

# Jacobi & Gauss-Seidel method

Created: Sep. 13. 2025.

Last Update: Sep. 14. 2025.

Tag: #Optimization/Distributed\_Optimization

## 0. Introduction

Jacobi method and Gauss-Seidel method are **classical iterative algorithms to determine the approximate solutions of a large-scale system of linear equations**. They are particularly useful when finding a direct solution by Gaussian elimination is impractical or computationally expensive. They computes the iterative solution from the initial guess until it converges to the exact solution within a desired tolerance.

The fundamental difference between them is the way they utilize information during the process. **The Jacobi method computes a new set of values for the solution vector using only the values from the previous iteration.** In contrast, **the Gauss-Seidel method immediately use the most recently computed values from the current iteration in subsequent calculations within that same step.**

While the Jacobi method is more convenient to formulate the problem in a parallel manner, the Gauss-Seidel method converges faster than the Jacobi method.

## 1. Problem formulation

### 🔗 Problem formulation

Solve  $Ax = b$

where  $A \in \mathbb{R}^{n \times n}$ ,  $x, b \in \mathbb{R}^n$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad \text{and} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

From a different point-of view, solving  $Ax = b$  is equivalent to minimizing  $\|Ax - b\|$ .

We can generalize this algorithm to the general optimization problem.

## 2. Algorithms

### 2.1. The Jacobi method

Separate  $A$  into 3 parts, i.e.  $A = D + L + U$  where  $D$  is diagonal,  $L$  is strictly lower triangular,  $U$  is strictly upper triangular parts of  $A$ .

Then, we have  $(D + L + U)x = b$  and hence

### 🔗 Jacobi Method

$$\begin{aligned}
Dx + (L + U)x &= b && \text{(given)} \\
Dx &= b - (L + U)x && \text{(move } (L + U)x\text{)} \\
x &= D^{-1}(b - (L + U)x) && \text{(left-multiply } D^{-1}\text{)}
\end{aligned}$$

So, from our initial guess  $x^{(0)}$ , we can update  $x$  using such an iterative algorithm.

$$x^{(k+1)} = D^{-1}(b - (L + U)x^{(k)})$$

The decomposed matrices are

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

and hence

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

Thus, we can rewrite the update equation in terms of elements like:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b - \sum_{j \neq i} a_{ij} x_j^{(k)} \right), \quad \forall i.$$

## 2.2. The Gauss-Seidel method

Same to the Jacobi method, separate  $A$  into 3 parts, i.e.  $A = D + L + U$  where  $D$  is diagonal,  $L$  is strictly lower triangular,  $U$  is strictly upper triangular parts of  $A$ .

Then, we have  $(D + L + U)x = b$  and hence

### Gauss-Seidel Method

$$\begin{aligned}
(D + L)x + Ux &= b && \text{(given)} \\
(D + L)x &= b - Ux && \text{(move } Ux\text{)} \\
x &= (D + L)^{-1}(b - Ux) && \text{(left-multiply } (D + L)^{-1}\text{)}
\end{aligned}$$

So, from our initial guess  $x^{(0)}$ , we can update  $x$  using such an iterative algorithm.

$$x^{(k+1)} = (D + L)^{-1}(b - Ux^{(k)})$$

The decomposed matrices are

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

and hence the element-based formula is

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

As we can see, both algorithms require **central factors** and hence not applicable to the fully decentralized optimization problems.

## 3. Convergence

### 3.1. Preliminaries

#### Def 3.1) Spectral radius

For a square matrix  $A \in \mathbb{C}^{n \times n}$ , the spectral radius of  $A$  is the maximum value of the absolute values of its eigenvalues and denoted by  $\rho(A)$ .

$$\rho(A) = \max\{|\lambda_1|, \dots, |\lambda_n|\}$$

Let  $A$  be a square matrix  $A \in \mathbb{C}^{n \times n}$  with entries  $a_{ij}$ . For  $i = 1, \dots, n$ , let  $R_i$  be the sum of the absolute values of the off-diagonal entries in the  $i^{\text{th}}$  row, i.e.  $R_i = \sum_{j \neq i} |a_{ij}|$ .

Then, a **closed disc centered at  $a_{ii}$  with radius  $R_i$  in the complex plane is called a Gershgorin disc**, which is denoted by  $D(a_{ii}, R_i) \subset \mathbb{C}$ .

