Iterative Solutions of Linear Systems

GOAL.

- Understand the norm of vectors and matrix
- Understand the conditional number of a matrix
- Understand, implement and analyze iterative methods

KEY WORDS. Condition number, iterative method, Jacobi method, Gauss-Seidel method, successive over-relaxation (SOR) method

In the last Chapter, we have seen that Gaussian elimination is the most general approach to solve a nonsingular linear system

$$A\boldsymbol{x} = \boldsymbol{b}. \tag{1}$$

The complexity of a Gaussian elimination (or with pivoting) procedure is of order n^3, where n is the size of the matrix. In this Chapter, we are exploring a completely different strategy of solving eq. (1). This approach is often used in enormous (of the sizes of hundreds or thousands of) systems arising from science and engineering applications. Before we step into the methodology, we need to introduce some basic concepts of vectors and matrix.

1 Norms of Vectors and Matrix

We first present the norm of vectors and matrix, because they are going to be useful in the discussion of stability of the algorithm and in the stopping criteria, convergence analysis of the iterative methods.

Definition 1.1. (Vector Norm) A vector norm $\|\boldsymbol{x}\|$ is any mapping from $\mathcal{R}^n$ to $\mathcal{R}$ with the following three properties.

1. $\|\boldsymbol{x}\| > 0$, if $\boldsymbol{x} \neq 0$
2. $\|\alpha\boldsymbol{x}\| = |\alpha|\|\boldsymbol{x}\|$, for any $\alpha \in \mathcal{R}$
3. $\|\boldsymbol{x} + \mathbf{y}\| \leq \|\boldsymbol{x}\| + \|\mathbf{y}\|$

for any vector $\boldsymbol{x}$, $\mathbf{y} \in \mathcal{R}^n$.

Example 1.2. One of the most commonly used vector norms is the Euclidean norm (or called l_2 norm)

$$\|\boldsymbol{x}\|_2 = (\sum_{i=1}^{n} x_i^2)^{1/2} = \sqrt{\boldsymbol{x}^T \cdot \boldsymbol{x}}, \quad l_2 \text{ norm},$$

which can be understood intuitively as the length or magnitude of a vector $\boldsymbol{x} \in \mathcal{R}^n$. The properties of the l^2 norm can be interpreted as

1. positivity: the length of a vector is always greater than 0, unless it is a zero vector
2. positive scalability: the length of the scalar product of a vector is the length of the vector multiplied by the absolute value of the scalar.
3. triangular inequality: the length of one side of triangular is always smaller than the sum of the length of the other two sides of a triangle.

Other examples of vector norms are l_1 norm, l_∞ norm,

$$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i|, \quad l_1 \text{ norm},$$

$$\|\boldsymbol{x}\|_\infty = \max_{1\le i\le n} |x_i|, \quad l_\infty \text{ norm},$$

It can be checked by Definition 1.1 that the l_1, l_2, l_∞ norm defined above are vector norms. Below we use the l_1 norm as an example.

Example 1.3. The l_1 norm is a vector norm.

It suffices to check that

1. $\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i| > 0$, if $\boldsymbol{x} \neq 0$
2. $\|\alpha\boldsymbol{x}\| = \sum_{i=1}^{n} |\alpha x_i| = \alpha \sum_{i=1}^{n} |x_i| = |\alpha|\|\boldsymbol{x}\|$, for any $\alpha \in \mathcal{R}$
3. $\|\boldsymbol{x} + \mathbf{y}\| = \sum_{i=1}^{n} |x_i + y_i| \le \sum_{i=1}^{n}(|x_i| + |y_i|) = \|\boldsymbol{x}\| + \|\mathbf{y}\|$.

Example 1.4. Let $\boldsymbol{x} = (1, 1, \cdots, 1)_{1\times n}$, then

$$\|\boldsymbol{x}\|_1 = n,$$

$$\|\boldsymbol{x}\|_2 = \sqrt{n}$$

$$\|\boldsymbol{x}\|_\infty = 1.$$

Definition 1.5. (Matrix Norm) A matrix norm of a matrix $\|A\|$ is any mapping from $\mathcal{R}^{n\times n}$ to $\mathcal{R}$ with the following three properties.

