac{\|\tilde f(x)-f(x)\|}{\|f(x)\|}\le\kappa(f,x,C_1\varepsilon_M)C_1\varepsilon_M.$$

The main result of these considerations is that large values of κ, C_1, and C_2 may very well destroy the accuracy for numeric computations. For instance, to compute eigenvalues as roots of a characteristic polynomial may not a good idea from the numerical point of view, since the condition number of root extraction can be very large as we demonstrated earlier.

1.2.14. Error analysis for back substitution. In 1.2.11 we investigated the condition of solving a system of equation and obtained the following result:

$$\frac{\|\delta x\|}{\|x\|}\le 2\kappa(A)\left(\frac{\|\delta b\|}{\|b\|}+\frac{\|\delta A\|}{\|A\|}\right).$$

In Theorem 1.2.7 we showed that back substitution is a backward stable algorithm and in particular that, using sup-norms,

$$\frac{\|\delta b\|}{\|b\|}\le(m+2)\varepsilon_M+(m+2)^2\varepsilon_M^2\quad\text{and}\quad\frac{\|\delta A\|}{\|A\|}\le(m+3)\varepsilon_M+(m+3)^2\varepsilon_M^2.$$

Hence

$$\frac{\|\delta x\|}{\|x\|}\le 4\kappa(A)((m+3)\varepsilon_M+(m+3)^2\varepsilon_M^2).$$

1.3. QR Factorization

1.3.1. Projections. A $m\times m$ matrix P is called *idempotent* or a *projection* if $P^2 = P$. Given a projection P the matrix $I-P$ is also a projection, called the *projection complementary to* P. Note that $\ker(P) = \operatorname{im}(I-P)$, $\ker(I-P) = \operatorname{im}(P)$, $\ker(P)\cap\ker(I-P) = \{0\}$, and $\operatorname{im}(P)\oplus\operatorname{im}(I-P) = \mathbb{C}^m$. Conversely, if $M\oplus N = \mathbb{C}^m$ then there is a unique projection P with $\operatorname{im}(P) = M$ and $\ker(P) = N$. This projection is the called the projection onto M along N. If M and N are orthogonal then P is called the orthogonal projection onto M.

If P is a projection then its eigenvalues are in $\{0, 1\}$.

THEOREM. Let P be a projection. Then the following statements are equivalent:
(1) P is an orthogonal projection.
(2) P is self-adjoint, i.e., $P = P^*$.
(3) P is normal.
(4) $I - P$ is an orthogonal projection.
(5) $\|P\| \leq 1$.

Sketch of proof: Suppose P is an orthogonal projection and $x = x_1 + x_2$ and $y = y_1 + y_2$ where $x_1, y_1 \in \text{im}(P)$ and $x_2, y_2 \in \ker(P)$. Then $x^*(P^*y) = (Px)^*y = x_1^*(y_1 + y_2) = (x_1 + x_2)^*y_1 = x^*(Py)$. Hence (1) implies (2). That (2) implies (3) is trivial. Next assume that P is normal. First note that P^* is a projection, since $(P^*)^2 = (PP)^* = P^*$. The normality of P implies $\|Px\| = \|P^*x\|$ for all x and hence $\ker(P) = \ker(P^*)$. If $x \in \ker(P) = \ker(P^*)$ then $0 = (P^*x)^*y = x^*(Py)$ which implies that $\text{im}(P) \perp \ker(P)$, i.e., P is an orthogonal projection. So (3) implies (1). The equivalence of (1) and (4) is immediate. Next assume $P = P^*$. The squares of the singular values of P are the eigenvalues of PP^* and hence the largest singular value, which equals $\|P\|$, is not more than one, i.e., (2) implies (5). Finally, assume that $\ker(P)$ and $\text{im}(P)$ are not orthogonal. Then there is an $x \in \text{im}(P)$ and a $y \in \ker(P)$ such that $x^*y \neq 0$ and hence there is a number α such that $\|y\|^2 + \alpha y^*x + \overline{\alpha}x^*y < 0$. But this implies that $\|P(y + \alpha x)\|/\|y + \alpha x\| > 1$. Thus (5) implies (1). □

The orthogonal projection onto the subspace spanned by a vector $a \neq 0$ is given by

$$P_a = \frac{aa^*}{a^*a}.$$

More generally, the orthogonal projection onto the subspace spanned by the linearly independent vectors a_1, ..., a_n is given by

$$P_A = A(A^*A)^{-1}A^* = AA^+$$

where A is the matrix consisting of the columns a_1, ..., a_n.

1.3.2. Matrices with orthonormal columns. Suppose that $n \leq m$ and that the columns of $Q \in \mathbb{C}^{m\times n}$ are orthonormal. Then Q^*Q is the $n \times n$ identity matrix and hence $P_Q = QQ^* \in \mathbb{C}^{m\times m}$ is the orthogonal projection onto the image of Q.

The following theorem is a generalization of the corresponding part of Theorem 1.1.4.

THEOREM. Suppose the columns of $Q \in \mathbb{C}^{m\times n}$ are orthonormal. If $x, y \in \mathbb{C}^n$ then $(Qx)^*(Qy) = x^*y$ and $\|Qx\| = \|x\|$. If $x, y \in \text{im}(Q) \subset \mathbb{C}^m$ then $(Q^*x)^*(Q^*y) = x^*y$ and $\|Q^*x\| = \|x\|$.

1.3.3. QR factorization. Suppose $A = \hat{Q}\hat{R}$ where $\hat{Q}$ is an $m \times n$ matrix with orthonormal columns and $\hat{R}$ is an $n \times n$ upper triangular matrix. Then $\hat{Q}\hat{R}$ is called a *reduced QR factorization* of A. Note that necessarily $m \geq n$ in this case.

If $A = QR$ where Q is an $m\times m$ matrix with orthonormal columns (i.e., a unitary matrix) and R is an $m \times n$ upper triangular matrix, then QR is called a *full QR factorization* or just a *QR factorization* of A.

If $m \geq n$ and if QR is a full QR factorization of a matrix A let $\hat{Q}$ be the matrix consisting of the first n columns of Q and $\hat{R}$ the matrix consisting of the first m rows of R. Then $\hat{Q}\hat{R}$ is a reduced QR factorization of A. Conversely, if $\hat{Q}\hat{R}$ is a reduced QR factorization of a matrix A, choose $m - n$ orthonormal vectors in $\text{im}(\hat{Q})^\perp$. Append these vectors (as

columns to $\hat{Q}$ to obtain a $m \times m$ unitary matrix Q. Let R be the matrix obtained from $\hat{R}$ by appending $m - n$ zero rows. Then QR is a full QR factorization of A.

1.3.4. Gram-Schmidt orthogonalization. Let $m \geq n$ and suppose that $A \in \mathbb{C}^{m \times n}$ has full rank. Then the image of A has an orthonormal basis which can be constructed by the Gram-Schmidt algorithm. This algorithm can be described as follows: suppose orthonormal vectors q_1, ..., q_{k-1} spanning $\langle a_1, ..., a_{k-1} \rangle$ have been constructed. Let $\hat{Q}_{k-1}$ be the matrix consisting of the columns q_1, ..., q_{k-1}. Then $\hat{Q}_{k-1}\hat{Q}_{k-1}^* = \sum_{\ell=1}^{k-1} q_\ell q_\ell^*$ is the orthogonal projection onto $\langle a_1, ..., a_{k-1} \rangle$. Define $v_k = (I - \hat{Q}_{k-1}\hat{Q}_{k-1}^*)a_k$ and $q_k = v_k / \|v_k\|$. Then q_1, ..., q_k are orthonormal vectors spanning $\langle a_1, ..., a_k \rangle$. Induction proves then that there are orthonormal vectors q_1, ..., q_n spanning $\langle a_1, ..., a_n \rangle$. Note that $a_k = \sum_{\ell=1}^{n} R_{\ell,k} q_\ell$ where

$$R_{\ell,k} = \begin{cases} q_\ell^* a_k & \text{if } \ell < k \\ \|v_k\| & \text{if } \ell = k \\ 0 & \text{if } \ell > k. \end{cases}$$

Now let $\hat{Q} = \hat{Q}_n$ and let R be the matrix formed by the numbers $R_{\ell,k}$. Then $A = \hat{Q}\hat{R}$. The following algorithm implements these steps:

ALGORITHM.
for $k = 1$ **to** n
  $v_k = a_k$
  for $\ell = 1$ **to** $k - 1$
    $r_{\ell,k} = q_\ell^* a_k$
    $v_k = v_k - r_{\ell,k} q_\ell$
  $r_{k,k} = \|v_k\|$
  $q_k = v_k / r_{k,k}$
end

The Gram-Schmidt algorithm requires no more than $2mn^2 + 3mn$ flops to compute a reduced QR factorization of an $m \times n$ matrix.

THEOREM. Let $m \geq n$ and suppose that $A \in \mathbb{C}^{m \times n}$ has full rank. Then A has a full and a reduced QR factorization.

Sketch of proof: This follows immediately from the Gram-Schmidt algorithm. □

1.3.5. Modified Gram-Schmidt algorithm. Another way to obtain the orthonormal basis is the so called modified Gram-Schmidt algorithm, which is numerically more suitable. It is based on the following fact

$$I - \hat{Q}_{k-1}\hat{Q}_{k-1}^* = \prod_{\ell=1}^{k-1} (I - q_\ell q_\ell^*).$$

This leads to the following algorithm:

ALGORITHM.
for $k = 1$ **to** n
  $v_k = a_k$
for $k = 1$ **to** n
  $r_{k,k} = \|v_k\|$
  $q_k = v_k / r_{k,k}$
  for $j = k + 1$ **to** n
    $r_{k,j} = q_k^* v_j$

$v_j = v_j - r_{k,j} q_k$

end

The modified Gram-Schmidt algorithm requires no more than $2mn^2 + 3mn$ flops to compute a reduced QR factorization of an $m \times n$ matrix.

1.3.6. Householder triangularization. The Gram-Schmidt and the modified Gram-Schmidt algorithms could be considered as methods of triangular orthogonalization. We now discuss a method which could be called orthogonal triangularization since its emphasis is on constructing the upper triangular matrix R.

R will be obtained by applying a sequence of unitary and self-adjoint matrices Q_k to A (from the left) so that $R = Q_n ... Q_1 A$ and $A = Q_1 ... Q_n R$. In step k zeros are produced below the diagonal element in column k while the zeros below the diagonal in columns 1 through $k-1$ are left untouched. To achieve this let

$$Q_k = \begin{pmatrix} I_{k-1} & 0 \\ 0 & F \end{pmatrix}$$

where F is a certain $(m-k+1) \times (m-k+1)$ unitary and self-adjoint matrix which is called a Householder reflector. Since Q_k is acting on a matrix of the type

$$A_k = \begin{pmatrix} A_{k;1,1} & A_{k;1,2} \\ 0 & A_{k,2,2} \end{pmatrix},$$

where $A_{k;1,1} \in \mathbb{C}^{(k-1)\times(k-1)}$ and $A_{k;2,2} \in \mathbb{C}^{(m-k+1)\times(n-k+1)}$, we obtain

$$Q_k A_k = \begin{pmatrix} A_{k;1,1} & A_{k;1,2} \\ 0 & F A_{k,2,2} \end{pmatrix}.$$

If x denotes the first column in $A_{k;2,2}$. If $x = 0$ let $F = I_{m+1-k}$. Otherwise F should be such that $Fx = \alpha \|x\| e_1$ where α is chosen such that $|\alpha| = 1$ and $\alpha \overline{x_1} \leq 0$. Define $v = \alpha \|x\| e_1 - x$ and let P be the orthogonal projection onto the orthogonal complement of $\langle v \rangle$, i.e.,

$$P = I - \frac{vv^*}{v^* v}$$

so that $Px = x + v/2 = Fx - v/2$. This is satisfied when $F = 2P - I$, i.e.,

$$F = I - 2\frac{vv^*}{v^* v}$$

which is indeed unitary and self-adjoint. The choice of α was made so that $\|v\|^2 = 2\|x\|^2 - 2\,\mathrm{Re}(\alpha x_1)\|x\|$ is as large as possible. Thus we arrive at the following algorithm:

ALGORITHM.
for $k = 1$ **to** n
  $x = A_{k:m,k}$
  if $x \neq 0$
    $v_k = \alpha \|x\| e_1 - x$
    $v_k = v_k / \|v_k\|$
    $A_{k:m,k:n} = A_{k:m,k:n} - 2v_k (v_k^* A_{k:m,k:n})$
end

Householder triangularization requires $\sim 2mn^2 - 2n^3/3$ flops to compute the upper triangular factor R in the QR factorization of an $m \times n$ matrix.

