

Unit 3 Iterative Solutions to Linear Systems

Numerical Analysis

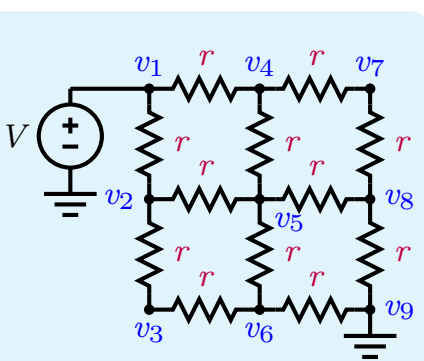
Mar. 26, 2015

A Resistor Network

- Let $g = 1/r$, then the system of equations for the resistor network can be formulated as

$$\begin{bmatrix} 1 & & & & & & & & \\ -g & 3g & -g & & -g & & & & \\ & -g & 2g & & -g & & & & \\ -g & & & 3g & -g & -g & & & \\ & -g & & -g & 4g & -g & & -g & \\ & & -g & & -g & 3g & & & \\ & & & -g & & & 2g & -g & -g \\ & & & & -g & & -g & 3g & -g \\ & & & & & & & & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \\ v_9 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (3.1.1)$$

- Since $v_1 = V$ and $v_9 = 0$, it can be reformulated as



$$\begin{bmatrix} 3g & -g & & -g & & & & & \\ -g & 2g & & -g & & & & & \\ & & 3g & -g & -g & & & & \\ -g & & -g & 4g & -g & -g & & & \\ & -g & & -g & 3g & & & & \\ & & -g & & & 2g & -g & & \\ & & & -g & & -g & 3g & & \end{bmatrix} \begin{bmatrix} v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{bmatrix} = \begin{bmatrix} gV \\ 0 \\ gV \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (3.1.2)$$

- Note the sparsity of the matrices for both Eqs. (3.1.1) and (3.1.2).
- Eq. (3.1.2) has two fewer variables and lower matrix dimension.
- Matrix of Eq. (3.1.2) is symmetric.
- The structure of the matrix is not dependent on the values of the resistors
- If the resistor mesh is large, the forms of the matrices remain the same.
 - Sparsity and symmetry.
- Gaussian elimination or any direct solution methods applicable to find the solution.
- Fill-ins will be created that decrease the sparsity of the matrices.
- Iterative solution methods can be effective in solving this kind of matrices.
 - Can provide higher degree of parallelism in solving the system of equations.

Jacobi Method

- For the resistor network, any node voltage, v_i , must satisfy Kirchhoff current law.

$$\sum_j g_{ij}(v_i - v_j) = 0. \quad (3.1.3)$$

The summation is carried out for any node j that is connected to node i through a resistor, $r_{ij} = 1/g_{ij}$.

- Assembly all the node variables, then the system of equations (3.1.1) or (3.1.2) is formed.
- The Eq. (3.1.3) can be reformulated as

$$\begin{aligned} \sum_j g_{ij}v_i &= \sum_j g_{ij}v_j \\ G_i v_i &= \sum_j g_{ij}v_j \\ v_i &= \sum_j \frac{g_{ij}}{G_i} v_j \end{aligned} \quad (3.1.4)$$

where $G_i = \sum_j g_{ij}$ and the summation is carried out where j is connected to node i through a resistor.

Jacobi Method, II

- Compare Eqs. (3.1.2) and (3.1.4), we found that (3.1.4) can be formed by letting $G_i = a_{ii}$, the diagonal matrix element, and moving all off-diagonal elements to the right hand side of the equations.
- For the resistor network, we have

$$v_2 = 1/3(V + v_3 + v_5)$$

$$v_3 = 1/2(v_2 + v_6)$$

$$v_4 = 1/3(V + v_5 + v_7)$$

$$v_5 = 1/4(v_2 + v_4 + v_6 + v_8)$$

$$v_6 = 1/3(v_3 + v_5)$$

$$v_7 = 1/2(v_4 + v_8)$$

$$v_8 = 1/3(v_5 + v_7)$$

$$v_2^{(k+1)} = 1/3(V + v_3^{(k)} + v_5^{(k)})$$

$$v_3^{(k+1)} = 1/2(v_2^{(k)} + v_6^{(k)})$$

$$v_4^{(k+1)} = 1/3(V + v_5^{(k)} + v_7^{(k)})$$

$$v_5^{(k+1)} = 1/4(v_2^{(k)} + v_4^{(k)} + v_6^{(k)} + v_8^{(k)})$$

$$v_6^{(k+1)} = 1/3(v_3^{(k)} + v_5^{(k)})$$

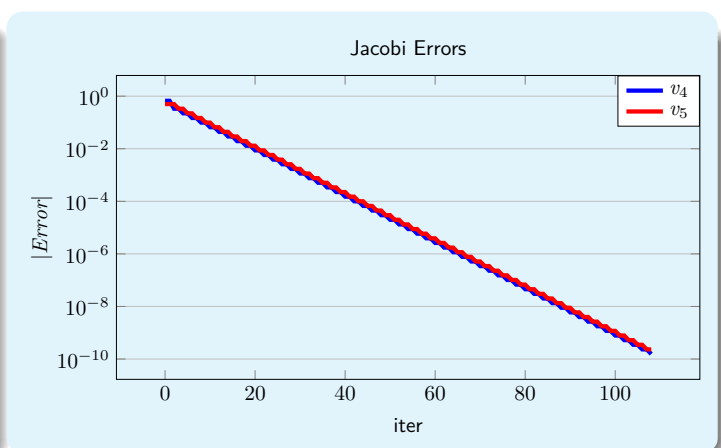
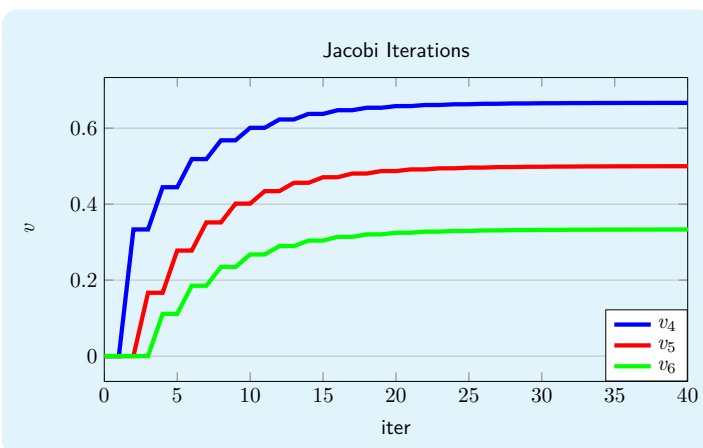
$$v_7^{(k+1)} = 1/2(v_4^{(k)} + v_8^{(k)})$$

