1.2 Factorization Based Methods - Continued

1.2.2 LUP Factorization

So, we can find an LU factorization for a matrix A, whenever row swaps are *not* necessary. What if row interchanges (pivoting) *are* necessary? A row interchange is a permutation of two rows. We keep track of those in a *permutation* matrix, which is simply a matrix obtained from the corresponding identity matrix I by permuting rows. So, for a matrix A we find its LUP **factorization (decomposition)**, i.e., a triplet (L, U, P), with L a lower triangular, U an upper triangular and P a permutation matrix, such that

$$PA = LU. \tag{1.1}$$

Remark 1.1.
1. Multiplication of a matrix A to the *left* by a permutation matrix P will yield the same *row* interchanges on the matrix A as in P, while multiplication on the *right* will result in the same *column* interchanges in A as in P.
2. Solving the system $Ax = b$ is now equivalent to solving two triangular systems

$$\begin{aligned} Ly &= Pb \text{ and} \\ Ux &= y. \end{aligned} \tag{1.2}$$

3. The procedure for obtaining an LUP factorization is similar to the previous one, while keeping track of the row interchanges in a permutation matrix P.

Example 1.2. Find an LUP factorization for the matrix

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

Solution. At the first step, we do partial pivoting and interchange $(R_1) \longleftrightarrow (R_3)$.
At each row interchange, instead of writing the entire matrix P, we only emphasize which rows are

permuted. Other than that, we proceed as before. We have

$$A \sim \left[\begin{array}{ccc} 3 & -1 & 1 \\ 1 & 1 & -1 \\ 2 & 1 & -2 \end{array}\right] = \left[\begin{array}{c|cc} 3 & -1 & 1 \\ \hline 1 & 1 & -1 \\ 2 & 1 & -2 \end{array}\right] \sim \left[\begin{array}{c|cc} 3 & -1 & 1 \\ \hline 1/3 & & \\ 2/3 & & \end{array}\right], \quad \left[\begin{array}{c} 3 \\ 2 \\ 1 \end{array}\right].$$

The Schur complement is

$$\left[\begin{array}{cc} 1 & -1 \\ 1 & -2 \end{array}\right] - \frac{1}{3}\left[\begin{array}{c} 1 \\ 2 \end{array}\right][-1 \;\; 1] = \left[\begin{array}{cc} 1 & -1 \\ 1 & -2 \end{array}\right] - \left[\begin{array}{cc} -1/3 & 1/3 \\ -2/3 & 2/3 \end{array}\right] = \left[\begin{array}{cc} 4/3 & -4/3 \\ 5/3 & -8/3 \end{array}\right],$$

so, at this point we have

$$A \sim \left[\begin{array}{c|cc} 3 & -1 & 1 \\ \hline 1/3 & 4/3 & -4/3 \\ 2/3 & 5/3 & -8/3 \end{array}\right] \sim \left[\begin{array}{c|cc} 3 & -1 & 1 \\ \hline 2/3 & 5/3 & -8/3 \\ 1/3 & 4/3 & -4/3 \end{array}\right], \quad \left[\begin{array}{c} 3 \\ 1 \\ 2 \end{array}\right],$$

because we interchanged $(R_2) \longleftrightarrow (R_3)$. Further, we have

$$A \sim \left[\begin{array}{c|c|c} 3 & -1 & 1 \\ \hline 2/3 & 5/3 & -8/3 \\ \hline 1/3 & 4/5 & \end{array}\right] \sim \left[\begin{array}{c|c|c} 3 & -1 & 1 \\ \hline 2/3 & 5/3 & -8/3 \\ \hline 1/3 & 4/5 & 4/5 \end{array}\right],$$

the last Schur complement being

$$-\frac{4}{3} - \frac{4}{5} \cdot \Big(-\frac{8}{3}\Big) = \frac{4}{5}.$$

So, we obtained

$$L = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 2/3 & 1 & 0 \\ 1/3 & 4/5 & 1 \end{array}\right], \; U = \left[\begin{array}{ccc} 3 & -1 & 1 \\ 0 & 5/3 & -8/3 \\ 0 & 0 & 4/5 \end{array}\right], \; P = \left[\begin{array}{ccc} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{array}\right].$$

Check that $PA = LU$.

■

Remark 1.3.

1. The computational cost for LU (and LUP) factorization is about the same as for Gaussian

elimination, $O(n^3)$ flops. However, for *tridiagonal* matrices, that cost drops to $O(n)$ operations. The *Thomas algorithm*, based on LUP decomposition is an efficient way of solving tridiagonal matrix systems. In addition, only three one-dimensional arrays for the three diagonals are needed to store the matrix. This means that very large systems can be solved rapidly and efficiently, and systems of order over $n = 10,000$ are not unusual in some applications, for example, in solving boundary value problems for differential equations.