THEOREM. Let the QR factorization of a matrix $A \in \mathbb{C}^{m\times n}$ be computed by Householder triangularization on a computer satisfying Axiom 1.2.1 and let the computed factors be $\tilde{Q}$ and $\tilde{R}$. Then we have $\tilde{Q}\tilde{R} = A + \delta A$ where $\|\delta A\|/\|A\| \leq C_1 m\varepsilon_M$ for some number C_1. In other words QR factorization by Householder triangularization is backward stable[3].

Note that the relative errors in $\tilde{Q}$ and $\tilde{R}$ can be huge when compared to $C_1 m\varepsilon_M$ (see Trefethen and Bau, Lecture 16).

1.3.7. Computation of Q in the Householder approach. To solve the system $Ax = b$ via a QR factorization we have to consider $QRx = b$ or, equivalently, $Rx = Q^*b$. If the matrix R is determined by the Householder algorithm one has to determine the vector $Q^*b = Q_n...Q_1 b$. This is achieved by the following algorithm utilizing the vectors v_k computed by algorithm 1.3.6:

ALGORITHM.
for $k = 1$ **to** n
$b_{k:m} = b_{k:m} - 2v_k(v_k^* b_{k:m})$
end

The work involved in this algorithm is no more than $4mn$.

Incidentally, this algorithm can be used to compute Q^* and hence Q itself, by applying it to the canonical basis vectors e_1, ..., e_m.

1.3.8. Backward stability of solving equations by Householder triangularization. Let A be an invertible $m \times m$ matrix and QR its QR factorization. Let $b \in \mathbb{C}^m$ and $x = A^{-1}b$. We want to compute x by Householder triangularization and back substitution. Even though the matrices $\tilde{Q}$ and $\tilde{R}$ computed by the Householder algorithm may not be very close to the actual factors Q and R of A they may — miraculously — still be used to solve the system $Ax = b$. To see this recall first from 1.3.6 that

$$\tilde{Q}\tilde{R} = A + \delta A$$

where $\|\delta A\|/\|A\| \leq C_1\varepsilon_M$. Here C_1 (as well as C_1' and C_1'' introduced below) depends on m. Secondly recall from 1.2.7 that for given $(\tilde{R}, \beta)$ there are δR and $\delta\beta$ such that $\|\delta R\|/\|\tilde{R}\|$ and $\|\delta\beta\|/\|\beta\|$ are bounded by $C_1'\varepsilon_M$ and

$$(\tilde{R} + \delta R)^{-1}(\beta + \delta\beta) = \tilde{x}$$

where $\tilde{x}$ is the output produced by the algorithm of back substitution instead of $\tilde{R}^{-1}\beta$. Finally, note that Algorithm 1.3.7 is also a backward stable algorithm, i.e., if $(\tilde{Q}, b) \mapsto \tilde{Q}^*b$ then there is a δQ and a $\delta Q'$ such that $\|\delta Q\|/\|\tilde{Q}\|$ and $\|\delta Q'\|/\|\tilde{Q}\|$ are bounded by $C_1''\varepsilon_M$ and

$$(\tilde{Q} + \delta Q')^* b = (\tilde{Q} + \delta Q)^{-1} b = \beta$$

where β is the output produced by Algorithm 1.3.7 instead of $\tilde{Q}^*b$.

Using these results we obtain

$$b = (\tilde{Q} + \delta Q)\beta = (\tilde{Q} + \delta Q)[(\tilde{R} + \delta R)\tilde{x} - \delta\beta] = (\tilde{Q}\tilde{R} + \tilde{Q}\delta R + \delta Q\tilde{R} + \delta Q\delta R)\tilde{x} - (\tilde{Q} + \delta Q)\delta\beta.$$

Hence

$$b + \Delta b = (A + \Delta A)\tilde{x}$$

where

$$\Delta b = (\tilde{Q} + \delta Q)\delta\beta$$

[3]Actually, this is to be taken with a grain of salt since, in general, $\tilde{Q}$ is not unitary and $\tilde{R}$ is not upper triangular and hence $\tilde{Q}\tilde{R}$ is not a QR factorization of anything.

and

$$\Delta A = \delta A + \tilde{Q}\delta R + \delta Q\tilde{R} + \delta Q\delta R.$$

One can then show that $\|\Delta A\|/\|A\|$ and $\|\Delta b\|/\|b\|$ are small.

This result has now to be combined with (5) which states

$$\frac{\|\tilde{x} - x\|}{\|x\|} \leq 2\kappa(A)\left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|}\right)$$

provided $\|\Delta A\|\|A^{-1}\| \leq 1/2$.

1.4. LU Factorization

1.4.1. Gaussian elimination. Suppose $A = LU$ where L is an invertible $m \times m$ lower triangular matrix and U is an $m \times n$ upper triangular matrix. Then LU is called an LU *factorization* of A.

The LU factorization of a matrix is computed via Gaussian elimination. As in the Householder triangularization zeros will be produced below the diagonal of A. However instead of applying a sequence of unitary matrices a sequence of lower triangular matrices is applied. Let $A_0 = A$ and suppose that after step $k-1$ we have $A_{k-1} = L_{k-1}...L_1A$ where the L_j are invertible lower triangular matrices and where the entries $A_{k-1;j,\ell}$ of A_{k-1} are zero if $j > \ell$ and $\ell \leq k-1$. In step k we now want to produce zeros below the diagonal element in column k while the zeros below the diagonal in columns 1 through $k-1$ are left untouched. To achieve this let $L_{k;r,1:m} = e_r^* - \ell_{k;r}e_k^*$ for $r = 1, ..., m$ where $\ell_k \in \mathbb{C}^m$ and $\ell_{k;1} = ... = \ell_{k;k} = 0$. The r-s entry of $A_k = L_kA_{k-1}$ is

$$A_{k;r,s} = L_{k;r,1:m}A_{k-1;1:m,s} = \begin{cases} A_{k-1;r,s} & \text{if } 1 \leq r \leq k \\ A_{k-1;r,s} - \ell_{k;r}A_{k-1;k,s} & \text{if } k+1 \leq r \leq m. \end{cases}$$

Note that the elements below the diagonal in the first $k-1$ columns of A_k are zero. The condition is now that $0 = A_{k;r,k} = A_{k-1;r,k} - \ell_{k;r}A_{k-1;k,k}$ for $r = k+1, ..., m$. Hence, if $A_{k-1;k,k} \neq 0$, choosing $\ell_{k;r} = A_{k-1;r,k}/A_{k-1;k,k}$ for $r = k+1, ..., m$ gives the required result. If $A_{k-1;k,k} = 0$ and also $A_{k-1;k+1,k} = ... = A_{k-1;m,k} = 0$ we may choose $\ell_{k;r} = 0$. However, if $A_{k-1;k,k} = 0$ and $A_{k-1;j,k} \neq 0$ for some $j \in \{k+1, ..., m\}$ Gaussian elimination fails. This case is considered in 1.4.2. Assume now that this does not happen. After $\mu = \min\{m-1, n\}$ steps we have, letting $A_\mu = U$, that

$$L_\mu...L_1A = U$$

is upper triangular. Now let $L = (L_\mu...L_1)^{-1} = L_1^{-1}...L_\mu^{-1}$. Then

$$A = LU$$

but we still have to show that L is lower triangular. To this end we note firstly that each L_k has only one as an eigenvalue and is therefore invertible. In fact, letting ℓ_k to be the column consisting of the numbers $\ell_{k;1}$, ..., $\ell_{k;m}$ (the first k of which are zero) we have that $L_k = I - \ell_k e_k^*$ and hence $L_k^{-1} = I + \ell_k e_k^*$. Therefore $L_k^{-1}L_{k+1}^{-1} = (I + \ell_k e_k^*)(I + \ell_{k+1}e_{k+1}^*) = I + \ell_k e_k^* + \ell_{k+1}e_{k+1}^*$ and, employing an induction proof,

$$L = \prod_{k=1}^{\mu}(I + \ell_k e_k^*) = I + \sum_{k=1}^{\mu}\ell_k e_k^*.$$

The algorithm for LU factorization is

ALGORITHM.
$U = A$
$L = I$
for $k = 1$ **to** μ
  for $j = k+1$ **to** m
    $\ell_{k;j} = L_{j,k} = U_{j,k}/U_{k,k}$
    $U_{j,k:m} = U_{j,k:m} - L_{j,k}U_{j,k:m}$
end

The work performed this algorithm is bounded by an expression which is asymptotically given by $2m^3/3$.

1.4.2. Partial pivoting. Let $Sx = b$ where $S \in \mathbb{C}^{m\times n}$ and $b \in \mathbb{C}^m$ represent a system of linear equations. One may then consider the $m \times (n+1)$ matrix obtained by pasting b as an $(n+1)$st column to S. The resulting matrix $A = (S, b)$ is called an augmented matrix. To solve the system $Sx = b$ by an LU factorization one has actually to factor the augmented matrix (S, b): if $(S, b) = A = LU$ let $U = (\tilde{U}, \tilde{b})$ where $\tilde{U} \in \mathbb{C}^{m\times n}$ and $\tilde{b} \in \mathbb{C}^m$ so that $L\tilde{U} = S$ and $L\tilde{b} = b$. Hence $Sx = b$ if and only if $\tilde{U}x = \tilde{b}$. Since $\tilde{U}$ is upper triangular this latter system is easy to solve (cf. 1.2.7).

Suppose now that Gaussian elimination fails to produce the factorization. Hence for some k the entry $A_{k-1;k,k}$ is zero but $A_{k-1;j,k}$ is not zero for some $j_k \in \{k+1, ..., m\}$ (we are using here the notation introduced in 1.4.1). Note that interchanging rows in $A = (S, b)$ corresponds to reordering the equations making up the system and is therefore entirely irrelevant for obtaining the solution. Therefore one interchanges rows k and j_k in A_{k-1}. This is achieved by applying a unitary self-adjoint matrix P_k to A_{k-1} (in fact the matrix which achieves the exchange of rows j and k is the matrix one obtains from the identity matrix by exchanging rows j and k). Hence PA_{k-1} has the property that $(PA_{k-1})_{k,k} \neq 0$ and therefore one may proceed with Gaussian elimination. Doing this whenever necessary one obtains

$$L_\mu P_\mu ....L_1 P_1 A = U.$$

Now define $L'_m = L_m$ and $L'_j = P_\mu ...P_{j+1}L_j P_{j+1}^{-1}...P_\mu^{-1}$ for $j = 1, ..., \mu - 1$. Then we have

$$L_\mu P_\mu ....L_1 P_1 A = L'_\mu ...L'_1 P_\mu ...P_1 A = U$$

or, denoting $P_\mu ...P_1$ by P and $(L'_\mu ...L'_1)^{-1}$ by L,

$$PA = LU.$$

It turns out that L is still lower triangular. This follows from the following fact: Suppose P is the transformation which interchanges rows j and j'. Then, if $j, j' > k$ we have $Pe_k = e_k$ and hence

$$PL_k P^{-1} = I - (P\ell_k)(Pe_k)^* = I - \ell'_k e_k$$

so that L_k and PL_kP^{-1} have the same structure and the arguments of 1.4.1 still apply.

The procedure just described is called *partial pivoting*. The nonzero numbers $U_{k,k}$ are called *pivots*.