Then, **every eigenvalue of  $A$  lies within at least one of the Gershgorin discs  $D(a_{ii}, R_i)$** .

In other words, **every eigenvalue of  $A$  is in the union of the Gershgorin discs, i.e.**

$$\lambda(A) \in \bigcup_i D(a_{ii}, R_i).$$

**proof.** Let  $\lambda$  be the eigenvalue of  $A$  and  $x$  be the eigenvector corresponds to  $\lambda$ , i.e.  $Ax = \lambda x$ .

Let  $x_{j^*}$  be the element whose magnitude is the largest among the elements of  $x$ , i.e.

$$|x_i| = \max_j |x_j|$$

Since  $x \neq 0$ ,  $|x_i| > 0$ .

Now, from the equation  $Ax = \lambda x$ , observe the  $i^{\text{th}}$  row only.

$$\lambda x_i = \sum_j a_{ij} \frac{x_j}{x_i}$$

and hence

$$\lambda - a_{ii} = \sum_{j \neq i} a_{ij} \frac{x_j}{x_i}$$

Take absolute value

$$|\lambda - a_{ii}| = \left| \sum_{j \neq i} a_{ij} \frac{x_j}{x_i} \right|$$

By triangle inequality,

$$\left| \sum_{j \neq i} a_{ij} \frac{x_j}{x_i} \right| = \sum_{j \neq i} \left| a_{ij} \frac{x_j}{x_i} \right|$$

and from the fact that  $|x_i| \geq |x_j|$ ,

$$\sum_{j \neq i} \left| a_{ij} \frac{x_j}{x_i} \right| = \sum_{j \neq i} |a_{ij}| \left| \frac{x_j}{x_i} \right| \leq \sum_{j \neq i} |a_{ij}| = R_i$$

Thus, we can conclude that  $|\lambda - a_{ii}| \leq R_i$  and hence, proved.  $\square$

### Cor 3.5) Nonsingularity of Strict Diagonally dominant matrix

A strictly diagonally dominant matrix (or an irreducibly diagonally dominant matrix is nonsingular.

## 3.2. Convergence Theorems

Both iterative methods are said to converge when  $\|x^{(k+1)} - x^{(k)}\| < \varepsilon$  for any  $\varepsilon > 0$ .

There are some conditions to guarantee the theoretical convergence of Jacobi method.

### Thm 3.6) Convergence for strictly diagonally dominant matrix

For a system of linear equations  $Ax = b$ , both the Jacobi and the Gauss-Seidel method converges to the unique solution if  $A$  is strictly diagonally dominant. (Sufficient condition)

**Proof for the Jacobi method** Since  $A$  is strictly diagonally dominant,  $A$  is nonsingular and hence invertible. This implies there exists an exact solution  $x^* = A^{-1}b$ .

From the element-based Jacobi method,

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right), \quad \forall i,$$

we can denote the exact solution by dropping the iteration index  $k$ .

$$x_i^* = \frac{1}{a_{ii}} \left( b_i - \sum_{j \neq i} a_{ij} x_j^* \right), \quad \forall i,$$

#### Claim: Error Converges to 0

In this setting, we can define an error at iteration  $k$  by taking an absolute value of the difference between the value at iteration  $k$  and the exact solution.

$$e_i^{(k+1)} = x_i^{(k+1)} - x_i^* = \frac{1}{a_{ii}} \sum_{j \neq i} a_{ij} (x_j^{(k)} - x_j^*) = \frac{1}{a_{ii}} \sum_{j \neq i} a_{ij} e_j^{(k)}$$

Taking absolute value and apply the triangle inequality.

$$|e_i^{(k+1)}| = \left| \frac{1}{a_{ii}} \sum_{j \neq i} a_{ij} e_j^{(k)} \right| \leq \left| \frac{1}{a_{ii}} \right| \sum_{j \neq i} |a_{ij} e_j^{(k)}| = \sum_{j \neq i} \left| \frac{a_{ij}}{a_{ii}} \right| |e_j^{(k)}|$$

Since  $A$  is strictly diagonally dominant,  $C_i = \sum_{j \neq i} \left| \frac{a_{ij}}{a_{ii}} \right| < 1$  and hence we have

$$|e_i^{(k+1)}| \leq C_i |e_i^{(k)}|$$

Then, by the definition of  $\ell_\infty$  norm,

$$\|e^{(k+1)}\|_\infty \leq C \|e^{(k)}\|_\infty \quad \text{where } C < 1$$

Thus, we can conclude that the error converges to zero.  $\square$

---

**Proof for the Gauss-Seidel method** The proof for the Gauss-Seidel method utilizes same idea to that of the Jacobi method.