2. More generally, a *band* or *banded* matrix is a sparse matrix whose non-zero entries are confined to a diagonal band, comprising of the main diagonal and zero or more diagonals on either side. If all matrix elements are zero outside a diagonally bordered band whose range is determined by constants $k_1, k_2 \geq 0$,

$$a_{ij} \quad = \quad 0, \text{ if } \; j < i - k_1 \; \text{ or } \; j > i + k_2$$

then the quantities k_1 and k_2 are called the *lower bandwidth* and *upper bandwidth*, respectively. The *bandwidth* of the matrix is then defined as

$$w \quad = \quad \max\{k_1, k_2\},$$

i.e., it is the number w such that

$$a_{ij} \quad = \quad 0, \text{ if } \; |i - j| > w.$$

It can be shown that LU factorization with partial pivoting for $n \times n$ banded matrices with bandwidth w requires $O(w^2 n)$ flops, while triangular solvers require $O(wn)$ flops.

1.2.3 QR Factorization

Definition 1.4. *A real square matrix Q is called **orthogonal** if*

$$Q \cdot Q^T \quad = \quad Q^T \cdot Q \; = \; I. \tag{1.3}$$

Theorem 1.5. *Let A be a real square matrix. Then there exist unique matrices Q and R such that*

$$A \quad = \quad QR, \tag{1.4}$$

*with Q orthogonal and R upper triangular with positive elements on the main diagonal, $r_{ii} > 0, \; \forall i$. The pair (Q, R) is called the **QR factorization** of A.*

Remark 1.6.
1. If $A = QR$, then solving the system $Ax = b$ is equivalent to solving the upper triangular systems

$$Rx \;=\; Q^T b. \tag{1.5}$$

2. Relation (1.3) automatically implies that any orthogonal matrix is nonsingular with $Q^{-1} \;=\; Q^T$. Orthogonal matrices are very useful in Numerical Analysis, as they preserve lengths, angles, and do not magnify errors.

1.2.4 Cholesky Factorization

Definition 1.7. *A real square matrix A is called **positive definite**, if*

$$x^T A\, x \;=\; \sum_{i,j=1}^{n} a_{ij} x_i x_j \;>\; 0,\; \forall x \in \mathbb{R}^n,\; x \neq 0. \tag{1.6}$$

Symmetric positive definite matrices can be decomposed into triangular factors twice as fast as general matrices. The standard algorithm for this, *Cholesky factorization*, is a variant of Gaussian elimination, which operates both on the left and the right of the matrix at once, preserving and exploiting the symmetry. These matrices have many interesting properties. Among them, the fact that a symmetric matrix is positive definite if and only if all its e-values are real and positive. Also, the e-vectors corresponding to distinct e-values of such a matrix, are orthogonal. Systems having symmetric positive definite matrices play an important role in Numerical Linear Algebra and its applications. Many matrices that arise in physical systems are symmetric and positive definite because of the fundamental physical laws.

Theorem 1.8. *Let A be a symmetric positive definite matrix. Then A has a unique **Cholesky factorization***

$$A \;=\; R^T R, \tag{1.7}$$

where R is an upper triangular matrix with positive elements on the main diagonal, $r_{ii} > 0,\; \forall i$.

Sketch of Proof. First off, let us use (1.6) for

$$x \;=\; e_1 \;=\; [1 \;\; 0 \;\; \ldots \;\; 0\,]^T.$$

We get

$$
\begin{aligned}
x^T A x &= [1\ 0\ \ldots\ 0] \begin{bmatrix} a_{11} & \ldots & a_{1n} \\ a_{21} & \ldots & a_{2n} \\ \vdots & & \vdots \\ a_{n1} & \ldots & a_{nn} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\
&= [1\ 0\ \ldots\ 0] \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix} = a_{11}.
\end{aligned}
$$

So, any positive definite matrix A has $a_{11} > 0$ and we can set $\alpha = \sqrt{a_{11}}$. Then we proceed in a similar way as with LU factorization, keeping in mind that A is also symmetric, so we work on the left and on the right at the same time.

$$
\begin{aligned}
A &= \begin{bmatrix} a_{11} & w^T \\ w & A' \end{bmatrix} \\
&= \begin{bmatrix} \alpha & 0 \\ w/\alpha & I_{n-1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & A' - ww^T/a_{11} \end{bmatrix} \begin{bmatrix} \alpha & w^T/\alpha \\ 0 & I_{n-1} \end{bmatrix} = R_1^T A_1 R_1.
\end{aligned}
$$

By induction, all matrices that appear during the factorization are positive definite and so, the process cannot break down. This procedure is repeated until

$$
A = \underbrace{R_1^T R_2^T \ldots R_n^T}_{R^T} \underbrace{R_n R_{n-1} \ldots R_1}_{R} = R^T R.
$$

The uniqueness follows from the fact that at each step, the value $\alpha = \sqrt{a_{11}}$ is uniquely determined from the factorization and once α is determined, all the rest of the R_i's are also uniquely determined.

