Chapter 3
Numerical linear algebra

Review of linear algebra

We consider the following system of linear equations which has n unknowns $x_1, \ldots, x_n$.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \quad \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases}$$

We can write the system as

$$A\mathbf{x} = \mathbf{b},$$

where

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}, \qquad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}.$$

We note that the ith row of $Ax = b$ is written as

$$\sum_{j=1}^{n} a_{ij}x_j = b_i,$$

where j is the column index.

If A is invertible, we can obtain $\mathbf{x}$ by

$$\mathbf{x} = A^{-1}\mathbf{b}.$$

In Matlab, you can type `x=A\b`. The next theorem states when A is invertible.[1]

Fall 2013 Math 471 Sec 2
Introduction to Numerical Methods

Manabu Machida (University of Michigan)

[1] Math 214/417/419

Theorem 1. *The following conditions are equivalent.*

1. $A\mathbf{x} = \mathbf{b}$ *has a unique solution for* $\forall \mathbf{b}$.
2. A *is invertible.*
3. $\det A \neq 0$.
4. $A\mathbf{x} = \mathbf{0}$ *has the unique solution* $\mathbf{x} = \mathbf{0}$.
5. *The columns of* A *are linearly independent.*
6. *The eigenvalues of* A *are nonzero.*

Suppose A is invertible. Note that $\mathbf{x} = A^{-1}\mathbf{b}$ is not the best way to numerically compute $\mathbf{x}$. There are two types of methods for solving $A\mathbf{x} = \mathbf{b}$: direct methods and iterative methods.

Gaussian elimination with back substitution

Suppose A is an upper triangular matrix. In 3, we will consider other matrices. The equation $A\mathbf{x} = \mathbf{b}$ is written as

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ \qquad\quad a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \qquad\qquad\qquad\quad \vdots \\ \qquad\qquad\qquad\quad a_{nn}x_n = b_n \end{cases}$$

We can readily obtain $x_n, x_{n-1}, \ldots, x_1$ by back substitution:

$$\begin{aligned} x_n &= \frac{b_n}{a_{nn}}, \\ x_{n-1} &= \frac{b_{n-1} - a_{n-1,n}x_n}{a_{n-1,n-1}}, \\ &\vdots \\ x_1 &= \frac{b_1 - (a_{12}x_2 + \cdots + a_{1n}x_n)}{a_{11}}. \end{aligned}$$

This procedure can be implemented as follows.

```
1 x(n)=b(n)/a(n,n)
2 for i=n-1:-1:1   % i: row index
3   tmp=b(i)
4   for j=i+1:n    % j: column index
5     tmp=tmp-a(i,j)*x(j)
```

```
6 ||    end
7 ||    x(i)=tmp/a(i,i)
8 || end
```

Operation counts (back substitution)

Let us consider how much work it takes to perform the back substitution. As a measure of work, we will use the number of arithmetic operations being performed.[2][3]

$$\begin{aligned}
&\text{the number of divisions } = n,\\
&\text{the number of multiplications} = \text{the number of additions}\\
&\qquad\qquad = 1+2+\cdots+(n-1)\\
&\qquad\qquad = \frac{1}{2}n(n-1) \sim \frac{1}{2}n^2 \text{ for large } n,
\end{aligned}$$

where we used

$$\begin{aligned}
&S = 1+2+\cdots+(n-1),\\
&2S = [1+2+\cdots+(n-1)]+[(n-1)+\cdots+2+1] = n+n+\cdots+n = n(n-1).\\
&\therefore \quad S = \frac{n(n-1)}{2}.
\end{aligned}$$

Hence the leading order term in the operation count for back substitution is n^2.

Elementary row operations

Elementary row operations consist of the following three operations.

- Interchanging two rows.
- Multiplying any row by a nonzero constant.
- Subtracting a multiple of one row from another row.

---

[2] On ancient computers, multiplication and division were significantly more time-consuming than addition and subtraction. Division was the slowest operation and we tend to write `0.5` instead of `1/2.0`. On modern architectures, however, multiplication is no more expensive and division is not twice as expensive as addition and subtraction. So, here we break from tradition and just count the total number of arithmetic operations.

[3] The word FLOPs (**FL**oating-point **OP**eration**s**) is sometimes used. FLOPS stands for (**FL**oating-point **O**perations **P**er **Second**).

Gaussian elimination is to transform the augmented matrix $(A\ \mathbf{b})$ into upper triangular form by repeatedly applying the third elementary row operation. Solutions to the system of equations $A\mathbf{x} = \mathbf{b}$ don't change by elementary row operations.

Example 1. Let us solve

$$\begin{cases} 2x_1 - x_2 = 1, \\ -x_1 + 2x_2 - x_3 = 0, \\ -x_2 + 2x_3 = 1. \end{cases} \tag{3.1}$$

We write the augmented matrix as

$$\left(\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right) = \left(\begin{array}{ccc|c} 2 & -1 & 0 & 1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & 1 \end{array}\right).$$

By the operation (2nd row) $-\frac{a_{21}}{a_{11}}$(1st row), we get

$$\left(\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a_{22} - m_{21}a_{12} & a_{23} - m_{21}a_{13} & b_2 - m_{21}b_1 \\ 0 & a_{32} & a_{33} & b_3 \end{array}\right) = \left(\begin{array}{ccc|c} 2 & -1 & 0 & 1 \\ 0 & \frac{3}{2} & -1 & \frac{1}{2} \\ 0 & -1 & 2 & 1 \end{array}\right),$$

where $m_{21} = a_{21}/a_{11} = -1/2$. We refer to m_{21} as a multiplier. Now we have

$$\left(\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a_{22} & a_{23} & b_2 \\ 0 & a_{32} & a_{33} & b_3 \end{array}\right) = \left(\begin{array}{ccc|c} 2 & -1 & 0 & 1 \\ 0 & \frac{3}{2} & -1 & \frac{1}{2} \\ 0 & -1 & 2 & 1 \end{array}\right).$$

Then by the operation (3rd row) $- m_{32}$(2nd row), where the multiplier $m_{32} = a_{32}/a_{22} = -1/\frac{3}{2} = -2/3$, we obtain

$$\left(\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a_{22} & a_{23} & b_2 \\ 0 & a_{32} - m_{32}a_{22} & a_{33} - m_{32}a_{23} & b_3 - m_{32}b_2 \end{array}\right) = \left(\begin{array}{ccc|c} 2 & -1 & 0 & 1 \\ 0 & \frac{3}{2} & -1 & \frac{1}{2} \\ 0 & 0 & \frac{4}{3} & \frac{4}{3} \end{array}\right).$$

Finally we obtain the upper triangular matrix

$$\left(\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & a_{22} & a_{23} & b_2 \\ 0 & 0 & a_{33} & b_3 \end{array}\right) = \left(\begin{array}{ccc|c} 2 & -1 & 0 & 1 \\ 0 & \frac{3}{2} & -1 & \frac{1}{2} \\ 0 & 0 & \frac{4}{3} & \frac{4}{3} \end{array}\right),$$

or the system

$$\begin{cases} 2x_1 - x_2 = 1, \\ \frac{3}{2}x_2 - x_3 = \frac{1}{2}, \\ \frac{4}{3}x_3 = \frac{4}{3}. \end{cases} \tag{3.2}$$

In a general $n \times n$ case, reduction to the upper triangular form can be implemented as follow.

```
1 for k=1:n-1   % k: step index
2 for i=k+1:n
3   m(i,k)=a(i,k)/a(k,k)
4   for j=k+1:n
5     a(i,j)=a(i,j)-m(i,k)*a(k,j)
6   end
7   b(i)=b(i)-m(i,k)*b(k)
8 end
```

Note that $\mathtt{a}(\mathtt{k},\mathtt{k}) \neq 0$ was assumed. This point will be discussed later.

The element $\mathtt{a}(\mathtt{k},\mathtt{k})$ in the kth step is called a pivot (these are the diagonal elements in the last step). In the previous example, the pivots are $2, 3/2, 4/3$.

Operation counts (Gaussian elimination)

The leading order term comes from line 5 of the above code.

$$\left.\begin{array}{l} k=1 \Longrightarrow 2(n-1)^2 \\ k=2 \Longrightarrow 2(n-2)^2 \\ \quad\vdots \\ k=n-2 \Longrightarrow 2\cdot 2^2 \\ k=n-1 \Longrightarrow 2\cdot 1^2 \end{array}\right\} \Longrightarrow 2\cdot\sum_{k=1}^{n-1} k^2 = 2\cdot\frac{1}{6}(n-1)n(2n-1),$$

where we used $\sum_{k=1}^{n} k^2 = \frac{1}{6}n(n+1)(2n+1)$, which can be derived using $n^3 = n^3 - (n-1)^3 + (n-1)^3 + \cdots - 2^3 + 2^3 - 1^3 + 1^3$. Hence the operation count for Gaussian elimination is $\frac{2}{3}n^3$.

Pivoting

Here we consider cases where one of the pivots is zero.

Partial pivoting

Consider the reduced matrix at the beginning of step k:

$$\left(\begin{array}{ccccc:c} a_{11} & \cdots & a_{1k} & \cdots & a_{1n} & b_1 \\ & \ddots & \vdots & & \vdots & \vdots \\ & & a_{kk} & \cdots & a_{kn} & b_k \\ & & \vdots & & \vdots & \vdots \\ & & a_{nk} & \cdots & a_{nn} & b_n \end{array}\right).$$

If $a_{kk} = 0$, find index l such that $|a_{lk}| = \max\{|a_{ik}|; k \le i \le n\}$, then interchange row l and row k, and proceed with the elimination.