$$v_8^{(k+1)} = 1/3(v_5^{(k)} + v_7^{(k)})$$

- Jacobi method assumes an initial guess $v^{(0)}$ for the node's voltages, and applies it to the right hand side to find the new values of the solution, $v^{(k+1)}$.
- This process is repeated until a converged solution is found.

Jacobi Method, III

- The Jacobi method is shown to be convergent for the resistor network problem.
- The convergent rate is shown to be constant.
 - It takes 100 iterations to reach $\epsilon < 1e-9$.
- Each iteration takes NZ' multiplications, where NZ' is the number of off-diagonal elements in the system matrix, and n divisions.
- This method takes full advantage of the sparsity of the matrix.



Jacobi Method, IV

- Given a linear system $\mathbf{Ax} = \mathbf{b}$, it can also be written as

$$\sum_{j=1}^n a_{ij}x_j = b_i, \quad i = 1, \dots, n. \quad (3.1.5)$$

Assuming $a_{ii} \neq 0$, $i = 1, \dots, n$, it can be reformulated as

$$x_i = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1, j \neq i}^n a_{ij}x_j \right], \quad i = 1, \dots, n. \quad (3.1.6)$$

Given an initial guess $\mathbf{x}^{(0)}$, then the Jacobi method is to compute new solution at iteration k by

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1, j \neq i}^n a_{ij}x_j^{(k)} \right], \quad i = 1, \dots, n. \quad (3.1.7)$$

Jacobi Method, V

- In matrix form, we write

$$\mathbf{A} = -\mathbf{E} + \mathbf{D} - \mathbf{F}, \quad (3.1.8)$$

$$\begin{aligned} \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} &= \begin{bmatrix} 0 & 0 & \cdots & 0 \\ a_{21} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & 0 \end{bmatrix} + \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} \\ &+ \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \end{aligned} \quad (3.1.9)$$

where \mathbf{E} is a strictly lower triangular matrix, \mathbf{D} is a diagonal matrix and \mathbf{F} is a strictly upper triangular matrix. Then Jacobi iteration is

$$\mathbf{x}^{(k+1)} = \mathbf{D}^{-1} [\mathbf{b} + (\mathbf{E} + \mathbf{F})\mathbf{x}^{(k)}]. \quad (3.1.10)$$

Over-Relaxation Method (JOR)

- The Jacobi method can be generalized as

$$x_i^{(k+1)} = \frac{\omega}{a_{ii}} \left[b_i - \sum_{j=1, j \neq i}^n a_{ij} x_j^{(k)} \right] + (1 - \omega) x_i^{(k)}, \quad i = 1, \dots, n. \quad (3.1.11)$$

Or in matrix form

$$\mathbf{x}^{(k+1)} = \omega \mathbf{D}^{-1} [\mathbf{b} + (\mathbf{E} + \mathbf{F}) \mathbf{x}^{(k)}] + (1 - \omega) \mathbf{x}^{(k)}. \quad (3.1.12)$$

Define the residue at k th iteration as

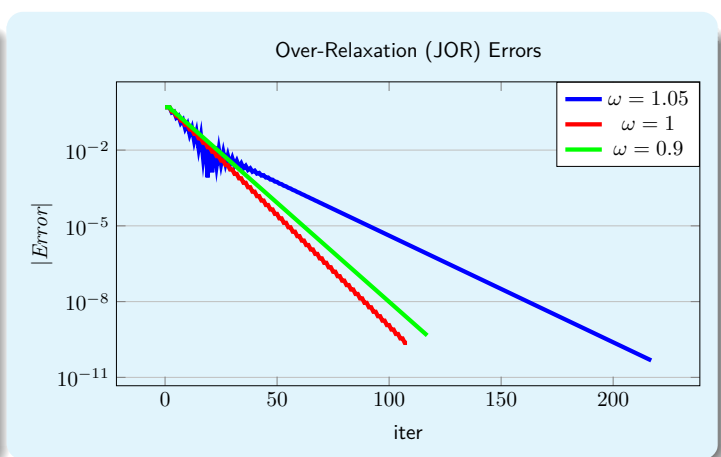
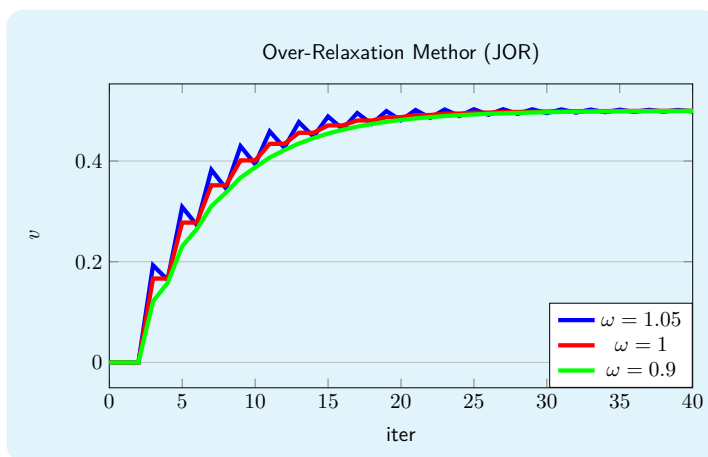
$$\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{A} \mathbf{x}^{(k)}, \quad (3.1.13)$$