1. $\|A\| > 0$, if $A \neq 0$
2. $\|\alpha A\| = |\alpha|\|A\|$, for any $\alpha \in \mathcal{R}$
3. $\|A + B\| \le \|A\| + \|B\|$ (triangular inequality)

for any matrix A, $B \in \mathcal{R}^{n\times n}$.

We usually prefer matrix norms that are related to a vector norm.

Definition 1.6. (subordinate matrix norm) The subordinate matrix norm based on a vector norm $\|\cdot\|$ is given by

$$\begin{aligned} \|A\| &= \sup\{\|A\boldsymbol{x}\| : \boldsymbol{x} \in \mathcal{R}^n \quad \text{and} \quad \|\boldsymbol{x}\| = 1\} \\ &= \sup_{\|x\|=1} \{\|A\boldsymbol{x}\|\} \end{aligned} \tag{2}$$

It can be checked that the subordinate matrix norm defined by eq.(2) is a norm.

1. $\|A\| > 0$, if $A \neq 0$
2. $$\|\alpha A\| = \sup_{\|x\|=1} \{\|\alpha A\boldsymbol{x}\|\} = \sup_{\|x\|=1} \{|\alpha|\|A\boldsymbol{x}\|\} = |\alpha| \sup_{\|x\|=1} \{\|A\boldsymbol{x}\|\} = |\alpha|\|A\|$$
3. $$\begin{aligned} \|A+B\| &= \sup_{\|x\|=1} \{\|(A+B)\boldsymbol{x}\|\} = \sup_{\|x\|=1} \{\|A\boldsymbol{x} + B\boldsymbol{x}\|\} \\ &\leq \sup_{\|x\|=1} \{\|A\boldsymbol{x}\| + \|B\boldsymbol{x}\|\} \\ &\leq \sup_{\|x\|=1} \{\|A\boldsymbol{x}\|\} + \sup_{\|x\|=1} \{\|B\boldsymbol{x}\|\} \\ &= \|A\| + \|B\| \end{aligned}$$

Why introduce subordinate matrix norms? Because of some additional properties that they enjoy,

- $$\|I\| = 1$$
- $$\|A\boldsymbol{x}\| \leq \|A\|\|\boldsymbol{x}\|$$
- $$\|AB\| \leq \|A\|\|B\|$$

To derive them,

- $$\|I\| = \sup_{\|\boldsymbol{x}\|=1} \{\|I\boldsymbol{x}\|\} = \sup_{\|\boldsymbol{x}\|=1} \{\|\boldsymbol{x}\|\} = 1$$
- - When $\|\boldsymbol{x}\| = 1$, $\|A\| = \sup_{\|\boldsymbol{x}\|=1}\{\|A\boldsymbol{x}\|\} \geq \|A\boldsymbol{x}\| \Rightarrow \|A\|\|\boldsymbol{x}\| \geq \|A\boldsymbol{x}\|$
  - When $\|\boldsymbol{x}\| = \alpha \neq 1$, let $\mathbf{y} = \frac{\mathbf{x}}{\alpha}$, then $\|y\| = 1$, therefore

    $$\|A\mathbf{y}\| \leq \|A\|\|\mathbf{y}\| \Rightarrow \alpha\|\mathbf{Ay}\| \leq \alpha\|\mathbf{A}\|\|\mathbf{y}\| \Rightarrow \|\mathbf{A}\alpha\mathbf{y}\| \leq \|\mathbf{A}\|\|\alpha\mathbf{y}\| \Rightarrow \|\mathbf{A}\boldsymbol{x}\| \leq \|\mathbf{A}\|\|\boldsymbol{x}\|$$
- Left as homework.

Examples of subordinate matrix norms for a matrix A, based on the l_1, l_2 and l_∞ vector norms respectively, are

$$\|A\|_1 = \max_{1\le j\le n} \sum_{i=1}^{n} |a_{ij}|, \quad l_1 \text{ norm}$$

$$\|A\|_2 = \max_{1\le j\le n} \sigma_{max}, \quad l_2 \text{ norm}$$

$$\|A\|_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|, \quad l_\infty \text{ norm}$$

where σ_i are the square root of eigenvalues of A^TA, which are called the singular values of A. σ_{max} is the largest in absolute value among σ_i.