If A is invertible, then Gaussian elimination with partial pivoting does not break down.[4]

In practice, pivoting is often applied even when the pivot is nonzero.

Example 2. The exact solutions of the following problem are $x_1 = x_2 = 1$.

$$\left(\begin{array}{cc:c} \varepsilon & 1 & 1+\varepsilon \\ 1 & 1 & 2 \end{array}\right) \to \left(\begin{array}{cc:c} \varepsilon & 1 & 1+\varepsilon \\ 0 & 1-\frac{1}{\varepsilon} & 1-\frac{1}{\varepsilon} \end{array}\right) \quad \therefore \quad \begin{cases} x_1 = \dfrac{1+\varepsilon-1}{\varepsilon} = 1, \\ x_2 = \dfrac{1-\frac{1}{\varepsilon}}{1-\frac{1}{\varepsilon}} = 1. \end{cases}$$

Since $1/\varepsilon$ is large, by taking the effect of roundoff error into account, we can write

$$\left(\begin{array}{cc:c} \varepsilon & 1 & 1 \\ 0 & -\frac{1}{\varepsilon} & -\frac{1}{\varepsilon} \end{array}\right) \quad \therefore \quad \begin{cases} x_1 = \dfrac{1-1}{\varepsilon} = 0, \\ x_2 = \dfrac{-\frac{1}{\varepsilon}}{-\frac{1}{\varepsilon}} = 1. \end{cases}$$

Thus, an inaccurate solution is obtained. Now we apply pivoting.

$$\left(\begin{array}{cc:c} 1 & 1 & 2 \\ \varepsilon & 1 & 1+\varepsilon \end{array}\right) \to \left(\begin{array}{cc:c} 1 & 1 & 2 \\ 0 & 1 & 1 \end{array}\right) \quad \therefore \quad \begin{cases} x_1 = 1, \\ x_2 = 1. \end{cases}$$

In this case, the computed solutions are accurate.

Vector and matrix norms

To do error analysis, it is convenient to introduce the size of a vector and the size of a matrix.

Definition 1. A vector norm is a function $\|\mathbf{x}\|$ satisfying the following properties.

[4] Math 571

1. $\|\mathbf{x}\| \geq 0$, and $\|\mathbf{x}\| = 0 \Leftrightarrow \mathbf{x} = \mathbf{0}$.
2. $\|\alpha\mathbf{x}\| = |\alpha| \, \|\mathbf{x}\|, \quad \alpha \in \mathbb{C}$.
3. $\|\mathbf{x}+\mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$.

We can think of different norms.

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}, \; 1 \leq p < \infty, \qquad \|\mathbf{x}\|_\infty = \max\{|x_i|; i = 1, \ldots, n\}.$$

For example,

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|, \qquad \|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} = |\mathbf{x}|.$$

Example 3. If $\mathbf{x} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$, then

$$\|\mathbf{x}\|_2 = \sqrt{5}, \quad \|\mathbf{x}\|_\infty = 2.$$

Definition 2. The matrix norm of a matrix A is given by

$$\|A\| = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|}.$$

We can regard $\mathbf{x}$ as input and $A\mathbf{x}$ as output. Then the ratio $\|A\mathbf{x}\|/\|\mathbf{x}\|$ (the amplification factor) shows how much the input gets large. The matrix norm satisfies the following properties.

1. $\|A\| \geq 0$, and $\|A\| = 0 \Leftrightarrow A = 0$.
2. $\|\alpha A\| = |\alpha| \, \|A\|, \quad \alpha \in \mathbb{C}$.
3. $\|A + B\| \leq \|A\| + \|B\|$.
4. $\|A\mathbf{x}\| \leq \|A\| \, \|\mathbf{x}\|$.
5. $\|AB\| \leq \|A\| \, \|B\|$.

The second half of the first property is proved as

$$\|A\| = 0 \Leftrightarrow \|A\mathbf{x}\| = 0 \text{ for all } \mathbf{x} \neq \mathbf{0} \Leftrightarrow A\mathbf{x} = \mathbf{0} \text{ for all } \mathbf{x} \neq \mathbf{0} \Leftrightarrow A = 0.$$

The last property can be proved as follows.

$$\|AB\mathbf{x}\| \leq \|A\| \, \|B\mathbf{x}\| \leq \|A\| \, \|B\| \, \|\mathbf{x}\|. \qquad \therefore \|AB\| \leq \|A\| \, \|B\|.$$

Theorem 2 (Maximum absolute row sum).

$$\|A\|_\infty = \max_i \sum_j |a_{ij}|.$$

Proof.

$$\|A\mathbf{x}\|_\infty = \max_i \left|\sum_{j=1}^n a_{ij}x_j\right| \le \max_i \sum_{j=1}^n |a_{ij}|\,|x_j| \le \max_j |x_j| \max_i \sum_{j=1}^n |a_{ij}|$$
$$= \|\mathbf{x}\|_\infty \max_i \sum_{j=1}^n |a_{ij}|.$$

Hence

$$\|A\|_\infty \le \max_i \sum_{j=1}^n |a_{ij}|. \tag{3.3}$$

Define $\mathbf{y}$ ($\|\mathbf{y}\|_\infty = 1$) by

$$y_j = \begin{cases} 1, & \text{if } a_{ij} \ge 0, \\ -1, & \text{if } a_{ij} < 0. \end{cases}$$

We have

$$\|A\mathbf{y}\|_\infty = \max_i \left|\sum_{j=1}^n a_{ij}y_j\right| = \max_i \sum_{j=1}^n |a_{ij}| = \|\mathbf{y}\|_\infty \max_i \sum_{j=1}^n |a_{ij}|.$$

Hence

$$\|A\|_\infty \ge \max_i \sum_{j=1}^n |a_{ij}|. \tag{3.4}$$

Equations (3.3) and (3.4) yields

$$\|A\|_\infty = \max_i \sum_{j=1}^n |a_{ij}|.$$

□

Example 4. Let us consider $A = \begin{pmatrix} 3 & -4 \\ 1 & 0 \end{pmatrix}$. According to the above theorem, we obtain

$$\|A\|_\infty = \max\{|3| + |-4|, |1| + |0|\} = 7.$$

Let us try a few $\mathbf{x}$'s.

$$A\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \quad \Rightarrow \quad \frac{\|A\mathbf{x}\|_\infty}{\|\mathbf{x}\|_\infty} = \frac{3}{1} = 3,$$
$$A\begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} -4 \\ 0 \end{pmatrix} \quad \Rightarrow \quad \frac{\|A\mathbf{x}\|_\infty}{\|\mathbf{x}\|_\infty} = \frac{4}{1} = 4,$$
$$A\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \quad \Rightarrow \quad \frac{\|A\mathbf{x}\|_\infty}{\|\mathbf{x}\|_\infty} = \frac{1}{1} = 1,$$
$$A\begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 7 \\ 1 \end{pmatrix} \quad \Rightarrow \quad \frac{\|A\mathbf{x}\|_\infty}{\|\mathbf{x}\|_\infty} = \frac{7}{1} = 7.$$

We also have

$$\|A\|_1 = \max_j \sum_i |a_{ij}| \quad \text{(maximum absolute column sum)},$$
$$\|A\|_2 = \sqrt{\text{max eigenvalue of } A^*A},$$

where A^* is the conjugate transpose of A.

Error analysis

We consider $A\mathbf{x} = \mathbf{b}$. Let $\mathbf{x}$ and $\tilde{\mathbf{x}}$ denote the exact solution and an approximate solution. The error $\mathbf{e} = \mathbf{x} - \tilde{\mathbf{x}}$ is usually unknown. The residual $\mathbf{r} = \mathbf{b} - A\tilde{\mathbf{x}}$ can be computed.

Example 5. Even if $\|\mathbf{r}\|$ is small, there is no guarantee that $\|\mathbf{e}\|$ is small. Consider

$$A = \begin{pmatrix} 1.01 & 0.99 \\ 0.99 & 1.01 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \quad \Rightarrow \quad \mathbf{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Suppose we get an approximate solution $\tilde{\mathbf{x}}_1 = \begin{pmatrix} 1.01 \\ 1.01 \end{pmatrix}$. We obtain

$$\mathbf{e}_1 = \mathbf{x} - \tilde{\mathbf{x}}_1 = \begin{pmatrix} -0.01 \\ -0.01 \end{pmatrix} \quad \Rightarrow \quad \|\mathbf{e}_1\| = 0.01,$$
$$\mathbf{r}_1 = \mathbf{b} - A\tilde{\mathbf{x}}_\mathbf{1} = \begin{pmatrix} 2 \\ 2 \end{pmatrix} - \begin{pmatrix} 2.02 \\ 2.02 \end{pmatrix} = \begin{pmatrix} -0.02 \\ -0.02 \end{pmatrix} \quad \Rightarrow \quad \|\mathbf{r}_1\| = 0.02.$$

Next let us suppose we get an approximate solution $\tilde{\mathbf{x}}_2 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$. We obtain

$$\mathbf{e}_2 = \mathbf{x} - \tilde{\mathbf{x}}_2 = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \quad \Rightarrow \quad \|\mathbf{e}_2\| = 1,$$
$$\mathbf{r}_2 = \mathbf{b} - A\tilde{\mathbf{x}}_\mathbf{2} = \begin{pmatrix} 2 \\ 2 \end{pmatrix} - \begin{pmatrix} 2.02 \\ 1.98 \end{pmatrix} = \begin{pmatrix} -0.02 \\ 0.02 \end{pmatrix} \quad \Rightarrow \quad \|\mathbf{r}_2\| = 0.02.$$

We want to know how large $\|\mathbf{e}\|$ can be.