then

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \omega \mathbf{D}^{-1} (\mathbf{r} + \mathbf{A} \mathbf{x}^{(k)} + (\mathbf{E} + \mathbf{F}) \mathbf{x}^{(k)} - \mathbf{D} \mathbf{x}^{(k)}) \\ &= \mathbf{x}^{(k)} + \omega \mathbf{D}^{-1} \mathbf{r}^{(k)}. \end{aligned} \quad (3.1.14)$$

- When $\omega = 1$ then it reduces to the Jacobi iteration.
- When $\omega < 1$ it is called under-relaxation.
- When $\omega > 1$ it is called over-relaxation.

Over-Relaxation Method (JOR), II



- In Jacobi iterations we see the solution converges by increasing the node voltages toward the solutions.
- Over-relaxation method predicts the solution with larger amount of voltage increases.
- For the resistor network problem, the over-relaxation method does not appear to improve the convergence rate.

Gauss-Seidel Method

- Note that in Jacobi iterations, some variables are updated before others. These updated values can be used in the current iteration to speed up the convergence.

$$\begin{aligned}
 v_2 &= 1/3(V + v_3 + v_5) \\
 v_3 &= 1/2(v_2 + v_6) \\
 v_4 &= 1/3(V + v_5 + v_7) \\
 v_5 &= 1/4(v_2 + v_4 + v_6 + v_8) \\
 v_6 &= 1/3(v_3 + v_5) \\
 v_7 &= 1/2(v_4 + v_8) \\
 v_8 &= 1/3(v_5 + v_7) \\
 v_2^{(k+1)} &= 1/3(V + v_3^{(k)} + v_5^{(k)}) \\
 v_3^{(k+1)} &= 1/2(v_2^{(k+1)} + v_6^{(k)}) \\
 v_4^{(k+1)} &= 1/3(V + v_5^{(k)} + v_7^{(k)}) \\
 v_5^{(k+1)} &= 1/4(v_2^{(k+1)} + v_4^{(k+1)} + v_6^{(k)} + v_8^{(k)}) \\
 v_6^{(k+1)} &= 1/3(v_3^{(k+1)} + v_5^{(k+1)}) \\
 v_7^{(k+1)} &= 1/2(v_4^{(k+1)} + v_8^{(k)}) \\
 v_8^{(k+1)} &= 1/3(v_5^{(k+1)} + v_7^{(k+1)})
 \end{aligned}$$

- Thus, Gauss-Seidel method has the same iteration process as Jacobi method, and it uses updated values whenever possible.
- Convergence rate is expected to be faster.

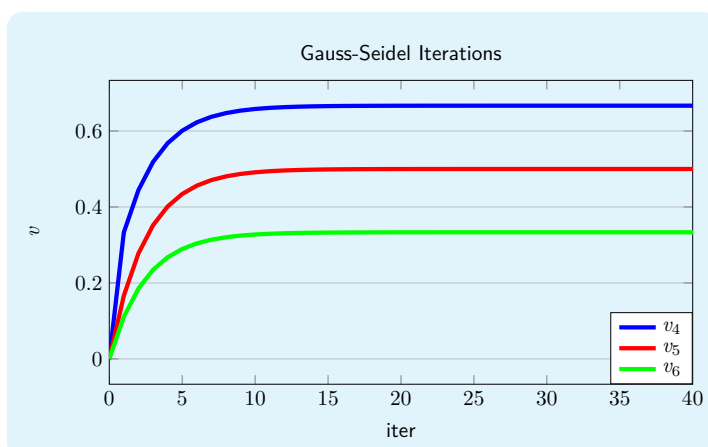
Gauss-Seidel Method, II

- Comparing to the Jacobi method, Eq. (3.1.7), Gauss-Seidel method can be formulated as

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right], \quad i = 1, \dots, n. \quad (3.1.15)$$

Or using matrix form, as compared to Eq. (3.1.10),

$$\begin{aligned}
 \mathbf{D}\mathbf{x}^{(k+1)} &= \mathbf{b} + \mathbf{E}\mathbf{x}^{(k+1)} + \mathbf{F}\mathbf{x}^{(k)}, \\
 (\mathbf{D} - \mathbf{E})\mathbf{x}^{(k+1)} &= \mathbf{b} + \mathbf{F}\mathbf{x}^{(k)}, \\
 \mathbf{x}^{(k+1)} &= (\mathbf{D} - \mathbf{E})^{-1}(\mathbf{b} + \mathbf{F}\mathbf{x}^{(k)}). \quad (3.1.16)
 \end{aligned}$$



Successive Over-Relaxation Method (SOR)

- The Gauss-Seidel appears to have better convergence rate than the Jacobi method.
- The Gauss-Seidel method can also be generalized as the following successive over-relaxation method.

$$x_i^{(k+1)} = \frac{\omega}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right] + (1 - \omega)x_i^{(k)}, \quad i = 1, \dots, n. \quad (3.1.17)$$

