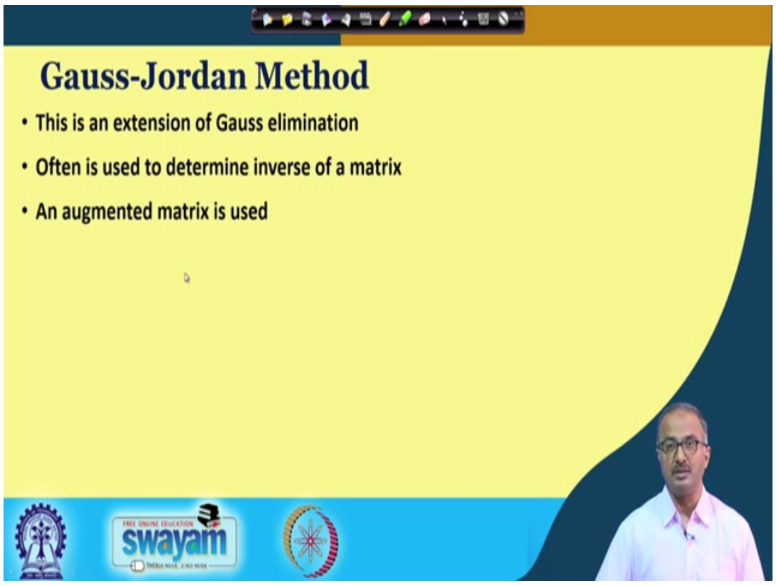
Mass, Momentum and Energy Balances in Engineering Analysis

Prof. Pavitra Sandilya
Cryogenics Engineering Center
Indian Institute of Technology, Kharagpur

Lecture - 37
Matrix Techniques – II

Welcome. We started with some matrix methods to use for solving the model equations we obtained. And we started with the methods which can give us the solutions without any iterations and that were LU decomposition and the tridiagonal matrix system. Now in this lecture we shall be going further with a few more methods and in this lecture we shall be covering both the direct methods and the iterative methods.

(Refer Slide Time: 00:48)

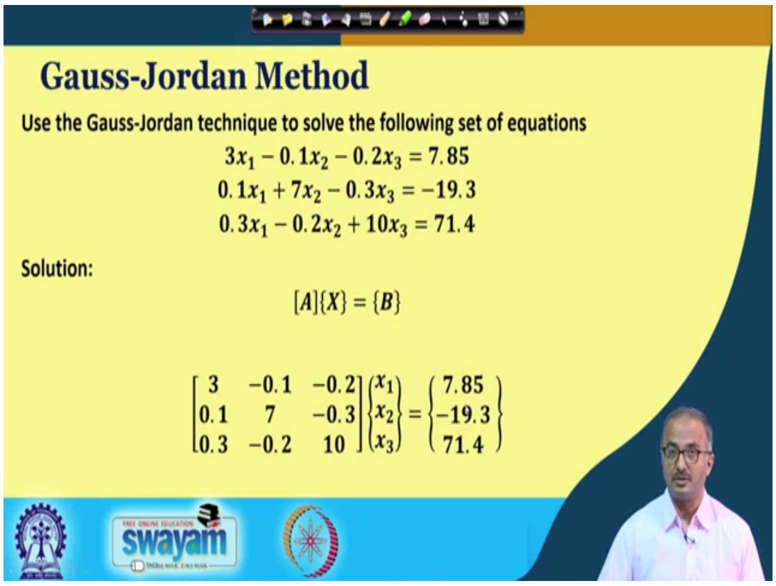


Gauss-Jordan Method

- This is an extension of Gauss elimination
- Often is used to determine inverse of a matrix
- An augmented matrix is used

So, first let us the Gauss Jordan method which is a modification of the Gauss elimination and it is often used to find out the inverse of a matrix and in this case we use an augmented matrix. And this particular method I shall be illustrating you with an example.