Example 1.7. Let the matrix A be

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

then

$$\|A\|_1 = 6,$$
$$\|A\|_2 = 5.4650,$$
$$\|A\|_\infty = 7.$$

To get $\|A\|_2$, use 'eig(A'*A)' in matlab to obtain $\sigma_1^2, \cdots, \sigma_n^2$, then pick the largest one among all σ_i's, σ_{max}.

The formulas for l_1, l_2 and l_∞ subordinate matrix norms can be derived by using Definition 1.6. For example,

$$\begin{aligned} \|A\|_2 &= \sup_{\|x\|_2=1} \|A\boldsymbol{x}\|_2 \\ &= \sup_{\|x\|_2=1} (\boldsymbol{x}^T A^T A \boldsymbol{x})^{\frac{1}{2}} \\ &= \sup_{\|x\|_2=1} (\boldsymbol{x}^T Q^T \Lambda Q \boldsymbol{x})^{\frac{1}{2}}, \\ &\overset{\mathbf{y}=Q\boldsymbol{x}}{=} \sup_{\|\mathbf{y}\|_2=1} (\mathbf{y}^T \Lambda \mathbf{y})^{\frac{1}{2}}, \\ &= \sigma_{max}. \end{aligned} \tag{3}$$

where $Q^T\Lambda Q$ is an eigenvalue decomposition of A^TA, where Q is an unitary matrix and $\Lambda = diag(\sigma_1^2, \cdots, \sigma_n^2)$.

2 Condition number and stability

Definition 2.1. (Condition number) Condition number of a matrix indicates if the solution of the linear system is sensitive to small changes. It turns out that this sensitivity can be measured by the condition number defined as

$$\kappa(A) = \|A\|_2 \|A^{-1}\|_2$$

To see condition number measures the sensitivity of the system, suppose that we want to solve an invertible linear system of equations $A\boldsymbol{x} = \boldsymbol{b}$. For a given A and $\boldsymbol{b}$, there may be some perturbations of the data owing to the uncertainty in measurement or the roundoff errors in computers. Suppose that the right hand side $\boldsymbol{b}$ is perturbed by δb and the corresponding solution to the problem is perturbed by an amount denoted by δx. Then we have

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

where we have

$$A\delta x = \delta b.$$

For the original linear system $A\boldsymbol{x} = \boldsymbol{b}$, we have

$$\|\boldsymbol{b}\| = \|A\boldsymbol{x}\| \leq \|A\| \|\boldsymbol{x}\|$$

which gives

$$\frac{1}{\|\boldsymbol{x}\|} \leq \frac{\|A\|}{\|\boldsymbol{b}\|}. \tag{4}$$

For the perturbed linear system $A\delta x = \delta b$, we have $\delta x = A^{-1}\delta b$ and therefore

$$\|\delta x\| = \|A^{-1}\delta b\| \leq \|A^{-1}\| \|\delta b\| \tag{5}$$

Combining eq.(4) and eq.(5) gives

$$\frac{\|\delta x\|}{\|\boldsymbol{x}\|} \leq \|A\| \|A^{-1}\| \frac{\|\delta b\|}{\|\boldsymbol{b}\|} = \kappa \frac{\|\delta b\|}{\|\boldsymbol{b}\|}. \tag{6}$$

Remark 2.2. $\frac{\|\delta b\|}{\|\boldsymbol{b}\|}$ is the relative perturbation we have on $\boldsymbol{b}$, and $\frac{\|\delta x\|}{\|\boldsymbol{x}\|}$ is the resulting relative perturbation we have on the solution $\boldsymbol{x}$ as a result of the perturbation on $\boldsymbol{b}$. Therefore the condition number of a matrix $\kappa(A)$ measures the sensitivity of the system to errors in data. When the condition number is large, the computed solution of the system may be dangerously in error. Further check should be made before accepting the solution as being accurate.

Remark 2.3. Condition number is always greater than 1. $\kappa(A) = \|A\| \|A^{-1}\| \geq \|AA^{-1}\| = \|I\| = 1$. Values of the condition number close to 1 indicate a well-conditioned matrix whereas large values indicate an ill-conditioned matrix.

