1 Preliminaries

1.1 Vector and Matrix Norms

The p-norm of a vector $x \in \mathbb{C}^n$ is defined by

$$\|x\|_p = \Big(\sum_{i=1}^{n} |x_i|^p\Big)^{1/p}, \quad \text{for } p \geq 1,$$
$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|.$$

For $p = 2$ we get the usual Euclidean norm.

The p-norm of a matrix $A \in \mathbb{C}^{m \times n}$ induced by the vector norm $\|\cdot\|_p$ is defined by

$$\|A\|_p = \max_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \max_{\|x\|_p \leq 1} \|Ax\|_p = \max_{\|x\|_p = 1} \|Ax\|_p.$$

Only the cases $p = 1$ and $p = \infty$ have closed form expressions.

$$\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{m} |a_{ij}|,$$
$$\|A\|_\infty = \max_{1 \leq i \leq m} \sum_{j=1}^{n} |a_{ij}|.$$

For $p = 2$ we have $\|A\|_2 = \sigma_{\max}$, the largest singular value of A (see Section 2.3).

For a square matrix A we define the *condition number* of A by $\kappa_p(A) = \|A\|_p \|A^{-1}\|_p$, where we set $\kappa_p(A) = \infty$ if A is singular. In particular, the condition number for the 2-norm is given by the ratio of the largest and smallest singular values

$$\kappa_2(A) = \frac{\sigma_{\max}}{\sigma_{\min}}.$$

This also holds for nonsquare matrices.

1.2 Some Facts About Floating Point Numbers

The *machine epsilon* of a floating point number system is the gap between 1 and the next larger floating point number. For IEEE double (single) precision $\varepsilon = 2^{-52}$ ($\varepsilon = 2^{-23}$). Hence there are no floating point numbers between 1 and $1 + \varepsilon$, so $1 + \delta\varepsilon$ rounds to either 1 or $1 + \varepsilon$ for $\delta \in (0, 1)$.

The *unit roundoff* is defined as $u = \varepsilon/2$. For IEEE double (single) precision $u = 2^{-53} \approx 1.1102 \cdot 10^{-16}$ ($u = 2^{-24} \approx 5.9605 \cdot 10^{-8}$). If $\mathrm{fl}(x)$ is the floating point representation of $x \in \mathbb{R}$, we have

$$\mathrm{fl}(x) = x(1 + \delta), \qquad |\delta| \leq u.$$

Let $\mathrm{round}(x)$ be the nearest floating point number to x (the default rounding mode in IEEE arithmetic). Then for $x \neq 0$

$$\frac{|\mathrm{round}(x) - x|}{|x|} \leq u.$$

Let op be any of the four basic arithmetic operations. The IEEE standard requires that the computed result is the correctly rounded version of the exact result and

$$\mathrm{fl}(x \operatorname{op} y) = (x \operatorname{op} y)(1 + \delta), \qquad |\delta| \leq u.$$

Therefore we have the relative error

$$\frac{|\text{fl}(x \text{ op } y) - (x \text{ op } y)|}{|x \text{ op } y|} \leq u,$$

provided that $x \text{ op } y \neq 0$. This also means that we can view the computed result as the exact result between perturbed operands. For example $\text{fl}(x+y) = (x+y)(1+\delta) = x(1+\delta) + y(1+\delta)$, so the computed sum of x and y is the exact sum of $x(1+\delta)$ and $y(1+\delta)$. This is called *backwards error analysis*.

2 Matrix Decompositions

2.1 LU Decomposition

The following theorem gives the conditions for the existence of the LU decomposition.

Theorem 1. *If $A \in \mathbb{C}^{n\times n}$ and the leading principal minors $A(1:k, 1:k)$ are nonsingular for $k = 1, \ldots, n-1$, then there is a unique lower triangular $L \in \mathbb{C}^{n\times n}$ with unit diagonal elements and a unique upper triangular $U \in \mathbb{C}^{n\times n}$ s.t. $A = LU$.*

Theorem 1 also holds for nonsquare matrices.

The LU decomposition is computed with Gaussian elimination by performing row operations on the matrix. If the conditions of Theorem 1 don't hold, we can still compute the LU-decomposition if we permute the rows suitably during the elimination to avoid zero pivots (we should do this anyway for numerical stability).

Theorem 2. *For every $A \in \mathbb{C}^{m\times n}$ there is a unit lower triangular $L \in \mathbb{C}^{m\times m}$, an upper triangular $U \in \mathbb{C}^{m\times n}$, and a permutation matrix $P \in \mathbb{C}^{m\times m}$ s.t. $PA = LU$.*

The computation of LU decompostion takes $O(n^3)$ flops.

2.2 Cholesky Decomposition

Recall that a matrix $A \in \mathbb{C}^{n\times n}$ is Hermitian (or self-adjoint) if $A = A^*$. A Hermitian matrix is positive definite (positive semi-definite) if $x^*Ax > 0$ for all nonzero $x \in \mathbb{C}^n$ ($x^*Ax \geq 0$ for all $x \in \mathbb{C}^n$). For Hermitian positive definite matrices the LU decomposition takes the following form.