□

Remark 1.9. This method requires only $n(n+1)/2$ storage locations for R, rather than the usual n^2 locations. Since only half the matrix needs to be stored, it follows that half of the arithmetic operations can be avoided and the number of operations is about $O\left(\frac{1}{3}n^3\right)$, rather than the number $O\left(\frac{2}{3}n^3\right)$ required for the usual LU decomposition.

Example 1.10. Find the Cholesky factorization (if it exists) of the matrix

$$A = \begin{bmatrix} 4 & 12 & -16 \\ 12 & 37 & -43 \\ -16 & -43 & 98 \end{bmatrix}.$$

Solution. The matrix is symmetric and its e-values are

$$0.0188, \;\; 15.5040, \;\; 123.4772,$$

real and positive. Therefore, A is positive definite and has a Cholesky decomposition. We will only work on the lower triangular part, the other will follow by symmetry. We have

$$A = \left[\begin{array}{c|cc} 4 & & \\ \hline 12 & 37 & \\ -16 & -43 & 98 \end{array}\right] \sim \left[\begin{array}{c|c} \sqrt{4} & \\ \hline 6 & \\ -8 & \end{array}\right].$$

The first Schur complement is

$$\begin{aligned} A' - ww^T/a_{11} &= \begin{bmatrix} 37 & \\ -43 & 98 \end{bmatrix} - \begin{bmatrix} 6 \\ -8 \end{bmatrix} [6 \;\; -8] \\ &= \begin{bmatrix} 37 & \\ -43 & 98 \end{bmatrix} - \begin{bmatrix} 36 & \\ -48 & 64 \end{bmatrix} = \begin{bmatrix} 1 & \\ 5 & 34 \end{bmatrix} \end{aligned}$$

and

$$A \sim \left[\begin{array}{c|cc} 2 & & \\ \hline 6 & 1 & \\ -8 & 5 & 34 \end{array}\right] \sim \left[\begin{array}{c|c|c} 2 & & \\ \hline 6 & \sqrt{1} & \\ \hline -8 & 5 & \end{array}\right] \sim \left[\begin{array}{c|c|c} 2 & & \\ \hline 6 & 1 & \\ \hline -8 & 5 & \sqrt{9} \end{array}\right] = \begin{bmatrix} 2 & & \\ 6 & 1 & \\ -8 & 5 & 3 \end{bmatrix},$$

with the last Schur complement being $34 - 5 \cdot 5 = 9$ and its square root 3. Then

$$R^T = \begin{bmatrix} 2 & 0 & 0 \\ 6 & 1 & 0 \\ -8 & 5 & 3 \end{bmatrix}, \;\; R = \begin{bmatrix} 2 & 6 & -8 \\ 0 & 1 & 5 \\ 0 & 0 & 3 \end{bmatrix} \;\; \text{and} \;\; A = R^T R.$$

■

2 Iterative Methods

The linear systems $Ax = b$ that occur in many applications can have very large orders ($10^3, 10^5, 10^6$). For such systems, the Gaussian elimination method (and consequent factorization methods) of the last section is often too expensive in either computation time or computer memory requirements, or possibly both. Moreover, the accumulation of round-off errors can sometimes prevent the numerical solution from being accurate. As an alternative, such linear systems are usually solved with *iteration methods*. In an iterative method, a sequence of progressively accurate iterates is produced to approximate the solution. Thus, in general, we do not expect to get the *exact* solution in a finite number of iteration steps, even if the round-off error effect is not taken into account. In the study of iteration methods, a most important issue is the *convergence property*. We will provide a framework for the convergence analysis of a general iteration method.

2.1 Jacobi and Gauss-Seidel Methods

We begin with some numerical examples that illustrate two popular iteration methods. Following that, we give a more general discussion of iteration methods.

