

**Computational Hydraulics**  
**Professor Anirban Dhar**  
**Department of Civil Engineering**  
**Indian Institute of Technology Kharagpur**  
**Lecture 29**  
**Algebraic Equation: Gauss - Seidel Method**

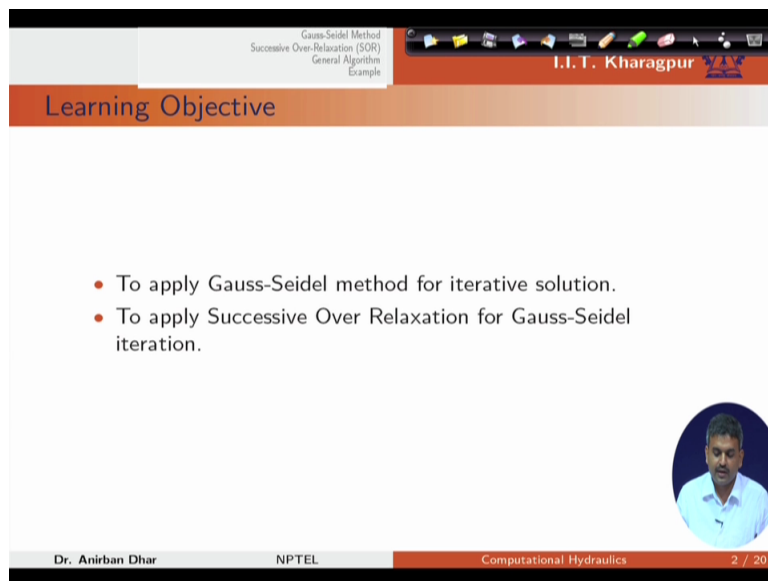
Welcome to this lecture number 29 of the course computational hydraulics. We are in module number 2, numerical methods. And in this particular lecture class I will be discussing algebraic equation, Gauss-Seidel method.

(Refer Slide Time 00:32)

The image shows a presentation slide with a white background and a dark blue header. The header contains the text "Gauss-Seidel Method", "Successive Over-Relaxation (SOR)", "General Algorithm", and "Example" on the left, and "I.I.T. Kharagpur" with a logo on the right. The main content area features a dark blue rounded rectangle with the text "Module 02: Numerical Methods" and "Unit 25: Algebraic Equation: Gauss-Seidel Method". Below this, the name "Anirban Dhar" is centered, followed by "Department of Civil Engineering" and "Indian Institute of Technology Kharagpur, Kharagpur". At the bottom, it says "National Programme for Technology Enhanced Learning (NPTEL)". The footer of the slide includes "Dr. Anirban Dhar", "NPTEL", "Computational Hydraulics", and "1 / 20".

This is again another iterative method like our Jacobian method discussed in unit number 24. Now what is the learning objective? At the end of this unit students will be able to apply Gauss-Seidel method for iterative solution. And they will be able to apply successive over relaxation for Gauss-Seidel iteration.

(Refer Slide Time 01:04)



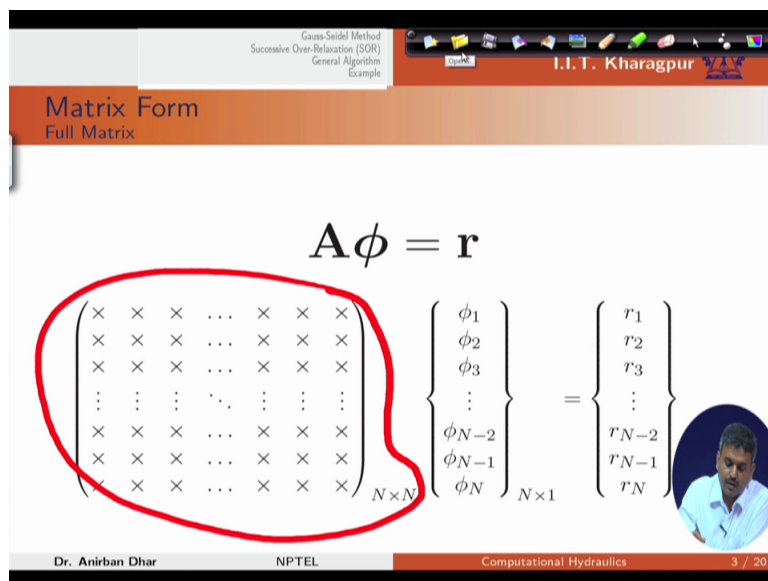
Slide 2: Learning Objective

- To apply Gauss-Seidel method for iterative solution.
- To apply Successive Over Relaxation for Gauss-Seidel iteration.

Dr. Anirban Dhar NPTEL Computational Hydraulics 2 / 20

In general like our Jacobian method we have our general structure. We are considering the full matrix here. We are not dividing it, or we are not storing it in reduced form.

(Refer Slide Time 01:28)



Slide 3: Matrix Form Full Matrix

$$\mathbf{A}\phi = \mathbf{r}$$
$$\begin{pmatrix} \times & \times & \times & \dots & \times & \times & \times \\ \times & \times & \times & \dots & \times & \times & \times \\ \times & \times & \times & \dots & \times & \times & \times \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \times & \times & \times & \dots & \times & \times & \times \\ \times & \times & \times & \dots & \times & \times & \times \\ \times & \times & \times & \dots & \times & \times & \times \end{pmatrix}_{N \times N} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \\ \phi_{N-2} \\ \phi_{N-1} \\ \phi_N \end{pmatrix}_{N \times 1} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_{N-2} \\ r_{N-1} \\ r_N \end{pmatrix}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 3 / 20

So let us start with the basic steps. Basic steps, this is similar to or first steps are exactly similar to our Jacobian method. We will divide our A matrix into lower, diagonal, upper. This lower one is strictly lower, diagonal is only diagonal terms and upper is with strictly upper matrix.

(Refer Slide Time 02:03)

Gauss-Seidel Method  
Successive Over-Relaxation (SOR)  
General Algorithm  
Example

I.I.T. Kharagpur


### Basic Steps

Gauss-Seidel Method

The coefficient matrix **A** can be written as

$$\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$$

where **L**, **D**, **U** are strictly lower triangular, diagonal, strictly upper triangular matrices respectively.



Dr. Anirban Dhar      NPTEL      Computational Hydraulics      4 / 20

So now if we consider the overall calculation, again LDU this again can be multiplied because this whole thing is A.

(Refer Slide Time 02:16)

Gauss-Seidel Method  
Successive Over-Relaxation (SOR)  
General Algorithm  
Example

I.I.T. Kharagpur

### Basic Steps



Gauss-Seidel Method

The coefficient matrix **A** can be written as

$$\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$$

where **L**, **D**, **U** are strictly lower triangular, diagonal, strictly upper triangular matrices respectively.

Overall calculation can be presented as

$$(\mathbf{L} + \mathbf{D} + \mathbf{U})\phi = r$$


Dr. Anirban Dhar      NPTEL      Computational Hydraulics      4 / 20

But in this process we can divide this into two parts. In case of our Jacobi iteration we have seen that this is D into phi p plus L plus U into phi p minus 1, this r. This was there.