Theorem 3.

$$\frac{\|\mathbf{e}\|}{\|\mathbf{x}\|} \leq \kappa(A) \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|},$$

where $\kappa(A) = \|A\| \, \|A^{-1}\|$ *is the condition number.*

Proof. We have

$$\|\mathbf{b}\| = \|A\mathbf{x}\| \le \|A\|\,\|\mathbf{x}\| \quad \Rightarrow \quad \|\mathbf{x}\| \ge \|\mathbf{b}\|/\|A\|.$$

Note that

$$A\mathbf{e} = A(\mathbf{x}-\tilde{\mathbf{x}}) = A\mathbf{x} - A\tilde{\mathbf{x}} = b - A\tilde{\mathbf{x}} = \mathbf{r} \quad \Rightarrow \quad A\mathbf{e} = \mathbf{r}.$$

Hence,

$$\mathbf{e} = A^{-1}\mathbf{r} \quad \Rightarrow \quad \|\mathbf{e}\| = \|A^{-1}\mathbf{r}\| \le \|A^{-1}\|\,\|\mathbf{r}\|.$$

Finally we obtain

$$\frac{\|\mathbf{e}\|}{\|\mathbf{x}\|} \le \frac{\|A^{-1}\|\,\|\mathbf{r}\|}{\|\mathbf{b}\|/\|A\|} = \frac{\|A\|\,\|A^{-1}\|\,\|\mathbf{r}\|}{\|\mathbf{b}\|} = \kappa(A)\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}.$$

□

If we write $A\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$, then

$$\frac{\|\mathbf{x}-\tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \le \kappa(A)\frac{\|\mathbf{b}-\tilde{\mathbf{b}}\|}{\|\mathbf{b}\|}.$$

If we write $\tilde{A}\tilde{\mathbf{x}} = \mathbf{b}$, then

$$\frac{\|\mathbf{x}-\tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \le \kappa(A)\frac{\|A-\tilde{A}\|}{\|A\|}.$$

Hence $\kappa(A)$ controls the change in $\mathbf{x}$ due to changes in A and $\mathbf{b}$.

Example 6. The exact solutions of the following problem are $x_1 = x_2 = 1$. Since $1/\varepsilon$ is large, by taking the effect of roundoff error into account, we have

$$\left(\begin{array}{cc:c} \varepsilon & 1 & 1+\varepsilon \\ 1 & 1 & 2 \end{array}\right) \to \left(\begin{array}{cc:c} \varepsilon & 1 & 1 \\ 0 & -\frac{1}{\varepsilon} & -\frac{1}{\varepsilon} \end{array}\right) \qquad \therefore \quad \begin{cases} x_1 = 0, \\ x_2 = 1. \end{cases}$$

We obtain

$$A = \begin{pmatrix} \varepsilon & 1 \\ 1 & 1 \end{pmatrix}, \quad A^{-1} = \frac{1}{\varepsilon - 1}\begin{pmatrix} 1 & -1 \\ -1 & \varepsilon \end{pmatrix} \quad \Rightarrow \quad \kappa_\infty(A) = 2 \cdot \frac{1}{|\varepsilon - 1|} \cdot 2 \approx 4.$$

However, Gaussian elimination reduces the system to upper triangular form.

$$U = \begin{pmatrix} \varepsilon & 1 \\ 0 & -\frac{1}{\varepsilon} \end{pmatrix}, \quad U^{-1} = \frac{1}{-1}\begin{pmatrix} -\frac{1}{\varepsilon} & -1 \\ 0 & \varepsilon \end{pmatrix}$$
$$\Rightarrow \quad \kappa_\infty(U) = \left|-\frac{1}{\varepsilon}\right| \cdot \frac{1}{|-1|} \cdot \left(\left|-\frac{1}{\varepsilon}\right| + 1\right) \approx \frac{1}{\varepsilon^2}.$$

Thus $\kappa(U) \gg \kappa(A)$.

Now by pivoting we obtain

$$\left(\begin{array}{cc:c} 1 & 1 & 2 \\ \varepsilon & 1 & 1+\varepsilon \end{array}\right) \to \left(\begin{array}{cc:c} 1 & 1 & 2 \\ 0 & 1 & 1 \end{array}\right) \qquad \therefore \quad \begin{cases} x_1 = 1, \\ x_2 = 1. \end{cases}$$

In this case, we have

$$U = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad U^{-1} = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix} \quad \Rightarrow \quad \kappa_\infty(U) \approx 4 \approx \kappa_\infty(A).$$

A small change in A or $\mathbf{b}$ (for example, due to roundoff error) can produce a large change in the computed solution. That is, Gaussian elimination is *unstable* for solving $A\mathbf{x} = \mathbf{b}$. However, since pivoting preserves the condition number of the original matrix A, Gaussian elimination with pivoting is *stable* (in most cases).

LU factorization

LU factorization or *LU* decomposition is a matrix form of Gaussian elimination. L is a **l**ower triangular matrix and U is an **u**pper triangular matrix. Let us consider the *LU* factorization of an $n \times n$ matrix A. For simplicity we assume $n = 3$:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}.$$

Step 1: Operate a lower triangular matrix.

$$\overbrace{\begin{pmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & 0 & 1 \end{pmatrix}}^{E_1} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & a'_{32} & a'_{33} \end{pmatrix},$$

where $m_{21} = a_{21}/a_{11}$ and $m_{31} = a_{31}/a_{11}$.

Step 2: Operate another lower triangular matrix.

$$\overbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -m_{32} & 1 \end{pmatrix}}^{E_2} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & a'_{32} & a'_{33} \end{pmatrix} = \overbrace{\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a''_{33} \end{pmatrix}}^{U},$$

where $m_{32} = a'_{32}/a'_{22}$.

Step 3: Rearrange matrices.

$$E_2E_1A = U \quad \Rightarrow \quad A = E_1^{-1}E_2^{-1}U.$$

We have

$$E_1^{-1}E_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & m_{32} & 1 \end{pmatrix} = \overbrace{\begin{pmatrix} 1 & 0 & 0 \\ m_{21} & 1 & 0 \\ m_{31} & m_{32} & 1 \end{pmatrix}}^{L}.$$

Therefore,

$$A = LU.$$

Example 7.

$$\overbrace{\begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}}^{A} = \overbrace{\begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & -\frac{2}{3} & 1 \end{pmatrix}}^{L} \overbrace{\begin{pmatrix} 2 & -1 & 0 \\ 0 & \frac{3}{2} & -1 \\ 0 & 0 & \frac{4}{3} \end{pmatrix}}^{U}.$$

We can solve $A\mathbf{x} = \mathbf{b}$ as follows.

Step 1 Factor $A = LU$ (the operation count is $\frac{2}{3}n^3$).
Step 2 Solve $L\mathbf{y} = \mathbf{b}$ by forward substitution (the operation count is n^2).
Step 3 Solve $U\mathbf{x} = \mathbf{y}$ by back substitution (the operation count is n^2).

(The operation count of finding A^{-1} by Gauss-Jordan elimination $\approx 8n^3/3$.)

Example 8. Consider $A\mathbf{x} = \mathbf{b}$, where

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}.$$

We have

$$L\mathbf{y} = \mathbf{b} \quad \Rightarrow \quad \begin{pmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & -\frac{2}{3} & 1 \end{pmatrix} \mathbf{y} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \quad \Rightarrow \quad \mathbf{y} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{4}{3} \end{pmatrix},$$

and

$$U\mathbf{x} = \mathbf{y} \quad \Rightarrow \quad \begin{pmatrix} 2 & -1 & 0 \\ 0 & \frac{3}{2} & -1 \\ 0 & 0 & \frac{4}{3} \end{pmatrix} \mathbf{x} = \begin{pmatrix} 1 \\ \frac{1}{2} \\ \frac{4}{3} \end{pmatrix} \quad \Rightarrow \quad \mathbf{x} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

Indeed, the exact solution is $\mathbf{x} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$.

Sometimes we want to solve $A\mathbf{x} = \mathbf{b}_i$ ($i = 1, 2, \dots$), i.e., for a given matrix A and a sequence of vectors $\mathbf{b}$. Once the LU factorization of A is known, we can apply forward and back substitution to the sequence of $\mathbf{b}_i$. We only need to do the LU factorization once in the beginning.

LU factorization and partial pivoting

Sometimes we need to interchange rows. In such a case, we construct $PA = LU$, where P is a permutation matrix. Then,

$$A\mathbf{x} = \mathbf{b} \quad \Rightarrow \quad PA\mathbf{x} = P\mathbf{b} \quad \Rightarrow \quad LU\mathbf{x} = P\mathbf{b}.$$

If pivoting is required in more than one step, we proceed as

$$E_2P_2E_1P_1A = U \quad \Rightarrow \quad E_2\tilde{E}_1P_2P_1A = U \quad \Rightarrow \quad PA = LU,$$

where $P = P_2P_1$, $L = \tilde{E}_1^{-1}E_2^{-1}$, and $\tilde{E}_1$ is the matrix such that $P_2E_1 = \tilde{E}_1P_2$.