Consider the linear system

$$\begin{array}{rcrcrcl} 9x_1 & + & x_2 & + & x_3 & = & b_1 \\ 2x_1 & + & 10x_2 & + & 3x_3 & = & b_2 \\ 3x_1 & + & 4x_2 & + & 11x_3 & = & b_3 \end{array} \tag{2.1}$$

We proceed as follows: in the equation numbered k, solve for x_k in terms of the remaining unknowns. In the above case,

$$\begin{array}{rcl} x_1 & = & \frac{1}{9}\big[b_1 - x_2 - x_3\big] \\ x_2 & = & \frac{1}{10}\big[b_2 - 2x_1 - 3x_3\big] \\ x_3 & = & \frac{1}{11}\big[b_3 - 3x_1 - 4x_2\big] \end{array} \tag{2.2}$$

Let

$$x^{(0)} = \big[x_1^{(0)}, x_2^{(0)}, x_3^{(0)}\big]^T$$

be an initial guess of the true solution x. Then define an iteration sequence:

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{9}\Big[b_1 - x_2^{(k)} - x_3^{(k)}\Big] \\ x_2^{(k+1)} &= \frac{1}{10}\Big[b_2 - 2x_1^{(k)} - 3x_3^{(k)}\Big] \\ x_3^{(k+1)} &= \frac{1}{11}\Big[b_3 - 3x_1^{(k)} - 4x_2^{(k)}\Big] \end{aligned} \tag{2.3}$$

for $k = 0, 1, \ldots$. This is called the **Jacobi iteration method** or the *method of simultaneous replacements (substitution).*

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	Error	Ratio
0	0	0	0	$2.00e+0$	
1	1.1111	1.9000	0	$1.00e+0$	0.500
2	0.9000	1.6778	-0.9939	$3.22e-1$	0.322
3	1.0351	2.0182	-0.8556	$1.44e-1$	0.448
4	0.9819	1.9496	-1.0162	$5.06e-2$	0.349
5	1.0074	2.0085	-0.9768	$2.32e-2$	0.462
6	0.9965	1.9915	-1.0051	$8.45e-3$	0.364
7	1.0015	2.0022	-0.9960	$4.03e-3$	0.477
8	0.9993	1.9985	-1.0012	$1.51e-3$	0.375
9	1.0003	2.0005	-0.9993	$7.40e-4$	0.489
10	0.9999	1.9997	-1.0003	$2.83e-4$	0.382
30	1.0000	2.0000	-1.0000	$3.01e-11$	0.447
31	1.0000	2.0000	-1.0000	$1.35e-11$	0.447

Table 1: Jacobi iteration for solving system (2.1)

In Table 1, we give a number of the iterations for the case that $b = [10, 19, 0]^T$, which yields the true solution

$$x = [1, 2, -1]^T$$

and where we started with the initial value $x^{(0)} = [0, 0, 0]^T$. In the table, the error is computed as

$$||x - x^{(k)}|| = \max_{1 \le i \le n} |x_i - x_i^{(k)}|.$$

Notice that the errors decrease as k increases and the values of the ratio eventually approach a

limiting constant of approximately 0.447 as k becomes much larger.

As another approach to the iterative solution of system (2.1) through the use of (2.2), we use *all* the information we obtain in the calculation of each new component. Specifically, let us define

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{9}\Big[b_1 - x_2^{(k)} - x_3^{(k)}\Big] \\ x_2^{(k+1)} &= \frac{1}{10}\Big[b_2 - 2x_1^{(k+1)} - 3x_3^{(k)}\Big] \\ x_3^{(k+1)} &= \frac{1}{11}\Big[b_3 - 3x_1^{(k+1)} - 4x_2^{(k+1)}\Big] \end{aligned} \tag{2.4}$$

for $k = 0, 1, \ldots$. This is called the **Gauss-Seidel iteration method** or the *method of successive replacements (substitution)*. This method is usually more rapidly convergent than the Jacobi method.

In Table 2, we give a number of iterations for solving the system (2.1). Compare these results to those in Table 1. The speed of convergence is much higher than with the Jacobi method (2.3). The values of the ratio, however, do not appear to approach a limiting value, even when looking at values of k larger than those in the table.

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	Error	Ratio
0	0	0	0	$2.00e+0$	
1	1.1111	1.6778	-0.9131	$3.22e-1$	0.161
2	1.0262	1.9687	-0.9958	$3.13e-2$	0.097
3	1.0030	1.9981	-1.0001	$3.00e-3$	0.096
4	1.0002	2.0000	-1.0001	$2.24e-4$	0.074
5	1.0000	2.0000	-1.0000	$1.65e-5$	0.074
6	1.0000	2.0000	-1.0000	$2.58e-6$	0.155

Table 2: Gauss-Seidel iteration for solving system (2.1)

2.2 Iterative Methods – General Theory

To understand the behavior of iteration methods, it is best to put them into a vector-matrix format. To this end, we recall some notions and results from Linear Algebra.