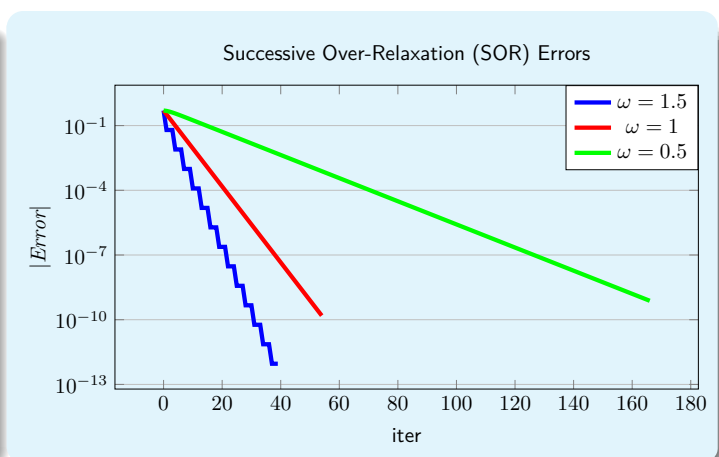
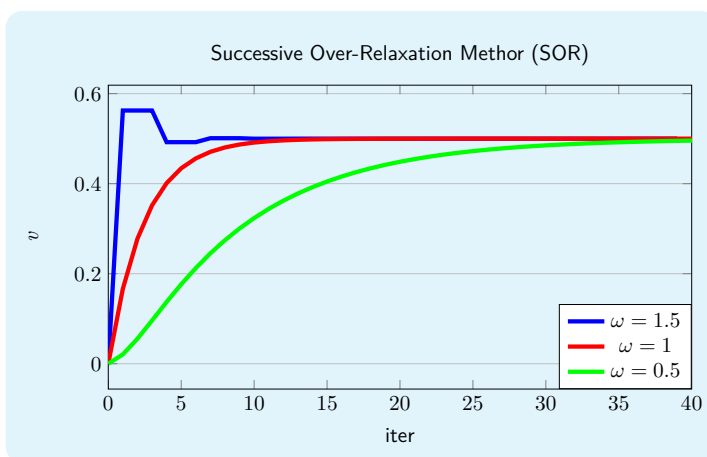
Or in matrix form

$$\begin{aligned} (\mathbf{D} - \omega\mathbf{E})\mathbf{x}^{(k+1)} &= \omega\mathbf{b} + [\omega\mathbf{F} + (1 - \omega)\mathbf{D}]\mathbf{x}^{(k)} \\ &= \omega[\mathbf{A}\mathbf{x}^{(k)} + \mathbf{r}^{(k)}] + [\omega\mathbf{F} + (1 - \omega)\mathbf{D}]\mathbf{x}^{(k)} \\ &= \omega\mathbf{r}^{(k)} + [\omega(-\mathbf{E} + \mathbf{D} - \mathbf{F}) + \omega\mathbf{F} + \mathbf{D} - \omega\mathbf{D}]\mathbf{x}^{(k)} \\ &= \omega\mathbf{r}^{(k)} + (\mathbf{D} - \omega\mathbf{E})\mathbf{x}^{(k)} \end{aligned} \quad (3.1.18)$$

Thus

$$\begin{aligned} \mathbf{x}^{(k+1)} &= \mathbf{x}^{(k)} + \omega(\mathbf{D} - \omega\mathbf{E})^{-1}\mathbf{r}^{(k)} \\ &= \mathbf{x}^{(k)} + \left(\frac{1}{\omega}\mathbf{D} - \mathbf{E} \right)^{-1} \mathbf{r}^{(k)}. \end{aligned} \quad (3.1.19)$$

Successive Over-Relaxation Method (SOR), II



- The successive over-relaxation method can improve the convergence rate.
- For the resistor network problem
 - $\omega < 1$ has lower convergence rate,
 - $\omega = 1.5$ has higher convergence rate,
 - when ω is too large, it does not converge.
- Implementation using Eq. (3.1.17) is easier than Eq. (3.1.19).

Iterative Methods

- In this Chapter, we solve the following linear system using iterative method

$$\mathbf{Ax} = \mathbf{b}. \quad (3.1.20)$$

- A general iterative method is to form a sequence

$$\begin{aligned} \mathbf{x}^{(0)} &= \mathbf{f}_0(\mathbf{A}, \mathbf{b}), \\ \mathbf{x}^{(k+1)} &= \mathbf{f}_{k+1}(\mathbf{x}^{(k)}, \mathbf{x}^{(k-1)}, \dots, \mathbf{x}^{(n-m)}, \mathbf{A}, \mathbf{b}), \text{ for } n \geq m, \end{aligned} \quad (3.1.21)$$

such that $\lim_{k \rightarrow \infty} \mathbf{x}^{(k)} = \mathbf{x}$, and \mathbf{x} satisfies Eq. (3.1.20).

- The number of steps that $\mathbf{x}^{(k+1)}$ depends on is called the **order of the method**.
- If \mathbf{f}_k is independent of k , then the method is called **stationary**, otherwise it is called **nonstationary**.
- If \mathbf{f}_k depends linearly on $\mathbf{x}^{(0)}, \dots, \mathbf{x}^{(m)}$, then the method is called **linear**, otherwise it is called **nonlinear**.
- In this section, our methods are stationary linear iterative methods of the first order that have the following form, given $\mathbf{x}^{(0)}$

$$\mathbf{x}^{(k+1)} = \mathbf{B}\mathbf{x}^{(k)} + \mathbf{f}, \quad k \geq 0, \quad (3.1.22)$$

where \mathbf{B} is an $n \times n$ matrix and is called the **iteration matrix**.

Iterative Methods, II

- Note that Eq. (3.1.22) needs to converge to the solution of Eq. (3.1.20). Thus, when converge

$$\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{f}$$

$$(\mathbf{I} - \mathbf{B})\mathbf{x} = \mathbf{f}$$

$$\mathbf{x} = (\mathbf{I} - \mathbf{B})^{-1}\mathbf{f} = \mathbf{A}^{-1}\mathbf{b}$$

Thus,

$$\mathbf{f} = (\mathbf{I} - \mathbf{B})\mathbf{A}^{-1}\mathbf{b}.$$

Definition 3.1.1.