(Refer Slide Time: 01:03)



Gauss-Jordan Method

Use the Gauss-Jordan technique to solve the following set of equations

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85$$
$$0.1x_1 + 7x_2 - 0.3x_3 = -19.3$$
$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4$$

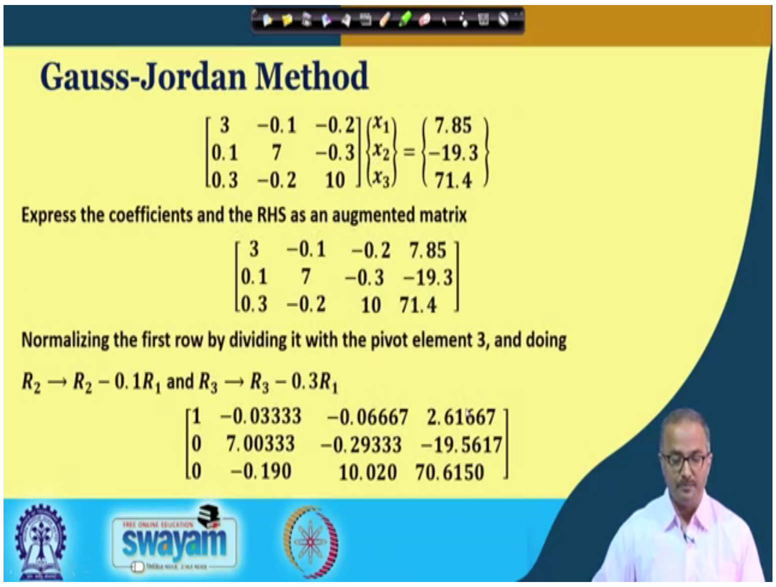
Solution:

$$[A]\{X\} = \{B\}$$

$$\begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.1 & 7 & -0.3 \\ 0.3 & -0.2 & 10 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 7.85 \\ -19.3 \\ 71.4 \end{Bmatrix}$$

So, here we see that we have a set of equations with the 3 variables x 1 x 2 and x 3 and these equations have to be solved to get the value of x 1 x 2 and x 3. So, let us first put this set of equations in terms of the matrixes. So, you have these are A matrix of the coefficients and the X matrix is a unknowns, the vector and B the vector of the forcing functions ok.

(Refer Slide Time: 01:32)



Gauss-Jordan Method

$$\begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.1 & 7 & -0.3 \\ 0.3 & -0.2 & 10 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} = \begin{Bmatrix} 7.85 \\ -19.3 \\ 71.4 \end{Bmatrix}$$

Express the coefficients and the RHS as an augmented matrix

$$\begin{bmatrix} 3 & -0.1 & -0.2 & 7.85 \\ 0.1 & 7 & -0.3 & -19.3 \\ 0.3 & -0.2 & 10 & 71.4 \end{bmatrix}$$

Normalizing the first row by dividing it with the pivot element 3, and doing

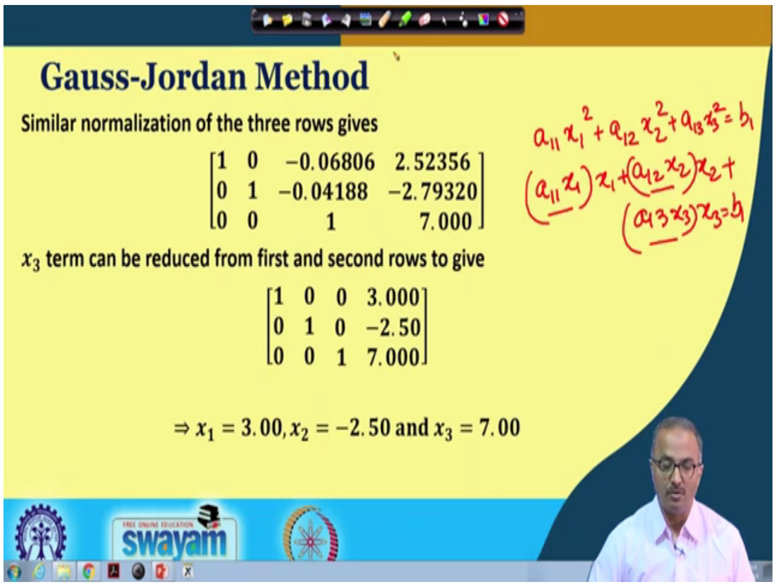
$R_2 \rightarrow R_2 - 0.1R_1$ and $R_3 \rightarrow R_3 - 0.3R_1$

$$\begin{bmatrix} 1 & -0.03333 & -0.06667 & 2.61667 \\ 0 & 7.00333 & -0.29333 & -19.5617 \\ 0 & -0.190 & 10.020 & 70.6150 \end{bmatrix}$$

Now, what we do that after this thing, we make the augmented matrix what we do that, we put another matrix in this we put the values of a first and the last column we put all these forcing functions over here. Now what we do that, we first normalize the first row by dividing with the pivot element; pivot element are the elements at the diagonal.