Definition 2.1. *Let* $A \in \mathbb{R}^{n\times n}$.
– The polynomial $p(\lambda) = \det(A - \lambda I_n)$ *is called the* ***characteristic polynomial of*** A *and the equa-*

tion $p(\lambda) = 0$ *the* ***characteristic equation of*** A.
– The roots of $p(\lambda)$ *are called* ***eigenvalues (e-values) of*** A.
– If $\lambda \in \mathbb{C}$ *is an e-value of* A*, a vector* $x \in \mathbb{R}^n, x \neq 0$ *satisfying* $(A - \lambda I_n)x = 0$ *is called an* ***eigenvector (e-vector) of*** A*, corresponding to the e-value* λ.
– The set of all e-values of A*, denoted by* $\lambda(A)$ *is called the* ***spectrum of*** A.
– The value $\rho(A) = \max\{|\lambda| \mid \lambda \in \lambda(A)\}$ *is called the* ***spectral radius of*** A.
– The value $tr(A) = a_{11} + \cdots + a_{nn}$ *is called the* ***trace of*** A.

Definition 2.2. *A* ***matrix norm*** *is a function* $|| \cdot || : \mathbb{R}^{n\times n} \to \mathbb{R}$ *satisfying the conditions:* $\forall A, B \in \mathbb{R}^{n\times n}, \forall \alpha \in \mathbb{R}$,
(i) $||A|| \geq 0,\ ||A|| = 0 \Leftrightarrow A = 0_n$.
(ii) $||\alpha A|| = |\alpha| \cdot ||A||$.
(iii) $||A + B|| \leq ||A|| + ||B||$.
(iv) $||AB|| \leq ||A|| \cdot ||B||$.

The first three conditions define *any* norm on a vector space. The fourth one is *specific* to matrix norms and it is necessary due to the fact that matrix multiplication is not done component-wise.

The easiest way of obtaining a matrix norm is from a vector one.

Definition 2.3. *Let* $|| \cdot ||$ *be a norm on* $\mathbb{R}^n$. *Then*

$$||A|| \ = \ \sup_{v\neq 0} \frac{||Av||}{||v||} \ = \ \sup_{||v||\leq 1} ||Av|| \ = \ \sup_{||v||=1} ||Av|| \tag{2.5}$$

is the ***natural (subordinate, induced)*** *matrix norm associated with the vector norm* $|| \cdot ||$.

Remark 2.4.
1. It can be easily checked that (2.5) satisfies the conditions of Definition 2.2 and is indeed a matrix norm.
2. A subordinate matrix norm is just a particular case for the norm of a linear mapping $A : \mathbb{R}^n \to \mathbb{R}^n$.
3. For any induced norm,

$$||I|| \ = \ 1. \tag{2.6}$$

Theorem 2.5. *Let $A \in \mathbb{R}^{n\times n}$. Then*
a)

$$\begin{aligned} ||A||_1 &= \sup_{v\neq 0} \frac{||Av||_1}{||v||_1} = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{ij}| \text{ (the } \textbf{Minkovski norm}\text{)}, \\ ||A||_\infty &= \sup_{v\neq 0} \frac{||Av||_\infty}{||v||_\infty} = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}| \text{ (the } \textbf{Chebyshev norm}\text{)}, \\ ||A||_2 &= \sup_{v\neq 0} \frac{||Av||_2}{||v||_2} = \sqrt{\rho(A^TA)} \text{ (the } \textbf{Euclidean norm}\text{)}. \end{aligned} \tag{2.7}$$

b) *The mapping $||\cdot||_F : \mathbb{R}^{n\times n} \to \mathbb{R}^n$ given by*

$$||A||_F = \left[\sum_{i=1}^{n}\sum_{j=1}^{n} |a_{ij}|^2\right]^{1/2} = \sqrt{tr(A^TA)} \tag{2.8}$$

*is a nonsubordinate ($||I_n||_F = \sqrt{n}$) matrix norm, called the **Frobenius norm**.*