Example 9. Consider

$$\begin{pmatrix} 0 & 4 & -15 \\ 10 & 0 & 15 \\ 1 & -1 & -1 \end{pmatrix} \mathbf{x} = \begin{pmatrix} -12 \\ 100 \\ 0 \end{pmatrix}.$$

We want to interchange rows 1 and 2. We define

$$P = \begin{pmatrix} 0\,1\,0 \\ 1\,0\,0 \\ 0\,0\,1 \end{pmatrix}.$$

We have

$$PA = \begin{pmatrix} 10 & 0 & 15 \\ 0 & 4 & -15 \\ 1 & -1 & -1 \end{pmatrix} = \overbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.1 & -0.25 & 1 \end{pmatrix}}^{L} \overbrace{\begin{pmatrix} 10 & 0 & 15 \\ 0 & 4 & -15 \\ 0 & 0 & -6.25 \end{pmatrix}}^{U}.$$

We obtain

$$L\mathbf{y} = P\mathbf{b} \quad \Rightarrow \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0.1 & -0.25 & 1 \end{pmatrix} \mathbf{y} = \begin{pmatrix} 100 \\ -12 \\ 0 \end{pmatrix} \quad \Rightarrow \quad \mathbf{y} = \begin{pmatrix} 100 \\ -12 \\ -13 \end{pmatrix},$$

$$U\mathbf{x} = \mathbf{y} \quad \Rightarrow \quad \begin{pmatrix} 10 & 0 & 15 \\ 0 & 4 & -15 \\ 0 & 0 & -6.25 \end{pmatrix} \mathbf{x} = \begin{pmatrix} 100 \\ -12 \\ -13 \end{pmatrix} \quad \Rightarrow \quad \mathbf{x} = \begin{pmatrix} 6.88 \\ 4.80 \\ 2.08 \end{pmatrix}.$$

Two-point boundary value problem

This section corresponds to §8.1 of the textbook.

Consider the temperature $y(x,t)$ in a material. It obeys the heat equation

$$\frac{\partial y}{\partial t} - k\frac{\partial^2 y}{\partial x^2} = r(x),$$

where k is the thermal diffusivity and $r(x)$ is the internal source. Let us suppose the temperature is in steady state and set $k = 1$ for simplicity. Let us find $y(x)$ on $0 \le x \le 1$. We assume the boundary values are known. We have

$$\begin{cases} -y'' = r(x), & x \in (0,1), \\ y = \alpha, & x = 0, \\ y = \beta, & x = 1. \end{cases}$$

We use *finite-difference scheme*. Choose $n \ge 1$ and set the mesh size $h = \dfrac{1}{n+1}$. Set mesh points

$$x_i = ih \quad \text{for } i = 0, 1, \ldots, n+1 \qquad (x_0 = 0,\ x_{n+1} = 1).$$

We write $y_i = y(x_i)$ and $r_i = r(x_i)$. For each x_i, the exact solution is $y(x_i)$. Recall

$$D_+ y_i = \frac{y_{i+1} - y_i}{h}, \quad D_- y_i = \frac{y_i - y_{i-1}}{h}.$$

We have (HW1 Prob. 7(a))

$$\begin{aligned} D_+ D_- y_i = D_+ \left(\frac{y_i - y_{i-1}}{h} \right) &= \frac{1}{h} \left[\left(\frac{y_{i+1} - y_i}{h} \right) - \left(\frac{y_i - y_{i-1}}{y} \right) \right] \\ &= \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} \approx y''(x_i). \end{aligned}$$

Using Taylor series about x_i, we obtain

$$\begin{aligned} y_{i+1} = y(x_i + h) &= y_i + hy_i' + \frac{h^2}{2} y_i'' + \frac{h^3}{3!} y_i''' + \frac{h^4}{4!} y_i^{(4)} + \frac{h^5}{5!} y_i^{(5)} + O(h^6), \\ y_{i-1} = y(x_i - h) &= y_i - hy_i' + \frac{h^2}{2} y_i'' - \frac{h^3}{3!} y_i''' + \frac{h^4}{4!} y_i^{(4)} - \frac{h^5}{5!} y_i^{(5)} + O(h^6). \end{aligned}$$

Therefore,

$$D_+ D_- y_i = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} = y''(x_i) + \frac{h^2}{12} y_i^{(4)} + O(h^4).$$

That is, the approximation is second-order accurate (HW1 Prob. 7(b)).

Let w_i denote a numerical solution ($w_i \approx y_i$). We set $w_0 = \alpha$ and $w_{n+1} = \beta$. We can implement $-y'' = r(x)$ as (finite-difference equations)

$$-\left(\frac{w_{i+1} - 2w_i + w_{i-1}}{h^2}\right) = r_i, \qquad i = 1, \ldots, n.$$

Thus we have the following matrix-vector equation.

$$\overbrace{\frac{1}{h^2}\begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}}^{A} \overbrace{\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_{n-1} \\ w_n \end{pmatrix}}^{\mathbf{w}} = \overbrace{\begin{pmatrix} r_1 + \frac{\alpha}{h^2} \\ r_2 \\ \vdots \\ r_{n-1} \\ r_n + \frac{\beta}{h^2} \end{pmatrix}}^{\mathbf{r}}.$$

LU factorization for a tridiagonal system (Thomas algorithm)

Let us consider in general how we can implement *LU* factorization for tridiagonal matrices:

$$\begin{pmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ & & & a_n & b_n \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & l_n & 1 \end{pmatrix} \begin{pmatrix} u_1 & c_1 & & & \\ & u_2 & c_2 & & \\ & & \ddots & \ddots & \\ & & & \ddots & c_{n-1} \\ & & & & u_n \end{pmatrix}$$

For $n = 3$, we can write

$$\begin{pmatrix} b_1 & c_1 & 0 \\ a_2 & b_2 & c_2 \\ 0 & a_3 & b_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ l_2 & 1 & 0 \\ 0 & l_3 & 1 \end{pmatrix} \begin{pmatrix} u_1 & c_1 & 0 \\ 0 & u_2 & c_2 \\ 0 & 0 & u_3 \end{pmatrix} = \begin{pmatrix} u_1 & c_1 & 0 \\ l_2 u_1 & l_2 c_1 + u_2 & c_2 \\ 0 & l_3 u_2 & l_3 c_2 + u_3 \end{pmatrix}.$$

In this case we can determine l_i, u_i, c_i as follows.

$$\begin{aligned} b_1 = u_1 \qquad & \Rightarrow u_1 = b_1, \\ a_2 = l_2 u_1 \qquad & \Rightarrow l_2 = a_2/u_1, \\ b_2 = l_2 c_1 + u_2 \; & \Rightarrow u_2 = b_2 - l_2 c_1, \\ & \vdots \end{aligned}$$

In general we can proceed as follows.

$$\begin{aligned} b_1 &= u_1 && \Rightarrow u_1 = b_1, \\ a_k &= l_k u_{k-1} && \Rightarrow l_k = a_k/u_{k-1}, \qquad k = 2,\dots,n, \\ b_k &= l_k c_{k-1} + u_k && \Rightarrow u_k = b_k - l_k c_{k-1}, \qquad k = 2,\dots,n. \end{aligned} \tag{3.5}$$

Numerical solutions

We can compute $\mathbf{w}$ as follows.

Step 1: Find L, U by (3.5).

Step 2: Solve $L\mathbf{z} = \mathbf{r}$.

$$\begin{aligned} & z_1 = r_1, \\ & l_k z_{k-1} + z_k = r_k \quad \Rightarrow \quad z_k = r_k - l_k z_{k-1}, \qquad k = 2,\dots,n. \end{aligned}$$

Step 3: Solve $U\mathbf{w} = \mathbf{z}$.

$$\begin{aligned} & u_n w_n = z_n && \Rightarrow w_n = z_n/u_n, \\ & u_k w_k + c_k w_{k+1} = z_k && \Rightarrow w_k = (z_k - c_k w_{k+1})/u_k, \qquad k = n-1, n-2, \dots, 2, 1. \end{aligned}$$

Note that the operation count for the above algorithm is $O(n)$ whereas solving $A\mathbf{x} = \mathbf{b}$ in general requires $O(n^3)$. If vectors are used instead of full matrices, the required memory is also $O(n)$.

Example 10. Let us solve the following two-point boundary value problem.

$$-y'' = 25\sin(\pi x), \quad 0 \le x \le 1, \quad y(0) = 0, \quad y(1) = 1.$$

The solution is $y(x) = \frac{25}{\pi^2}\sin(\pi x) + x$ but we seek numerical solutions. We may write the following codes. Numerical results are shown in Fig. 3.1. Error analysis is done in the table below. Note that the error decreases by $\approx \frac{1}{4}$ if h decreases by half. Thus $\|\mathbf{y} - \mathbf{w}\| = O(h^2)$ and the method is second order accurate.

```
% -y''=r,  y(0)=alpha,  y(1)=beta
clear; clf;
alpha=0;
beta=1;
n=3;
h=1/(n+1);
x_exact=0:0.0025:1;
```

```
y_exact=25/pi^2*sin(pi*x_exact)+x_exact;
for i=1:n
   x(i)=i*h;
   y(i)=25/pi^2*sin(pi*x(i))+x(i);
   a(i)=-1/h^2;
   b(i)=2/h^2;
   c(i)=-1/h^2;
   r(i)=25*sin(pi*x(i));
end
r(1)=r(1)+alpha/h^2;
r(n)=r(n)+beta/h^2;
w=LU_factorization(a,b,c,r);
% output
table(1)=h;
table(2)=norm(y-w,inf);
table(3)=norm(y-w,inf)/h;
table(4)=norm(y-w,inf)/h^2;
table(5)=norm(y-w,inf)/h^3;
table
xplot=[0 x 1];
wplot=[alpha w beta];
plot(x_exact,y_exact,xplot,wplot,'g-',xplot,wplot,'ro')
axis([0 1 0 4])
title(sprintf('n=%d, h=1/%d',n,n+1),'FontSize',24)
set(gca,'FontSize',24)
```

```
function w = LU_factorization(a,b,c,r)
n=length(r);
u(1)=b(1);
for k=2:n
   l(k)=a(k)/u(k-1);
   u(k)=b(k)-l(k)*c(k-1);
end
z(1)=r(1);
for k=2:n
   z(k)=r(k)-l(k)*z(k-1);
end
w(n)=z(n)/u(n);
for k=n-1:-1:1
   w(k)=(z(k)-c(k)*w(k+1))/u(k);
end
```