Since  $A$  is strictly diagonally dominant,  $A$  is nonsingular and hence invertible. This implies there exists an exact solution  $x^* = A^{-1}b$ .

From the element-based Gauss-Seidel method,

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

we can denote the exact solution by dropping the iteration index  $k$ .

$$x_i^* = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j^* - \sum_{j=i+1}^n a_{ij} x_j^* \right), \quad \forall i,$$

### Claim: Error Converges to 0

In this setting, we can define an error at iteration  $k$  by taking a difference between the value at iteration  $k$  and the exact solution,  $e_i^{(k)} = x_i^{(k)} - x_i^*$ . Subtracting the second equation from the first yields the error propagation formula:

$$e_i^{(k+1)} = x_i^{(k+1)} - x_i^* = -\frac{1}{a_{ii}} \left( \sum_{j=1}^{i-1} a_{ij} e_j^{(k+1)} + \sum_{j=i+1}^n a_{ij} e_j^{(k)} \right)$$

Taking the absolute value and applying the triangle inequality, we get:

$$|e_i^{(k+1)}| \leq \frac{1}{|a_{ii}|} \left( \sum_{j=1}^{i-1} |a_{ij}| |e_j^{(k+1)}| + \sum_{j=i+1}^n |a_{ij}| |e_j^{(k)}| \right)$$

Let  $\|e^{(k+1)}\|_\infty = \max_i |e_i^{(k+1)}| = |e_p^{(k+1)}|$  for some index  $p$ . For this specific component  $p$ , the inequality becomes:

$$\|e^{(k+1)}\|_\infty \leq \frac{1}{|a_{pp}|} \left( \left( \sum_{j=1}^{p-1} |a_{pj}| \right) \|e^{(k+1)}\|_\infty + \left( \sum_{j=p+1}^n |a_{pj}| \right) \|e^{(k)}\|_\infty \right)$$

We can now solve for  $\|e^{(k+1)}\|_\infty$ :

$$\|e^{(k+1)}\|_{\infty} \left(1 - \frac{\sum_{j=1}^{p-1} |a_{pj}|}{|a_{pp}|}\right) \leq \left(\frac{\sum_{j=p+1}^n |a_{pj}|}{|a_{pp}|}\right) \|e^{(k)}\|_{\infty}$$

$$\|e^{(k+1)}\|_{\infty} \leq \left(\frac{\sum_{j=p+1}^n |a_{pj}|}{|a_{pp}| - \sum_{j=1}^{p-1} |a_{pj}|}\right) \|e^{(k)}\|_{\infty}$$

Since  $A$  is strictly diagonally dominant, we know that  $|a_{pp}| > \sum_{j \neq p} |a_{pj}| = \sum_{j=1}^{p-1} |a_{pj}| + \sum_{j=p+1}^n |a_{pj}|$ .

This implies  $|a_{pp}| - \sum_{j=1}^{p-1} |a_{pj}| > \sum_{j=p+1}^n |a_{pj}|$ . Let us define the constant as  $C_p = \frac{\sum_{j=p+1}^n |a_{pj}|}{|a_{pp}| - \sum_{j=1}^{p-1} |a_{pj}|}$ , it follows that  $C_p < 1$ .

Then, by the definition of  $l_{\infty}$  norm,

$$\|e^{(k+1)}\|_{\infty} \leq C \|e^{(k)}\|_{\infty} \quad \text{where } C = \max_p C_p < 1$$

Thus, we can conclude that the error converges to zero.  $\square$

### Thm 3.7) Convergence of general iterative method

An iterative method of the form  $x^{(k+1)} = Tx^{(k)} + c$  converges to the unique solution of  $x = Tx + c$  for any initial guess vector  $x^{(0)}$  if and only if the spectral radius  $\rho(T)$  of the iteration matrix  $T$  is less than 1.

$$\rho(T) < 1$$

|

In this setting,  $T_J = D^{-1}(L + U)$  and  $T_G = (D + L)^{-1}U$  for the Jacobi and Gauss-Seidel methods, respectively.