Now, to solve the system $Ax = b$, for a nonsingular matrix $A \in \mathbb{R}^{n\times n}$, suppose there exist $T \in \mathbb{R}^{n\times n}$ and $c \in \mathbb{R}^n$, such that $I - T$ is invertible and the solution x of $Ax = b$ is the unique fixed point of the equation

$$x = Tx + c. \tag{2.9}$$

Let x^* be the solution and $x^{(0)}$ be an arbitrary vector (the initial approximation). Then, we use (2.9) to define an iterative method by

$$x^{(k+1)} = Tx^{(k)} + c,\ k \in \mathbb{N}. \tag{2.10}$$

The matrix T should be chosen such that the system $Tx = f$ is "easily solvable" (diagonal, triangular, tridiagonal, etc.)
Regarding the convergence of such methods, we have the following results from Calculus and Linear Algebra:

Lemma 2.6 (Geometric Series)**.** *Let $X \in \mathbb{R}^{n\times n}$. If $\rho(X) < 1$, then $(I - X)^{-1}$ exists and*

$$(I - X)^{-1} = I + X + \cdots + X^k + \ldots. \tag{2.11}$$

Conversely, if the series in (2.11) *is convergent, then* $\rho(X) < 1$.

Theorem 2.7. *The following are equivalent:*
a) *The iteration method* (2.10) *is convergent;*
b) $\rho(T) < 1$*;*
c) $||T|| < 1$ *for some matrix norm* $||\cdot||$.

Theorem 2.8. *If* $||T|| < 1$ *for some matrix norm* $||\cdot||$*, then the sequence* $\{x^{(k)}\}_{k\in\mathbb{N}}$ *defined in* (2.10) *converges to the unique fixed point* x^**, starting with any* $x^{(0)} \in \mathbb{R}^n$*, and the error bounds*

$$||x^* - x^{(k)}|| \;\leq\; \frac{||T||}{1-||T||}||x^{(k)} - x^{(k-1)}|| \tag{2.12}$$

and

$$||x^* - x^{(k)}|| \;\leq\; \frac{||T||^k}{1-||T||}||x^{(1)} - x^{(0)}|| \;\leq\; \frac{||T||}{1-||T||}||x^{(1)} - x^{(0)}||, \tag{2.13}$$

hold for every $k \in \mathbb{N}^*$.

Remark 2.9.
1. By Theorem 2.8, for a given error ε, we compute iterations until

$$||x^{(k)} - x^{(k-1)}|| \;\leq\; \frac{1-||T||}{||T||}\,\varepsilon. \tag{2.14}$$

2. In particular, if $||T|| < 1/2$, then

$$||x^* - x^{(k)}|| \;\leq\; ||x^{(k)} - x^{(k-1)}||$$

and the stopping criterion can be

$$||x^{(k)} - x^{(k-1)}|| \;\leq\; \varepsilon.$$

Now, *how* to actually find the matrix T and the scalar c, satisfying (2.9)? Suppose we can write A as

$$A \;=\; M - N. \tag{2.15}$$

This is called a *splitting* of A. If M is easily invertible (diagonal, triangular, etc.), then we can write

$$Ax = b \iff (M-N)x = b \iff Mx = Nx+b \iff x = M^{-1}Nx + M^{-1}b,$$

which is of the form (2.9), with

$$\begin{aligned} T &= M^{-1}N = M^{-1}(M-A) = I - M^{-1}A, \\ c &= M^{-1}b. \end{aligned}$$

We then define the iteration method by

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b,\ k \in \mathbb{N}, \tag{2.16}$$

with $x^{(0)}$ an arbitrary vector.

Assume A is nonsingular, with $a_{ii} \neq 0, i = \overline{1,n}$. We can write

$$A = D - L - U,$$

with

$$D = \begin{bmatrix} a_{11} & & & 0 \\ & a_{22} & & \\ & & \ddots & \\ 0 & & & a_{nn} \end{bmatrix},\ -L = \begin{bmatrix} 0 & & & 0 \\ a_{21} & 0 & & \\ \vdots & & \ddots & \\ a_{n1} & a_{n2} & \dots & 0 \end{bmatrix},\ -U = \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ & 0 & \dots & a_{2n} \\ & & \ddots & \vdots \\ 0 & & & 0 \end{bmatrix},$$

the diagonal, the lower triangular (without the diagonal) and the upper triangular (without the diagonal) parts of A.