Theorem 3. *Let $A \in \mathbb{C}^{n\times n}$ be Hermitian and positive definite. Then there is a unique lower triangular matrix $L \in \mathbb{C}^{n\times n}$ with positive diagonal elements s.t. $A = LL^*$.*

If A is only semi-definite, Theorem 3 still holds but some diagonal elements of L will be zero and L might not be unique.

2.3 Singular Value Decomposition

Theorem 4. *Every matrix $A \in \mathbb{C}^{m\times n}$ has a singular value decomposition (SVD) $A = U\Lambda V^*$, where $U \in \mathbb{C}^{m\times m}$ and $V \in \mathbb{C}^{n\times n}$ are unitary, $\Lambda = \operatorname{diag}(\sigma_1, \ldots, \sigma_p) \in \mathbb{C}^{m\times n}$, where $p = \min\{m, n\}$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p \geq 0$.*

The numbers σ_i are the singular values of A and the columns of U and V are the left and right singular vectors of A, respectively.

Notice that Λ has the same dimensions as A. It has $m - n$ zero rows if $m > n$, and $n - m$ zero columns if $n > m$.

The matrix A has full rank if and only if $\sigma_p > 0$, otherwise the rank is the number of nonzero singular values.

3 Direct Solution of Linear Equations

Consider the linear equation

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

where $A \in \mathbb{C}^{n\times n}$ and $b \in \mathbb{C}^n$. You should never solve (1) by multiplying the right hand side with the inverse of A. In MATLAB you should always use the backslash operator \ and compute `x = A\b`.

3.1 Solution Using the LU Decomposition

Suppose that A is nonsingular and has the LU-decomposition $PA = LU$. Multiplying (1) from the left by P we get

$$Pb = PAx = LUx.$$

Letting $Ux = y$ we solve (1) by solving the two triangular systems

$$Ly = Pb = \tilde{b} \quad \text{and} \quad Ux = y.$$

The lower triangular system $Ly = \tilde{b}$ is solved as follows. The ith equation is clearly

$$\tilde{b}_i = \sum_{j=1}^{i} l_{ij} y_j = \sum_{j=1}^{i-1} l_{ij} y_j + y_i.$$

Hence y_1 can be solved from the first equation, y_2 from the second and so on. This gives us the following forward substitution algorithm, which we give in a slightly more general form, where the diagonal elements of L can be a any nonzero numbers.

Algorithm 3.1: Forward Substitution

Input: A nonsingular lower triangular matrix $L \in \mathbb{C}^{n\times n}$ and a vector $b \in \mathbb{C}^n$.
Output: The solution x of the equation $Lx = b$.

for $i = 1$ **to** n **do**

$$x_i = \Big(b_i - \sum_{j=1}^{i-1} l_{ij} x_j\Big)/l_{ii}$$

end

The upper triangular system $Ux = y$ has the ith equation

$$y_i = \sum_{j=i}^{n} u_{ij}x_j = \sum_{j=i+1}^{n} u_{ij}x_j + u_{ii}x_i.$$

It follows from the nonsingularity of A that $u_{ii} \neq 0$ for $i = 1,\ldots,n$, so x_n can be solved from the last equation, then x_{n-1} from the second to last and so on. This gives us the backward substitution algorithm 3.2.

Algorithm 3.2: Backward Substitution

Input: A nonsingular upper triangular matrix $U \in \mathbb{C}^{n\times n}$ and a vector $b \in \mathbb{C}^n$.
Output: The solution x of the equation $Ux = b$.

for $i = n$ **downto** 1 **do**

$$x_i = \Big(b_i - \sum_{j=i+1}^{n} u_{ij}x_j\Big)/u_{ii}$$

end

The solution of each triangular system takes $O(n^2)$ flops.

Suppose we want to solve (1) with multiple right hand sides b and the same A. Collecting all the right hand sides as columns of a matrix B leads to the matrix equation $AX = B$. As before, we get the triangular equations

$$LY = PB = \tilde{B} \quad \text{and} \quad UX = Y.$$

These can be solved by replacing b_i in Algorithms 3.1 and 3.2 with the ith rows of $\tilde{B}$ and Y, respectively. Notice that we have to find the LU decomposition of A only once. In particular, we can find the inverse of A by setting $B = I$.

3.2 Accuracy Considerations

The next theorem shows the effect of roundoff to the computation of the LU decomposition. For a matrix $A = (a_{ij})$ we define $|A|$ to be the matrix of absolute values $|A| = (|a_{ij}|)$.

Theorem 5. *If $A \in \mathbb{C}^{m\times n}$ has an LU decomposition, then the computed L and U satisfy $A + E = LU$, where*

$$|E| \leq 2(n-1)u(|A| + |L||U|) + O(u^2),$$

and u is the unit roundoff.