3 Basic iterative method

The iterative method produces a sequence of approximate solution vector $\boldsymbol{x}^{(0)}$, $\boldsymbol{x}^{(1)}$, $\boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(k)}, \cdots$ for system of equations $A\boldsymbol{x} = \boldsymbol{b}$. The numerical procedure is designed such that, in principle, the sequence of approximate vectors converge to the actual solution, and as rapidly as possible. The process could be stop when the approximate solution is sufficiently close to the true solution or close to each other. This is in contract with the Gaussian elimination, which has no provisional solution. A general iterative procedure goes as follows:

1. Select a initial guess $\boldsymbol{x}^{(0)}$.

2. Design an iterative procedure:

$$Q\boldsymbol{x}^{(k)} = (Q - A)\boldsymbol{x}^{(k-1)} + \boldsymbol{b}, \forall\ k = 1, \cdots \tag{7}$$

To see that the iterative procedure eq.(7) actually is consistent with the original $A\boldsymbol{x} = \boldsymbol{b}$, we let $k \to \infty$ and presume that the approximate sequence converges to $\boldsymbol{x}$, then we have

$$Q\boldsymbol{x} = (Q - A)\boldsymbol{x} + \boldsymbol{b}$$

which leads to $A\boldsymbol{x} = \boldsymbol{b}$. Thus, if the sequence converge, its limit is the solution to the $A\boldsymbol{x} = \boldsymbol{b}$.

To have a method that is efficient, we hope to have the Q satisfying the following properties for the general iterative procedure (from eq. (7)),

1. Q is easy to invert.

2. The sequence $\boldsymbol{x}^{(k)}$ will converge to $\boldsymbol{x}$, no matter what the initial guess is.

3. The sequence $\boldsymbol{x}^{(k)}$ converges to $\boldsymbol{x}$ as rapidly as possible.

In the following, we will introduce three iterative methods: Jacobi method, the Gauss-Seidel method and the successive over-relaxation (SOR) method.

Jacobi method. Let's first write the system of equations $A\boldsymbol{x} = \boldsymbol{b}$ in its detailed form

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad 1 \leq i \leq n. \tag{8}$$

In the kth iteration, we solve the ith equation for the ith unknown $x_i^{(k)}$, assuming that the other x_j comes from the previous iteration $x_j^{(k-1)}$, we obtain an equation that describes the Jacobi method:

$$\sum_{j=1}^{i-1} a_{ij} x_j^{(k-1)} + a_{ii} x_i^{(k)} + \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)} = b_i, \quad 1 \leq i \leq n. \tag{9}$$

or

$$\sum_{j=1, j \neq i}^{n} a_{ij} x_j^{(k-1)} + a_{ii} x_i^{(k)} = b_i, \quad 1 \leq i \leq n. \tag{10}$$

which could be rearranged as

$$\begin{aligned} x_i^{(k)} &= (b_i - \sum_{j=1, j\neq i}^{n} a_{ij} x_j^{(k-1)})/a_{ii} \\ &= \frac{b_i}{a_{ii}} - \sum_{j=1, j\neq i}^{n} \frac{a_{ij}}{a_{ii}} x_j^{(k-1)}. \end{aligned} \tag{11}$$

Here we assume that all diagonal entries are nonzero. If this is not the case, we can usually rearrange the equation so that it is.

The equation (9) could be written in the following matrix form

$$\left(\begin{array}{llll} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{array}\right) \boldsymbol{x}^{(k)} + \left(\begin{array}{llll} 0 & a_{12} & \cdots & a_{1n} \\ a_{21} & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & 0 \end{array}\right) \boldsymbol{x}^{(k-1)} = \boldsymbol{b}$$

If we decompose the matrix A as $A = D - C_L - C_U$, where D is the diagonal part of A, C_L is the negative lower triangular part of A and C_U is the negative upper triangular part of A,

$$D = diag(A), \quad C_L = (-a_{ij})_{i>j}, \quad C_U = (-a_{ij})_{i<j},$$

then the above matrix representation of Jacobi matrix is

$$D\boldsymbol{x}^{(k)} + (A - D)\boldsymbol{x}^{(k-1)} = \boldsymbol{b}.$$

Rearrange it a little bit, we have

$$D\boldsymbol{x}^{(k)} = (D - A)\boldsymbol{x}^{(k-1)} + \boldsymbol{b},$$

in the form of eq.(7) with $Q = D$.