A stationary linear iterative method of the equation form (3.1.22) is said to be **consistent** with (3.1.20) if it satisfies

$$\mathbf{f} = (\mathbf{I} - \mathbf{B})\mathbf{A}^{-1}\mathbf{b}. \quad (3.1.23)$$

Definition 3.1.2.

The **error at the k -th iteration** is given by

$$\mathbf{e}^{(k)} = \mathbf{x}^{(k)} - \mathbf{x}. \quad (3.1.24)$$

If the iterative method is convergent then $\lim_{k \rightarrow \infty} \mathbf{e}^{(k)} = \mathbf{0}$ for any choice of initial guess $\mathbf{x}^{(0)}$.

Iterative Methods, III

Theorem 3.1.3.

Given a consistent iteration method as

$$\mathbf{x}^{(k+1)} = \mathbf{B}\mathbf{x}^{(k)} + \mathbf{f}, \quad k \geq 0,$$

then the sequence of vectors $\{\mathbf{x}^{(k)}\}$ converges to the solution of

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

for any initial guess $\mathbf{x}^{(0)}$ if and only if $\rho(\mathbf{B}) < 1$.

Definition 3.1.4.

Let \mathbf{B} be the iteration matrix, then

1. $\|\mathbf{B}^m\|$ is the **convergence factor** after m steps of the iteration;
2. $\|\mathbf{B}^m\|^{1/m}$ is the **average convergence factor** after m steps;
3. $R_m(\mathbf{B}) = -\frac{1}{m} \log \|\mathbf{B}^m\|$ is the **average convergence rate** after m steps;
4. $R(\mathbf{B}) = \lim_{k \rightarrow \infty} R_k(\mathbf{B}) = -\log \rho(\mathbf{B})$ is the **asymptotic convergence rate**.

- The convergence behavior is determined by the matrix \mathbf{B} .

Iterative Methods, IV

- A general approach to solve the linear system of $\mathbf{A}\mathbf{x} = \mathbf{b}$ is to split matrix \mathbf{A} additively to the form $\mathbf{A} = \mathbf{P} - \mathbf{N}$ with \mathbf{P} nonsingular. Matrix \mathbf{P} is called **preconditioning matrix** or **preconditioner**. Then the iteration is

$$\mathbf{P}\mathbf{x}^{(k+1)} = \mathbf{N}\mathbf{x}^{(k)} + \mathbf{b}, \quad k \geq 0. \quad (3.1.25)$$

Comparing to Eq. (3.1.22), we have $\mathbf{B} = \mathbf{P}^{-1}\mathbf{N}$ and $\mathbf{f} = \mathbf{P}^{-1}\mathbf{b}$.

Definition 3.1.5.

Given an iterative method as Eq. (3.1.22) to solve the linear equation (3.1.20), the **residue** at iteration k is

$$\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}. \quad (3.1.26)$$

- Substitute Eq. (3.1.26) into Eq. (3.1.25), we have

$$\begin{aligned} \mathbf{P}\mathbf{x}^{(k+1)} &= \mathbf{N}\mathbf{x}^{(k)} + \mathbf{r}^{(k)} + \mathbf{A}\mathbf{x}^{(k)} \\ &= \mathbf{N}\mathbf{x}^{(k)} + \mathbf{r}^{(k)} + \mathbf{P}\mathbf{x}^{(k)} - \mathbf{N}\mathbf{x}^{(k)} \\ &= \mathbf{P}\mathbf{x}^{(k)} + \mathbf{r}^{(k)} \end{aligned}$$

Thus, we have

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{P}^{-1}\mathbf{r}^{(k)}. \quad (3.1.27)$$

Theorem 3.1.6.

Let $\mathbf{A} = \mathbf{P} - \mathbf{N}$, with \mathbf{A} and \mathbf{P} symmetric and positive definite. If the matrix $2\mathbf{P} - \mathbf{A}$ is positive definite, then the iterative method defined by (3.1.25) or (3.1.27) is convergent for any choice of initial guess $\mathbf{x}^{(0)}$ and

$$\rho(\mathbf{B}) = \|\mathbf{B}\|_{\mathbf{A}} = \|\mathbf{B}\|_{\mathbf{P}} < 1.$$

Moreover, the convergence of the iteration is monotone with respect to the norms $\|\cdot\|_{\mathbf{P}}$ and $\|\cdot\|_{\mathbf{A}}$.

Theorem 3.1.7.

Let $\mathbf{A} = \mathbf{P} - \mathbf{A}$ with \mathbf{A} being symmetric and positive definite. If the matrix $\mathbf{P} + \mathbf{P}^T - \mathbf{A}$ is positive definite, then \mathbf{P} is invertible, the iterative method defined by (3.1.25) or (3.1.27) is monotonically convergent with respect to norm $\|\cdot\|_{\mathbf{A}}$ and $\rho(\mathbf{B}) \leq \|\mathbf{B}\|_{\mathbf{A}} < 1$.