(Refer Slide Time 02:54)

Gauss-Seidel Method  
Successive Over-Relaxation (SOR)  
General Algorithm  
Example

I.I.T. Kharagpur

### Basic Steps

Gauss-Seidel Method

The coefficient matrix  $\mathbf{A}$  can be written as

$$\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$$

where  $\mathbf{L}$ ,  $\mathbf{D}$ ,  $\mathbf{U}$  are strictly lower triangular, diagonal, strictly upper triangular matrices respectively.

Overall calculation can be presented as

$$(\mathbf{L} + \mathbf{D} + \mathbf{U})\phi = \mathbf{r}$$

Iterative form can be written as

$$(\mathbf{L} + \mathbf{D})\phi^{(p)} + \mathbf{U}\phi^{(p-1)} = \mathbf{r}$$

$\mathbf{D}\phi^{(p)} + (\mathbf{L} + \mathbf{U})\phi^{(p-1)} = \mathbf{r}$

Dr. Anirban Dhar      NPTEL      Computational Hydraulics      4 / 20

But in this case we are considering the lower diagonal or lower triangular matrix here. And we are updating it. So  $\mathbf{L}$  plus  $\mathbf{D}$  and  $\mathbf{U}$  is there. So  $\phi$  again is the available value. And in this process we can write the final form,  $\mathbf{L}$  plus  $\mathbf{D}$  this inverse into  $\mathbf{U}\phi^{p-1}$ , this term is there. Like our Jacobi iteration. But the difference is the inclusion of  $\mathbf{L}$  term which is the lower part or the updated part of the matrix. Now this will be clear with example.

(Refer Slide Time 03:51)

Gauss-Seidel Method  
Successive Over-Relaxation (SOR)  
General Algorithm  
Example

I.I.T. Kharagpur

### Basic Steps

Gauss-Seidel Method

Iterative form can be written as

$$(\mathbf{L} + \mathbf{D})\phi^{(p)} + \mathbf{U}\phi^{(p-1)} = \mathbf{r}$$

$$(\mathbf{L} + \mathbf{D})\phi^{(p)} = -\mathbf{U}\phi^{(p-1)} + \mathbf{r}$$

Final form can be written as

$$\phi^{(p)} = -(\mathbf{L} + \mathbf{D})^{-1}\mathbf{U}\phi^{(p-1)} + (\mathbf{L} + \mathbf{D})^{-1}\mathbf{r}$$

where  $p$  is the iteration counter ( $p \geq 1$ ).

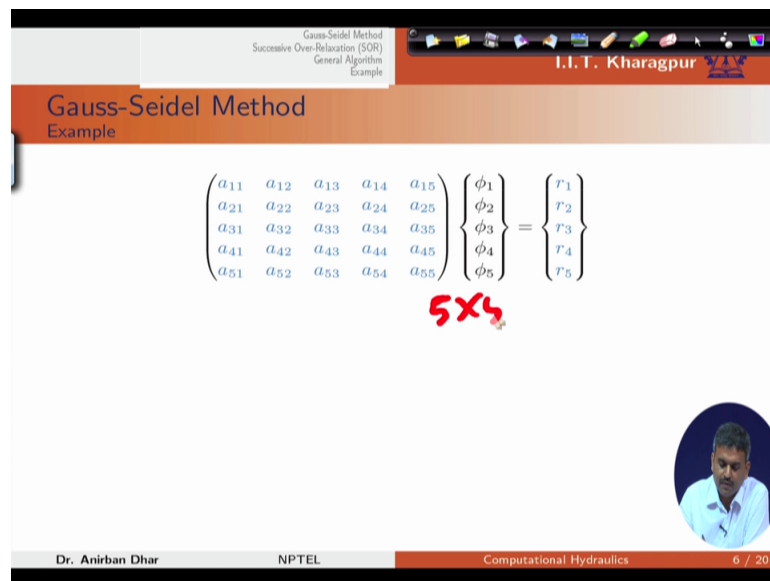
Iteration starts with a guess value  $\phi^{(0)}$

Dr. Anirban Dhar      NPTEL      Computational Hydraulics      5 / 20

So in Gauss Seidel we let us say that we will apply the same example with 5 by 5 matrix.



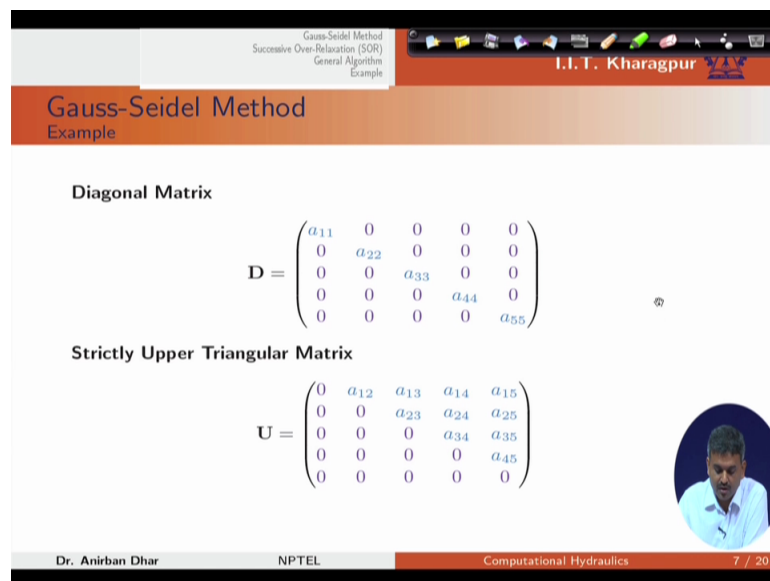
(Refer Slide Time 04:07)



Slide 6: Gauss-Seidel Method Example. The slide shows a 5x5 matrix equation: 
$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{pmatrix}$$
 A red handwritten '5x5' is written below the matrix. The slide also includes a small video inset of the speaker and a navigation bar at the top with the text 'Gauss-Seidel Method', 'Successive Over-Relaxation (SOR)', 'General Algorithm', 'Example', and 'I.I.T. Kharagpur'.

Now in this case again we can write our coefficient as lower, strictly lower, diagonal, strictly upper diagonal. Up to this we are not disturbing any element within the A matrix. And we will not disturb.

(Refer Slide Time 04:29)



Slide 7: Gauss-Seidel Method Example. The slide shows the decomposition of the matrix A into a Diagonal Matrix D and a Strictly Upper Triangular Matrix U. The Diagonal Matrix D is defined as: 
$$D = \begin{pmatrix} a_{11} & 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 & 0 \\ 0 & 0 & a_{33} & 0 & 0 \\ 0 & 0 & 0 & a_{44} & 0 \\ 0 & 0 & 0 & 0 & a_{55} \end{pmatrix}$$
 The Strictly Upper Triangular Matrix U is defined as: 
$$U = \begin{pmatrix} 0 & a_{12} & a_{13} & a_{14} & a_{15} \\ 0 & 0 & a_{23} & a_{24} & a_{25} \\ 0 & 0 & 0 & a_{34} & a_{35} \\ 0 & 0 & 0 & 0 & a_{45} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$
 The slide also includes a small video inset of the speaker and a navigation bar at the top with the text 'Gauss-Seidel Method', 'Successive Over-Relaxation (SOR)', 'General Algorithm', 'Example', and 'I.I.T. Kharagpur'.