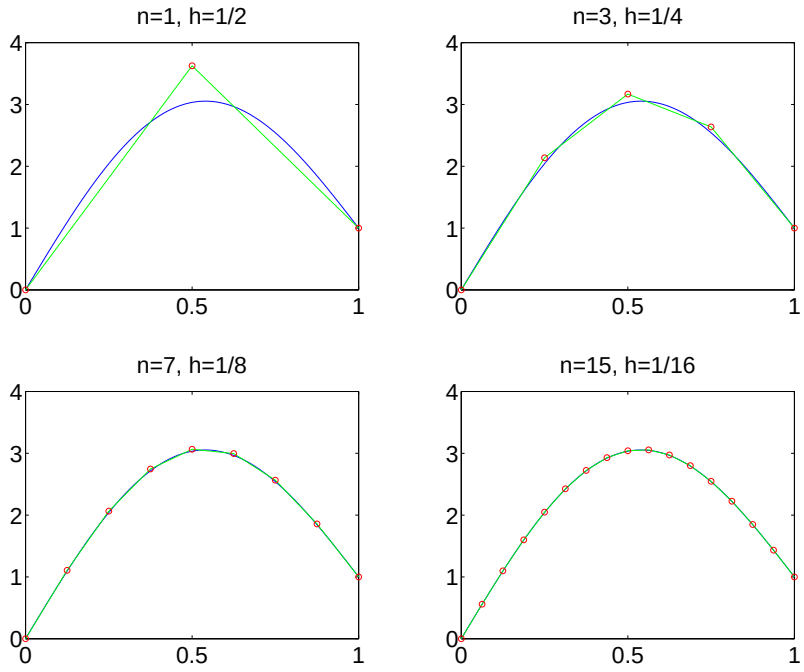

Fig. 3.1 Numerical solutions to $-y'' = 25\sin(\pi x)$, $0 \le x \le 1$, $y(0) = 0$, $y(1) = 1$. The exact solution is plotted as a solid curve.

h	$\|\mathbf{y}-\mathbf{w}\|_\infty$	$\|\mathbf{y}-\mathbf{w}\|_\infty/h$	$\|\mathbf{y}-\mathbf{w}\|_\infty/h^2$	$\|\mathbf{y}-\mathbf{w}\|_\infty/h^3$
0.5000	0.5920	1.1839	2.3679	4.7358
0.2500	0.1343	0.5373	2.1492	8.5968
0.1250	0.0328	0.2624	2.0995	16.7960
0.0625	0.0082	0.1305	2.0874	33.3977

Iterative methods

We will solve $A\mathbf{x} = \mathbf{b}$ by iterative methods. We rewrite the equation to an equivalent linear system.

$$A\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad \mathbf{x} = B\mathbf{x} + \mathbf{c}.$$

Then for given $\mathbf{x}_0$, we compute $\mathbf{x}_1, \mathbf{x}_2, \ldots$ by fixed-point iteration:

$$\mathbf{x}_{k+1} = B\mathbf{x}_k + \mathbf{c}.$$

We choose the iteration matrix B so that the sequence converges ($\lim_{k\to\infty} \mathbf{x}_k = \mathbf{x}$).

Jacobi method

Consider A with nonzero diagonal elements ($a_{ii} \neq 0$, $i = 1,\dots,n$). We write A as $A = L + D + U$, where

$$L = \begin{pmatrix} 0 & & & \\ a_{21} & 0 & & \\ \vdots & \ddots & \ddots & \\ \vdots & & \ddots & \ddots \\ a_{n1} & \cdots & \cdots & a_{n,n-1} \; 0 \end{pmatrix}, \quad D = \begin{pmatrix} a_{11} & & & \\ & a_{22} & & \\ & & \ddots & \\ & & & a_{nn} \end{pmatrix}, \quad U = \begin{pmatrix} 0 & a_{12} & \cdots & \cdots & a_{1n} \\ & 0 & \ddots & & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & a_{n-1,n} \\ & & & & 0 \end{pmatrix}.$$

We write the system as

$$\begin{aligned} A\mathbf{x} = \mathbf{b} \quad &\Leftrightarrow \quad (L+D+U)\mathbf{x} = \mathbf{b} \\ &\Leftrightarrow \quad D\mathbf{x} = -(L+U)\mathbf{x} + \mathbf{b} \\ &\Leftrightarrow \quad \mathbf{x} = -D^{-1}(L+U)\mathbf{x} + D^{-1}\mathbf{b}. \end{aligned}$$

Thus the iteration matrix is $B_{\mathrm{J}} = -D^{-1}(L+U)$. In this case, the following iteration is more convenient.

$$D\mathbf{x}_{k+1} = -(L+U)\mathbf{x}_k + \mathbf{b}.$$

In component form we have

$$a_{ii}x_i^{(k+1)} = -\sum_{j=i+1}^{n} a_{ij}x_j^{(k)} - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} + b_i \qquad (i = 1,2,\dots,n).$$

Example 11. Let us consider the following system of equations.

$$\begin{cases} 2x_1 - x_2 = 1, \\ -x_1 + 2x_2 = 1. \end{cases}$$

The exact solution is $x_1 = x_2 = 1$. We have

$$L = \begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix}, \quad D = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, \quad U = \begin{pmatrix} 0 & -1 \\ 0 & 0 \end{pmatrix}.$$

Therefore,

$$\begin{aligned} 2x_1^{(k+1)} &= x_2^{(k)} + 1, \\ 2x_2^{(k+1)} &= x_1^{(k)} + 1, \end{aligned}$$

where $k = 0, 1, 2, \ldots$. Let the initial guess be $x_1^{(0)} = x_2^{(0)} = 0$.

k	$x_1^{(k)}$	$x_2^{(k)}$
0	0	0
1	1/2	1/2
2	3/4	3/4
3	7/8	7/8

Numerical solutions converge to the exact solution as $k \to \infty$.

Let us consider

$$\mathbf{e}_k = \mathbf{x} - \mathbf{x}_k,$$

which is the error at step k. In the above example, we have

$$\|\mathbf{e}_0\|_\infty = 1,\ \|\mathbf{e}_1\|_\infty = \frac{1}{2},\ \|\mathbf{e}_2\|_\infty = \frac{1}{4}, \ldots,\ \|\mathbf{e}_{k+1}\|_\infty = \frac{1}{2}\|\mathbf{e}_k\|_\infty.$$

Theorem 4. *Consider a fixed-point iteration $\mathbf{x}_{k+1} = B\mathbf{x}_k + \mathbf{c}$ to solve a linear system $A\mathbf{x} = \mathbf{b}$. Then*

$$\mathbf{e}_{k+1} = B\mathbf{e}_k,$$

for all $k \geq 0$. Furthermore, if $\|B\| < 1$, then $\mathbf{x}_k \to \mathbf{x}$ as $k \to \infty$ for any $\mathbf{x}_0$.

Proof. The first half is proved as

$$\mathbf{e}_{k+1} = \mathbf{x} - \mathbf{x}_{k+1} = (B\mathbf{x} + \mathbf{c}) - (B\mathbf{x}_k + \mathbf{c}) = B(\mathbf{x} - \mathbf{x}_k) = B\mathbf{e}_k.$$

To prove the second half, let us consider

$$\|\mathbf{e}_{k+1}\| = \|B\mathbf{e}_k\| \leq \|B\|\,\|\mathbf{e}_k\| = \|B\|\,\|B\mathbf{e}_{k-1}\| \leq \|B\|\,\|B\|\,\|\mathbf{e}_{k-1}\|$$

Thus we obtain

$$\|\mathbf{e}_{k+1}\| \leq \|B\|^{k+1}\,\|\mathbf{e}_0\|,$$

which goes to zero as $k \to \infty$. □

Example 12. In the previous example, we had the matrix

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

Hence,

$$B_{\mathrm{J}} = -D^{-1}(L+U) = -\begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}\begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix} \quad \Rightarrow \quad \|B_{\mathrm{J}}\|_\infty = \frac{1}{2} < 1.$$

The theorem implies that the Jacobi method converges and the proof shows that $\|\mathbf{e}_k\|$ decreases by a factor of at least $1/2$ in each step.

Gauss-Seidel method

We write the system as

$$\begin{aligned} A\mathbf{x}=\mathbf{b} \quad &\Leftrightarrow \quad (L+D+U)\mathbf{x}=\mathbf{b} \\ &\Leftrightarrow \quad (L+D)\mathbf{x}=-U\mathbf{x}+\mathbf{b} \\ &\Leftrightarrow \quad \mathbf{x}=-(L+D)^{-1}U\mathbf{x}+(L+D)^{-1}\mathbf{b}. \end{aligned}$$

Thus the iteration matrix is $B_{\mathrm{GS}}=-(L+D)^{-1}U$. In this case, the following iteration is more convenient.

$$(L+D)\mathbf{x}_{k+1}=-U\mathbf{x}_k+\mathbf{b}.$$

Note that we can rewrite the above relation as

$$D\mathbf{x}_{k+1}=-U\mathbf{x}_k-L\mathbf{x}_{k+1}+\mathbf{b}.$$

In component form we have

$$a_{ii}x_i^{(k+1)}=-\sum_{j=i+1}^{n}a_{ij}x_j^{(k)}-\sum_{j=1}^{i-1}a_{ij}x_j^{(k+1)}+b_i \qquad (i=1,2,\ldots,n).$$

Example 13. Let us consider the following system of equations again.

$$\begin{cases} 2x_1-x_2=1, \\ -x_1+2x_2=1. \end{cases}$$

We have

$$\begin{aligned} 2x_1^{(k+1)} &= x_2^{(k)}+1, \\ 2x_2^{(k+1)} &= x_1^{(k+1)}+1, \end{aligned}$$

where $k=0,1,2,\ldots$. Let the initial guess be $x_1^{(0)}=x_2^{(0)}=0$.