Example 3.1. (Jacobi iterative method) Let

$$A = \left(\begin{array}{lll} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{array}\right), \quad b = \left(\begin{array}{l} 1 \\ 8 \\ -5 \end{array}\right).$$

Carry out a number of Jacobi iteration, starting with zero initial vector.
Solution: Rewriting the equation, we have

$$\begin{aligned} x_1^{(k)} &= \frac{1}{2} x_2^{(k-1)} + \frac{1}{2} \\ x_2^{(k)} &= \frac{1}{3} x_1^{(k-1)} + \frac{1}{3} x_3^{(k-1)} + \frac{8}{3} && (12) \\ x_3^{(k)} &= \frac{1}{2} x_2^{(k-1)} - \frac{5}{2}. && (13) \end{aligned}$$

Taking the initial vector to be $\boldsymbol{x}^{(0)} = [0,0,0]'$, we find that

$$\begin{aligned}
\boldsymbol{x}^{(1)} &= [0.5000, 2.6667, -2.500]' \\
\boldsymbol{x}^{(2)} &= [1.8333, 2.0000, -1.1667]' \\
&\cdots \\
\boldsymbol{x}^{(21)} &= [2.0000, 3.0000, -1.0000]'
\end{aligned}$$

After 21 iterations, the actual solution is obtained within some fixed precision.

In the Jacobi method, the matrix Q is taken to be the diagonal part of A,

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

With this Q, we know that the Jacobi method could also be implemented as

$$\boldsymbol{x}^{(k)} = B\boldsymbol{x}^{(k-1)} + \mathbf{h}$$

with the Jacobi iterative matrix B and constant vector $\mathbf{h}$ are

$$B = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 \end{pmatrix}, \quad \mathbf{h} = \begin{pmatrix} 1/2 \\ 8/3 \\ -5/2 \end{pmatrix}.$$

Gauss-Seidel method. Let's first write the system of equations $A\boldsymbol{x} = \boldsymbol{b}$ in its detailed form

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \quad 1 \leq i \leq n.$$

In the Jacobi method, the equations are solved in order. When solving the ith equation, the component $x_j^{(k)}$ $(1 \leq j < i)$ can be immediately in their place, and is expected to be more accurate than $x_j^{(k-1)}$ $(1 \leq j < i)$. Taking into account of this, we obtain an equation that describes the Gauss-Seidel (GS) method:

$$\sum_{j=1}^{i-1} a_{ij}x_j^{(k)} + a_{ii}x_i^{(k)} + \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)} = b_i, \quad 1 \leq i \leq n. \tag{14}$$

which could be rearranged as

$$\begin{aligned}
x_i^{(k)} &= (b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)})/a_{ii} \\
&= \frac{b_i}{a_{ii}} - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}}x_j^{(k)} - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}}x_j^{(k-1)} \qquad (15)
\end{aligned}$$

Here we assume that all diagonal entries are nonzero. If this is not the case, we can usually rearrange the equation so that it is. The equation (14) could be written in the following matrix form

$$\begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \boldsymbol{x}^{(k)} + \begin{pmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix} \boldsymbol{x}^{(k-1)} = \boldsymbol{b}$$

Use the notation of decomposing $A = D - C_L - C_U$, then the above matrix representation of GS matrix is

$$(D - C_L)\boldsymbol{x}^{(k)} + (A - D + C_L)\boldsymbol{x}^{(k-1)} = \boldsymbol{b}.$$

Rearrange it a little bit, we have

$$(D - C_L)\boldsymbol{x}^{(k)} = (D - C_L - A)\boldsymbol{x}^{(k-1)} + \boldsymbol{b},$$

in the form of eq.(7) with $Q = D - C_L$.