Iteration starts with guess values. Now in iteration 1, this is clear that for row 1, whatever available value starting from 2 to 5 is available from previous iteration. So except this diagonal term we are calculating this updated value based on previous time level value.

(Refer Slide Time 05:06)

Gauss-Seidel Method  
Successive Over-Relaxation (SOR)  
General Algorithm  
Example

I.I.T. Kharagpur

### Gauss-Seidel Method

Example

Iteration starts with the guess value  $\phi^{(0)}$

$$\phi^{(0)} = [\phi_1^{(0)} \quad \phi_2^{(0)} \quad \phi_3^{(0)} \quad \phi_4^{(0)} \quad \phi_5^{(0)}]^T$$

**Iteration 1:**  
Row 1:

$$\phi_1^{(1)} = \frac{1}{a_{11}} \left[ r_1 - \sum_{j=2}^5 a_{1j} \phi_j^{(0)} \right]$$

Dr. Anirban Dhar    NPTEL    Computational Hydraulics    8 / 20

But interesting feature is there for row 2. Now we have this phi2 calculation and with phi2 calculation we can omit this 2 term, but we have already one updated value available for phi 1 from row 1. So this comes directly from the iteration 1 or present iteration level. So r2 minus a21. So a21 is the coefficient of our lower triangular matrix.

(Refer Slide Time 06:00)

Gauss-Seidel Method  
Successive Over-Relaxation (SOR)  
General Algorithm  
Example

I.I.T. Kharagpur

### Gauss-Seidel Method

Example

Iteration starts with the guess value  $\phi^{(0)}$

$$\phi^{(0)} = [\phi_1^{(0)} \quad \phi_2^{(0)} \quad \phi_3^{(0)} \quad \phi_4^{(0)} \quad \phi_5^{(0)}]^T$$

**Iteration 1:**  
Row 1:

$$\phi_1^{(1)} = \frac{1}{a_{11}} \left[ r_1 - \sum_{j=2}^5 a_{1j} \phi_j^{(0)} \right]$$

Row 2:

$$\phi_2^{(1)} = \frac{1}{a_{22}} \left[ r_2 - a_{21} \phi_1^{(1)} - \sum_{j=3}^5 a_{2j} \phi_j^{(0)} \right]$$

$L = \begin{bmatrix} & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{bmatrix}$

Dr. Anirban Dhar    NPTEL    Computational Hydraulics    8 / 20

It's clear, that's why we are including the lower triangular matrix and you can see that we are writing it with present time level not from the previous time level. That's why L plus D into phi p was written in our calculation process.

(Refer Slide Time 06:26)

Gauss-Seidel Method  
Successive Over-Relaxation (SOR)  
General Algorithm  
Example

I.I.T. Kharagpur

### Gauss-Seidel Method

Example

Iteration starts with the guess value  $\phi^{(0)}$

$$\phi^{(0)} = [\phi_1^{(0)} \quad \phi_2^{(0)} \quad \phi_3^{(0)} \quad \phi_4^{(0)} \quad \phi_5^{(0)}]^T$$

**Iteration 1:**

Row 1:

$$\phi_1^{(1)} = \frac{1}{a_{11}} \left[ r_1 - \sum_{j=2}^5 a_{1j} \phi_j^{(0)} \right]$$

Row 2:

$$\phi_2^{(1)} = \frac{1}{a_{22}} \left[ r_2 - a_{21} \phi_1^{(1)} - \sum_{j=3}^5 a_{2j} \phi_j^{(0)} \right]$$

*(L+D)phi<sup>n</sup>*

Dr. Anirban Dhar      NPTEL      Computational Hydraulics      8 / 20

Now if we write the third one, again we have two values available for updating phi3. This is j 1 and j 2. And starting from j 4 to 5 we are utilizing previous time level value. Interesting part is that phi1 we have changed color, phi j we have changed color, because we are getting updated values and we are utilizing the updated values in the calculation. So red colored indicates that we are using updated values within the calculation process.

(Refer Slide Time 07:11)

Gauss-Seidel Method  
Successive Over-Relaxation (SOR)  
General Algorithm  
Example

I.I.T. Kharagpur

### Gauss-Seidel Method

Example

Iteration starts with the guess value  $\phi^{(0)}$

$$\phi^{(0)} = [\phi_1^{(0)} \quad \phi_2^{(0)} \quad \phi_3^{(0)} \quad \phi_4^{(0)} \quad \phi_5^{(0)}]^T$$

**Iteration 1:**

Row 1:

$$\phi_1^{(1)} = \frac{1}{a_{11}} \left[ r_1 - \sum_{j=2}^5 a_{1j} \phi_j^{(0)} \right]$$

Row 2:

$$\phi_2^{(1)} = \frac{1}{a_{22}} \left[ r_2 - a_{21} \phi_1^{(1)} - \sum_{j=3}^5 a_{2j} \phi_j^{(0)} \right]$$

Row 3:

$$\phi_3^{(1)} = \frac{1}{a_{33}} \left[ r_3 - \sum_{j=1}^2 a_{3j} \phi_j^{(1)} - \sum_{j=4}^5 a_{3j} \phi_j^{(0)} \right]$$

Dr. Anirban Dhar      NPTEL      Computational Hydraulics      8 / 20

But the blue values we are not changing the coefficients on the right hand side vector. Gauss Seidel example again fourth, fifth. In the fifth one except this fifth one all values are updated ones. So starting from 1 to 4 we have all values updated ones. In case of fourth one, except one, all are updated ones.

(Refer Slide Time 07:52)

Gauss-Seidel Method  
Successive Over-Relaxation (SOR)  
General Algorithm  
Example

I.I.T. Kharagpur

### Gauss-Seidel Method

Example

Row 4:


$$\phi_4^{(1)} = \frac{1}{a_{44}} \left[ r_4 - \sum_{j=1}^3 a_{4j} \phi_j^{(1)} - a_{45} \phi_5^{(0)} \right]$$

Row 5:

$$\phi_5^{(1)} = \frac{1}{a_{55}} \left[ r_5 - \sum_{j=1}^4 a_{5j} \phi_j^{(1)} \right]$$

**Iteration 2:**

Row 1:

$$\phi_1^{(2)} = \frac{1}{a_{11}} \left[ r_1 - \sum_{j=2}^5 a_{1j} \phi_j^{(1)} \right]$$


Dr. Anirban Dhar      NPTEL      Computational Hydraulics      9 / 20

So in iteration 2 again if we consider the same thing you will find that this is same. We are using the present available value at the present iteration level itself.