**Proof.** The error vector  $e^{(k)} = x^{(k)} - x^*$  follows the recurrence  $e^{(k+1)} = Te^{(k)}$ , which implies  $e^{(k)} = T^k e^{(0)}$ . The method converges if and only if  $\lim_{k \rightarrow \infty} e^{(k)} = 0$  for any initial error  $e^{(0)}$ . This is equivalent to the condition that  $\lim_{k \rightarrow \infty} T^k = 0$  (the zero matrix).

**(Necessity:**  $\lim_{k \rightarrow \infty} T^k = 0 \implies \rho(T) < 1$ )

Let  $\lambda$  be any eigenvalue of  $T$  with a corresponding eigenvector  $v \neq 0$ . By definition,  $Tv = \lambda v$ .

Applying  $T$  repeatedly, we get  $T^k v = \lambda^k v$ . Taking the limit as  $k \rightarrow \infty$ , and using our assumption that  $T^k \rightarrow 0$ :

$$\lim_{k \rightarrow \infty} (T^k v) = \left(\lim_{k \rightarrow \infty} T^k\right) v = 0 \cdot v = 0$$

This means we must have  $\lim_{k \rightarrow \infty} (\lambda^k v) = 0$ . Since  $v$  is (the zero matrix).

Since  $v$  is a non-zero vector, the scalar sequence  $\lambda^k$  must converge to 0. This is only possible if  $|\lambda| < 1$ . Since this must hold for every eigenvalue of  $T$ , it must hold for the one with the largest magnitude. Thus,  $\rho(T) < 1$ .

**(Sufficiency:**  $\rho(T) < 1 \implies \lim_{k \rightarrow \infty} T^k = 0$ )

This direction of the proof relies on a lemma of the relationship between norm and spectral radius.

By our assumption,  $\rho(T) < 1$ . We can choose an  $\epsilon > 0$  that is small enough such that  $\rho(T) + \epsilon < 1$ . For instance, we can choose  $\epsilon = (1 - \rho(T))/2$ .

According to the lemma, there exists a matrix norm for which  $\|T\| \leq \rho(T) + \epsilon = C$ , where  $C$  is a constant less than 1.

Using the submultiplicative property of matrix norms  $\|A^k\| \leq \|A\|^k$ , we have:

$$\|T^k\| \leq \|T\|^k \leq C^k$$

As  $k \rightarrow \infty$ , since  $C < 1$ , we have  $C^k \rightarrow 0$ .

This implies that  $\lim_{k \rightarrow \infty} \|T^k\| = 0$ . If the norm of a matrix converges to zero, the matrix itself must converge to the zero matrix. Therefore,  $\lim_{k \rightarrow \infty} T^k = 0$ .  $\square$

---

### Thm 3.8) Convergence of the Gauss-Seidel method

If  $A^\top = A \succeq 0$ , then the Gauss-Seidel iterates converge to  $x^* = A^{-1}b$  for any initial guess  $x^{(0)}$ .

---

### Thm 3.9) Stein–Rosenberg theorem (Convergence rate)

Let  $A = (a_{ij}) \in \mathbb{R}^{n \times n}$  and let  $\rho(T)$  be the spectral radius of a matrix  $T$ . Let  $T_J, T_G$  be the matrix splitting for the Jacobi method and the Gauss-Seidel method, respectively.

If  $a_{ij} \leq 0$ , for  $i \neq j$  and  $a_{ii} > 0$ , for  $i = 1, 2, \dots, n$ , then one and only one of the following statements holds (four statements are mutually exclusive):

- i)  $0 \leq \rho(T_G) < \rho(T_J) < 1$
  - ii)  $1 < \rho(T_J) < \rho(T_G)$
  - iii)  $\rho(T_J) = \rho(T_G) = 0$
  - iv)  $\rho(T_J) = \rho(T_G) = 1$
- 

## 4. Variants

### 4.1. Method of Successive over-relaxation (SOR)

The method of successive over-relaxation is a variant of the Gauss-Seidel method for solving a system of linear equations for faster convergence by introducing a relaxation factor.

#### 4.1.1. Algorithm

As with the Gauss-Seidal method, separate  $A$  into 3 parts, i.e.  $A = D + L + U$  where  $D$  is diagonal,  $L$  is strictly lower triangular,  $U$  is strictly upper triangular parts of  $A$ .