Even if Gaussian elimination does not fail it may be necessary to pivot when one does not use perfect arithmetic. Recall that one must perform division by the pivots and one wants to avoid not only dividing by zero but also dividing by small numbers. For example consider the matrix

$$A = \begin{pmatrix} 10^{-20} & 1 \\ 1 & 1 \end{pmatrix}.$$

Gaussian elimination without pivoting gives

$$A = \begin{pmatrix} 1 & 0 \\ 10^{20} & 1 \end{pmatrix} \begin{pmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{pmatrix}.$$

In double precision arithmetic the number $1 - 10^{20}$ would be replaced by 10^{20} which seems like a small error. However computing the solution of $Ax = b$ where $b = (1, 0)^*$ gives something which is approximately equal to $(1, 1)^*$ while the solution computed from the rounded factorization is $(0, 1)^*$ which gives an intolerable error.

The algorithm for Gaussian elimination with partial pivoting is

ALGORITHM.
$U = A$
$L = I$
$P = I$
for $k = 1$ **to** μ
  Select $j \in \{k, ..., m\}$ to maximize $|U_{j,k}|$
  Exchange rows j and k in U
  Exchange rows j and k in L
  Exchange rows j and k in P
  for $j = k + 1$ **to** m
    $L_{j,k} = U_{j,k}/U_{k,k}$
    $U_{j,k:m} = U_{j,k:m} - L_{j,k}U_{j,k:m}$
end

To leading order Gaussian elimination with partial pivoting takes as many operations as the one without pivoting ($\sim 2m^3/3$).

1.4.3. Complete pivoting. Recall that an exchange of rows in S would correspond to permuting (or relabeling) the independent variables and is therefore also rather unimportant if appropriate care is taken. One could therefore also perform complete pivoting by finding the element of the submatrix $S_{k:m,k:m}$ which has the largest absolute value and move it by a row exchange and a column exchange in to the k-k position. Doing this one obtains finally a factorization of the form

$$PAQ = LU$$

where P and Q are matrices whose entries are zero except that the number one occurs precisely once in each row and each column.

In practice the cost of finding the pivot in this way is too big and the method is therefore rarely used even though it does improve stability.

1.4.4. Stability of LU factorization. The stability of LU factorization without pivoting is described by the following theorem.

THEOREM. Let the LU factorization of a nonsingular matrix $A \in \mathbb{C}^{m\times m}$ be computed on a computer satisfying Axiom 1.2.1 by Gaussian elimination without pivoting. If A has an LU factorization then Algorithm 1.4.1 computes matrices $\tilde{L}$ and $\tilde{U}$ which satisfy

$$\tilde{L}\tilde{U} = A + \delta A \quad \text{and} \quad \frac{\|\delta A\|}{\|A\|} \leq \frac{\|L\|\|U\|}{\|A\|} C_1 \varepsilon_M$$

for some m-dependent constant C_1.

Hence if $\|L\|$ or $\|U\|$ is large compared to $\|A\|$ then Gaussian elimination with pivoting is not useful.

Employing partial pivoting results in a matrix L of which one can show that each of its entries has absolute value no larger than one and hence $\|L\| \leq C$ where the constant C depends only on m but not A. One then has still to compare $\|U\|$ with $\|A\|$ and one obtains the following corollary immediately from the previous theorem.

COROLLARY. Let the LU factorization of a nonsingular matrix $A \in \mathbb{C}^{m\times m}$ be computed on a computer satisfying Axiom 1.2.1 by Gaussian elimination with partial pivoting. Then Algorithm 1.4.2 computes matrices $\tilde{L}$, $\tilde{U}$, and $\tilde{P}$ which satisfy

$$\tilde{L}\tilde{U} = \tilde{P}A + \delta A \quad \text{and} \quad \frac{\|\delta A\|}{\|A\|} \leq \frac{\|U\|}{\|A\|} C_1 \varepsilon_M$$

for some m-dependent constant C_1.

While one can construct matrices for which the ratio $\|U\|/\|A\|$ becomes arbitrarily large the set of matrices where it actually happens seems to be extremely small so that in practice Gaussian elimination with partial pivoting is successfully employed. See Lecture 22 of Trefethen and Bau for more details.

1.4.5. Cholesky Factorization. A self-adjoint matrix A is called *positive definite* if x^*Ax is positive for all $x \neq 0$. This implies immediately that all eigenvalues of A are positive.

Let $n \leq m$ and let $B \in \mathbb{C}^{m\times n}$ be a matrix of full rank n. If A is positive definite then so is B^*AB. In particular the diagonal entries of A are then positive.

If there exists an upper triangular square matrix R with positive diagonal elements such that $R^*R = A$ then this factorization of A is called the *Cholesky factorization*.

THEOREM. Every positive definite matrix has a unique Cholesky factorization.

Sketch of proof: Call the matrix in question A and suppose it is an $m \times m$ matrix. If $m = 1$ then A is a positive number and $R = \sqrt{A}$. Assume that all positive definite matrices in $\mathbb{C}^{(m-1)\times(m-1)}$ have a unique Cholesky factorization. Let $\alpha = \sqrt{A_{1,1}}$ and $w = A_{2:m,1} \in \mathbb{C}^{m-1}$. Define R_1 by

$$R_1 = \begin{pmatrix} \alpha & w^*/\alpha \\ 0 & I \end{pmatrix}.$$

Then $A = R_1^* A_1 R_1$ where

$$A_1 = \begin{pmatrix} 1 & 0 \\ 0 & A_{2:m,2:m} - ww^*/A_{1,1} \end{pmatrix}.$$

Since $A_1 = R_1^{-1*} A R_1^{-1}$ is positive definite so is $A_{2:m,2:m} - ww^*/A_{1,1} \in \mathbb{C}^{(m-1)\times(m-1)}$. Hence we obtain $A_{2:m,2:m} - ww^*/A_{1,1} = \tilde{R}_2^*\tilde{R}_2$ where $\tilde{R}_2$ is upper triangular and has positive diagonal elements. Therefore

$$A_1 = \begin{pmatrix} 1 & 0 \\ 0 & \tilde{R}_2^*\tilde{R}_2 \end{pmatrix} = R_2^* R_2$$

where

$$R_2 = \begin{pmatrix} 1 & 0 \\ 0 & \tilde{R}_2 \end{pmatrix}.$$

Since $R_2 R_1$ is upper triangular this proves existence of the Cholesky factorization. If S^*S is another Cholesky of A then we have that $S_{1,1:m} = R_{1,1:m}$ and hence that $S = S_2 R_1$ where S_2 has the same form as R_2. An induction argument proves that, in fact, $S_2 = R_2$. □

The algorithm for the factorization is the following:

ALGORITHM.
$R = A$
for $k = 1$ **to** m
  for $j = k + 1$ **to** m
    $R_{j,j:m} = R_{j,j:m} - R_{k,j:m} R_{j,k}/R_{k,k}$
  $R_{k,k:m} = R_{k,k:m}/\sqrt{R_{k,k}}$
end

The algorithm takes asymptotically $m^3/3$ flops to perform. The algorithm is backward stable, i.e., the computed matrix $\tilde{R}$ satisfies

$$\tilde{R}^* \tilde{R} = A + \delta A \quad \text{and} \quad \frac{\|\delta A\|}{\|A\|} \leq C_1 \varepsilon_M.$$

To solve a system $Ax = b$ one solves successively to triangular systems: $R^* y = b$ and $Rx = y$ which requires a total work of $\sim m^3/3$ flops.

1.5. Least Squares Problems

1.5.1. Overdetermined systems of equations. Consider a system $Ax = b$ of m equations in n unknowns where $m > n$. If the rank of A is smaller than the rank of the augmented matrix (A, b) then the system has no solution since this means that b is not in the image of A. Define the vector $r(x) = b - Ax$ which is called the *residual*. If the system has no solution then the residual becomes never zero. The next best thing is then to determine x in such a way that the residual becomes as small as possible with respect to some norm. Choosing the 2-norm the problem is to find x such that $\|b - Ax\|_2$ becomes minimal. Since the 2-norm corresponds to Euclidean distance the geometric interpretation of this is to find x such that Ax is the point in $\mathrm{im}(A)$ which is closest to b. From this geometric picture it is clear that we have to choose x such that r is perpendicular to $\mathrm{im}(A)$. In fact, we have the following theorem

THEOREM. Let $n \leq m$, $A \in \mathbb{C}^{m \times n}$, and $b \in \mathbb{C}^m$. A vector x_0 satisfies

$$\|r(x_0)\|_2 = \|b - Ax_0\|_2 = \min\{\|b - Ax\|_2 : x \in \mathbb{C}^n\}$$

if and only if $r(x_0)$ is perpendicular to $\mathrm{im}(A)$. The minimizer x_0 is unique if and only if A has full rank n. The condition $r(x_0) \perp \mathrm{im}(A)$ is equivalent to any of the equations $A^* r(x_0) = 0$, $A^* A x_0 = A^* b$, or $Pb = Ax_0$ where P is the orthogonal projection onto $\mathrm{im}(A)$.

Sketch of proof: Abbreviate $r(x_0)$ by r. Then we have $0 = r^* P x = (Pr)^* x = (Pb - Ax_0)^* x$ for all $x \in \mathbb{C}^n$ if and only if $Pb = Ax_0$. Next denote the columns of A by a_1, ..., a_n. Then we have $r \perp \mathrm{im}(A)$ if and only if $a_j^* r = 0$ for all $j = 1, ..., n$. But the latter condition is equivalent to $0 = A^* r = A^*(b - Ax_0) = A^* b - A^* A x_0$. This proves the mentioned equivalences.

Now let r be perpendicular to $\mathrm{im}(A)$ and let x be any element in $\mathbb{C}^n$. Then $r \perp A(x_0 - x)$ and by the Pythagorean theorem

$$\|b - Ax\|_2 = \|r + A(x_0 - x)\|_2 = \|r\|_2 + \|A(x_0 - x)\|_2 \geq \|r\|_2.$$

Conversely if $\|r\| = \|b - Ax_0\|$ minimizes $\|b - Ax\|$ let $Pr = Ax_1$. Then

$$\|r - Ax_1\|^2 = \|r\|^2 - (r, Ax_1) - (Ax_1, r) + \|Ax_1\|^2 = \|r\|^2 - \|Ax_1\|^2 < \|r\|^2$$

unless $Pr = 0$.

To prove the uniqueness statement recall first that A has full rank if and only if $A^* A$ is invertible. Suppose $A^* A x_0 = A^* b = A^* A x_1$, i.e., that both x_0 and x_1 are minimizers of $\|b - Ax\|_2$. Then we have that $x_0 - x_1 \in \ker(A^* A)$. □

If A has full rank then A^*A is a positive definite $n \times n$ matrix and hence has a Cholesky factorization R^*R. To find the minimizer one has therefore to find the factorization and to solve the equation $R^*Rx = A^*b$ (by solving two triangular systems).

Alternatively one can use the reduced QR factorization of A. If $A = \hat{Q}\hat{R}$ then $P = \hat{Q}\hat{Q}^*$ and hence $Pb = Ax_0$ is equivalent to the triangular system $Rx_0 = \hat{Q}^*b$.

Finally, the reduced SVD decomposition of A may also be employed: If $A = \hat{U}\hat{\Sigma}V^*$ then $P = \hat{U}\hat{U}^*$ and hence $Pb = Ax_0$ is equivalent to $\hat{\Sigma}V^*x_0 = \hat{U}^*b$. Hence solve $\hat{\Sigma}y = \hat{U}^*b$ and set $x = Vy$.

1.5.2. Least squares curve fitting. What is the best way to approximate three points in the plane by a straight line. One may argue that one wants to find that line for which the sum of the square of the distances of the points from the line is minimal. This method (and its obvious generalizations) is called *least squares data fitting* and was invented around 1800 by Gauss and Legendre.

Suppose m points (x_j, y_j) in the plane are given and we want to find a polynomial curve passing through or approximating these points. If the polynomial has degree $n-1$ and is given by

$$p(x) = \sum_{k=0}^{n-1} a_k x^k$$

the problem is represented by the following system of equations

$$\begin{pmatrix} 1 & x_1 & ... & x_1^{n-1} \\ \vdots & & & \vdots \\ 1 & x_m & ... & x_m^{n-1} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_{n-1} \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix}$$

which we abbreviate by $Va = y$. A matrix of the form appearing on the left is called a Vandermonde matrix and one can show that it has rank n if $n \leq m$.