Example 3.2. (GS iterative method) Let A and $\boldsymbol{b}$ the same as in Example 3.1. Carry out a number of GS iterations, starting with zero initial vector.
Solution: Rewriting the equation, we have

$$\begin{aligned} x_1^{(k)} &= \frac{1}{2}x_2^{(k-1)} + \frac{1}{2} \\ x_2^{(k)} &= \frac{1}{3}x_1^{(k)} + \frac{1}{3}x_3^{(k-1)} + \frac{8}{3} && (16) \\ x_3^{(k)} &= \frac{1}{2}x_2^{(k)} - \frac{5}{2}. && (17) \end{aligned}$$

Taking the initial vector to be $\boldsymbol{x}^{(0)} = [0, 0, 0]'$, we find that

$$\begin{aligned} \boldsymbol{x}^{(1)} &= [0.5000, 2.8333, -1.0833]' \\ \boldsymbol{x}^{(2)} &= [1.9167, 2.9444, -1.0278]' \\ &\ \ \cdots \\ \boldsymbol{x}^{(9)} &= [2.0000, 3.0000, -1.0000]' \end{aligned}$$

After 9 iterations, the actual solution is obtained within some fixed precision.

In the GS method, the matrix Q is taken to be the lower triangular part of A,

$$\begin{pmatrix} 2 & 0 & 0 \\ -1 & 3 & 0 \\ 0 & -1 & 2 \end{pmatrix}$$

With this Q, we know that the GS method could also be implemented as

$$\boldsymbol{x}^{(k)} = B\boldsymbol{x}^{(k-1)} + \mathbf{h}$$

with the GS iterative matrix B and constant vector $\mathbf{h}$ are

$$B = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{6} & \frac{1}{3} \\ 0 & \frac{1}{12} & \frac{1}{6} \end{pmatrix}, \quad \mathbf{h} = \begin{pmatrix} 1/2 \\ 17/6 \\ -13/12 \end{pmatrix}. \qquad (18)$$

Successive Overrelaxation (SOR) method. Let's first write the system of equations $A\boldsymbol{x} = \boldsymbol{b}$ in its detailed form

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \quad 1 \le i \le n.$$

The idea of the SOR method is essentially the same as the GS method, except that it also use $x_i^{(k-1)}$ to solve for $x_i^{(k)}$. The algorithm is the following

$$\sum_{j=1}^{i-1} a_{ij}x_j^{(k)} + a_{ii}(\frac{1}{w}x_i^{(k)} + (1-\frac{1}{w})x_i^{(k-1)}) + \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)} = b_i, \quad 1 \le i \le n. \tag{19}$$

which could be rearranged as

$$\begin{aligned} x_i^{(k)} &= w(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k-1)})/a_{ii} + (1-w)x_i^{(k-1)} \\ &= w(\frac{b_i}{a_{ii}} - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}}x_j^{(k)} - \sum_{j=i+1}^{n} \frac{a_{ij}}{a_{ii}}x_j^{(k-1)}) + (1-w)x_i^{(k-1)} \qquad (20) \end{aligned}$$

Again we assume that all diagonal entries are nonzero. The equation (19) could be written in the following matrix form

$$\begin{pmatrix} a_{11}/w & 0 & \cdots & 0 \\ a_{21} & a_{22}/w & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn}/w \end{pmatrix} \boldsymbol{x}^{(k)} + \begin{pmatrix} (1-\frac{1}{w})a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & (1-\frac{1}{w})a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & (1-\frac{1}{w})a_{nn} \end{pmatrix} \boldsymbol{x}^{(k-1)} = \boldsymbol{b}$$

Use the notation of decomposing $A = D - C_L - C_U$, then the above matrix representation of SOR matrix is

$$(D/w - C_L)\boldsymbol{x}^{(k)} + (A - D/w + C_L)\boldsymbol{x}^{(k-1)} = \boldsymbol{b}.$$

Rearrange it a little bit, we have

$$(D/w - C_L)\boldsymbol{x}^{(k)} = (D/w - C_L - A)\boldsymbol{x}^{(k-1)} + \boldsymbol{b},$$

in the form of eq.(7) with $Q = D/w - C_L$.

Example 3.3. (SOR iterative method) Let A and $\boldsymbol{b}$ the same as in Example 3.1. Carry out a number of SOR iterations with $w = 1.1$, starting with zero initial vector.