Then, we have  $(D + L + U)x = b$  and multiply the relaxation factor  $\omega > 1$  both sides.

$$\omega \cdot (D + L + U)x = wb$$

By moving terms properly, we have

 SOR

$$(D + \omega L)x = \omega b - [\omega U + (\omega - 1)D]x$$

$$x = (D + \omega L)^{-1}(\omega b - [\omega U + (\omega - 1)D]x)$$

Thus, the iterative update can be expressed by:

$$x^{(k+1)} = (D + \omega L)^{-1}(\omega b - [\omega U + (\omega - 1)D]x^{(k)})$$

and its element-based expression is

$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \frac{\omega}{a_{ii}} \left( b_i - \sum_{j < i} a_{ij}x_j^{(k+1)} - \sum_{j > i} a_{ij}x_j^{(k)} \right), \quad i = 1, 2, \dots, n.$$

and it can be also directly expressed by

$$x^{(k+1)} = (1 - \omega)x^{(k)} + \omega D^{-1}(b - Lx^{(k+1)} - Ux^{(k)}).$$

this expression is more convenient because it doesn't require the computation of  $(D + \omega L)^{-1}$ .

If  $A$  is symmetric, i.e.  $L = U^T$ , such a method is called the Symmetric Successive over-relaxation (SSOR).

#### 4.1.2. Convergence

##### Thm) Kahan

If  $a_{ii} \neq 0$  for  $i = 1, \dots, n$ , then the SOR iteration matrix  $T_\omega$  satisfies

$$\rho(T_\omega) \geq |\omega - 1|$$

Consequently, the SOR iterates converge for every  $x^{(0)}$  only if  $0 < \omega < 2$ .

##### Thm) Ostrowski–Reich

If  $A^T = A \succeq 0$  and  $0 < \omega < 2$ , then the SOR iterates converge to  $A^{-1}b$  for every  $x^{(0)}$ .

##### Thm) Determining the relaxation factor

If  $A$  is symmetric positive-definite and tridiagonal, then

$$\rho(T_G) = \rho(T_J)^2 < 1,$$

and the  $\omega$  that minimizes  $\rho(T_\omega)$  is

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

For this  $\omega$ ,  $\rho(T_\omega) = \omega - 1$ .

## 5. Applications

## 5.1. 2D Laplace's equation for steady-state analysis

The problem of calculating the steady-state temperature distribution or electrostatic potential on a 2D plane is described by Laplace's equation:

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

Approximating this with the finite difference method, the second partial derivative term at each grid point  $(i, j)$  is expressed as

$$\begin{aligned}\frac{\partial^2 u}{\partial x^2} &\approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \\ \frac{\partial^2 u}{\partial y^2} &\approx \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2}\end{aligned}$$

and thus the entire equation is transformed into a simple algebraic equation by substituting them into the original PDE.

$$4u_{i,j} - u_{i-1,j} - u_{i+1,j} - u_{i,j-1} - u_{i,j+1} = 0$$

By setting up this equation for every interior grid point, a large, diagonally dominant matrix is formed where the diagonal element of each row is 4 and the off-diagonal elements are -1 or 0. It can be solved using the Gauss-Seidel method.

$$u_{i,j}^{(k+1)} = \frac{1}{4} \left( u_{i-1,j}^{(k+1)} + u_{i,j-1}^{(k+1)} + u_{i+1,j}^{(k)} + u_{i,j-1}^{(k)} \right)$$

## 5.2. Time-dependent 1D Heat Equation

1D heat equation which describes time-dependent phenomena is

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

Using an implicit method for a stable solution means the time derivative is approximated as

$$\frac{\partial u}{\partial t} \approx \frac{u_i^{n+1} - u_i^n}{\Delta t}$$

and the spatial derivative is approximated using the values at the next time step ( $n + 1$ ) as

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{h^2}$$

By substituting them into the original equation and rearrange it for the unknown  $u_i^{n+1}$  at the next time step yields a tridiagonal matrix system, we have

$$-ru_{i-1}^{n+1} + (1 + 2r)u_i^{n+1} - ru_{i+1}^{n+1} = u_i^n$$

where  $r = \alpha\Delta t/h^2$ . It can also be solved at each time step using the Gauss-Seidel method.