If $n = m$ this problem has a unique solution, i.e., the polynomial passes through all m of the given points. In practice this may give however unsatisfactory answers and one really desires a lower order curve (e.g., a straight line) passing nearby the points. To find it one minimizes, according to the least squares concept, the quantity

$$\sum_{j=0}^{m} |y_j - p(x_j)|^2 = \|y - Va\|_2^2,$$

i.e., one finds that point in $\mathrm{im}(V) \subset \mathbb{C}^m$ closest to $(y_1, ..., y_m)^t$.

Note that here polynomials are thought of as linear combination of power functions. One can choose any other class of functions, consider their linear combinations, and obtain a problem in linear algebra. One such class widely used in applications are the so called splines, which are piecewise polynomials spliced at their endpoints so that they have a certain amount of differentiability.

1.6. Eigenvalues

1.6.1. Eigenvalue revealing factorizations. We know from the Jordan decomposition theorem that for every square matrix A there is an invertible matrix T such that $A = T^{-1}JT$ where J is in Jordan normal form, i.e., all entries of J are zero except the diagonal entries (which are the eigenvalues repeated according to their algebraic multiplicity) and the entries in the superdiagonal (which can be zero or one). More precisely J is a block diagonal matrix whose blocks B, which are called *Jordan blocks*, have the following

form: $B = \lambda I + S$ where the S are nilpotent matrices whose entries satisfy $S_{j,k} = \delta_{j+1,k}$. Recall that A and J have the same characteristic polynomial and that λ is an eigenvalue of A of geometric multiplicity m if and only it is an eigenvalue of J with the same geometric multiplicity.

Suppose $J = \mathrm{diag}(B_1, ..., B_r)$ where, for $j = 1, ...r$, the matrix B_j is an $d_j \times d_j$ Jordan blocks with eigenvalue λ_j. The geometric multiplicity of an eigenvalue λ of A is given by the number of blocks which have it as an eigenvalue. The algebraic multiplicity of λ is the sum of the dimensions d_j of all blocks which have λ as an eigenvalue. The index of an eigenvalue is the dimension of the largest of these blocks. Recall that the characteristic polynomial of A is given by

$$\prod_j (\mu - \lambda_j)^{m_j}$$

where the m_j are the algebraic multiplicities of the λ_j. The minimal polynomial of A is given by

$$\prod_j (\mu - \lambda_j)^{\nu_j}$$

where the ν_j are the indices of the λ_j. The algebraic and geometric multiplicities of λ coincide precisely when the index of λ is equal to one.

An eigenvalue whose index is larger than one is called *defective* since there are not enough eigenvectors to span its algebraic eigenspace. A matrix is called *defective matrix* if it has a defective eigenvalue. A matrix is not defective if and only if all of its Jordan blocks are one-dimensional, i.e., if it is a diagonal matrix. Matrices which are not defective are therefore called *diagonalizable*. The columns of T are the eigenvectors of A (in general the columns of T are the eigenvectors and generalized eigenvectors). If the eigenvectors are pairwise orthogonal T can be chosen to be unitary. In this case A is called unitarily diagonalizable.

THEOREM. A matrix is unitarily diagonalizable if and only if it is normal.

1.6.2. Schur factorization. Let $A \in \mathbb{C}^{m\times m}$. If $A = QTQ^*$ where Q is unitary and T is upper triangular is called a *Schur factorization* of A.

THEOREM. Every square matrix has a Schur factorization.

Sketch of proof: Let $A \in \mathbb{C}^{m\times m}$. The theorem holds when $m = 1$. Suppose $m > 1$ and that it holds for $m - 1$. Let q_1 be any normalized eigenvector of A associated with some eigenvalue λ and Q_1 a unitary matrix whose first column is q_1. Then

$$Q_1^* A Q_1 = \begin{pmatrix} \lambda & w^* \\ 0 & Q_2 T_2 Q_2^* \end{pmatrix}$$

for some $w \in \mathbb{C}^{m-1}$, $Q_2, T_2 \in \mathbb{C}^{(m-1)\times(m-1)}$, Q_2 unitary, and T_2 upper triangular. Now let

$$Q = Q_1 \begin{pmatrix} 1 & 0 \\ 0 & Q_2 \end{pmatrix}$$

to obtain

$$T = \begin{pmatrix} \lambda & w^* Q_2 \\ 0 & T_2 \end{pmatrix}$$

and $A = QTQ^*$.

1.6.3. The Rayleigh quotient. Let $A \in \mathbb{C}^{m\times m}$ and $0 \neq x \in \mathbb{C}^m$. The quantity

$$\frac{x^*Ax}{x^*x}$$

is called a *Rayleigh quotient*. Since $x^*Ax/(x^*x) = (\alpha x)^*A(\alpha x)/((\alpha x)^*(\alpha x))$ for all scalars $\alpha \neq 0$, the Rayleigh quotient may be considered as a function on the unit vectors (on the unit sphere).

THEOREM. Let $A \in \mathbb{C}^{m\times m}$ and x a unit vector in $\mathbb{C}^m$. Then

$$\|Ax - x(x^*Ax)\|_2 = \min\{\|Ax - \mu x\|_2 : \mu \in \mathbb{C}\}.$$

That is, the vector $x(x^*Ax)$ is the orthogonal projection of the vector Ax on the line spanned by x.

Sketch of proof: Consider the equation $x\mu = Ax$ has a least squares problem where x is the given matrix, Ax the given vector, and μ the unknown variable. From Theorem 1.5.1 we know that the minimizer satisfies $PAx = x\mu$ where P is the orthogonal projection onto $\langle x\rangle$, i.e., $P = xx^*$. Hence $\mu = x^*PAx = x^*Ax$. □

Now let x be an eigenvector of A of length one corresponding to an eigenvalue λ. Let y be another vector of length one and define $\rho = y^*Ay$. Then $\lambda - \rho = x^*A(x-y) + (x-y)^*Ay$ and hence, using the triangle inequality and Schwarz's inequality,

$$|\lambda - \rho| \leq 2\|A\|_2 \ \|x - y\|_2.$$

If A is self-adjoint one may even prove that

$$|\lambda - \rho| \leq C(A)\|x - y\|_2^2$$

for some positive constant $C(A)$.

The set

$$\Sigma = \{x^*Ax : \|x\| = 1\}$$

is called the *numerical range* of A. If A is self-adjoint and if λ_1 and λ_m are the smallest and largest eigenvalue of A respectively, then

$$\lambda_1 = \min(\Sigma) \quad \text{and} \quad \lambda_m = \max(\Sigma).$$

1.6.4. The minimax theorem. The minimax theorem gives a characterization of all eigenvalues of self-adjoint matrix:

THEOREM. Let $A \in \mathbb{C}^{m\times m}$ be self-adjoint with eigenvalues $\lambda_1 \leq ... \leq \lambda_m$. Then, for every $j = 1, ..., m$,

$$\lambda_j = \min\{\max\{\frac{x^*Ax}{x^*x} : 0 \neq x \in L\} : \dim(L) = j\}.$$

Sketch of proof: Let $\{u_1, ..., u_m\}$ be an orthonormal basis of eigenvectors of A corresponding respectively to the eigenvalues $\lambda_1, ..., \lambda_m$. Suppose L is a subspace of $\mathbb{C}^m$ of dimension j. Then there is a nonzero vector $x \in L$ orthogonal to $\langle u_1, ..., u_{j-1}\rangle$ since otherwise the dimension of $L + \langle u_1, ..., u_{j-1}\rangle \perp$ would be $m + 1$. Hence $x = \sum_{k=j}^m c_k u_k$ so that

$$\frac{x^*Ax}{x^*x} = \frac{\sum_{k=j}^m |c_k|^2\lambda_k}{\sum_{k=j}^m |c_k|^2} \geq \lambda_j.$$

Therefore,

$$\max\{\frac{x^*Ax}{x^*x} : 0 \neq x \in L\} \geq \lambda_j.$$

Now let $L = \langle u_1, ..., u_j \rangle$. Obviously $\dim(L) = j$ and

$$\frac{x^*Ax}{x^*x} = \frac{\sum_{k=1}^{j} |c_k|^2 \lambda_k}{\sum_{k=1}^{j} |c_k|^2} \leq \lambda_j$$

for every nonzero vector $x \in L$. □

1.6.5. Left eigenvectors. A row vector y^* is called a *left eigenvector* of A associated with λ if $y^*A = \lambda y^*$. Note that this is equivalent to the condition $A^*y = \overline{\lambda}y$. Since column rank and row rank of a matrix are equal we have that $A - \lambda I$ is invertible if and only if $A^* - \overline{\lambda}I$ is invertible. In other words $\overline{\lambda}$ is an eigenvalue of A^* if and only if λ is an eigenvalue of A. Also their algebraic and geometric multiplicities are respectively the same. Therefore it is not necessary to introduce the notion of left eigenvalue.

THEOREM. Let $A \in \mathbb{C}^{m \times m}$. Then the following two statements hold:
(1) If $Ax = \lambda x$, $y^*A = \mu y^*$, and $\lambda \neq \mu$ then $y^*x = 0$.
(2) If λ is a simple eigenvalue of A with eigenvector x and left eigenvector y^* then $y^*x \neq 0$.

Sketch of proof: The equalities $\mu y^*x = y^*Ax = \lambda y^*x$ prove (1).

To prove (2), assume that $A = QRQ^*$ is a Schur factorization of A where q_1, the first column of Q is an eigenvector of A associated with λ so that

$$R = \begin{pmatrix} \lambda & h^* \\ 0 & R_1 \end{pmatrix}$$

for some $h \in \mathbb{C}^{m-1}$ and some upper triangular matrix R'. Define $w = Q^*y = (w_1, w_2)^t$ where $w_1 \in \mathbb{C}$ and $w_2 \in \mathbb{C}^{m-1}$. Then $y^*A = \lambda y^*$ implies

$$(\overline{\lambda} - R_1^*)w_2 = w_1 h. \tag{6}$$

Since $\overline{\lambda}$ is a simple eigenvalue of R and since equation (6) has a nontrivial solution we must have $w_1 \neq 0$. The proof is finished after noticing that $y^*x = w^*Q^*Qe_1 = \overline{w_1}$. □

1.6.6. Perturbations of eigenvalues. Let Ω be a domain in the complex plane. A function $a : \Omega \to \mathbb{C}$ is called an algebraic function if there exists a polynomial $c_0 w^n + ... + c_n$ with coefficients $c_j \in \mathbb{C}[z]$ such that

$$c_0(z)a(z)^n + ... + c_n(z) = 0$$

for every $z \in \Omega$. It is always possible to give an explicit formula for the function a involving root extractions if $n \leq 4$. However, it is a famous theorem of Ruffini and Abel that this is not always possible if $n > 4$.

Let $A_0, A_1 \in \mathbb{C}^{m \times m}$ and consider the matrix $A(\mu) = A_0 + \mu A_1$ where μ is a complex parameter. It is then obvious that the eigenvalues of A are algebraic functions of μ. Also the eigenvectors of A are algebraic functions of μ since these form a field. This implies that the eigenvalues are and that the eigenvectors may be considered as continuous functions of μ.

If λ_0 is a simple eigenvalue of A_0 with associated eigenvector x_0 and if $\Omega = B(0, r)$ is a disk centered a zero of sufficiently small radius r, then there are differentiable (i.e., analytic) functions $\lambda : \Omega \to \mathbb{C}$ and $x : \Omega \to \mathbb{C}^m$ such that $A(\mu)x(\mu) = \lambda(\mu)x(\mu)$, $x(0) = x_0$, and $\lambda(0) = \lambda_0$.