This shows that the computed L and U are the exact LU decomposition of the perturbation $A + E$ of A.

It can also be shown that with partial pivoting (row exchanges) the computed solution $\hat{x}$ to (1) satisfies $(A + E)\hat{x} = b$, where

$$\|E\|_\infty \leq 6n^3\rho\|A\|_\infty u + O(u^2),$$

where the *growth factor* ρ measures how large the elements of A become during the computation. In the worst case ρ can be as large as 2^{n-1}. However, experiments show that usually $\rho \approx n^{2/3}$ and serious growth is rare. Hence we have the following useful approximation: $\|E\|_\infty \approx u\|A\|_\infty$.

It can be shown that the *residual* $r = b - A\hat{x}$ satisfies $\|r\|_\infty \approx u\|A\|_\infty\|\hat{x}\|_\infty$. This gives the following heuristic.

Theorem 6 (Heuristic I). *Gaussian elimination produces a solution $\hat{x}$ with a relatively small residual.*

Small residual doesn't imply an accurate solution. We have

$$\frac{\|\hat{x} - x\|_\infty}{\|x\|_\infty} \approx u\kappa_\infty(A). \tag{2}$$

This gives us the second heuristic.

Theorem 7 (Heuristic II). *If the unit roundoff and condition number satisfy $u \approx 10^{-d}$ and $\kappa_\infty(A) \approx 10^q$, respectively, then Gaussian elimination produces a solution $\hat{x}$ that has about $d - q$ correct decimal digits.*

4 Iterative Solution of Linear Equations

Iterative methods for solving (1) generate a sequence $(x_k)_{k=1}^\infty$ starting from an initial guess x_0 s.t. $x_k \to A^{-1}b$ as $k \to \infty$.

4.1 Classical Iterations

In classical iterative methods we form a *splitting* of A by writing $A = M - N$ and then the iteration takes the form

$$Mx_{k+1} = Nx_k + b, \qquad k \geq 0, \quad x_0 \text{ given.} \tag{3}$$

If $x_k \to x$ as $k \to \infty$, then taking limits of the both sides of (3) we get $Mx = Nx + b$, which implies $b = (M - N)x = Ax$. Hence x is a solution of (1).

Recall that the *spectral radius* of $A \in \mathbb{C}^{n\times n}$ is defined as

$$\rho(A) = \max\{\, |\lambda| \mid \lambda \text{ is an eigenvalue of } A \,\}.$$

The following theorem characterizes convergence of iteration (3).

Theorem 8. *Let $A = M - N$ be a splitting of a nonsingular $A \in \mathbb{C}^{n\times n}$. If M is nonsingular, then iteration (3) converges to a solution of equation(1) for all initial values x_0 if and only if $\rho(M^{-1}N) < 1$.*

The splitting has to be chosen in such a way that equations of the form $Mz = c$ are relatively easy to solve and the condition $\rho(M^{-1}N) < 1$ holds.

4.1.1 Jacobi Iteration

Here we use the splitting $M = D_A$, $N = -(L_A + U_A)$, where D_A is the diagonal of A and L_A and U_A are the strict lower and upper triangular parts of A, respectively.

Clearly in this case the equation $Mz = c$ is easy to solve. The following theorem gives a sufficient condition for the spectral radius condition to hold.

Theorem 9. *If $A \in \mathbb{C}^{n\times n}$ is strictly diagonally dominant, i.e.,*

$$\sum_{\substack{j=1\\ j\neq i}}^{n} |a_{ij}| < |a_{ii}|, \qquad i = 1, \ldots, n,$$

then the Jacobi iteration converges to a solution of equation (1) for all initial values x_0.

4.1.2 Gauss-Seidel Iteration

Here we use the splitting $M = D_A + L_A$, $N = -U_A$, where D_A, L_A, and U_A are as in Section 4.1.1. Now M is lower triangular, so the equation $Mz = c$ is easy to solve.

It can be shown that the Gauss-Seidel iteration converges if A is strictly diagonally dominant. The following theorem gives an alternate convergence criterion.

Theorem 10. *If $A \in \mathbb{R}^{n\times n}$ is symmetric and positive definite, then the Gauss-Seidel iteration converges to a solution of equation (1) for all initial values x_0.*

4.1.3 Successive Over Relaxation (SOR)

Here we use a parametriced splitting $A = M(\omega) - N(\omega)$ for $\omega \in \mathbb{R}$ and

$$M(\omega) = \frac{1}{\omega} D_A + L_A, \qquad N(\omega) = \left(\frac{1}{\omega} - 1\right) D_A - U_A.$$

Clearly for $\omega = 1$ we get the Gauss-Seidel iteration.

We have the following convergence result.

Theorem 11. *If $A \in \mathbb{R}^{n\times n}$ is symmetric and positive definite and $\omega \in (0, 2)$, then the SOR iteration converges to a solution of equation (1) for all initial values x_0.*