For **Jacobi iteration**, take

$$\begin{aligned} M &= D,\ N = L+U, \text{ so} \\ T_J &= D^{-1}(L+U),\ c_J = D^{-1}b. \end{aligned}$$

The method is defined by

$$x^{(k+1)} = D^{-1}(L+U)x^{(k)} + D^{-1}b,\ k \in \mathbb{N},\ x^{(0)} \in \mathbb{R}^n, \tag{2.17}$$

or, component-wise,

$$x_i^{(k+1)} \;=\; \frac{1}{a_{ii}}\Big[b_i - \sum_{\substack{j=1\\ j\neq i}}^{n} a_{ij}x_j^{(k)}\Big],\ i=\overline{1,n}. \tag{2.18}$$

What can be said about the convergence of the method? By Theorem 2.7, we need a matrix norm such that $||T_J|| < 1$. Using Theorem 2.5, we want

$$||T_J||_\infty \;=\; \max_{1\leq i\leq n}\sum_{j=1}^{n}\Big|\frac{a_{ij}}{a_{ii}}\Big| \;<\; 1,$$

which means

$$|a_{ii}| \;>\; \sum_{\substack{j=1\\ j\neq i}}^{n}|a_{ij}|,\ i=\overline{1,n},$$

so, a *diagonally dominant* matrix A. Thus, for any diagonally dominant system, the Jacobi iterative method converges and the error estimates from Theorem 2.8 can be used. More generally, a necessary and sufficient condition for the convergence of the Jacobi iteration is

$$\rho(T_J) \;<\; 1.$$

For **Gauss-Seidel iteration**, we take

$$\begin{aligned} M \;&=\; D-L,\ \ N \;=\; U,\ \text{so}\\ T_{GS} \;&=\; (D-L)^{-1}U,\ c_{GS} \;=\; (D-L)^{-1}b. \end{aligned}$$

Then the method is defined by

$$x^{(k+1)} \;=\; (D-L)^{-1}Ux^{(k)} + (D-L)^{-1}b,\ k\in\mathbb{N},\ x^{(0)}\in\mathbb{R}^n, \tag{2.19}$$

and each component, by

$$x_i^{(k+1)} \;=\; \frac{1}{a_{ii}}\Big[b_i - \sum_{j=1}^{i-1}a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n}a_{ij}x_j^{(k)}\Big],\ i=\overline{1,n}. \tag{2.20}$$

Although it is not so trivial, it can be shown that for a diagonally dominant matrix, $||T_{GS}|| < 1$ and

so the Gauss-Seidel iterative method converges at least as fast as the Jacobi one.

Acceleration methods; SOR Method

Most iterative methods have a regular pattern in which the error decreases. This can often be used to *accelerate* the convergence. Rather than giving a general theory for the acceleration of iteration methods for solving $Ax = b$, we just describe an acceleration of the Gauss-Seidel method. This is one of the main cases of interest in applications.
We introduce an *acceleration parameter* ω and consider the following modification of the method:

$$\begin{aligned} M &= \frac{D}{\omega} - L, \;\; N = \Big(\frac{1-\omega}{\omega}D + U\Big), \text{ so} \\ T_\omega &= \Big(\frac{D}{\omega} - L\Big)^{-1}\Big(\frac{1-\omega}{\omega}D + U\Big), \; c_\omega = \Big(\frac{D}{\omega} - L\Big)^{-1} b. \end{aligned}$$

The acceleration method is defined by

$$x_i^{(k+1)} = \frac{\omega}{a_{ii}}\Big[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\Big] + (1-\omega)x_i^{(k)}, \; i = \overline{1,n}. \tag{2.21}$$

This is called the *relaxation method*. We have the following cases:
– $\omega < 1$ is called *subrelaxation*;
– $\omega = 1$ is the Gauss-Seidel method;
– $\omega > 1$ is called *overrelaxation*, the **SOR method**, an abbreviation for *successive overrelaxation*.

It can be shown that, if $a_{ii} \neq 0, \; i = \overline{1,n}$, then $\rho(T_\omega) \geq |\omega - 1|$. Thus, by Theorem 2.7, a necessary condition for the convergence of the SOR method is

$$0 < \omega < 2. \tag{2.22}$$

Also, the following holds:

Theorem 2.10 (**Ostrowski-Reich**). *If A is a positive definite matrix and $0 < \omega < 2$, then the SOR iteration method converges for any choice of the initial approximation $x^{(0)} \in \mathbb{R}^n$.*

The parameter ω is to be chosen to minimize the error, in order to make $x^{(k)}$ converge to x as rapidly as possible. It was found that the optimal value for ω is

$$\omega^* = \frac{2}{1 + \sqrt{1 - \big(\rho(T_J)\big)^2}}. \tag{2.23}$$

Remark 2.11. Iterative methods are rarely used for systems of small order, because they are inefficient, since the time needed to get the desired precision exceeds the time required for Gaussian elimination. But for large systems ($n \geq 10^3$), especially for sparse matrices, they can really make a huge difference in the implementation and computational cost.

3 Conditioning of a Linear System

Recall (from Lecture 1) the issue of *stability* (sensitivity to errors/perturbations) and *conditioning* (a measure of that sensitivity) of a mathematical problem.