THEOREM. Let λ_0 be a simple eigenvalue of A_0 and x_0 and y_0^* the associated normalized eigenvectors and normalized left eigenvector of A_0, respectively. Let $\lambda(\mu)$ denote the (simple)

eigenvalue of $A_0 + \mu A_1$ when λ is sufficiently small. Then

$$|\lambda'(0)| \leq \frac{\|A_1\|}{|y_0^* x_0|}.$$

Sketch of proof: Differentiate the equation

$$(A_0 + \mu A_1)x(\mu) = \lambda(\mu)x(\mu)$$

with respect to μ and set $\mu = 0$ to get

$$A_0 x'(0) + A_1 x(0) = \lambda'(0)x(0) + \lambda(0)x'(0).$$

Multiplying this equation by y_0^* on the left and using that $x(0) = x_0$ and that $y^* x \neq 0$ gives

$$\lambda'(0) = \frac{y_0^* A_1 x_0}{y_0^* x_0}$$

which immediately implies the desired result. □

The proof of the previous theorem and the Taylor expansion theorem show actually that

$$\lambda(\mu) = \lambda_0 + \frac{y_0^* A_1 x_0}{y_0^* x_0}\mu + O(\mu^2)$$

which is useful if A_1 itself rather than just $\|A_1\|$ is explicitly known.

1.6.7. The condition number of an eigenvalue. Let $\lambda \neq 0$ be a simple eigenvalue of the matrix A. Theorems 1.2.8 and 1.6.6 imply that

$$\lim_{\varepsilon \to 0} \kappa(\lambda, A, \varepsilon) = \frac{\|A\|}{|\lambda|} \frac{1}{|y^* x|}$$

where x is a normalized eigenvector and y^* a normalized left eigenvector of A associated with λ. This number is denoted by $\kappa(\lambda, A)$ and is called the relative condition number of λ.

Note that $|y^* x|$ is never larger than one. If A is normal and λ is simple then any eigenvector is also a left eigenvector and hence $|y^* x| = 1$.

1.6.8. Power iteration. Eigenvalues are the roots of the characteristic polynomial. However, we saw in 1.2.12 that in general the problem of computing roots is ill conditioned. The problem is related to the fact that, for polynomials of degree higher than four, there is no formula generalizing the quadratic formula. In practice eigenvalues are therefore computed iteratively.

Let A be an $m \times m$ matrix with algebraically simple eigenvalues $\lambda_1, ..., \lambda_m$. Let $u_1, ..., u_m$ be the associated eigenvectors. Suppose that there is a number ρ such that $|\lambda_j|/|\lambda_1| \leq \rho < 1$ for $j = 2, ..., m$. Pick any vector $x_0 = c_1 u_1 + ... + c_m u_m$ and define

$$x_k = \frac{A^k x_0}{\sigma_k}$$

for certain numbers σ_k. Then

$$x_k = \frac{1}{\sigma_k} \sum_{j=1}^{m} c_j \lambda_j^k u_j = \frac{\lambda_1^k}{\sigma_k} \left(c_1 u_1 + \sum_{j=2}^{m} c_j \left(\frac{\lambda_j}{\lambda_1} \right)^k u_j \right).$$

The resulting sequence x_k does perhaps not converge. However, if $c_1 \neq 0$, one finds that

$$\|x_k^{\perp}\|_p \leq \left| \frac{\lambda_1^k}{\sigma_k} \right| \rho^k \sum_{j=2}^{m} |c_j| \ \|u_j - \frac{u_1^* u_j}{u_1^* u_1} u_1\|_p$$

where $x_k^\perp = x_k - u_1(u_1^* x_k)/\|u_1\|^2$ is the component of x_k which is orthogonal to u_1. Choosing σ_k such that $\|x_k\|_p = 1$ gives

$$\left|\frac{\lambda_1^k}{\sigma_k}\right| \leq \frac{1}{\|c_1 u_1\|_p - \rho^k \sum_{j=2}^m \|c_j u_j\|_p} \leq \frac{2}{\|c_1 u_1\|_p}$$

for sufficiently large k. In this case we have therefore that $\|x_k^\perp\|_p$ approaches zero as k tends to infinity.

If we choose $p = \infty$, if exactly one component, say the first, of u_1 has absolute value one and if σ_k is chosen to be that component of $A^k x_0$ which has the largest absolute value then eventually $\|x_k\|_\infty = x_{k;1} = 1$.

$$\lim_{k\to\infty} \frac{\sigma_k}{\sigma_{k-1}} = \lambda_1$$

and

$$\lim_{k\to\infty} x_k = \alpha u_1$$

where $\alpha = 1/u_{1;1}$.

1.6.9. The inverse power method. Suppose A is an $m \times m$ matrix and that μ is not an eigenvalue of A. Note that x is an eigenvector of A associated with the eigenvalue λ if and only if x is an eigenvector of $(A - \mu I)^{-1}$ associated with the eigenvalue $1/(\lambda - \mu)$. Also, if λ is a simple eigenvalue and μ is properly chosen $|1/(\lambda - \mu)|$ is much larger than $|1/(\lambda' - \mu)|$ whenever λ' is another eigenvalue of A. Combining this observation with the power method gives then an algorithm which rapidly produces approximate eigenvectors. The approximate eigenvalue can then be computed as a Rayleigh quotient.

ALGORITHM.
Choose a vector v with norm 1
for $k = 1$ **to** ...
  Solve $(A - \mu)w = v$
  $v = w/\|w\|$
  $\lambda = v^* A v$
end

1.6.10. Rayleigh quotient iteration. The drawback of the Algorithm 1.6.9 is that in order to make a sensible choice for μ one has to have some a priori knowledge about the eigenvalues. If that is not given one may interlace the computation of approximate eigenvector and approximate eigenvalue (and use one for the computation of the other):

ALGORITHM.
Choose a vector v with norm 1
$\lambda = v^* A v$
for $k = 1$ **to** ...
  Solve $(A - \lambda)w = v$
  $v = w/\|w\|$
  $\lambda = v^* A v$
end

1.6.11. Simultaneous power iteration. Suppose $A \in \mathbb{C}^{m\times m}$ has m linearly independent eigenvectors u_1, ..., u_m associated with the eigenvalues λ_1, ... λ_m which satisfy

$$|\lambda_1| \geq |\lambda_2| \geq ... \geq |\lambda_n| > |\lambda_{n+1}| \geq |\lambda_{n+2}| \geq ... \geq |\lambda_m|$$

for some $n \in \{1, ..., m-1\}$. Assume that S_n is an n-dimensional subspace of $\mathbb{C}^m$ such that $S_n \cap U_n = \{0\}$ where $U_n = \langle u_{n+1}, ..., u_m\rangle$. Let P_n be the orthogonal projection onto $\langle u_1, ..., u_n\rangle^\perp$. Then one may show that the spaces $A^k S_n = \{A^k x : x \in S_n\}$ are n-dimensional and that there is a constant C_n such that

$$\max\{\|P_n A^k x\| : x \in S_n, \|A^k x\| = 1\} \leq C_n \left|\frac{\lambda_{n+1}}{\lambda_n}\right|^k. \tag{7}$$

Moreover, if s_1, ..., s_n is a basis of S_n then $A^k s_1$, ..., $A^k s_n$ is a basis of $A^k S_n$. But $A^k s_1$, ..., $A^k s_n$ is usually not an orthonormal basis even if s_1, ..., s_n are orthonormal. An orthonormal basis for $A^k S_n$ is therefore computed by the following recursive scheme: Let $s_1^{(0)} = s_1$, ..., $s_n^{(0)} = s_n$ (assuming these are orthonormal), compute the basis $As_1^{(\ell)}$, ..., $As_n^{(\ell)}$ of $A^{\ell+1}$, and orthonormalize these vectors to obtain $s_1^{(\ell+1)}$, ..., $s_n^{(\ell+1)}$.

When $n = 1$ we obtain the method of 1.6.9.

ALGORITHM.
Pick $\underline{\hat{S}} \in \mathbb{C}^{m\times n}$ with orthonormal columns
for $k = 1$ **to** ...
  $Z = A\underline{\hat{S}}$
  Let $\underline{\hat{S}}$ be the left factor of the reduced QR factorization of Z
end

1.6.12. The basic QR algorithm. The QR algorithm is the most widely used algorithm to compute the complete set of eigenvalues and eigenvectors of a matrix. The *QR algorithm* is defined as follows:

ALGORITHM.
$A_0 = A$
for $k = 1$ **to** ...
  Find the QR factorization of A_{k-1}, i.e., $A_{k-1} = Q_k R_k$
  Let $A_k = R_k Q_k$
end

One iteration of this algorithm is called a QR step or QR iteration.

Two observations are important.

(1) The matrices A_k produced in this way are unitarily equivalent since

$$A_k = Q_k^* A_{k-1} Q_k = Q_k^* ... Q_1^* A Q_1 ... Q_k = (Q_1 ... Q_k)^* A Q_1 ... Q_k.$$

(2) The k-th power of A is given by

$$A^k = Q_1 ... Q_k R_k ... R_1.$$

1.6.13. The QR algorithm and simultaneous iteration. The matrices R_k and Q_k produced by the QR algorithm for a matrix A can be found by simultaneous iteration starting from $S_0 = I$:

ALGORITHM.
$S_0 = I$
for $k = 1$ **to** ...
  $Z_k = AS_{k-1}$

$S_k\tilde{R}_k = Z_k$ (a full QR factorization of Z_k)
$Q_k = S_{k-1}^* S_k$
$R_k = \tilde{R}_k$
end

This is proved by induction on k. It is obvious for $k = 1$. Assume it is true for $k > 1$. Then

$$S_k Q_{k+1} R_{k+1} ... R_1 = A^{k+1} = AA^k = AS_k R_k ... R_1 = Z_{k+1} R_k ... R_1 = S_{k+1}\tilde{R}_{k+1} R_k ... R_1.$$

Upon choosing the factorization of Z_{k+1} appropriately we obtain that $\tilde{R}_{k+1} = R_{k+1}$ and that $S_k Q_{k+1} = S_{k+1}$.

1.6.14. Invariant subspaces. Recall that a subspace S is called *invariant* under the linear transformation A if $AS \subset S$.

THEOREM. Let $S \subset \mathbb{C}^m$ be invariant under $A \in \mathbb{C}^{m\times m}$. Suppose x_1, ..., x_n is a basis of S and x_1, ..., x_m is a basis of $\mathbb{C}^m$. Let X_1 denote the $m \times n$ matrix whose columns are x_1, ..., x_n and let X denote the $m \times m$ matrix whose columns are x_1, ..., x_m. Then $B = X^{-1}AX$ is block upper triangular, i.e.,

$$B = \begin{pmatrix} B_{1,1} & B_{1,2} \\ 0 & B_{2,2} \end{pmatrix}$$

where $B_{1,1} \in \mathbb{C}^{n\times n}$, etc.

Sketch of proof: The equation $B = X^{-1}AX$ is equivalent to $AX = XB$. Hence $Ax_j = \sum_{r=1}^m x_r B_{r,j}$. If $j \leq n$ then Ax_j is a linear combination of x_1, ..., x_n since S is invariant. This implies that $B_{r,j} = 0$ for all $r \in \{n + 1, ..., m\}$. □

1.6.15. Convergence of the QR algorithm.

THEOREM. Let λ_1, ..., λ_m be the eigenvalues of A satisfying

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_m|$$

and let u_1, ..., u_m be the respectively associated linearly independent eigenvectors. Assume that, for $n = 1, ..., m - 1$, $\langle e_1, ..., e_n\rangle \cap \langle u_{n+1}, ..., u_m\rangle = \{0\}$, i.e., the matrix U whose j-th column is u_j satisfies $\det(U_{1:n,1:n}) \neq 0$ for $n = 1, ..., m - 1$. Let A_k be the sequence of matrices unitarily equivalent to A produced by the QR algorithm. Then the subdiagonal entries of A_k converges to zero while the diagonal entries converge to the eigenvalues of A.

This theorem will not be proven here. Instead we give a heuristic concerning the convergence of the subdiagonal entries of the A_k. Sketch of proof: Suppose that, for $n = 1, ..., m-1$,

$$\langle Q_{k;1:m,1}, ..., Q_{k;1:m,n}\rangle = \langle u_1, ..., u_n\rangle$$

then A_k would be upper triangular by Theorem 1.6.14, since the lower left $(m - n) \times n$ block of A_k would be zero for $n = 1, ..., m - 1$. In actuality this may not be true but in any case the conditions of the theorem allow the application of the inequality (7) which in turn shows that entries of A_k tend to zero as k tends to infinity.