(Refer Slide Time 08:11)

Gauss-Seidel Method  
Successive Over-Relaxation (SOR)  
General Algorithm  
Example

I.I.T. Kharagpur


### Gauss-Seidel Method

Example

Row 2:

$$\phi_2^{(2)} = \frac{1}{a_{22}} \left[ r_2 - a_{21} \phi_1^{(2)} - \sum_{j=3}^5 a_{2j} \phi_j^{(1)} \right]$$

Row 3:

$$\phi_3^{(2)} = \frac{1}{a_{33}} \left[ r_3 - \sum_{j=1}^2 a_{3j} \phi_j^{(2)} - \sum_{j=4}^5 a_{3j} \phi_j^{(1)} \right]$$


Dr. Anirban Dhar      NPTEL      Computational Hydraulics      10 / 20

So anyway we are providing updated information to the algorithm. So that's why obviously we expect that the performance will be better compared to our Jacobi technique where we are utilizing the previous iteration level value or guess value at the beginning steps. So row 3, row 5 again this approach is similar.

(Refer Slide Time 08:48)

Gauss-Seidel Method  
Successive Over-Relaxation (SOR)  
General Algorithm  
Example

I.I.T. Kharagpur

**Gauss-Seidel Method**  
Example

Row 2:  

$$\phi_2^{(2)} = \frac{1}{a_{22}} \left[ r_2 - a_{21}\phi_1^{(2)} - \sum_{j=3}^5 a_{2j}\phi_j^{(1)} \right]$$

Row 3:  

$$\phi_3^{(2)} = \frac{1}{a_{33}} \left[ r_3 - \sum_{j=1}^2 a_{3j}\phi_j^{(2)} - \sum_{j=4}^5 a_{3j}\phi_j^{(1)} \right]$$

Row 4:  

$$\phi_4^{(2)} = \frac{1}{a_{44}} \left[ r_4 - \sum_{j=1}^3 a_{4j}\phi_j^{(2)} - a_{45}\phi_5^{(1)} \right]$$

Row 5:  

$$\phi_5^{(2)} = \frac{1}{a_{55}} \left[ r_5 - \sum_{j=1}^4 a_{5j}\phi_j^{(2)} \right]$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 10 / 20

Now if we generalize the algorithm this becomes like this. Except that  $i$ th term that means up to  $i$  minus 1 we have already calculated the updated value. So  $j$  starting from  $i$  minus 1, we are using updated value. And from  $i$  plus 1 to  $N$  we are using previous time level value.

(Refer Slide Time 09:16)

Gauss-Seidel Method  
Successive Over-Relaxation (SOR)  
General Algorithm  
Example

I.I.T. Kharagpur

**Gauss-Seidel Method**  
General Algorithm

Iteration starts with the guess value  $\phi^{(0)}$

$$\phi^{(0)} = [\phi_1^{(0)} \quad \phi_2^{(0)} \quad \dots \quad \phi_{N-1}^{(0)} \quad \phi_N^{(0)}]^T$$

$$\phi_i^{(p)} = \frac{1}{a_{ii}} \left[ r_i - \sum_{j=1}^{i-1} a_{ij}\phi_j^{(p)} - \sum_{j=i+1}^N a_{ij}\phi_j^{(p-1)} \right], \quad \forall i \in \{1, \dots, N\}, p \geq 1$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 11 / 20

Now like our previous Jacobi method we can add  $\phi_i^{p-1}$ . So we can just add  $\phi_i^{p-1}$ . And we can write this as residual again. So whatever value is available either in updated form or in non-updated form, we can calculate the right hand side minus left hand side. So this is our residual and divided by coefficient of the diagonal term. This is  $a_{ii}$ .

(Refer Slide Time 10:01)

Gauss-Seidel Method  
General Algorithm

Iteration starts with the guess value  $\phi^{(0)}$

$$\phi^{(0)} = [\phi_1^{(0)} \quad \phi_2^{(0)} \quad \dots \quad \phi_{N-1}^{(0)} \quad \phi_N^{(0)}]^T$$

$$\phi_i^{(p)} = \frac{1}{a_{ii}} \left[ r_i - \sum_{j=1}^{i-1} a_{ij} \phi_j^{(p)} - \sum_{j=i+1}^N a_{ij} \phi_j^{(p-1)} \right], \quad \forall i \in \{1, \dots, N\}, p \geq 1$$

By adding and subtracting  $\phi_i^{(p-1)}$  in right hand side

$$\phi_i^{(p)} = \phi_i^{(p-1)} + \frac{1}{a_{ii}} \left[ r_i - \sum_{j=1}^{i-1} a_{ij} \phi_j^{(p)} - a_{ii} \phi_i^{(p-1)} - \sum_{j=i+1}^N a_{ij} \phi_j^{(p-1)} \right], \quad \forall i, p \geq 1$$

In compact form

$$\phi_i^{(p)} = \phi_i^{(p-1)} + \frac{Res_i}{a_{ii}} \quad \forall i, p \geq 1$$

*RHS - LHS*

Dr. Anirban Dhar NPTEL Computational Hydraulics 11 / 20

So we can use this concept where we are updating the variable values at the current time steps. So obviously convergence will be better. Like our previous Jacobi iteration we can utilize residual error for this one. Maximum absolute error or RMSE for this purpose where epsilon max, these are actually allowable epsilon values.

(Refer Slide Time 10:47)

Gauss-Seidel Method  
Stopping Criterion

Residual Error in a particular iteration can be expressed as

$$e^{(p)} = \mathbf{A}\phi^{(p)} - \mathbf{r}$$

Maximum Absolute Error:

$$\max_{i \in \{1, \dots, N\}} |e_i^{(p)}| \leq \epsilon_{max}$$

Root Mean Square Error:

$$\sqrt{\frac{1}{N} \sum_{i=1}^N (e_i^{(p)})^2} \leq \epsilon_{max}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 12 / 20

Now in this case we need again this diagonal dominance. Without diagonal dominance these convergence are not possible.

(Refer Slide Time 11:04)

Gauss-Seidel Method  
Successive Over-Relaxation (SOR)  
General Algorithm  
Example

I.I.T. Kharagpur

### Convergence Criteria

#### Diagonal Dominance

**Diagonal Dominance:**

$$|a_{ii}| = \sum_{\substack{j=1 \\ j \neq i}}^N |a_{ij}|$$
$$\exists l : |a_{ll}| > \sum_{\substack{j=1 \\ j \neq l}}^N |a_{lj}|$$

**Weak Diagonal Dominance:**

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^N |a_{ij}|$$

**Strict Diagonal Dominance:**

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^N |a_{ij}|$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 13 / 20

Another method we can apply that is called as successive over relaxation. So we can apply this method and in successive over relaxation we can control the update using Gauss Seidel step. So let us say that this is  $\phi^p - \phi^{p-1}$ . So this  $\omega$  is there, into this  $\phi$  Gauss Seidel step whatever we are getting minus  $\phi^p$  that is old value, this step.

So we will take smallest step compared to the original smaller or larger step compared to the proposed modification from the Gauss Seidel step.