For a general problem of the type

$$y = f(x),\ f : \mathbb{R}^m \to \mathbb{R}^n,$$

we define

$$\gamma_{ij} = (\mathrm{cond}_{ij} f)(x) = \frac{x_i \dfrac{\partial f_j}{\partial x_i}}{f_j(x)},\ i = \overline{1, m},\ j = \overline{1, n}$$

and

$$\Gamma(x) = \left[\gamma_{ij}\right] = \begin{bmatrix} \dfrac{x_1 \dfrac{\partial f_1}{\partial x_1}}{f_1(x)} & \cdots & \dfrac{x_m \dfrac{\partial f_1}{\partial x_m}}{f_1(x)} \\ \vdots & & \vdots \\ \dfrac{x_1 \dfrac{\partial f_n}{\partial x_1}}{f_n(x)} & \cdots & \dfrac{x_m \dfrac{\partial f_n}{\partial x_m}}{f_n(x)} \end{bmatrix}, \tag{3.1}$$

called the **conditioning matrix**. Then, the **condition number** of f at x is defined by

$$(\mathrm{cond}\ f)(x) = ||\Gamma(x)||, \tag{3.2}$$

for a matrix norm $||\cdot||$. If f is a linear function, then

$$(\mathrm{cond}\ f)(x) = \frac{||x|| \left|\left| \dfrac{\partial f}{\partial x} \right|\right|}{||f(x)||}. \tag{3.3}$$

Now, for a linear system, we have $A \in \mathbb{R}^{n\times n}$, nonsingular and $b \in \mathbb{R}^n$ given. The problem is finding $x \in \mathbb{R}^n$ such that

$$Ax \;=\; b.$$

So, in this case, the input data consists of A and b and the output data is the vector x. Then, we can regard this problem as

$$x \;=\; f(b) \;=\; A^{-1}b,\; f : \mathbb{R}^n \to \mathbb{R}^n. \tag{3.4}$$

Since f is linear and $\dfrac{\partial f}{\partial b} \;=\; A^{-1}$, the condition number is

$$(\text{cond } f)(b) \;=\; \frac{||b||\,||A^{-1}||}{||A^{-1}b||} \;=\; \frac{||Ax||\,||A^{-1}||}{||x||} \;=\; ||A^{-1}||\frac{||Ax||}{||x||}.$$

Then

$$\max_{\substack{b\in\mathbb{R}^n\\ b\neq 0}} (\text{cond} f)(b) \;=\; ||A^{-1}||\,\max_{\substack{x\in\mathbb{R}^n\\ x\neq 0}} \frac{||Ax||}{||x||} \;=\; ||A^{-1}||\,||A||.$$

This is the **conditioning number of the matrix** A (and of the system):

$$\text{cond}(A) \;=\; ||A||\,||A^{-1}||. \tag{3.5}$$

If the matrix A is singular, by convention, $\text{cond}(A) \;=\; \infty$.

The number $\text{cond}(A)$ will vary with the norm being used, but it is always bounded below by one, since

$$1 \;=\; ||I|| \;=\; ||AA^{-1}|| \;\leq\; ||A||\,||A^{-1}|| \;=\; \text{cond}(A).$$

If the condition number is nearly 1, then small relative perturbations in b will lead to similarly small relative perturbations in the solution x. But if $\text{cond}(A)$ is large, then there may be small relative perturbations of b that will lead to large relative perturbations in x.

Example 3.1. (Ill-conditioned Matrices)

1. Hilbert Matrix

$$H_n = \left[\frac{1}{i+j-1}\right]_{i,j=\overline{1,n}} = \begin{bmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \vdots & & & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \cdots & \frac{1}{2n-1} \end{bmatrix}. \tag{3.6}$$

This is a symmetric and positive definite matrix, so it is nonsingular. However, it is very ill-conditioned, and increasingly so as n increases.

n	$\text{cond}_2(H_n)$
10	$1.6e+13$
20	$2.45e+28$
40	$7.65e+58$

Table 3: Condition numbers of Hilbert matrix

2. Vandermonde Matrix

$$V_n = \begin{bmatrix} 1 & 1 & \dots & 1 \\ t_1 & t_2 & \dots & t_n \\ \vdots & \vdots & \ddots & \vdots \\ t_1^{n-1} & t_2^{n-1} & \dots & t_n^{n-1} \end{bmatrix}. \tag{3.7}$$

For $t_i = \frac{1}{i}, i = \overline{1,n}$, it can be shown that

$$\text{cond}_\infty(V_n) > n^{n+1}.$$