1.6.16. Hessenberg form. A matrix $A \in \mathbb{C}^{m\times m}$ is said to be in *Hessenberg form* if $A_{j,k} = 0$ for all $j > k + 1$, i.e., if it is zero below the subdiagonal.

It was shown earlier that every matrix $A \in \mathbb{C}^{m\times m}$ is unitarily equivalent to an upper triangular matrix, i.e., $A = QUQ^*$ where Q is unitary and U is upper triangular. Of course, U is also in Hessenberg form. However, to construct U one needs to know the eigenvectors of A. We will show now that a matrix H in Hessenberg form can be constructed which is unitarily equivalent to A without any knowledge of the eigenvectors of A. This will be done by induction on the size of A. The statement is obviously true when $m = 1$ (or

$m = 2$). Let now $m > 1$ and let $x = A_{2:m,1}$, $v = \alpha\|x\|e_1 - x$ for some α of modulus one, $F = I - 2vv^*/(v^*v)$ (a Householder reflector), and

$$Q_1 = \begin{pmatrix} 1 & 0 \\ 0 & F \end{pmatrix}.$$

Then Q_1 is unitary and self-adjoint. Since, by the very definition of F, $Fx = \alpha\|x\|e_1$, we find

$$Q_1 A Q_1 = \begin{pmatrix} A_{1,1} & A_{1,2:m}F \\ \alpha\|x\|e_1 & A_1 \end{pmatrix}$$

where $A_1 = FA_{2:m,2:m}F \in \mathbb{C}^{(m-1)\times(m-1)}$. To perform this step we need to know only the entries of A. By the induction hypothesis we know that we can construct a Hessenberg matrix H_2 and a unitary matrix Q_2 satisfying $A_1 = Q_2^* H_2 Q_2$ just from knowing the entries of A_1. Now let

$$Q = Q_1 \begin{pmatrix} 1 & 0 \\ 0 & Q_2 \end{pmatrix}.$$

Then Q is unitary and Q^*AQ is in Hessenberg form.

Recall that one chooses α so that $\alpha x_1 \leq 0$ in order to improve stability when implementing the algorithm below (in perfect arithmetic the value of α is irrelevant as long as $|\alpha| = 1$).

ALGORITHM.
for $k = 1$ **to** $m - 2$
$x = A_{k+1:m,k}$
Determine α
$v_k = \alpha\|x\|e_1 - x$
$v_k = v_k/\|v_k\|$
$A_{k+1:m,k:m} = A_{k+1:m,k:m} - 2v_k(v_k^* A_{k+1:m,k:m})$
$A_{1:m,k+1:m} = A_{1:m,k+1:m} - 2(A_{1:m,k+1:m}v_k)v_k^*$
end

This algorithm requires $\sim 10m^3/3$ flops. When A is self adjoint $H_{j,k}$ will be zero if $|j - k| \geq 2$. By taking this into account in the algorithm we can reduce the operation count to $\sim 4m^3/3$ flops.

1.6.17. Applying the Hessenberg transformation. The most important property of the Hessenberg transformation is that it is invariant under the QR algorithm if A_0 is invertible.

THEOREM. If the matrix A_0 is Hessenberg form and invertible, then all the matrices A_k generated by the QR algorithm are also in Hessenberg form.

Sketch of proof: Suppose that A_{k-1} is in Hessenberg form. Then $Q_k = A_{k-1}R_k^{-1}$ and $A_k = R_kQ_k$ are also in Hessenberg form. □

Each step of the QR algorithm requires $O(m^3)$ flops when A_k is a full matrix. However, when A_k is in Hessenberg form only $O(m^2)$ flops are needed. This explains the importance of reducing A first to Hessenberg form.

1.6.18. The QR algorithm with shift. An improvement on the convergence of the QR algorithm can be achieved by introducing so called shifts at each QR step. The QR algorithm with shift works as follows:

ALGORITHM.
Let A_0 be the Hessenberg transform of A
for $k = 1$ **to** ...
$A_{k-1} = A_{k-1} - \rho_{k-1} I$
Find the QR factorization of A_{k-1}, i.e., $A_{k-1} = Q_k R_k$
Let $A_k = R_k Q_k + \rho_{k-1} I$
end

In this algorithm the numbers ρ_k have to be chosen appropriately. A standard choice is the Rayleigh quotient of the vector e_m, i.e.,

$$\rho_k = e_m^* A_k e_m = A_{k;m,m}$$

(regarding e_m as the approximate eigenvector corresponding to λ_m). This is called the *Rayleigh quotient shift*. Experience shows that this choice of the shift works quite well to approximate the zero in row m and column $m-1$ and the eigenvalue λ_m in row m and column m.

However, the other subdiagonal entries, $A_{k,j+1,j}$, $1 \le j \le m-2$, approach zero slowly. To speed things up, one uses the following trick. After making $A_{k;m,m-1}$ is practically equal to zero and $A_{k;m,m}$ is practically equal to λ_m, one partitions the matrix A_k as

$$A_k = \begin{pmatrix} \hat{A}_k & b_k \\ 0 & \lambda_m \end{pmatrix}$$

where $\hat{A}_k$ is an $(m-1) \times (m-1)$ Hessenberg matrix, whose eigenvalues are (obviously) λ_1, ..., λ_{m-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_{m-1}, which can be split off as above, etc. This procedure is called the *deflation* of the matrix A.

In practice, each eigenvalue of A requires 3-5 iterations (QR steps), on the average. The algorithm is rather fast and very accurate.

1.6.19. Computing the SVD. The singular and singular vectors of a matrix $A \in \mathbb{C}^{m \times n}$ (where $m \ge n$) are given as the roots of the eigenvalues and the eigenvectors of the $n \times n$ matrix $A^* A$.

However, it is not a good idea to use this procedure in the presence of imperfect arithmetic. A backward stable algorithm for computation of eigenvalues would compute approximations $\tilde{\lambda}_k$ satisfying

$$|\tilde{\lambda}_k - \lambda_k| \le C \varepsilon_M \|A^* A\| \le C' \varepsilon_M \|A\|^2.$$

Since $\tilde{\lambda}_k - \lambda_k = (\tilde{\sigma}_k - \sigma_k)(\tilde{\sigma}_k + \sigma_k)$ one obtains

$$\frac{|\tilde{\sigma}_k - \sigma_k|}{\sigma_k} \le C'' \varepsilon_M \frac{\|A\|^2}{\sigma_k^2} = C'' \varepsilon_M \frac{\sigma_1^2}{\sigma_k^2}.$$

For singular values much smaller than σ_1 this gives a very large error bound. For numerical purposes one proceeds in a different way to compute singular values and singular vectors.

First one applies alternately a sequence of Householder reflectors to the left and right of A so that in the end one arrives at a matrix B which is bidiagonal, i.e., all entries outside its diagonal and its superdiagonal are zero. More precisely

$$B = U_n ... U_1 A V_1 ... V_{n-2}$$

(recall that Householder reflectors are both unitary and self-adjoint). This process is called *Golub-Kahan bidiagonalization*. Note that $B^* B$ and $A^* A$ are unitarily equivalent which

implies that the singular values of B are precisely those of A. Further note that

$$B = \begin{pmatrix} B_1 \\ 0 \end{pmatrix}$$

where B_1 is an $n \times n$ matrix and where $B_1^* B_1 = B^* B$. Hence in order to find the singular values of A we may look for the singular values of the square matrix B_1.

Next one forms the self-adjoint matrix

$$H = \begin{pmatrix} 0 & B_1^* \\ B_1 & 0 \end{pmatrix}$$

and note that the eigenvalue decomposition of H is

$$\begin{pmatrix} 0 & B_1^* \\ B_1 & 0 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} V & V \\ U & -U \end{pmatrix} \begin{pmatrix} \Sigma & 0 \\ 0 & -\Sigma \end{pmatrix} \begin{pmatrix} V^* & U^* \\ V^* & -U^* \end{pmatrix}$$

when $B_1 = U\Sigma V^*$ is the SVD of B_1. In other words, the singular of A (or of B_1) are the nonnegative eigenvalues of H. Since $\|H\| = \sigma_1$ these can be computed with a relative error proportional to σ_1/σ_k (as compared to an error proportional to σ_1^2/σ_k^2).

1.7. Iterative Methods

In this section we study algorithms designed to deal with very large (or even infinite) matrices where one has to be content with partial information on eigenvalues and eigenvectors. If A is the matrix under consideration it is only required that Ax is relatively cheaply computed for any vector x. This happens, for instance, when A is a so called *sparse* matrix, i.e., a matrix where the vast majority of the entries in any row is zero (e.g., a $10^5 \times 10^5$ matrix for which about 10 entries per row are different from zero) so that computation of Ax is much cheaper than m^2 flops.

The basic idea of the methods to be discussed in this section is to study an m-dimensional problem on an n-dimensional subspace.

1.7.1. Krylov subspaces. Let $A \in \mathbb{C}^{m\times m}$ and $b \in \mathbb{C}^m$. The sequence $n \mapsto A^{n-1}b$ is called the *Krylov sequence* associated with A and b and the spaces $\mathcal{K}_n = \langle b, Ab, ..., A^{n-1}b\rangle$ are called *Krylov subspaces* (of $\mathbb{C}^m$).

1.7.2. Arnoldi iteration. A complete reduction of $A \in \mathbb{C}^{m\times m}$ to Hessenberg form is given by $A = QHQ^*$ where Q is unitary and H is Hessenberg. If m is so large that the computation of the complete reduction is out of the question one might be interested in partial reductions. Note that $A = QHQ^*$ is equivalent to $AQ = QH$. One now wants to compute the first n columns of Q and H. This is done by the Arnoldi algorithm. Let $\tilde{H}_n$ be the $(n+1) \times n$ upper left block of H and let Q_n denote the $m \times n$ matrix consisting of the first n columns of Q. Then the first n columns of the equation $AQ = QH$ read $AQ_n = Q_{n+1}\tilde{H}_n$. The n-th column, in particular, is

$$Aq_n = H_{1,n}q_1 + ... + H_{n,n}q_n + H_{n+1,n}q_{n+1}.$$

If q_1 is given this allows to compute Q_{n+1} and $\tilde{H}_n$ recursively, as long as $Aq_n \notin \mathcal{K}_n = \langle q_1, ..., q_n\rangle$. This is seen by induction: If Aq_1 is not a multiple of q_1 then $\langle q_1, Aq_1\rangle$ has an orthonormal basis q_1, q_2 and $H_{1,1}$ and $H_{1,2}$ are the coefficients of Aq_1 with respect to that basis. Now suppose that Q_n and $\tilde{H}_{n-1}$ have been computed and that Aq_n is not in $\mathcal{K}_n$.

Then $\langle q_1, ..., q_n, Aq_n\rangle$ has an orthonormal basis $q_1, ..., q_{n+1}$ and the coefficients of Aq_n in terms of that basis form the entries of the last column of $\tilde{H}_n$ so that

$$Q_{n+1} = (Q_n, q_{n+1}) \quad \text{and} \quad \tilde{H}_n = \begin{pmatrix} \tilde{H}_{n-1} & h \\ 0 & H_{n+1,n} \end{pmatrix}$$

where $h \in \mathbb{C}^n$ satisfies $h_k = H_{k,n}$. At step n of the process one has computed an orthonormal basis $q_1, ..., q_n$ of the Krylov subspace $\langle q_1, Aq_1, ..., A^{n-1}q_1\rangle$.

If $Aq_n \in \mathcal{K}_n$ then $\mathcal{K}_n$ is an invariant subspace under A and hence one can split off a problem on an n-dimensional subspace.