Iterative Methods, VI

- In Jacobi method, the matrix \mathbf{A} is splatted as

$$\mathbf{A} = \mathbf{D} - \mathbf{E} - \mathbf{F},$$

where \mathbf{D} is the diagonal matrix, \mathbf{E} is the strictly lower triangle matrix and \mathbf{F} is the strictly upper triangle matrix. And the iteration is

$$\mathbf{x}^{(k+1)} = \mathbf{D}^{-1} [\mathbf{b} + (\mathbf{E} + \mathbf{F})\mathbf{x}^{(k)}]$$

$$\mathbf{D}\mathbf{x}^{(k+1)} = \mathbf{b} + (\mathbf{E} + \mathbf{F})\mathbf{x}^{(k)}$$

Compared to Eq. (3.1.25), we have

$$\mathbf{P} = \mathbf{D}, \mathbf{N} = \mathbf{E} + \mathbf{F},$$

and the iteration matrix of Jacobi method is

$$\mathbf{B}_J = \mathbf{D}^{-1}(\mathbf{E} + \mathbf{F}) = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}. \quad (3.1.28)$$

- In the over-relaxation method,

$$\mathbf{x}^{(k+1)} = \omega \mathbf{D}^{-1} [\mathbf{b} + (\mathbf{E} + \mathbf{F})\mathbf{x}^{(k)}] + (1 - \omega)\mathbf{x}^{(k)},$$

and the iteration matrix is

$$\mathbf{B}_{J_\omega} = \omega \mathbf{B}_J + (1 - \omega)\mathbf{I}. \quad (3.1.29)$$

Iterative Methods, VII

- The Gauss-Seidel method has

$$\mathbf{x}^{(k+1)} = \mathbf{D}^{-1} \left[\mathbf{b} + \mathbf{E}\mathbf{x}^{(k+1)} + \mathbf{F}\mathbf{x}^{(k)} \right],$$

thus, the splitting of \mathbf{A} is $\mathbf{P} = \mathbf{D} - \mathbf{E}$, $\mathbf{N} = \mathbf{F}$, and the iteration matrix is

$$\mathbf{B}_{GS} = (\mathbf{D} - \mathbf{E})^{-1} \mathbf{F}. \quad (3.1.30)$$

- The successive over-relaxation method has

$$(\mathbf{I} - \omega \mathbf{D}^{-1} \mathbf{E}) \mathbf{x}^{(k+1)} = [(1 - \omega) \mathbf{I} + \omega \mathbf{D}^{-1} \mathbf{F}] \mathbf{x}^{(k)} + \omega \mathbf{D}^{-1} \mathbf{b}, \quad (3.1.31)$$

and the iteration matrix is

$$\mathbf{B}(\omega) = (\mathbf{I} - \omega \mathbf{D}^{-1} \mathbf{E})^{-1} [(1 - \omega) \mathbf{I} + \omega \mathbf{D}^{-1} \mathbf{F}]. \quad (3.1.32)$$

Iterative Methods, VIII

Theorem 3.1.8.

If \mathbf{A} is a strictly diagonally dominant matrix by rows, i.e., $|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|$, $i = 1, \dots, n$, then the Jacobi and Gauss-Seidel methods are convergent.

Theorem 3.1.9.

If \mathbf{A} and $2\mathbf{D} - \mathbf{A}$ are symmetric and positive definite matrices, then the Jacobi is convergent and $\rho(\mathbf{B}_J) = \|\mathbf{B}_J\|_{\mathbf{A}} = \|\mathbf{B}_J\|_{\mathbf{D}}$.

Theorem 3.1.10.

If \mathbf{A} is symmetric positive definite, then the JOR method is convergent if

$$0 < \omega < \frac{2}{\rho(\mathbf{D}^{-1} \mathbf{A})}. \quad (3.1.33)$$

Iterative Methods, IX

Theorem 3.1.11.

If \mathbf{A} is symmetric positive definite, the Gauss-Seidel method is monotonically convergent with respect to the norm $\|\cdot\|_{\mathbf{A}}$.

- If \mathbf{A} is tridiagonal (or block tridiagonal), then it can be shown that

$$\rho(\mathbf{B}_{GS}) = \rho^2(\mathbf{B}_J). \quad (3.1.34)$$

Thus, both methods converge or diverge at the same time. If they converge, then the Gauss-Seidel method converge faster than Jacobi method, and the asymptotic convergence rate of the Gauss-Seidel method is twice than that of Jacobi method.

Theorem 3.1.12.

If the Jacobi method is convergent, then the JOR method converges if $0 < \omega \leq 1$.

Iterative Methods, X

Theorem 3.1.13. (Ostrowski)

If \mathbf{A} is symmetric and positive definite, then the SOR method is convergent if and only if $0 < \omega < 2$. Moreover, its convergence is monotone with respect to $\|\cdot\|_{\mathbf{A}}$.

- If \mathbf{A} is strictly diagonally dominant by rows, then SOR method converges if $0 < \omega \leq 1$.

Theorem 3.1.14.

If the matrix \mathbf{A} enjoys the A-property and if \mathbf{B}_J has real eigenvalues, then the SOR method converges for any choice of $\mathbf{x}^{(0)}$ if and only if $\rho(\mathbf{B}_J) < 1$ and $0 < \omega < 2$. Moreover,

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho^2(\mathbf{B}_J)}} \quad (3.1.35)$$

and the corresponding asymptotic convergence factor is

$$\rho(\mathbf{B}(\omega_{opt})) = \frac{1 - \sqrt{1 - \rho^2(\mathbf{B}_J)}}{1 + \sqrt{1 - \rho^2(\mathbf{B}_J)}}. \quad (3.1.36)$$

Symmetric Gauss-Seidel Method

- In the Gauss-Seidel method, the values of the updated variables can be used immediately provided those variables are ordered first.
- The variables are updated only in one direction.
- The symmetric Gauss-Seidel method add another step in each iteration that updates each variable in backward order.
- Using the resistor network example,

