

### 1.2.3 QR Factorization

**Definition 1.1.** A real square matrix  $Q$  is called **orthogonal** if

$$Q \cdot Q^T = Q^T \cdot Q = I. \quad (1.1)$$

**Theorem 1.2.** Let  $A$  be a real square matrix. Then there exist unique matrices  $Q$  and  $R$  such that

$$A = QR, \quad (1.2)$$

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 1.3.**

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

$$Rx = Q^T b. \quad (1.3)$$

2. Relation (1.1) 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 1.4.** A real square matrix  $A$  is called **positive definite**, if

$$x^T A x = \sum_{i,j=1}^n a_{ij} x_i x_j > 0, \quad \forall x \in \mathbb{R}^n, \quad x \neq 0. \quad (1.4)$$

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 1.5.** *Let  $A$  be a symmetric positive definite matrix. Then  $A$  has a unique **Cholesky factorization***

$$A = R^T R, \quad (1.5)$$

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 (1.4) for

$$x = e_1 = [1 \ 0 \ \dots \ 0]^T.$$

We get

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

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.

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

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

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

The uniqueness follows from the fact that at each step, the value  $\alpha = \sqrt{a_{11}}$  is uniquely determined

from the factorization and once  $\alpha$  is determined, all the rest of the  $R_i$ 's are also uniquely determined.  $\square$

**Remark 1.6.** 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 1.7.** 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

$$0.0188, \quad 15.5040, \quad 123.4772,$$

real and positive. Therefore,  $A$  is positive definite and has a Cholesky decomposition. We have

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

The first Schur complement is

$$\begin{aligned} A' - ww^*/a_{11} &= \begin{bmatrix} 37 & -43 \\ -43 & 98 \end{bmatrix} - \begin{bmatrix} 6 \\ -8 \end{bmatrix} [6 \quad -8] \\ &= \begin{bmatrix} 37 & -43 \\ -43 & 98 \end{bmatrix} - \begin{bmatrix} 36 & -48 \\ -48 & 64 \end{bmatrix} = \begin{bmatrix} 1 & 5 \\ 5 & 34 \end{bmatrix} \end{aligned}$$

and

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

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

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

■

## 2 Iterative Methods

The linear systems  $Ax = b$  that occur in many applications can have a very large order ( $10^3, 10^5, 10^6$ ). 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 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

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

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

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

Let

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

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

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

for  $k = 0, 1, \dots$ . 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

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

In the table, the error is computed as

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

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

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

for  $k = 0, 1, \dots$ . 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 faster 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 equation  $p(\lambda) = 0$  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  $\lambda$ .
- 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  $\text{tr}(A) = a_{11} + \cdots + a_{nn}$  is called the **trace** of  $A$ .

**Definition 2.2.** A **matrix norm** is a function  $\|\cdot\| : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$  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

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

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  $A : \mathbb{R}^n \rightarrow \mathbb{R}^n$ .
3. For any induced norm,

$$\|I\| = 1. \quad (2.6)$$

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

a)

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

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

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

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

$$x = Tx + c. \quad (2.9)$$

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

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

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

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



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*

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

and

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

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

**Remark 2.9.**

1. By Theorem 2.8, for a given error  $\varepsilon$ , we compute iterations until

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

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

$$A = M - N. \quad (2.15)$$

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

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

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

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

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

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

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

For **Jacobi iteration**, take

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

The method is defined by

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

or, component-wise,

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

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

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

which means

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

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

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

Then the method is defined by

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

and each component, by

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

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*  $\omega$  and consider the following modification of the method:

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

The acceleration method is defined by

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

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

$$0 < \omega < 2. \quad (2.22)$$

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  $\omega$  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  $\omega$  is

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

**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 Condition of a Linear System

A number of mathematical problems have solutions that are quite sensitive to small computational errors, for example rounding errors. To deal with this phenomenon, we use the concepts of *stability* and *condition number*.

Consider a general problem of the type

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

We are interested in how “sensitive” is  $f$  to small perturbations of  $x$ , that is, do small perturbations of  $x$  lead to small perturbations in  $y$  – which means the problem is *stable*, or not – *unstable* problem. In the latter case, approximation is not very helpful. We would like to “measure” the degree of sensitivity (how stable or unstable a problem is) by a single number, called the **condition number** of  $f$  at  $x$ .

The function  $f$  is assumed to be evaluated exactly, with infinite precision, as we perturb  $x$ . The condition number of  $f$ , therefore, is an inherent property of the map  $f$  and does not depend on any algorithmic considerations concerning its implementation.

Recall from the propagation of error formula (in Lecture 1) that when  $f : \mathbb{R}^m \rightarrow \mathbb{R}$ , we found

$$\delta f \approx \sum_{i=1}^m x_i \frac{f'_{x_i}}{f} \delta x_i.$$

Now, for the general case  $f : \mathbb{R}^m \rightarrow \mathbb{R}^n, x = [x_1, \dots, x_m]^T \in \mathbb{R}^m, f(x) = (f_1(x), \dots, f_n(x))^T \in \mathbb{R}^n$ , let

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

and

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

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

$$(\text{cond } f)(x) = \|\Gamma(x)\|, \quad (3.2)$$

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

$$(\text{cond } f)(x) = \frac{\|x\| \left\| \frac{\partial f}{\partial x} \right\|}{\|f(x)\|}. \quad (3.3)$$

If the condition number of a problem is large ( $(\text{cond } f)(x) \gg 1$ ), then even for small (relative) errors in the input data, we can expect large errors in the output data. Such problems are called *ill-conditioned* problems. There is no clear line between ill- and well-conditioned problems, it all depends on precision specifications.

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

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

Since  $f$  is linear and  $\frac{\partial f}{\partial b} = A^{-1}$ , 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):

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

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

$$H_n = \begin{bmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \vdots & & & \vdots \\ \frac{1}{n} & \frac{1}{n+1} & \cdots & \frac{1}{2n-1} \end{bmatrix}. \quad (3.6)$$

This is a symmetric 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

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

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

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