(Refer Slide Time 12:24)

Gauss-Seidel Method  
Successive Over-Relaxation (SOR)  
General Algorithm  
Example

I.I.T. Kharagpur

### Gauss-Seidel Method (GS)

#### Successive Over-Relation (SOR)

Convergence can be achieved by increasing or reducing the step size

$$\phi^{(p)} - \phi^{(p-1)} = \omega [\phi_{GS}^{(p)} - \phi^{(p-1)}]$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 14 / 20

So if we directly write it in iterative form, so by transferring this part in the right hand side we can write omega into GS plus 1 minus omega into phi p minus 1. So in this case we can use the Gauss Seidel approximation.

(Refer Slide Time 12:55)

Gauss-Seidel Method (GS)  
Successive Over-Relation (SOR)

Convergence can be achieved by increasing or reducing the step size

$$\phi^{(p)} - \phi^{(p-1)} = \omega [\phi_{GS}^{(p)} - \phi^{(p-1)}]$$

In iterative form

$$\phi^{(p)} = \omega \phi_{GS}^{(p)} + (1 - \omega) \phi^{(p-1)}$$

Gauss-Seidel approximation can be written as

$$D \phi_{GS}^{(p)} = -L \phi^{(p)} - U \phi^{(p-1)} + r$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 14 / 20

In Gauss Seidel approximation what we are getting? We are getting this D plus L. So right hand side we are writing this for lower triangular one, we are transferring. Now in this case if we multiply D on both sides.

(Refer Slide Time 13:37)

Gauss-Seidel Method (GS)  
Successive Over-Relation (SOR)

Convergence can be achieved by increasing or reducing the step size

$$\phi^{(p)} - \phi^{(p-1)} = \omega [\phi_{GS}^{(p)} - \phi^{(p-1)}]$$

In iterative form

$$D \phi^{(p)} = \omega D \phi_{GS}^{(p)} + (1 - \omega) D \phi^{(p-1)}$$

Gauss-Seidel approximation can be written as

$$D \phi_{GS}^{(p)} = -L \phi^{(p)} - U \phi^{(p-1)} + r$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 14 / 20



Then we can get this form. This is actually our desired level into D. But this D is our Gauss Seidel step like this. So omega into this Gauss Seidel step plus 1 minus omega D phi p minus 1. So if we rearrange this, what we are getting?

(Refer Slide Time 14:11)

Gauss-Seidel Method (GS)  
Successive Over-Relation (SOR)

By combining expressions

$$\begin{aligned}
 \underline{D\phi^{(p)}} &= \omega \underline{D\phi_{GS}^{(p)}} + (1 - \omega) D\phi^{(p-1)} \\
 &= \omega \left[ \underline{-L\phi^{(p)} - U\phi^{(p-1)} + r} \right] + (1 - \omega) \underline{D\phi^{(p-1)}} \\
 &= -\omega L\phi^{(p)} + (1 - \omega) D\phi^{(p-1)} - \omega U\phi^{(p-1)} + \omega r
 \end{aligned}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 15 / 20

From this two terms we are getting this first term. This is our diagonal one and this part is coming from here and omega into r, this is our constant term. Although omega value may vary. But we can use a fixed value for iteration.

(Refer Slide Time 14:30)

Gauss-Seidel Method (GS)  
Successive Over-Relation (SOR)

By combining expressions

$$\begin{aligned}
 D\phi^{(p)} &= \omega D\phi_{GS}^{(p)} + (1 - \omega) D\phi^{(p-1)} \\
 &= \omega \left[ -L\phi^{(p)} - U\phi^{(p-1)} + r \right] + (1 - \omega) D\phi^{(p-1)} \\
 &= -\omega L\phi^{(p)} + (1 - \omega) D\phi^{(p-1)} - \omega U\phi^{(p-1)} + \omega r
 \end{aligned}$$

Dr. Anirban Dhar NPTEL Computational Hydraulics 15 / 20

So rearrangement is this one. We can change the side as per our desired one. D plus L that inverse we need to utilize for Gauss Seidel. But in this case we are not using this one. We are multiplying this omega.

(Refer Slide Time 14:57)

I.I.T. Kharagpur

### Gauss-Seidel Method (GS)

Successive Over-Relation (SOR)

By combining expressions

$$\begin{aligned} \mathbf{D}\phi^{(p)} &= \omega\mathbf{D}\phi_{GS}^{(p)} + (1-\omega)\mathbf{D}\phi^{(p-1)} \\ &= \omega[-\mathbf{L}\phi^{(p)} - \mathbf{U}\phi^{(p-1)} + \mathbf{r}] + (1-\omega)\mathbf{D}\phi^{(p-1)} \\ &= -\omega\mathbf{L}\phi^{(p)} + (1-\omega)\mathbf{D}\phi^{(p-1)} - \omega\mathbf{U}\phi^{(p-1)} + \omega\mathbf{r} \end{aligned}$$

Rearrangement yields

$$(\mathbf{D} + \omega\mathbf{L})\phi^{(p)} = [(1-\omega)\mathbf{D} - \omega\mathbf{U}]\phi^{(p-1)} + \omega\mathbf{r}$$

Finally in matrix form

$$\phi^{(p)} = (\mathbf{D} + \omega\mathbf{L})^{-1} [(1-\omega)\mathbf{D} - \omega\mathbf{U}]\phi^{(p-1)} + \omega(\mathbf{D} + \omega\mathbf{L})^{-1}\mathbf{r}$$

Dr. Anirban Dhar      NPTEL      Computational Hydraulics      15 / 20

We are bit cautious, we are reducing or increasing steps. So finally in matrix form this can be written like this.

(Refer Slide Time 15:08)

I.I.T. Kharagpur

### Gauss-Seidel Method (GS)

Successive Over-Relation (SOR)

By combining expressions

$$\begin{aligned} \mathbf{D}\phi^{(p)} &= \omega\mathbf{D}\phi_{GS}^{(p)} + (1-\omega)\mathbf{D}\phi^{(p-1)} \\ &= \omega[-\mathbf{L}\phi^{(p)} - \mathbf{U}\phi^{(p-1)} + \mathbf{r}] + (1-\omega)\mathbf{D}\phi^{(p-1)} \\ &= -\omega\mathbf{L}\phi^{(p)} + (1-\omega)\mathbf{D}\phi^{(p-1)} - \omega\mathbf{U}\phi^{(p-1)} + \omega\mathbf{r} \end{aligned}$$

Rearrangement yields

$$(\mathbf{D} + \omega\mathbf{L})\phi^{(p)} = [(1-\omega)\mathbf{D} - \omega\mathbf{U}]\phi^{(p-1)} + \omega\mathbf{r}$$

Finally in matrix form

$$\phi^{(p)} = (\mathbf{D} + \omega\mathbf{L})^{-1} [(1-\omega)\mathbf{D} - \omega\mathbf{U}]\phi^{(p-1)} + \omega(\mathbf{D} + \omega\mathbf{L})^{-1}\mathbf{r}$$

Dr. Anirban Dhar      NPTEL      Computational Hydraulics      15 / 20

But implementation wise we already have information about this system. Now this is in compact form, from our Gauss Seidel. This is our actually Gauss Seidel step. Now with this if

we proceed and if we write in this format so that we can achieve convergence by increasing or decreasing the time step, we can simply input this value here.