Solution: Rewriting the equation, we have

$$\begin{aligned} x_1^{(k)} &= w(\frac{1}{2}x_2^{(k-1)} + \frac{1}{2}) + (1-w)x_1^{(k-1)} \\ x_2^{(k)} &= w(\frac{1}{3}x_1^{(k)} + \frac{1}{3}x_3^{(k-1)} + \frac{8}{3}) + (1-w)x_2^{(k-1)} \qquad (21) \\ x_3^{(k)} &= w(\frac{1}{2}x_2^{(k)} - \frac{5}{2}) + (1-w)x_3^{(k-1)}. \qquad (22) \end{aligned}$$

Taking the initial vector to be $\boldsymbol{x}^{(0)} = [0,0,0]'$, we find that

$$\begin{aligned}
\boldsymbol{x}^{(1)} &= [0.5500, 3.1350, -1.0257]' \\
\boldsymbol{x}^{(2)} &= [2.2193, 3.0574, -0.9658]' \\
& \ldots \\
\boldsymbol{x}^{(7)} &= [2.0000, 3.0000, -1.0000]'
\end{aligned}$$

After 7 iterations, the actual solution is obtained within some fixed precision.

In the SOR method, the matrix Q is taken to be

$$\begin{pmatrix} 2/w & 0 & 0 \\ -1 & 3/w & 0 \\ 0 & -1 & 2/w \end{pmatrix}$$

With this Q, we know that the SOR(w) method could also be implemented as

$$\boldsymbol{x}^{(k)} = B\boldsymbol{x}^{(k-1)} + \mathbf{h}$$

with the SOR iterative matrix B and constant vector $\mathbf{h}$ are

$$B = \begin{pmatrix} -1/10 & 11/20 & 0 \\ -11/300 & 61/600 & 11/30 \\ -121/6000 & -671/12000 & 61/600 \end{pmatrix}, \quad \mathbf{h} = \begin{pmatrix} 11/20 \\ 627/200 \\ -4103/4000 \end{pmatrix}. \tag{23}$$

with $w = 1.1$.

Remark 3.4. (Complexity analysis) In each of the iterative step in the methods described above, the complexity is of order $\mathcal{O}(n^2)$.

Remark 3.5. (Stopping criteria) The stopping criteria in the iterative methods for solving $A\boldsymbol{x} = \boldsymbol{b}$ is to make sure that the distance, measure in norms, between approximations are bounded by some prescribed tolerance,

$$\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\|_2 < \epsilon,$$

where ϵ is the tolerance of the error.

4 Convergence of iterative methods

For the analysis of the method described by eq.(7), we write

$$\boldsymbol{x}^{(k)} = Q^{-1}[(Q - A)\boldsymbol{x}^{(k-1)} + \boldsymbol{b}]. \tag{24}$$

Let the error at the kth iteration be $\mathbf{e}^{(k)} = \boldsymbol{x} - \boldsymbol{x}^{(k)}$. Let $\boldsymbol{x}$ mines both sides of eq.(24), we have

$$\begin{aligned}
\mathbf{e}^{(k)} &= \boldsymbol{x} - Q^{-1}[(Q-A)\boldsymbol{x}^{(k-1)} + \boldsymbol{b}] \\
&= \boldsymbol{x} - (I - Q^{-1}A)\boldsymbol{x}^{(k-1)} - Q^{-1}\boldsymbol{b} \\
&= \boldsymbol{x} - \boldsymbol{x}^{(k-1)} + Q^{-1}A\boldsymbol{x}^{(k-1)} - Q^{-1}\boldsymbol{b} \\
&= \mathbf{e}^{(k-1)} + Q^{-1}A\boldsymbol{x}^{(k-1)} - Q^{-1}A\boldsymbol{x} \\
&= \mathbf{e}^{(k-1)} + Q^{-1}A(\boldsymbol{x}^{(k-1)} - \boldsymbol{x}) \\
&= \mathbf{e}^{(k-1)} - Q^{-1}A\mathbf{e}^{(k-1)} \\
&= (I - Q^{-1}A)\mathbf{e}^{(k-1)}. \qquad (25)
\end{aligned}$$

We want to have $\mathbf{e}^{(k)}$ to become smaller as we increase the k. The above derivation shows that $\mathbf{e}^{(k)}$ will be smaller than $\mathbf{e}^{(k-1)}$ if $I - Q^{-1}A$ is small in some sense. Indeed, from eq.(25), we have

$$\begin{aligned}
\|\mathbf{e}^{(k)}\| &= \|(I - Q^{-1}A)\mathbf{e}^{(k-1)}\| \\
&\leq \|(I - Q^{-1}A)\|\|\mathbf{e}^{(k-1)}\| \qquad (26)
\end{aligned}$$