The Gauss-Seidel converges faster than the Jacobi. In the above example, we have

$$\|\mathbf{e}_0\|_\infty=1,\ \|\mathbf{e}_1\|_\infty=\frac{1}{2},\ \|\mathbf{e}_2\|_\infty=\frac{1}{8},\ \|\mathbf{e}_3\|_\infty=\frac{1}{32},\ldots,\ \|\mathbf{e}_{k+1}\|_\infty=\frac{1}{4}\|\mathbf{e}_k\|_\infty.$$

k	$x_1^{(k)}$	$x_2^{(k)}$
0	0	0
1	1/2	3/4
2	7/8	15/16
3	31/32	63/64

Example 14. In the previous example, we obtain

$$B_{\mathrm{GS}} = -(L+D)^{-1}U = -\frac{1}{4}\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}\begin{pmatrix} 0 & -1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{4} \end{pmatrix} \quad \Rightarrow \quad \|B_{\mathrm{GS}}\|_\infty = \frac{1}{2} < 1.$$

The theorem implies that the Gauss-Seidel converges, but we see that $\|\mathbf{e}_k\|$ decreases by a factor of $1/4 < \|B_{\mathrm{GS}}\|$ in each step.

Spectral radius

Definition 3. If $A\mathbf{v} = \lambda\mathbf{v}$ with a vector $\mathbf{v} \neq \mathbf{0}$ and a scalar λ, then λ is an eigenvalue of A and $\mathbf{v}$ is a corresponding eigenvector.

Definition 4. We call $f_A(\lambda) = \det(A - \lambda I)$ the characteristic polynomial of A.

Theorem 5. *A scalar λ is an eigenvalue of A if and only if λ is a solution to the characteristic equation:*

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

Proof.

$$\begin{aligned} \lambda \text{ is an eigenvalue} \quad &\Leftrightarrow \quad \text{There exists } \mathbf{v} \neq \mathbf{0} \text{ such that } A\mathbf{v} = \lambda\mathbf{v} \\ &\Leftrightarrow \quad \mathbf{v} = \mathbf{0} \text{ is not the unique solution to } (A - \lambda I)\mathbf{v} = \mathbf{0} \\ &\Leftrightarrow \quad (A - \lambda I) \text{ is not invertible} \\ &\Leftrightarrow \quad \det(A - \lambda I) = 0. \end{aligned}$$

□

Theorem 6. *If A is upper triangular, then the eigenvalues are the diagonal elements.*

Proof.

$$
\begin{aligned}
f_A(\lambda) &= \det(A-\lambda I) \\
&= \begin{vmatrix} a_{11}-\lambda & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}-\lambda & \cdots & a_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & & & a_{nn}-\lambda \end{vmatrix} \\
&= (a_{11}-\lambda) \begin{vmatrix} a_{22}-\lambda & \cdots & a_{2n} \\ & \ddots & \vdots \\ 0 & & a_{nn}-\lambda \end{vmatrix} = \cdots \\
&= (a_{11}-\lambda)(a_{22}-\lambda)\cdots(a_{nn}-\lambda) = 0.
\end{aligned}
$$

Therefore, $\lambda = a_{ii}$ $(i = 1,2,\ldots,n)$. □

Example 15. In the previous example, we considered

$$B_{\mathrm{GS}} = \begin{pmatrix} 0 & \frac{1}{2} \\ 0 & \frac{1}{4} \end{pmatrix}.$$

By solving,

$$\det(B_{\mathrm{GS}} - \lambda I) = \begin{vmatrix} -\lambda & \frac{1}{2} \\ 0 & \frac{1}{4}-\lambda \end{vmatrix} = \lambda\left(\lambda - \frac{1}{4}\right) = 0,$$

we obtain $\lambda = 0 = \lambda_1$ and $\lambda = 1/4 = \lambda_2$. The corresponding eigenvectors are obtained as

$$\mathbf{v}_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \mathbf{v}_2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}.$$

Let us analyze the error.

$$
\begin{aligned}
\mathbf{e}_0 &= \mathbf{x} - \mathbf{x}_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \mathbf{v}_2 - \mathbf{v}_1, \\
\mathbf{e}_1 &= B_{\mathrm{GS}}\mathbf{e}_0 = B_{\mathrm{GS}}(\mathbf{v}_2 - \mathbf{v}_1) = \lambda_2\mathbf{v}_2 - \lambda_1\mathbf{v}_1 \\
\mathbf{e}_2 &= B_{\mathrm{GS}}\mathbf{e}_1 = \lambda_2^2\mathbf{v}_2 - \lambda_1^2\mathbf{v}_1 \\
&\vdots \\
\mathbf{e}_k &= \lambda_2^k\mathbf{v}_2 - \lambda_1^k\mathbf{v}_1 = \lambda_2^k\mathbf{v}_2.
\end{aligned}
$$

Therefore,

$$\|\mathbf{e}_k\| = \left(\frac{1}{4}\right)^k \|\mathbf{v}_2\|.$$

This is why we had $\|\mathbf{e}_{k+1}\|_\infty = \frac{1}{4}\|\mathbf{e}_k\|_\infty$ ($\|\mathbf{e}_{k+1}\|_\infty \le \|B_{\mathrm{GS}}\|_\infty \|\mathbf{e}_k\|_\infty = \frac{1}{2}\|\mathbf{e}_k\|_\infty$ is correct but not a very sharp estimate).

Definition 5. The spectral radius $\rho(A)$ is defined by

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

Theorem 7. *We have*

$$\|\mathbf{e}_{k+1}\| \sim \rho(B)\|\mathbf{e}_k\|,$$

as $k \to \infty$.

Proof. Suppose $\lambda_n \geq \lambda_{n-1} \geq \cdots \geq \lambda_1$. The initial vector $\mathbf{e}_0$ is given as $\mathbf{e}_0 = c_n\mathbf{v}_n + c_{n-1}\mathbf{v}_{n-1} + \cdots + c_1\mathbf{v}_1$ with constants $c_n, c_{n-1}, \ldots, c_1$. Hence,

$$\begin{aligned}\mathbf{e}_k &= c_n\lambda_n^k\mathbf{v}_n + c_{n-1}\lambda_{n-1}^k\mathbf{v}_{n-1} + \cdots + c_1\lambda_1^k\mathbf{v}_1 \sim c_n\lambda_n^k\mathbf{v}_n,\\ \mathbf{e}_{k+1} &= c_n\lambda_n^{k+1}\mathbf{v}_n + c_{n-1}\lambda_{n-1}^{k+1}\mathbf{v}_{n-1} + \cdots + c_1\lambda_1^{k+1}\mathbf{v}_1 \sim c_n\lambda_n^{k+1}\mathbf{v}_n,\end{aligned}$$

as $k \to \infty$. Therefore we have $\mathbf{e}_{k+1} \sim \lambda_n\mathbf{e}_k$. This proves the theorem. □

We note that the theorem means

$$\lim_{k\to\infty} \frac{\|\mathbf{e}_{k+1}\|}{\|\mathbf{e}_k\|} = \rho(B).$$

Thus the spectral radius $\rho(B)$ of the iteration matrix determines the convergence rate of an iterative method.

Example 16. Recall we had $B_{\mathrm{J}} = \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix}$ for the matrix $A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}$. In this case, $\rho(B_{\mathrm{J}}) = \dfrac{1}{2} = \|B_{\mathrm{J}}\|_\infty$.

SOR

We can accelerate the convergence of the Gauss-Seidel method by writing

$$A = L + D + U = L + \frac{1}{\omega}D + \left(1 - \frac{1}{\omega}\right)D + U,$$

where $\omega > 1$ is a relaxation parameter. We have

$$\begin{aligned} A\mathbf{x} = \mathbf{b} \quad &\Leftrightarrow \quad \left(L + \frac{1}{\omega}D\right)\mathbf{x} = \left[\left(\frac{1}{\omega} - 1\right)D - U\right]\mathbf{x} + \mathbf{b} \\ &\Leftrightarrow \quad (\omega L + D)\,\mathbf{x} = [(1-\omega)D - \omega U]\,\mathbf{x} + \omega\mathbf{b} \\ &\Leftrightarrow \quad \mathbf{x} = (\omega L + D)^{-1}\,[(1-\omega)D - \omega U]\,\mathbf{x} + \omega\,(\omega L + D)^{-1}\,\mathbf{b}. \end{aligned}$$

Thus,

$$\begin{aligned}(\omega L+D)\,\mathbf{x}_{k+1} &= [(1-\omega)D-\omega U]\,\mathbf{x}_k+\omega\mathbf{b}\\ \Leftrightarrow\quad (D+\omega L)\mathbf{x}_{k+1} &= D\mathbf{x}_k-\omega\,[(D+U)\mathbf{x}_k-\mathbf{b}]\,.\end{aligned}$$

We can rewrite the above relation as follows.

$$D\mathbf{x}_{k+1}=D\mathbf{x}_k+\omega\,[-L\mathbf{x}_{k+1}-(D+U)\mathbf{x}_k+\mathbf{b}]\,.$$

In component form we have

$$a_{ii}x_i^{(k+1)}=a_{ii}x_i^{(k)}+\omega\left[-a_{ii}x_i^{(k)}-\sum_{j=i+1}^{n}a_{ij}x_j^{(k)}-\sum_{j=1}^{i-1}a_{ij}x_j^{(k+1)}+b_i\right]\qquad(i=1,2,\ldots,n).$$

This method is called the SOR (successive over-relaxation) method. When $\omega=1$, the SOR method reduces to the Gauss-Seidel method. The iteration matrix is given by

$$B_{\mathrm{SOR}}=B_\omega=(\omega L+D)^{-1}\,[(1-\omega)D-\omega U]$$

The selection of the parameter ω is crucial.