(Refer Slide Time 15:52)

Gauss-Seidel Method  
Successive Over-Relaxation (SOR)  
General Algorithm  
Example

I.I.T. Kharagpur

### Gauss-Seidel Method

General Algorithm

Iteration starts with the guess value  $\phi^{(0)}$

$$\phi^{(0)} = [\phi_1^{(0)} \quad \phi_2^{(0)} \quad \dots \quad \phi_{N-1}^{(0)} \quad \phi_N^{(0)}]^T$$

By adding and subtracting  $\phi_i^{(p-1)}$  in right hand side

$$\phi_{i,GS}^{(p)} = \phi_i^{(p-1)} + \frac{1}{a_{ii}} \left[ r_i - \sum_{j=1}^{i-1} a_{ij} \phi_j^{(p)} - a_{ii} \phi_i^{(p-1)} - \sum_{j=i+1}^N a_{ij} \phi_j^{(p-1)} \right], \quad \forall i, p \geq 1$$

In compact form

$$\phi_{i,GS}^{(p)} = \phi_i^{(p-1)} + \frac{Res_i}{a_{ii}}, \quad \forall i, p \geq 1$$

Convergence can be achieved by increasing or reducing the step size

$$\phi_i^{(p)} - \phi_i^{(p-1)} = \omega [\phi_{i,GS}^{(p)} - \phi_i^{(p-1)}]$$

Dr. Anirban Dhar      NPTEL      Computational Hydraulics      16 / 20

And rewrite this. So final form is omega into residual i. So a ii, this is our general form and omega is written here. So rearrangement is that phi i equals to old value plus residual i, a ii, omega. So in a way this value omega varies between zero to 2.

(Refer Slide Time 16:40)

Gauss-Seidel Method  
Successive Over-Relaxation (SOR)  
General Algorithm  
Example

I.I.T. Kharagpur

### Gauss-Seidel Method

General Algorithm

Final form can be written as

$$\begin{aligned} \phi_i^{(p)} &= \phi_i^{(p-1)} + \omega [\phi_{i,GS}^{(p)} - \phi_i^{(p-1)}] \\ &= \phi_i^{(p-1)} + \omega \frac{Res_i}{a_{ii}} \end{aligned}$$

Rearrangement yields

$$\phi_i^{(p)} = \phi_i^{(p-1)} + \frac{Res_i}{(a_{ii}/\omega)}, \quad 0 < \omega < 2$$

Dr. Anirban Dhar      NPTEL      Computational Hydraulics      17 / 20

Now if we have zero to 1, we call it as under relaxation. That means whatever value is coming here, we are reducing that value under relaxation.

(Refer Slide Time 17:00)

Gauss-Seidel Method  
Successive Over-Relaxation (SOR)  
General Algorithm  
Example

I.I.T. Kharagpur

### Gauss-Seidel Method

General Algorithm

Final form can be written as

$$\phi_i^{(p)} = \phi_i^{(p-1)} + \omega [\phi_{i,GS}^{(p)} - \phi_i^{(p-1)}]$$
$$= \phi_i^{(p-1)} + \omega \frac{Res_i}{a_{ii}}$$

Rearrangement yields

$$\phi_i^{(p)} = \phi_i^{(p-1)} + \frac{Res_i}{(a_{ii}/\omega)}, \quad 0 < \omega < 2$$

- $0 < \omega < 1 \Rightarrow$  Under-relaxation

Dr. Anirban Dhar      NPTEL      Computational Hydraulics      17 / 20

And over relaxation if it is between 1 to 2. But virtually what is happening, we are playing with the diagonal term. If we rewrite this thing like this, this is  $a_{ii}$  divided by  $\omega$ . So in case of under relaxation we are actually increasing the diagonal term. And in case of over relaxation we are reducing the diagonal term. So by changing this  $\omega$  values we can control the convergence for Gauss Seidel over relaxation as our method.

(Refer Slide Time 17:48)

Gauss-Seidel Method  
Successive Over-Relaxation (SOR)  
General Algorithm  
Example

I.I.T. Kharagpur

### Gauss-Seidel Method

General Algorithm

Final form can be written as

$$\phi_i^{(p)} = \phi_i^{(p-1)} + \omega [\phi_{i,GS}^{(p)} - \phi_i^{(p-1)}]$$
$$= \phi_i^{(p-1)} + \omega \frac{Res_i}{a_{ii}}$$

Rearrangement yields

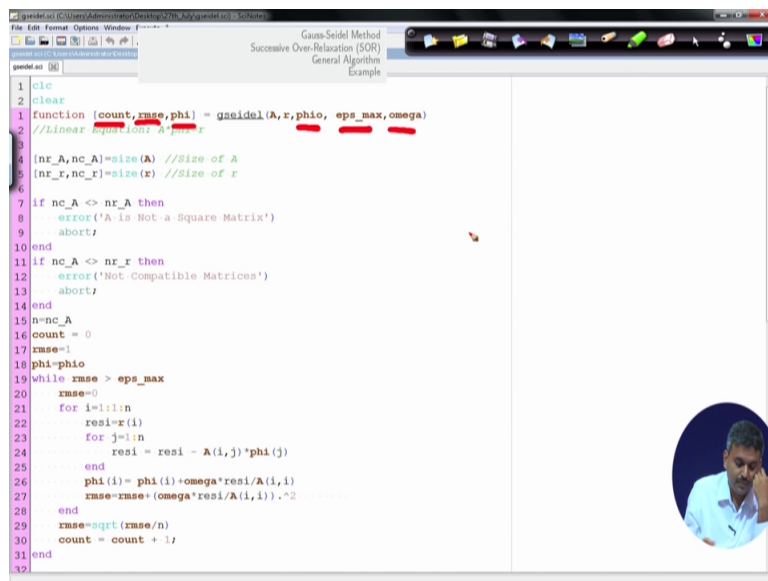
$$\phi_i^{(p)} = \phi_i^{(p-1)} + \frac{Res_i}{(a_{ii}/\omega)}, \quad 0 < \omega < 2$$

- $0 < \omega < 1 \Rightarrow$  Under-relaxation
- $1 < \omega < 2 \Rightarrow$  Over-relaxation

Dr. Anirban Dhar      NPTEL      Computational Hydraulics      17 / 20

Now let us consider our standard example in this case. In this case if we have Gauss Seidel up to this. This is general, we are again utilizing count, r, m, s, c,  $\phi_i$ ,  $\phi_i$  initial guess. So  $\epsilon_{max}$ ,  $\omega$ .  $\omega$  means now we need to provide this  $\omega$  value for this case.