ALGORITHM.
Choose an arbitrary normalized vector q_1
for $n = 1$ **to** ...
    $v = Aq_n$
    for $k = 1$ **to** n
        $H_{k,n} = q_k^* v$
        $v = v - H_{k,n} q_k$
    $H_{n+1,n} = \|v\|$
    if $H_{n+1,n} = 0$
        Terminate process
    $q_{n+1} = v/H_{n+1,n}$
end

1.7.3. Projection onto Krylov subspaces. If the size of the matrix A is too large to be tractable one is interested in a reduction of A onto the Krylov subspaces. Hence one restricts the domain of A to $\mathcal{K}_n$ but the image $A(\mathcal{K}_n)$ is generally not in $\mathcal{K}_n$ so that $A|_{\mathcal{K}_n}$ cannot be represented by a square matrix. Since $Q_n Q_n^*$ is the orthogonal projection onto $\mathcal{K}_n$ one considers therefore the matrix

$$T_n = Q_n Q_n^* A|_{\mathcal{K}_n}$$

which maps $\mathcal{K}_n$ to itself. Note that $Q_n^* Q_{n+1}$ is the $n \times (n+1)$ matrix with ones on the diagonal and zeros everywhere else. Hence the Hessenberg matrix $H_n = Q_n^* Q_{n+1} \tilde{H}_n$ is the upper left $n \times n$ block of H. Since $AQ_n = Q_{n+1}\tilde{H}_n$ and $Q_n^* Q_n = I$ we have $H_n = Q_n^* A Q_n$ and hence

$$T_n = Q_n Q_n^* A|_{\mathcal{K}_n} = Q_n H_n Q_n^*|_{\mathcal{K}_n}$$

i.e., H_n is the matrix associated with T_n when choosing the basis $q_1, ..., q_n$ in domain and range.

The eigenvalues of T_n (or H_n) are called the *Arnoldi eigenvalue estimates* or *Ritz values* (at step n). They can approximate some of the eigenvalues of A extraordinarily accurately even if n is a lot smaller than m. They are computed using the QR algorithm.

1.7.4. Arnoldi iteration and polynomial approximation. The set of polynomials of degree n (in one indeterminate) with complex coefficients is denote by $\mathbb{C}[z]_n$. Let M_n be the subset of $\mathbb{C}[z]_n$ whose elements are monic, i.e., whose elements have leading coefficient one. Suppose a matrix $A \in \mathbb{C}^{m\times m}$ and a vector $b \in \mathbb{C}^m$ are given. The associated Krylov subspace $\mathcal{K}_n$ can then be written as

$$\mathcal{K}_n = \{p(A)b : p \in \mathbb{C}[z]_{n-1}\}.$$

The Arnoldi approximation problem is to find a polynomial p_0 in M_n such that

$$\|p_0(A)b\| = \min\{\|p(A)b\| : p \in M_n\}.$$

THEOREM. If $\mathcal{K}_n$ has dimension n the Arnoldi approximation problem has a unique solution, namely the characteristic polynomial of H_n.

Sketch of proof: First note that $p(A)b - A^n b \in \mathcal{K}_n$. Hence there is a $y \in \mathbb{C}^n$ such that $p(A)b = A^n b - Q_n y$ and the Arnoldi approximation problem is equivalent to the least squares problem of minimizing $\|A^n b - Q_n y\|$ over $y \in \mathbb{C}^n$ where $A^n b$ is the given right hand side. By Theorem 1.5.1 we have to solve the equation $Q_n^* Q_n y = Q_n^* A^n b$ which is equivalent to $Q_n^* p(A)b = 0$. Since $A = QHQ^*$ we have $p(A) = Qp(H)Q^*$. Note that $Q_n^* Q$ is the $n \times m$ matrix whose first n columns form the $n \times n$ identity matrix and whose remaining $m - n$ columns are zero. Further note that $Q^* b = e_1 \in \mathbb{C}^m$. Hence $0 = Q_n^* p(A)b = Q_n^* Q p(H) Q^* b = p(H)_{1:n,1}$. Because of the structure of a Hessenberg matrix one finds that $p(H)_{1:n,1} = p(H_n)_{1:n,1}$. In conclusion we find that p gives rise to a minimum if and only if the first column of $p(H_n)$ is zero. That is certainly the case for (an appropriate multiple of) the characteristic polynomial of H_n by the Cayley-Hamilton theorem. To prove uniqueness assume that the minimum is attained for either of the polynomials p_1 and p_2 in M_n. Then $0 = Q_n^* p_1(A)b = Q_n^* p_2(A)b$ which implies that $Q_n^*(p_1 - p_2)(A)b = 0$. Since $q = p_1 - p_2$ has degree at most $n - 1$ we have that $q(A)b \in \mathcal{K}_n$. Thus $q(A)b = Q_n Q_n^* q(A)b = 0$. If q is not the zero polynomial the equation $q(A)b = 0$ implies that the vectors $b, Ab, ..., A^{n-1}b$ are linearly dependent, which is impossible. □

1.7.5. Computing eigenvalues by Arnoldi iteration. The following theorem is a special case of the so called spectral theorem:

THEOREM. Let $A \in \mathbb{C}^{m \times m}$ be a matrix and f a polynomial. Then

$$f(A) = \sum_{\lambda \in \sigma(A)} \sum_{j=0}^{\nu(\lambda)-1} \frac{(A - \lambda I)^j}{j!} f^{(j)}(\lambda) E_\lambda$$

where $\sigma(A)$ is the set of eigenvalues of A, $\nu(\lambda)$ denotes the index of λ, and E_λ is the eigenprojection associated with λ. In particular, if $\nu(\lambda) = 1$ for every eigenvalue λ then

$$f(A) = \sum_{\lambda \in \sigma(A)} f(\lambda) E_\lambda.$$

Hence if one picks f such that $f(\lambda)$, ..., $f^{(\nu(\lambda)-1)}(\lambda)$ are small for every eigenvalue λ then $\|f(A)\|$ and hence the norms of $\|f(A)x\|$ when $\|x\| = 1$ become small. Conversely after finding a polynomial such that $\|f(A)b\|$ is small one may hope that f is small near the eigenvalues of A or that the roots of f approximate (some of) the eigenvalues of A. For instance if A is diagonalizable (i.e., all indices are equal to one), if the number of distinct eigenvalues of A is equal to n, and if b has components in all eigenspaces, then after n steps the Arnoldi iteration has computed the minimal polynomial of A and hence all eigenvalues of A exactly.

1.7.6. Arnoldi lemniscates. Let p be a polynomial and C a positive constant. A curve (or a collection of curves) given by

$$\{z \in \mathbb{C} : |p(z)| = C\}$$

is called a *lemniscate*. If one chooses $C = \|p(A)b\|/\|b\|$ one calls the resulting curve an Arnoldi lemniscate. The Arnoldi lemniscates tend to split in components encircling or nearly encircling (groups of) eigenvalues. When a component of an Arnoldi lemniscate has captured a single eigenvalue experience shows that convergence to that eigenvalue is of the form ρ^n for some $\rho < 1$.

1.7.7. Generalized minimal residuals (GMRES). We next describe how Arnoldi iteration is used to compute solutions of the equation $Ax = b$ where $A \in \mathbb{C}^{m\times m}$ is nonsingular. Let $\mathcal{K}_n = \langle b, Ab, ..., A^{n-1}b\rangle$ be a Krylov subspace. The approximation of the true solution $x_* = A^{-1}b$ at step n is then taken to be the unique solution x_n of the least squares problem

$$\|r_n\| = \|b - Ax_n\| = \min\{\|b - Ax\| : x \in \mathcal{K}_n\}.$$

Since $x \in \mathcal{K}_n$ there is a $y \in \mathbb{C}^n$ such that $x = Q_n y$. Therefore, and because $AQ_n = Q_{n+1}\tilde{H}_n$, we have

$$\|r_n\| = \|b - Ax_n\| = \min\{\|b - Q_{n+1}\tilde{H}_n y\| : y \in \mathbb{C}^n\}.$$

Next note that $b - Q_{n+1}\tilde{H}_n y \in \mathcal{K}_{n+1}$. For any $z \in \mathcal{K}_{n+1} = \mathrm{im}(Q_{n+1})$ we have that $\|Q^*_{n+1}z\| = \|z\|$. This and the fact that $Q^*_{n+1}Q_{n+1}$ is the identity on $\mathbb{C}^{n+1}$ shows that

$$\|b - Q_{n+1}\tilde{H}_n y\| = \|Q^*_{n+1}b - \tilde{H}_n y\|.$$

Finally since $b = \|b\|q_1$ we have that

$$\|r_n\| = \|b - Ax_n\| = \min\{\|\tilde{H}_n y - \|b\|e_1\| : y \in \mathbb{C}^n\}$$

providing the following algorithm:

Algorithm.
$q_1 = b/\|b\|$
for $n = 1$ **to** ...
Perform step n of the Arnoldi algorithm to find Q_{n+1} and $\tilde{H}_n$
Find the vector $y_0 \in \mathbb{C}^n$ which minimizes $\|\tilde{H}_n y - \|b\|e_1\|$
$x_n = Q_n y_0$
end

Note that the value of $\|r_n\| = \|\tilde{H}_n y_0 - \|b\|e_1\|$ is a by-product of the algorithm.

1.7.8. Convergence of GMRES. Note that one computes the norms of the residuals r_n as the minima of the functional $x \mapsto \|b - Ax\|$ over a sequence of sets $\mathcal{K}_1 \subset \mathcal{K}_2 \subset ....$ Hence $\|r_n\|$ is a nonincreasing sequence. In the absence of rounding errors we will have $\|r_n\| = 0$ at the latest when $n = m$ and perhaps sooner. Thus the convergence of the algorithm is guaranteed but in practice one looks for a sufficiently small residual for values of n much smaller than m.

The relative error of the solution x_n is estimated in the following way:

$$\frac{|x_n - x_*|}{|x_*|} = \frac{A^{-1}(b - r_n) - A^{-1}b}{A^{-1}b} \leq \kappa(A)\frac{\|r_n\|}{\|b\|}.$$

Since, as mentioned above, $\|r_n\|$ is computed at each step of the iteration one has control over the error if information on the condition number of A is available.

1.7.9. GMRES and polynomial approximation. Since the approximation x_n to the solution x_* of the equation $Ax = b$ is the unique solution of the least squares problem associated with minimizing $r = b - Ax$ over $x \in \mathcal{K}_n$ we can write x_n as a linear combination of b, Ab, ..., $A^{n-1}b$, i.e., there is a unique polynomial $\tilde{p} \in \mathbb{C}[z]_{n-1}$ such that $x_n = \tilde{p}(A)b$. Hence $r_n = (I - Ax_n) = (I - A\tilde{p}(A))b = p(A)b$ where $p(z) = 1 - z\tilde{p}(z)$ is a polynomial of degree n for which the coefficient of z^0 is equal to one. Hence the following problem is equivalent to the problem of finding x_n:

$$\|r_n\| = \min\{\|p(A)b\| : p \in P_n\}$$

where P_n is the the subset of $\mathbb{C}[z]_n$ for which the coefficient of z^0 is equal to one.

1.8. Problems

Please find below the assigned homework. A bullet (•) indicates that the problem will be graded, the first due date is listed in parentheses.