Theorem 8 (Young (1950)). *If $\rho(B_\omega)<1$, then $0<\omega<2$. Assume A is symmetric, block tridiagonal, and positive definite. Then*

$$\omega_*=\frac{2}{1+\sqrt{1-\rho(B_{\mathrm{J}})^2}}$$

is the optimal SOR parameter and we have

$$\rho(B_{\omega_*})=\min_{0<\omega<2}\rho(B_\omega)=\omega_*-1<\rho(B_{\mathrm{GS}})<\rho(B_{\mathrm{J}})<1.$$

Note that a symmetric matrix A is said to be positive definite if $\mathbf{x}\cdot A\mathbf{x}$ is positive for all nonzero $\mathbf{x}$. A symmetric matrix A is positive definite if and only if all of its eigenvalues are positive.

Example 17. Consider matrices $A_1=\begin{pmatrix}2&-1\\-1&2\end{pmatrix}$ and $A_2=\begin{pmatrix}1&2\\2&1\end{pmatrix}$. The matrix A_1 is positive definite and the matrix A_2 is indefinite (not positive definite). Let us consider A_1.

$$\begin{aligned}\mathbf{x}\cdot A_1\mathbf{x}&=\begin{pmatrix}x_1\\x_2\end{pmatrix}\cdot\begin{pmatrix}2&-1\\-1&2\end{pmatrix}\begin{pmatrix}x_1\\x_2\end{pmatrix}=\begin{pmatrix}x_1\\x_2\end{pmatrix}\cdot\begin{pmatrix}2x_1-x_2\\-x_1+2x_2\end{pmatrix}\\&=2\left(x_1^2+x_2^2\right)-2x_1x_2=x_1^2+x_2^2+(x_1-x_2)^2>0.\end{aligned}$$

Hence A_1 is positive definite. The eigenvalues of A_1 are 1 and 3. Next consider A_2.

$$\begin{aligned}\mathbf{x}\cdot A_2\mathbf{x} &= \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \cdot \begin{pmatrix} x_1+2x_2 \\ 2x_1+x_2 \end{pmatrix} \\ &= x_1^2+x_2^2+4x_1x_2.\end{aligned}$$

If $\mathbf{x} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, then $x_1^2+x_2^2+4x_1x_2 = 1$. If $\mathbf{x} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$, then $x_1^2+x_2^2+4x_1x_2 = -2$. Thus A_2 is indefinite. The eigenvalues of A_2 are -1 and 3.

Example 18. Let us consider the following system of equations once again.

$$\begin{cases} 2x_1 - x_2 = 1, \\ -x_1 + 2x_2 = 1. \end{cases}$$

The exact solution is $x_1 = x_2 = 1$ and we have

$$L = \begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix}, \quad D = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}, \quad U = \begin{pmatrix} 0 & -1 \\ 0 & 0 \end{pmatrix}, \quad B_\omega = \begin{pmatrix} 1-\omega & \frac{1}{2}\omega \\ \frac{1}{2}\omega(1-\omega) & \frac{1}{4}\omega^2-\omega+1 \end{pmatrix}.$$

We obtain

$$\begin{aligned} 2x_1^{(k+1)} &= 2x_1^{(k)} + \omega\left(-2x_1^{(k)} + x_2^{(k)} + 1\right), \\ 2x_2^{(k+1)} &= 2x_2^{(k)} + \omega\left(x_1^{(k+1)} - 2x_2^{(k)} + 1\right), \end{aligned}$$

where $k = 0, 1, 2, \ldots$. We obtain

$$\omega_* = \frac{2}{1+\sqrt{1-\rho(B_{\mathrm{J}})^2}} = \frac{2}{1+\sqrt{1-\left(\frac{1}{2}\right)^2}} = \frac{4}{2+\sqrt{3}} = 1.0718.$$

Let the initial guess be $x_1^{(0)} = x_2^{(0)} = 0$. Numerical results are obtained as We see that

k	$x_1^{(k)}$	$x_2^{(k)}$	$\|\mathbf{e}_k\|$	$\|\mathbf{e}_k\|/\|\mathbf{e}_{k-1}\|$
0	0	0	1	
1	0.5359	0.8231	0.4641	0.4641
2	0.9385	0.9798	0.0615	0.1325
3	0.9936	0.9980	0.0064	0.1047

$x_1^{(k)} \to 1$, $x_2^{(k)} \to 1$, $\|\mathbf{e}_k\| \to 0$, and $\|\mathbf{e}_k\|/\|\mathbf{e}_{k-1}\| \to \rho(B_{\omega_*}) = \omega_* - 1 = 0.0718$. The optimal SOR converges faster than the Gauss-Seidel method.

Two-dimensional boundary value problems

This section corresponds to §9.1 of the textbook.

Consider a square metal plate. The plate is heated by internal sources and the edges are held at a given temperature. Let us find the temperature $\phi(x,y)$ inside the plate. We denote the plate (the plate domain) by $D = \{(x,y); 0 \le x,y \le 1\}$. Let $f(x,y)$ and $g(x,y)$ be heat sources and the boundary temperature, respectively. Then $\phi(x,y)$ satisfies the following Poisson equation.

$$\begin{aligned} -\Delta\phi = -\nabla^2\phi = -\left(\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2}\right) &= f, \qquad (x,y) \in D, \\ \phi &= g, \qquad (x,y) \in \partial D. \end{aligned}$$

Here Δ is called the Laplace operator or Laplacian. In the boundary condition, boundary values are specified. This type of boundary condition is called the Dirichlet boundary condition (cf, the Neumann boundary condition specifies the derivative of ϕ).

We can use the finite-difference scheme. We set the mesh size and mesh points as

$$h = \frac{1}{n+1}, \qquad (x_i, y_j) = (ih, jh), \quad i,j = 0,1,\ldots,n+1.$$

Let w_{ij} be a numerically obtained $\phi(x_i, y_j)$. The finite-difference equations are written as

$$\begin{aligned} & -\left(D_+^x D_-^x w_{ij} + D_+^y D_-^y w_{ij}\right) = f_{ij} \\ \Leftrightarrow \quad & -\left(\frac{w_{i+1,j} - 2w_{ij} + w_{i-1,j}}{h^2} + \frac{w_{i,j+1} - 2w_{ij} + w_{i,j-1}}{h^2}\right) = f_{ij} \\ \Leftrightarrow \quad & \frac{1}{h^2}\left(4w_{ij} - w_{i+1,j} - w_{i-1,j} - w_{i,j+1} - w_{i,j-1}\right) = f_{ij}. \end{aligned}$$

Attention is needed near the boundary ($i = 1,n$, $j = 1,n$). For example when $(i,j) = (1,1)$, the finite-difference equation is written as

$$\begin{aligned} & \frac{1}{h^2}\left(4w_{11} - w_{21} - w_{01} - w_{12} - w_{10}\right) = f_{11} \\ \Leftrightarrow \quad & \frac{1}{h^2}\left(4w_{11} - w_{21} - w_{12}\right) = f_{11} + \frac{1}{h^2}\left(g_{01} + g_{10}\right). \end{aligned}$$

Thus we have the following matrix-vector equation.

$$A\mathbf{w} = \mathbf{f}.$$

Here

$$A=\frac{1}{h^2}\begin{pmatrix} T & -I & & & \\ -I & T & -I & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & -I \\ & & & -I & T \end{pmatrix},\quad \mathbf{w}=\begin{pmatrix} \vdots \\ w_{i,j-1} \\ w_{ij} \\ w_{i,j+1} \\ \vdots \end{pmatrix},\quad \mathbf{f}=\begin{pmatrix} f_{11}+\frac{1}{h^2}(g_{01}+g_{10}) \\ f_{12}+\frac{1}{h^2}g_{02} \\ \vdots \\ f_{ij} \\ \vdots \end{pmatrix},$$

where T is an $n\times n$ tridiagonal symmetric matrix whose diagonal elements are 4 and off-diagonal elements are -1. The matrix A is an $n^2\times n^2$ block-tridiagonal symmetric matrix. The matrix A is positive definite.