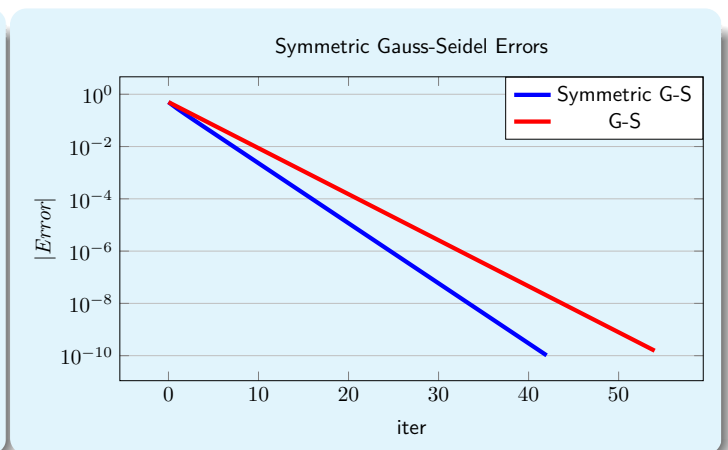
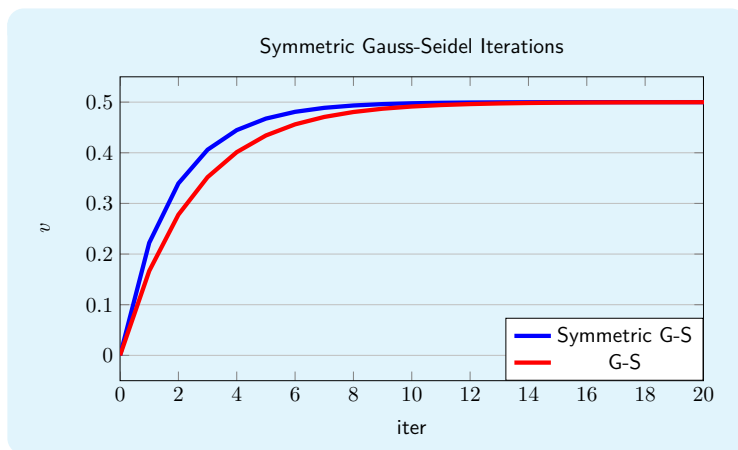
• Forward Gauss-Seidel

$$\begin{aligned}
 v_2^{(k+1/2)} &= 1/3(V + v_3^{(k)} + v_5^{(k)}) \\
 v_3^{(k+1/2)} &= 1/2(v_2^{(k+1/2)} + v_6^{(k)}) \\
 v_4^{(k+1/2)} &= 1/3(V + v_5^{(k)} + v_7^{(k)}) \\
 v_5^{(k+1/2)} &= 1/4(v_2^{(k+1/2)} + v_4^{(k+1/2)} + v_6^{(k)} + v_8^{(k)}) \\
 v_6^{(k+1/2)} &= 1/3(v_3^{(k+1/2)} + v_5^{(k+1/2)}) \\
 v_7^{(k+1/2)} &= 1/2(v_4^{(k+1/2)} + v_8^{(k)}) \\
 v_8^{(k+1/2)} &= 1/3(v_5^{(k+1/2)} + v_7^{(k+1/2)})
 \end{aligned}$$

• Backward Gauss-Seidel

$$\begin{aligned}
 v_8^{(k+1)} &= 1/3(v_5^{(k+1/2)} + v_7^{(k+1/2)}) \\
 v_7^{(k+1)} &= 1/2(v_4^{(k+1/2)} + v_8^{(k+1)}) \\
 v_6^{(k+1)} &= 1/3(v_3^{(k+1/2)} + v_5^{(k+1/2)}) \\
 v_5^{(k+1)} &= 1/4(v_2^{(k+1/2)} + v_4^{(k+1/2)} + v_6^{(k+1)} + v_8^{(k+1)}) \\
 v_4^{(k+1)} &= 1/3(V + v_5^{(k+1)} + v_7^{(k+1)}) \\
 v_3^{(k+1)} &= 1/2(v_2^{(k+1/2)} + v_6^{(k+1)}) \\
 v_2^{(k+1)} &= 1/3(V + v_3^{(k+1)} + v_5^{(k+1)})
 \end{aligned}$$

Symmetric Gauss-Seidel Method, II



- By combining forward and backward Gauss-Seidel method, symmetric Gauss-Seidel method improves the convergence rate.
- The symmetric Gauss-Seidel method appears to converge faster than the Gauss-Seidel method.
- But, each iteration of the symmetric Gauss-Seidel method needs more effort.

Symmetric Gauss-Seidel Method, III

- The forward Gauss-Seidel method is

$$\begin{aligned}\mathbf{D}\mathbf{x}^{(k+1/2)} &= \mathbf{b} + \mathbf{E}\mathbf{x}^{(k+1/2)} + \mathbf{F}\mathbf{x}^{(k)} \\ \mathbf{x}^{(k+1/2)} &= (\mathbf{D} - \mathbf{E})^{-1}(\mathbf{b} + \mathbf{F}\mathbf{x}^{(k)}).\end{aligned}\quad (3.1.37)$$

- The backward Gauss-Seidel method is then

$$\begin{aligned}\mathbf{D}\mathbf{x}^{(k+1)} &= \mathbf{b} + \mathbf{E}\mathbf{x}^{(k+1/2)} + \mathbf{F}\mathbf{x}^{(k+1)} \\ \mathbf{x}^{(k+1)} &= (\mathbf{D} - \mathbf{F})^{-1}(\mathbf{b} + \mathbf{E}\mathbf{x}^{(k+1/2)}).\end{aligned}\quad (3.1.38)$$

- Combining these two equations, we have

$$\mathbf{x}^{(k+1)} = (\mathbf{D} - \mathbf{F})^{-1}\mathbf{E}(\mathbf{D} - \mathbf{E})^{-1}\mathbf{F}\mathbf{x}^{(k)} + (\mathbf{D} - \mathbf{F})^{-1}[\mathbf{E}(\mathbf{D} - \mathbf{E})^{-1} + \mathbf{I}]\mathbf{b}. \quad (3.1.39)$$