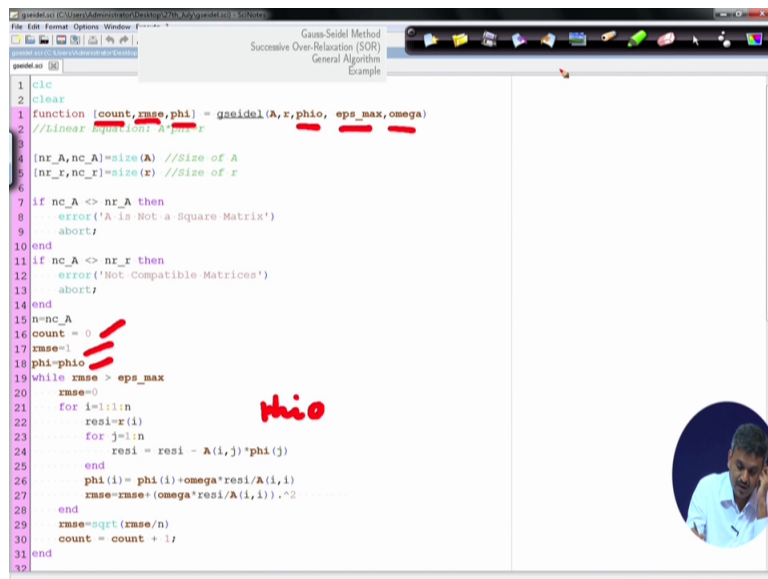
(Refer Slide Time 18:33)



```
1 clear
2 clear
3 function [count,rmse,phi] = gausidel(A,r,phio, eps_max,omega)
4 //Linear equation: A*x=r
5
6 [nr,nc_A]=size(A) //Size of A
7 [nr_r,nc_r]=size(r) //Size of r
8
9 if nc_A <> nr_A then
10     error('A is Not a Square Matrix')
11     abort;
12 end
13 if nc_A <> nr_r then
14     error('Not Compatible Matrices')
15     abort;
16 end
17 n=nc_A
18 count = 0
19 rmse=1
20 phi=phio
21 while rmse > eps_max
22     rmse=0
23     for i=1:n
24         resi=r(i)
25         for j=1:n
26             resi = resi - A(i,j)*phi(j)
27             phi(i) = phi(i)+omega*resi/A(i,i)
28             rmse=rmse+(omega*resi/A(i,i)).^2
29         end
30     end
31     rmse=sqrt(rmse/n)
32     count = count + 1;
33 end
34
```

Now this is count equals to zero, rmse equals to 1 to execute this while loop. And phi is equals to phi o. That means previously in Jacobi iteration we were using phi o or old values only.

(Refer Slide Time 18:51)



```
1 clear
2 clear
3 function [count,rmse,phi] = gausidel(A,r,phio, eps_max,omega)
4 //Linear equation: A*x=r
5
6 [nr,nc_A]=size(A) //Size of A
7 [nr_r,nc_r]=size(r) //Size of r
8
9 if nc_A <> nr_A then
10     error('A is Not a Square Matrix')
11     abort;
12 end
13 if nc_A <> nr_r then
14     error('Not Compatible Matrices')
15     abort;
16 end
17 n=nc_A
18 count = 0
19 rmse=1
20 phi=phio
21 while rmse > eps_max
22     rmse=0
23     for i=1:n
24         resi=r(i)
25         for j=1:n
26             resi = resi - A(i,j)*phi(j)
27             phi(i) = phi(i)+omega*resi/A(i,i)
28             rmse=rmse+(omega*resi/A(i,i)).^2
29         end
30     end
31     rmse=sqrt(rmse/n)
32     count = count + 1;
33 end
34
```

Now whatever value is updated we will directly try to utilize it. So that's why I have transferred these values here directly in phi. Now I am directly updating this phi. So updating and we are multiplying this omega, this residual, residual i again starting with  $r_i$  minus  $a_{ij}$  and  $\phi_j$ . And this is for all  $j$ . That means whether it is for lower or upper triangular matrix whatever maybe the coefficient. Whatever value is available with present or past iteration level, we are utilizing that information.

(Refer Slide Time 19:39)

```
1 clear
2 clear
3
4 function [count,rmse,phi] = gseidel(A,r,phio, eps_max,omega)
5 //Linear Equation: A*phi=r
6
7 [nr,nc_A]=size(A) //Size of A
8 [nr_r,nc_r]=size(r) //Size of r
9
10 if nc_A <> nr_A then
11     error('A is Not a Square Matrix')
12     abort;
13 end
14
15 if nc_A <> nr_r then
16     error('Not Compatible Matrices')
17     abort;
18 end
19 n=nc_A
20 count = 0
21 rmse=1
22 phi=phio
23 while rmse > eps_max
24     rmse=0
25     for i=1:n
26         resi=r(i)
27         for j=1:n
28             resi = resi - A(i,j)*phi(j)
29         end
30         phi(i) = phi(i) + omega*resi/A(i,i)
31         rmse=rmse + (omega*resi/A(i,i)).^2
32     end
33     end
34     rmse=sqrt(rmse/n)
35     count = count + 1;
36 end
37 end
```

Now with this I am also calculating rmse. Rmse again, omega because this is actually the increment that we are giving between 2 iterations.

(Refer Slide Time 19:56)

```
1 clear
2 clear
3
4 function [count,rmse,phi] = gseidel(A,r,phio, eps_max,omega)
5 //Linear Equation: A*phi=r
6
7 [nr,nc_A]=size(A) //Size of A
8 [nr_r,nc_r]=size(r) //Size of r
9
10 if nc_A <> nr_A then
11     error('A is Not a Square Matrix')
12     abort;
13 end
14
15 if nc_A <> nr_r then
16     error('Not Compatible Matrices')
17     abort;
18 end
19 n=nc_A
20 count = 0
21 phi=phio
22 while rmse > eps_max
23     rmse=0
24     for i=1:n
25         resi=r(i)
26         for j=1:n
27             resi = resi - A(i,j)*phi(j)
28         end
29         phi(i) = phi(i) + omega*resi/A(i,i)
30         rmse=rmse + (omega*resi/A(i,i)).^2
31     end
32     end
33     rmse=sqrt(rmse/n)
34     count = count + 1;
35 end
36 end
```

So again this rmse equals to rmse divided by n and we're taking square root. So this is actual calculation and count equals to count plus 1. And with this, this while loop ends.

(Refer Slide Time 20:13)

```
gaussid.m (C:\Users\Amruth\Desktop\21_01\gaussid.m)
File Edit Format Options Window Help
Gauss-Seidel Method
Successive Over-Relaxation (SOR)
General Algorithm
Example
gaussid.m (8)
1 clear
2 clear
3
4 function [count,rmse,phi] = gaussid(A,r,phio, eps_max,omega)
5 //Linear Equation: A*phi=r
6
7 [nr,nc_A]=size(A) //Size of A
8 [nr_r,nc_r]=size(r) //Size of r
9
10 if nc_A <> nr_A then
11     error('A Is Not a Square Matrix')
12     abort;
13 end
14 if nc_A <> nr_r then
15     error('Not Compatible Matrices')
16     abort;
17 end
18 n=nc_A
19 count = 0
20 rmse=1
21 phi=phio
22 while rmse > eps_max
23     rmse=0
24     for i=1:n
25         resi=r(i)
26         for j=1:n
27             resi = resi - A(i,j)'phi(j)
28         end
29         phi(i) = phi(i)+omega*resi/A(i,i)
30         rmse=rmse+(omega*resi/A(i,i)).^2
31     end
32     rmse=sqrt(rmse/n)
33     count = count + 1;
34 end
35 endfunction
```

And this is our algorithm. Compared to our direct solvers like LU decomposition or Gauss elimination, we need to write less numbers of lines for this one. In case of this our standard problem, we can use this concept of our case. And in this case let us say that A1, this is for diagonally dominant our standard matrix.