Example 19. For $n=3$, we obtain

$$A\mathbf{w}=\mathbf{f} \quad\Leftrightarrow$$

$$\frac{1}{h^2}\left(\begin{array}{ccc:ccc:ccc} 4 & -1 & & -1 & & & & & \\ -1 & 4 & -1 & & -1 & & & & \\ & -1 & 4 & & & -1 & & & \\ \hdashline -1 & & & 4 & -1 & & -1 & & \\ & -1 & & -1 & 4 & -1 & & -1 & \\ & & -1 & & -1 & 4 & & & -1 \\ \hdashline & & & -1 & & & 4 & -1 & \\ & & & & -1 & & -1 & 4 & -1 \\ & & & & & -1 & & -1 & 4 \end{array}\right)\begin{pmatrix} w_{11} \\ w_{12} \\ w_{13} \\ w_{21} \\ w_{22} \\ w_{23} \\ w_{31} \\ w_{32} \\ w_{33} \end{pmatrix}=\begin{pmatrix} f_{11}+\frac{1}{h^2}(g_{01}+g_{10}) \\ f_{12}+\frac{1}{h^2}g_{02} \\ f_{13}+\frac{1}{h^2}(g_{03}+g_{14}) \\ f_{21}+\frac{1}{h^2}g_{20} \\ f_{22} \\ f_{23}+\frac{1}{h^2}g_{24} \\ f_{31}+\frac{1}{h^2}(g_{30}+g_{41}) \\ f_{32}+\frac{1}{h^2}g_{42} \\ f_{33}+\frac{1}{h^2}(g_{34}+g_{43}) \end{pmatrix}.$$

Example 20. Suppose there is no internal heat source and the metal plate is heated only on one side. Let us calculate the temperature distribution ϕ. The equation for ϕ is written as

$$\begin{aligned} &\phi_{xx}+\phi_{yy}=0, \qquad (x,y)\in(0,1)\times(0,1),\\ &\phi(x,1)=1,\\ &\phi(x,0)=\phi(0,y)=\phi(1,y)=0. \end{aligned}$$

We consider the Jacobi method:

$$a_{ii}x_i^{(k+1)}=-\sum_{j=i+1}^{n}a_{ij}x_j^{(k)}-\sum_{j=1}^{i-1}a_{ij}x_j^{(k)}+b_i.$$

Let us write $\{\mathbf{w}\}_{pq}=w_{pq}$ $(p,q=1,2,\ldots,n)$ and

$$\{A\}_{ij}=a_{ij}=a_{p,q;p',q'}, \qquad i=n(p-1)+q, \quad j=n(p'-1)+q'.$$

Then the iteration for the p,q-th equation is written as

$$
\begin{aligned}
& a_{pq;pq} w_{pq}^{(k+1)} = \\
& \qquad -\left(a_{pq;p,q+1} w_{p,q+1}^{(k)} + a_{pq;p+1,q} w_{p+1,q}^{(k)}\right) - \left(a_{pq;p-1,q} w_{p-1,q}^{(k)} + a_{pq;p,q-1} w_{p,q-1}^{(k)}\right) + f_{pq} \\
\Leftrightarrow \quad & 4 w_{pq}^{(k+1)} = w_{p,q+1}^{(k)} + w_{p+1,q}^{(k)} + w_{p-1,q}^{(k)} + w_{p,q-1}^{(k)} + h^2 f_{pq},
\end{aligned}
$$

where

$$
h^2 f_{pq} = \begin{cases} 1, & q = 1, \\ 0, & \text{otherwise.} \end{cases}
$$

The results are shown in Fig. 3.2. In the calculation the zero vector was chosen for the initial guess. The main part of the code is written as follows. As the stopping criterion, `tol=10e-4` was used[5]

```
while ratio>tol
  k=k+1;
  for i=2:n+1
  for j=2:n+1
    res(i,j)=(4*w(i,j)-w(i+1,j)-w(i-1,j)-w(i,j+1)-w(i,j-1))/(h^2);
  end
  end
  rn(k)=norm(res,'fro');
  ratio=rn(k)/rn(1);
  for i=2:n+1
  for j=2:n+1
    w_old(i,j)=(w(i+1,j)+w(i-1,j)+w(i,j+1)+w(i,j-1))/4;
  end
  end
  w=w_old;
end
```

The number of iterations k required for different methods is summarized as follows.

Jacobi

h	k	$\rho(B_J)$
1/4	26	0.7071
1/8	96	0.9239
1/16	334	0.9808

Gauss-Seidel

h	k	$\rho(B_{GS})$
1/4	15	0.5000
1/8	51	0.8536
1/16	172	0.9619

Let us consider what happens if Gaussian elimination is used instead of iterative methods. In the above example A is a band matrix, i.e., $a_{ij} = 0$ for $|i-j| > m$, where

[5] Note that `10e-4` and `1e-3` are the same.

optimal SOR

h	k	$\rho(B_{\omega_*})$
1/4	9	0.1716
1/8	18	0.4465
1/16	34	0.6735

m is the bandwidth ($m = 3$ in the example). As the elimination proceeds, zeros inside the band can become nonzero, but zeros outside the band are preserved. Hence we can adjust the limits on the loops to reduce the operation count for Gaussian elimination from $O(n^3)$ to $O(nm^2)$. See the matrix A below.

$$\frac{1}{h^2}\begin{pmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & * & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & * & * & -1 & 0 & 0 & 0 \\ -1 & * & * & 4 & -1 & * & -1 & 0 & 0 \\ 0 & -1 & * & -1 & 4 & -1 & * & -1 & 0 \\ 0 & 0 & -1 & * & -1 & 4 & * & * & -1 \\ 0 & 0 & 0 & -1 & * & * & 4 & -1 & * \\ 0 & 0 & 0 & 0 & -1 & * & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & * & -1 & 4 \end{pmatrix}$$

Since zeros inside the band are replaced by nonzero values, more memory needs to be allocated than is required for the original matrix A. This is a disadvantage in comparison with iterative methods such as the Jacobi, Gauss-Seidel, and SOR, which preserve the sparsity of A.

We note that A is an $n^2 \times n^2$ matrix and the size quickly grows as the mesh size $h = 1/(n+1)$ decreases. The operation counts for solving $A\mathbf{w} = \mathbf{f}$ are summarized as follows.

- A^{-1} or general Gaussian elimination: $O\left((n^2)^3\right) = O(n^6)$,
- banded Gaussian elimination: $O(n^2m^2) = O(n^4)$,
- Jacobi and Gauss-Seidel: $O(n^2) \times \left[O(n^2) \text{ iterations}\right] = O(n^4)$,
- SOR: $O(n^2) \times [O(n) \text{ iterations}] = O(n^3)$.

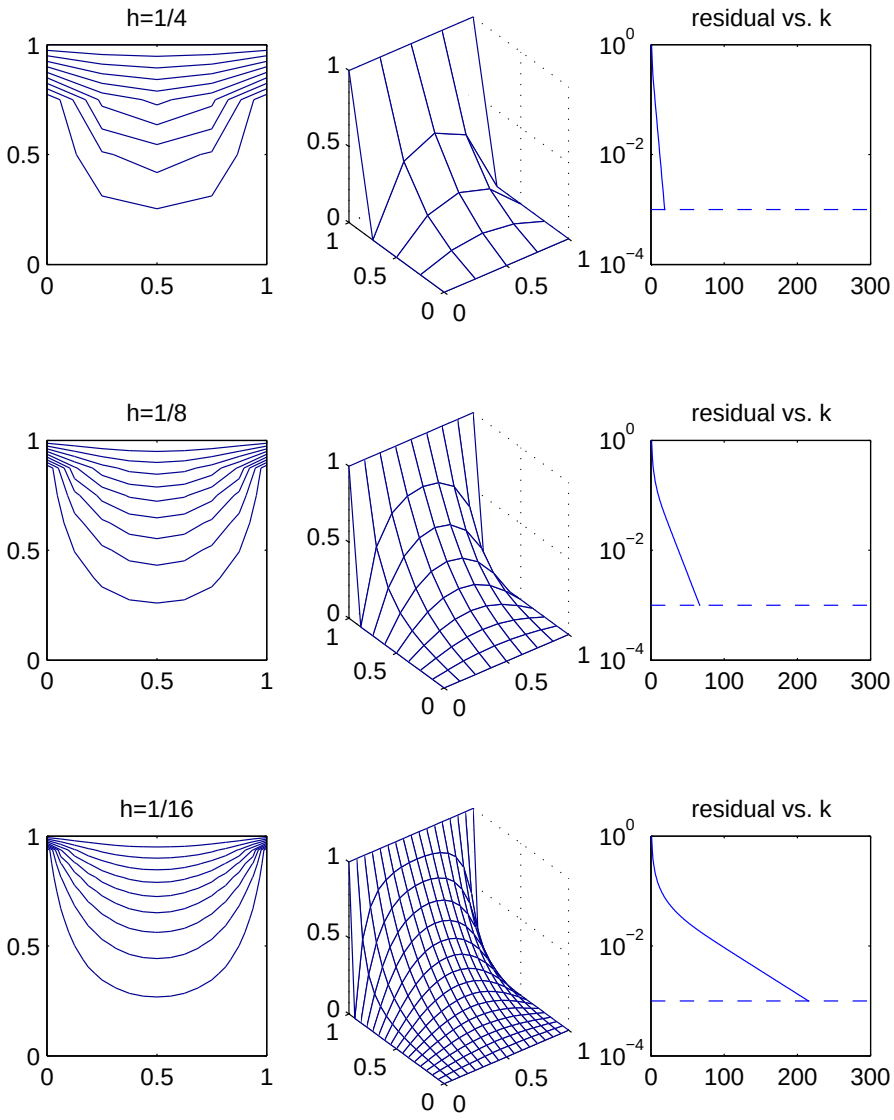

Fig. 3.2 Numerical solutions to $\phi_{xx} + \phi_{yy} = 0$, $\phi(x,1) = 1$, $\phi(x,0) = \phi(0,y) = \phi(1,y) = 0$. The Jacobi method is used.