Thus, the iterative matrix is

$$\mathbf{B}_{SGS} = (\mathbf{D} - \mathbf{F})^{-1}\mathbf{E}(\mathbf{D} - \mathbf{E})^{-1}\mathbf{F}. \quad (3.1.40)$$

And

$$\mathbf{f}_{SGS} = (\mathbf{D} - \mathbf{F})^{-1}[\mathbf{E}(\mathbf{D} - \mathbf{E})^{-1} + \mathbf{I}]\mathbf{b}. \quad (3.1.41)$$

And the preconditioning matrix is

$$\begin{aligned}\mathbf{P}_{SGS} &= \{(\mathbf{D} - \mathbf{F})^{-1}[\mathbf{E}(\mathbf{D} - \mathbf{E})^{-1} + \mathbf{I}]\}^{-1} \\ &= (\mathbf{D} - \mathbf{E})\mathbf{D}^{-1}(\mathbf{D} - \mathbf{F}).\end{aligned}\quad (3.1.42)$$

Symmetric Gauss-Seidel Method, IV

Theorem 3.1.15.

If \mathbf{A} is a symmetric positive definite matrix, the symmetric Gauss-Seidel method is convergent, and, moreover, \mathbf{B}_{SGS} is symmetric positive definite.

- Note that in Eq. (3.1.18), the forward SOR is

$$(\mathbf{D} - \omega\mathbf{E})\mathbf{x}^{(k+1)} = [\omega\mathbf{F} + (1 - \omega)\mathbf{D}]\mathbf{x}^{(k)} + \omega\mathbf{b}.$$

- The backward SOR is then

$$(\mathbf{D} - \omega\mathbf{F})\mathbf{x}^{(k+1)} = [\omega\mathbf{E} + (1 - \omega)\mathbf{D}]\mathbf{x}^{(k)} + \omega\mathbf{b}.$$

- Combine these two equations, the symmetric SOR (SSOR) method is

$$\mathbf{x}^{(k+1)} = \mathbf{B}_S(\omega)\mathbf{x}^{(k)} + \mathbf{b}_\omega,$$

where

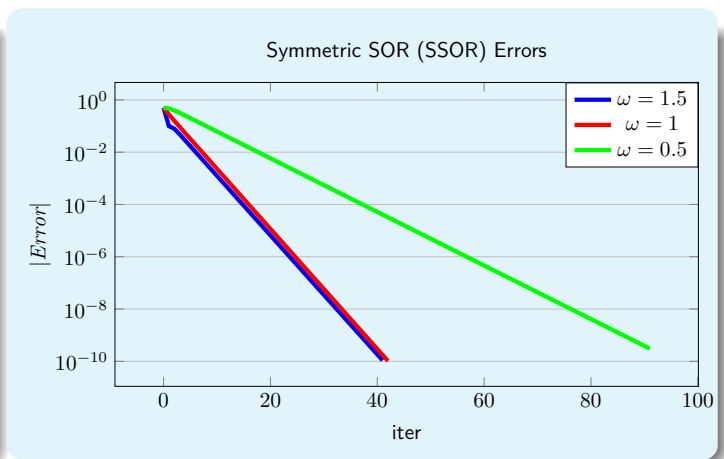
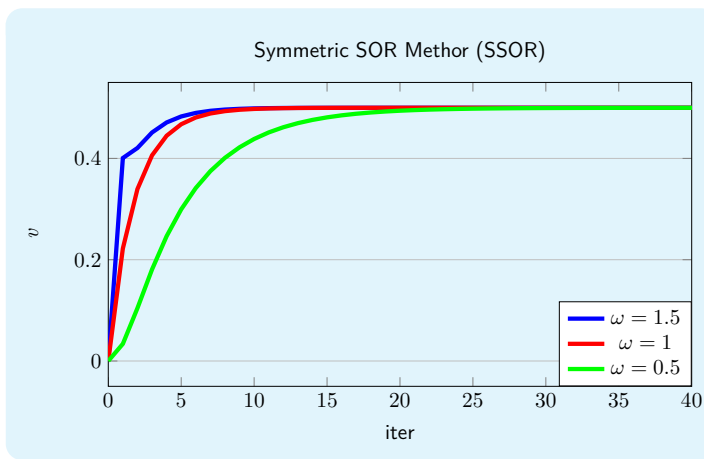
$$\begin{aligned}\mathbf{B}_S(\omega) &= (\mathbf{D} - \omega\mathbf{F})^{-1}(\omega\mathbf{E} + (1 - \omega)\mathbf{D})(\mathbf{D} - \omega\mathbf{E})^{-1}(\omega\mathbf{F} + (1 - \omega)\mathbf{D}), \\ \mathbf{b}_\omega &= \omega(2 - \omega)(\mathbf{D} - \omega\mathbf{F})^{-1}\mathbf{D}(\mathbf{D} - \omega\mathbf{E})^{-1}\mathbf{b}.\end{aligned}$$

Symmetric Gauss-Seidel Method, IV

- And the preconditioning matrix is

$$\mathbf{P}_{SSOR}(\omega) = \left(\frac{1}{\omega} \mathbf{D} - \mathbf{E} \right) \frac{\omega}{2 - \omega} \mathbf{D}^{-1} \left(\frac{1}{\omega} \mathbf{D} - \mathbf{F} \right). \quad (3.1.43)$$

- If \mathbf{A} is symmetric and positive definite, the SSOR method is convergent if $0 < \omega < 2$.
- Typically, the SSOR method with the optimal ω converges slower than the corresponding SOR method. But, $\rho(\mathbf{B}_S(\omega))$ is less sensitive to ω .
- Also, each iteration of SSOR method needs more efforts.



Summary

- Resistor network example
- Jacobi's method
- Over-relaxation method
- Gauss-Seidel Method
- Successive over-relaxation method
- Mathematical treatments
- Symmetric Gauss-Seidel method