(1) JPE, Spring 2000, #1
(2) • (September 18) JPE, Fall 1999, #5
(3) • (September 20) Let $\varepsilon \in \mathbb{R}$ be positive and assume that $A \in \mathbb{C}^{m\times m}$ satisfies $\|A\| < 1/\varepsilon$. Show that $I + \varepsilon A$ is invertible. Hint: a linear operator $T : \mathbb{C}^m \to \mathbb{C}^m$ is invertible if and only if $\ker(T) = \{0\}$.
(4) Suppose that λ_1, ..., λ_n are the distinct eigenvalues of a matrix $A : \mathbb{C}^m \to \mathbb{C}^m$ with respective algebraic multiplicities k_1, ..., k_n. The quantity $\det(\lambda I - A)$ is a polynomial with respect to λ of degree m and leading coefficient one, i.e.,

$$\det(\lambda I - A) = \lambda^m - \sigma_1\lambda^{m-1} + ... + (-1)^m\sigma_m.$$

The number $\sigma_1 = k_1\lambda_1 + ... + k_n\lambda_n$ is called the trace of A and denoted by $\operatorname{tr}(A)$. Prove the following statements:
   (a) $\operatorname{tr}(A) = \sum_{\ell=1}^m A_{\ell,\ell}$.
   (b) $\operatorname{tr}(AB) = \operatorname{tr}(BA)$ for all $A, B \in \mathbb{C}^{m\times m}$.
   (c) If $T \in \mathbb{C}^{m\times m}$ is invertible then $\operatorname{tr}(T^{-1}AT) = \operatorname{tr}(A)$.
   (d) $\|A\|_2^2 = \operatorname{tr}(A^*A) = \operatorname{tr}(AA^*)$.
(5) • (September 25) Prove

$$\|A\|_{1,1} = \max\{\sum_{j=1}^m |A_{j,k}| : 1 \le k \le m\} \quad \text{and} \quad \|A\|_{\infty,\infty} = \max\{\sum_{k=1}^m |A_{j,k}| : 1 \le j \le m\}.$$

(6) JPE, Spring 1999, #3
(7) Show that a matrix which is both triangular and unitary is diagonal.
(8) • (September 27) Let $x \in \mathbb{C}^m$, $A \in \mathbb{C}^{m\times n}$, and $p \in [1, \infty]$. Show

$$\|x\|_\infty \le \|x\|_p \le \sqrt[p]{m}\|x\|_\infty,$$
$$\|A\|_{\infty,\infty} \le \sqrt[p]{n}\|A\|_{p,p} \le \sqrt[p]{mn}\|A\|_{\infty,\infty}.$$

Is equality ever achieved?
(9) Determine the SVDs of the following matrices

(a) $\begin{pmatrix} 3 & 0 \\ 0 & 2 \end{pmatrix}$, (b) $\begin{pmatrix} 2 & 0 \\ 0 & 3 \end{pmatrix}$, (c) $\begin{pmatrix} 0 & 2 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$, (d) $\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$, (e) $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$.

(10) • (October 2) Two matrices $A, B \in \mathbb{C}^{m\times m}$ are called unitarily equivalent, if there is a unitary matrix Q such that $B = Q^*AQ$. Prove or disprove: A and B are unitarily equivalent if and only if they have the same singular values.
(11) How many IEEE double precision real numbers are between two adjacent nonzero IEEE single precision numbers?
(12) • (October 4) The floating point number system $F = F_p$ includes many integers but not all of them. Give an exact formula for the smallest integer in $\mathbb{N} - F_p$. In particular, what are these numbers for the IEEE single and double precision standards.
(13) For each of the following algorithms implemented on a computer satisfying Axiom 1.2.1 decide whether it is backward stable or stable but not backward stable. Compute C_1 and C_2.
   (a) $\tilde{f} : \mathbb{R} \to \mathbb{R} : x \mapsto x \oplus x$.

(b) $\tilde{f} : \mathbb{R} \to \mathbb{R} : x \mapsto x \otimes x$.

(14) • (October 9) Suppose you had an algorithm which determines for every $m \times n$ matrix its SVD. Explain what it would mean for this algorithm to be stable and backward stable. Is it possible that the algorithm is backward stable?

(15) Suppose $A \in \mathbb{C}^{m\times m}$ is self-adjoint (hermitian) and has the (not necessarily distinct) eigenvalues λ_1, ..., λ_m. What are the singular values of A.

(16) • (October 11) Give a sequence of 2×2 matrices A_n whose eigenvalues are $\lambda_1, \lambda_2 \in \mathbb{R}$ regardless of n but for which $\|A_n\|_{2,2} = \sigma_1(A_n)$ tends to infinity.

(17) Show that $\kappa(A) = \sigma_1/\sigma_n$ if $m \geq n$ and $A \in \mathbb{C}^{m\times n}$ has full rank and singular values $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n$.

(18) • (October 16) Suppose A is a 202×202 matrix with $\|A\|_{2,2} = 100$ and $\|A\|_2 = 101$. Give the best possible lower bound on $\kappa_{2,2}(A)$.

(19) JPE, Fall 1999, #2.

(20) • (October 18) JPE, Spring 2000, #5.

(21) Let $v \in \mathbb{C}^m$ be nonzero and P_v the orthogonal projection onto $\langle v\rangle$. Show that $F = I - 2P_v$ is unitary and self-adjoint.

(22) • (October 23) JPE, Fall 1998, #1.

(23) JPE, Fall 1999, #4 (second try).

(24) • (October 25) Let A be an $m \times m$ matrix and let a_j be the j-th column of A. Prove Hadamard's inequality which states that

$$|\det(A)| \leq \prod_{k=1}^{m} \|a_k\|_2.$$

Give a geometric interpretation of this result making use of the relation ship between $\det(A)$ and the volume of the parallelepiped spanned by a_1, ..., a_m.

(25) Show that multiplication of a vector by a matrix is backward stable. More precisely, if $A \in \mathbb{R}^{m\times n}$ and $x \in \mathbb{C}^n$, give an algorithm which computes Ax and show that there is a δA such that $(A + \delta A)x$ equals the vector computed by that algorithm. Assume that idealized computer arithmetic is utilized.

(26) • (October 30) Show that the matrices L'_j defined in 1.4.2 are lower triangular and that each of their diagonal entries is one.

(27) If a $LU = PA$ is a LU factorization obtained from A by Gaussian elimination with partial pivoting, prove that $\|L\|_\infty \leq 1$.

(28) • (November 1) Let $A \in \mathbb{C}^{m\times m}$ be nonsingular. Show that A has an LU factorization if and only if $A_{1:k,1:k}$ is nonsingular for $k = 1, ..., m$.

(29) JPE, Spring 2000, #3.

(30) • (November 6) Suppose that the $m \times n$ matrix A has the form

$$\begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$$

where A_1 is a nonsingular $n \times n$ matrix and A_2 is an arbitrary $(m-n)\times n$ matrix. Prove that $\|A^+\|_{2,2} \leq \|A_1^{-1}\|_{2,2}$.

(31) JPE, Spring 2000, #4.

(32) • (November 8) JPE, Spring 2000, #6.

(33) Let $A \in \mathbb{C}^{m\times m}$ have eigenvalues λ_1, ..., λ_m. Define $\rho = \max\{|\lambda_1|, ..., |\lambda_m|\}$ (this quantity is called the *spectral radius*). Show that $\lim_{n\to\infty} \|A^n\| = 0$ if and only if $\rho < 1$.

(34) • (November 13) Prove the following theorem:
Let A and B be selfadjoint matrices in $\mathbb{C}^{m\times m}$. Let $a_1 \leq ... \leq a_m$, $b_1 \leq ... \leq b_m$,

and $c_1 \leq ... \leq c_m$ denote respectively the eigenvalues of A, B, and $A+B$. Show that

$$a_k + b_1 \leq c_k \leq a_k + b_m$$

for $k = 1, ..., m$.

Remark: This theorem is particularly interesting if B is small in the sense that $|b_1|$ and $|b_m|$ are small compared to the smallest of the $|a_j|$.

(35) JPE, Fall 1999, #8.

(36) • (November 15) Let

$$A = \begin{pmatrix} 9 & 1 \\ 1 & 2 \end{pmatrix}$$

and $x_0 = (1,1)^*$. Compute $y_n = Ax_{n-1}$,

$$\sigma_n = \begin{cases} y_{n;1} & \text{if } |y_{n;1}| \geq |y_{n;2}| \\ y_{n;2} & \text{if } |y_{n;1}| < |y_{n;2}| \end{cases},$$

and $x_n = y_n/\sigma_n$ for $n = 1, ..., 5$. Also compute the eigenvalues and eigenvectors of A.

(37) Let $A \in \mathbb{C}^{m\times m}$ be tridiagonal and self-adjoint with all its sub- and superdiagonal entries nonzero. Prove that the eigenvalues of A are pairwise distinct. Hint: Show that, for any $\lambda \in \mathbb{C}$, the matrix $A - \lambda I$ has rank at least $m-1$.

(38) • (January 10) Let $A \in \mathbb{C}^{m\times m}$ have eigenvalues λ_1, ..., λ_m and respectively associated linearly independent eigenvectors u_1, ..., u_m such that

$$|\lambda_1| \geq ... \geq |\lambda_n| > |\lambda_{n+1}| \geq ... \geq |\lambda_m|$$

for some $n \in \{1, ..., m-1\}$. Suppose that there are linearly independent vectors s_1, ..., s_n such that

$$\langle s_1, ..., s_n \rangle \cap \langle u_{n+1}, ..., u_m \rangle = \{0\}.$$

Show that As_1, ..., As_n are linearly independent and that

$$\langle As_1, ..., As_n \rangle \cap \langle u_{n+1}, ..., u_m \rangle = \{0\}.$$

(39) Suppose that $H \in \mathbb{C}^{m\times m}$ is in Hessenberg form and that $R \in \mathbb{C}^{m\times m}$ is upper triangular. Show that RH and HR are in Hessenberg form.

(40) • (January 15) JPE, Spring 2000, #7.

(41) JPE, Fall 1999, #7.

(42) • (January 17) Describe how to find eigenvectors with the QR algorithm (under the assumptions of Theorem 1.6.15).

(43) Suppose the Arnoldi algorithm is executed for a given matrix A and vector b until at some step n one encounters the case that the entry $H_{n+1,n}$ is equal to zero. Prove the following statements:

(a) $\mathcal{K}_n$ is invariant under A and $\mathcal{K}_{n+j} = \mathcal{K}_n$ for every natural number j.

(b) Every eigenvalue of H_n is an eigenvalue of A.

(c) If A is nonsingular, then the solution of $Ax = b$ lies in $\mathcal{K}_n$.

(44) • (January 24) Assume that $n \leq m$, that $H \in \mathbb{C}^{m\times m}$ is Hessenberg and that H_n is the upper left $n \times n$ block of H. Show that

$$(H^k)_{1:n,1:(n+1-k)} = (H_n^k)_{1:n,1:(n+1-k)}$$

when $1 \leq k \leq n$.

(45) Suppose that the columns of $Q \in \mathbb{C}^{m\times n}$ are orthonormal. Show the truth of the following statement: If x and y are in $\mathrm{im}(Q)$ then $(Q^*x, Q^*y) = (x, y)$ and $\|Q^*x\| = \|x\|$.

(46) • (January 29) Suppose $A \in \mathbb{C}^{m\times m}$ is nonsingular. Let b be a nonzero vector in $\mathbb{C}^m$ and let $\mathcal{K}_j$ denote the associated Krylov spaces. Suppose that $\dim \mathcal{K}_{n+1} = n$. Show that $\|r_n\| = \min\{\|b - Ax\| : x \in \mathcal{K}_n\} = 0$.

1.9. Programming Assignments

(1) (Due: October 18) Let F' be the set of IEEE double precision numbers. There is a smallest integer N in $\mathbb{N} - F'$. It was computed in Problem 12. Write a program which reads a value for the integer p, computes the values of $2^p + k$, and prints the values of k and $2^p + k$ for $k = -10, ..., 10$. The program must include the `FORMAT` statement and a `DO` loop. Run the program at least for that value of p where you expect a problem. Are your expectations right? Explain what happens.

(2) (Due: November 14) Write subroutines for each of the following algorithms: solution of an upper triangular system by back substitution, Gram-Schmidt orthogonalization, modified Gram-Schmidt orthogonalization, Householder triangularization (including the algorithm for the computation of Q^*b), and LU factorization with partial pivoting. Use these algorithms to compute the solution of $Ax = b$ where

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 7 \\ 4 & 2 & 3 \\ 4 & 2 & 2 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} -3 \\ -3 \\ 1 \\ 0 \\ 2 \end{pmatrix}.$$

Write the programs for single precision arithmetic. Check that $(1, 1, -2)$ is the true solution and compute the relative errors in each case.

(3) (Due February 7) Write a program which computes the eigenvalues of a matrix (of size up to 10×10) by transforming it first to Hessenberg form and applying then the QR algorithm. Apply your program to the matrix

$$A = \begin{pmatrix} 190 & 356 & 522 & 92 & 150 \\ -92 & -172 & -248 & -40 & -64 \\ -11 & -22 & -29 & -6 & -11 \\ -32 & -64 & -96 & -8 & -32 \\ 51 & 102 & 133 & 6 & 35 \end{pmatrix}.$$

Index