(Refer Slide Time 21:01)

```
gaussid.m (C:\Users\Amruth\Desktop\21_01\gaussid.m)
File Edit Format Options Window Help
Gauss-Seidel Method
Successive Over-Relaxation (SOR)
General Algorithm
Example
gaussid.m (8)
19 while rmse > eps
20     rmse=0
21     for i=1:n
22         resi=r(i)
23         for j=1:n
24             resi = resi - A(i,j)'phi(j)
25         end
26         phi(i) = phi(i)+omega*resi/A(i,i)
27         rmse=rmse+(omega*resi/A(i,i)).^2
28     end
29     rmse=sqrt(rmse/n)
30     count = count + 1;
31 end
32 endfunction
33
34 A1=[
35     1 0 0 0 0
36     1 2 1 0 0
37     0 1 3 -1 0
38     0 0 1 2 1
39     0 0 0 0 1];
40
41 r1=[
42     1
43     12
44     11
45     28
46     9];
47
48 A2=[1 2 -3 4 5
49     3 -5 -7 9
50     -4 3 -2 1
51     4 -7 -10 13
52     -15 13 11 -9 2];
53
54
```

So let us start with A1 and r1. Now omega equals to 1. Omega equals to 1 means we are at Gauss Seidel step. We are not using SOR that means over relaxation or under relaxation. So in this case omega is 1, this means we are at exactly Gauss Seidel step.





(Refer Slide Time 22:40)

```

32 endfunction
33
34
35 A1=[1 0 0 0 0
36 2 1 0 0
37 0 1 3 -1 0
38 0 0 1 2 1
39 0 0 0 0 1];
40
41
42
43 r1=[1
44 12
45 11
46 28
47 9];
48
49 A2=[1 2 -3 4 5
50 0 3 -5 -7 9
51 5 -4 3 -2 1
52 1 4 -7 -10 13
53 -15 13 11 -9 21];
54
55 r2=[37
56 8
57 3
58 13
59 10];
60 phi0=[0
61 0
62 0
63 0
64 0];
65 eps_max=1e-6;
66 omega=1.5;
67 [count, rmse, phi] = gseidel(A1,r1,phi0, eps_max, omega)

```

Then what is happening here? If 1.5, we need 38 iterations.

(Refer Slide Time 22:53)

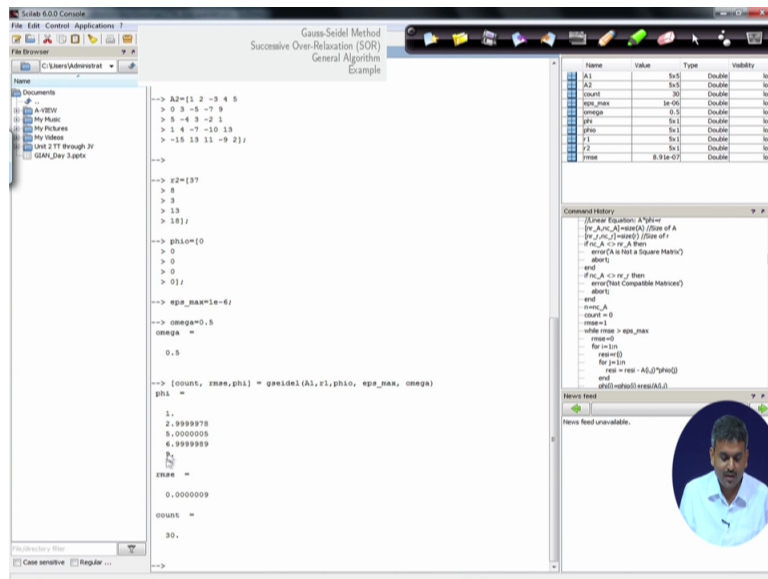
```

--> A2=[1 2 -3 4 5
--> 0 3 -5 -7 9
--> 5 -4 3 -2 1
--> 1 4 -7 -10 13
--> -15 13 11 -9 21];
-->
--> r2=[37
--> 8
--> 3
--> 13
--> 10];
--> phi0=[0
--> 0
--> 0
--> 0
--> 0];
--> eps_max=1e-6;
--> omega=1.5;
-->
--> [count, rmse, phi] = gseidel(A1,r1,phi0, eps_max, omega)
count =
    38.
rmse =
    0.0000009
phi =
    1.
    3.0000006
    6.9999996
    9.
    0.0000009

```

And if we have point 5 that is under relaxation, then we need 30 iteration for this one.

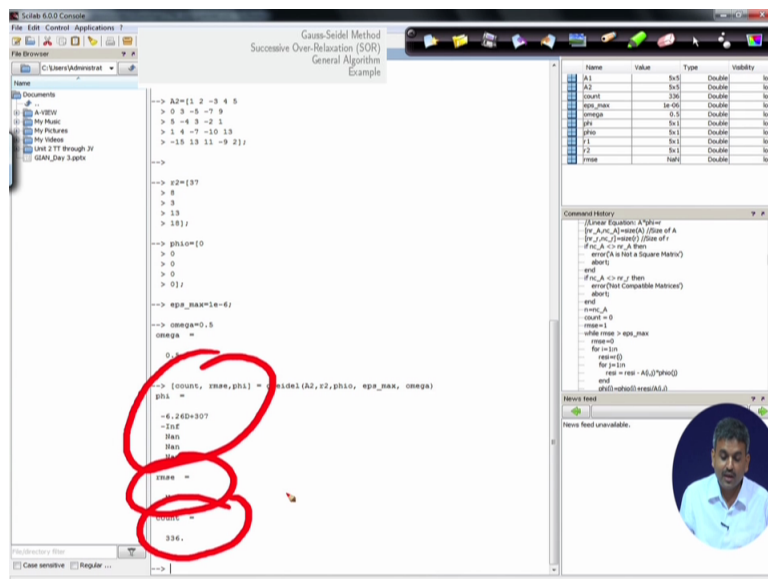
(Refer Slide Time 23:10)



And this values are closed. For this one within five iteration with Gauss Seidel we are getting the solution. But if you are having large matrix and sparse matrix we need then the utility of this approach will be very much visible. Now if we consider our second matrix which is not diagonally dominant and we utilize this algorithm. That means this is a2 and r2. Only change is in terms of a2, r2.

We are keeping the initial value epsilon max and omega same. So if we do that we can easily see that solutions are not converging. Rmsc equals to Nan. Although these many iterations we are not getting converge solution for this one.

(Refer Slide Time 24:26)



So for that our diagonal dominance is very much required for this iterative system. Otherwise we will not get solution from this system of equations. So with this we can end our successive GS SOR methods. And next lecture we will devote with nonlinear iterative techniques. Thank you.