As can be seen from eq.(26), if $\|(I - Q^{-1}A)\| < 1$, the error becomes smaller and smaller as the iteration goes on, therefore the iterative method converges. What is more, the smaller the $\|(I - Q^{-1}A)\|$ is, the faster convergence we would expect. A very classical theorem about the convergence of the iterative method is the following

Theorem 4.1. *(Spectral Radius Theorem) In order that the sequence generated by eq.(7) to converge, no matter what the starting point $\boldsymbol{x}^{(0)}$ is selected, it is necessary and sufficient that all eigenvalues of the matrix $I - Q^{-1}A$ lies in the open unit disc, $|z| < 1$, in the complex plane.*

Proof. Let $B = I - Q^{-1}A$, then

$$\mathbf{e}^{(k)} = B^k \mathbf{e}^{(0)},$$

hence

$$\|\mathbf{e}^{(k)}\| \leq \|B^k\|\|\mathbf{e}^{(0)}\|.$$

By the spectral radius Theorem,

$$\rho(B) < 1 \Leftrightarrow \lim_{k\to\infty} B^k = 0.$$

The iterative method converges if and only if $\rho(B) < 1$. □

The conclusion of the theorem can also be written as

$$\rho(I - Q^{-1}A) < 1$$

where ρ is the spectral radius function of a matrix: for a n-by-n matrix A, with eigenvalues λ_i, the

$$\rho(A) = \max_i \{|\lambda_i|\}$$

In Example 3.1, we have use Jacobi, GS and SOR method to iteratively solve it. We have observed that they take 21, 9 and 7 iterations respectively to obtain solutions within the same tolerance. Actually, this behavior could be predicted by the eigenvalues of $I - Q^{-1}A$.

Example 4.2. Determine whether the Jacobi, GS and SOR method will converge for the matrix A and $\boldsymbol{b}$ in Example 3.1, no matter what the initial condition is.

Solution: For the Jacobi method, we can easily compute the eigenvalues of the relevant matrix $I - Q^{-1}A$ (the matrix B in Example 3.1). The steps are

$$det(B - \lambda I) = det \begin{pmatrix} -\lambda & 1/2 & 0 \\ 1/3 & -\lambda & 1/3 \\ 0 & 1/2 & -\lambda \end{pmatrix} = -\lambda^3 + \frac{1}{3}\lambda = 0.$$

Solving for λ gives us the three eigenvalues are $0, \pm 0.5774$, all of which lies in the open unit disk. Thus, the Jacobi method converges.

Similarly, for the GS method, the eigenvalues of the relevant matrix $I - Q^{-1}A$ (the B from Example 3.2 eq.(18)) are determined by

$$det(B - \lambda I) = det \begin{pmatrix} -\lambda & 11/20 & 0 \\ 0 & 1/6 - \lambda & 1/3 \\ 0 & 1/12 & 1/6 - \lambda \end{pmatrix} = -\lambda(1/6 - \lambda)^2 + \frac{1}{36}\lambda = 0.$$

Solving for λ gives us the three eigenvalues are $0, 0, 0.3333$. Thus, the GS method converges.

Similarly, for the SOR method with $w = 1.1$, the eigenvalues of the relevant matrix $I - Q^{-1}A$ (the B from Example 3.3 eq.(23)) are determined by

$$\begin{aligned} det(B - \lambda I) &= det \begin{pmatrix} -1/10 - \lambda & 11/20 & 0 \\ -11/300 & 61/600 - \lambda & 11/30 \\ -121/6000 & 671/12000 & 61/600 - \lambda \end{pmatrix} \\ &= -1/1000 + 31/3000\lambda + 31/3000\lambda^2 - \lambda^3 = 0. \end{aligned}$$

Solving for λ gives us the three eigenvalues are $\approx 0.1200, 0.0833, -0.1000$. Thus, the SOR method converges.

Also from the magnitude of those eigenvalues, it is not surprise that the SOR performs better than GS, then Jacobi, in terms of the efficiency.