So, if I divide the first equation with the diagonal element, we get this particular thing. And a second equation we divide by the seven. So, we get this thing and the third one we get this. Now you see that we are trying to reduce the whole system into that we want to have a upper triangular matrix. So, we are doing what that R 2 and this is going to R 2 minus 0.1 into R 1 so, that we are getting this as 0. So, this way we are able to slows 1 by stepwise, we are able to get the 0s at the lower triangular thing. So, this we continue.

(Refer Slide Time: 02:40)



Gauss-Jordan Method

Similar normalization of the three rows gives

$$\begin{bmatrix} 1 & 0 & -0.06806 & 2.52356 \\ 0 & 1 & -0.04188 & -2.79320 \\ 0 & 0 & 1 & 7.000 \end{bmatrix}$$

x_3 term can be reduced from first and second rows to give

$$\begin{bmatrix} 1 & 0 & 0 & 3.000 \\ 0 & 1 & 0 & -2.50 \\ 0 & 0 & 1 & 7.000 \end{bmatrix}$$

$$\Rightarrow x_1 = 3.00, x_2 = -2.50 \text{ and } x_3 = 7.00$$

$$a_{11}x_1^2 + a_{12}x_2^2 + a_{13}x_3^2 = b_1$$

$$(\underline{a_{11}x_1})x_1 + (\underline{a_{12}x_2})x_2 + (\underline{a_{13}x_3})x_3 = b_1$$

This is also done in the Gauss elimination method and in Gauss elimination we also convert the parent matrix into an upper triangular matrix and then we go do a backward sweep to get the values of the unknowns.

Now, same thing we are doing, now here what we are doing that as in the Gauss elimination we are also making all these diagonal elements to be 1. So, once we have done this thing, you can see that we have converted the parent equation into an identity matrix. Identity matrix you can see that in which the diagonal elements are 1 and all the super diagonal elements and the sub diagonal elements will be 0.

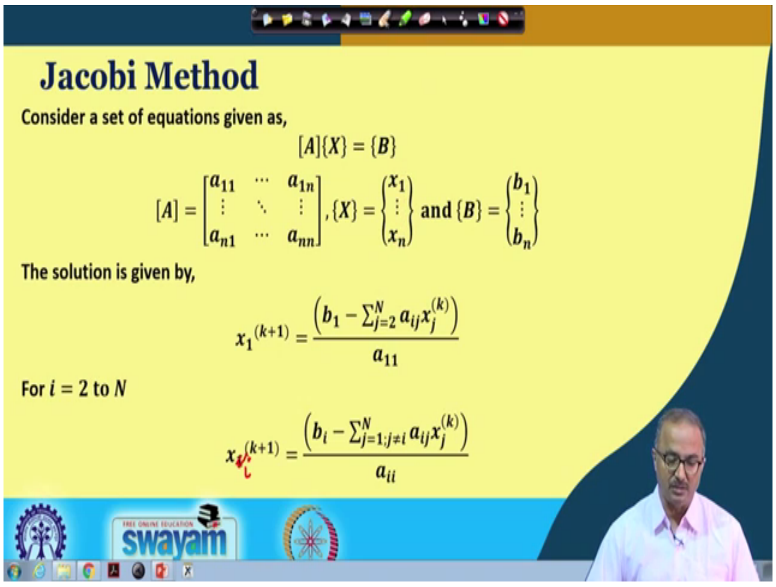
So, we have getting the identity matrix and when we get the identity matrix, now it is very easy for us to solve for the unknowns. So, we find that x 1 will be equal to 3, x 2 will be equal to minus 2.5 and x 3 will be equal to 7. So, this way you can carry out this particular method and you can also program it to get the solution. But as I was telling that it is not always possible to get the use these direct methods and they prove inefficient as the dimension of the matrices become larger. So, we go for indirect method. So, now, I shall be going to some of them indirect methods.

And one thing I must tell you that whenever you are for actual systems, whenever you are linearizing the equations many a times even without linearization also we start with non-linear equations. In the sense that suppose I have an equation like this an equation like this, that a 1 1 say x 1 square and say a 1 2 x 2 square and suppose a 1 3 x 3 square equals a b 1. So, this is just illustration in many a times what we do if it is non-linear we can take this x 1 into x 1, and then 1 to x 2 into x 2 and then 1 3 x 3 into x 3 equal to b 1.

So, you can see here these coefficients are themselves function of the unknown. So, this is a also one way of treating these things, but when you do this kind of system then what you see that you have to go for iteration because you start with some guess value of x 1 x 2 x 3 so, that you can find the values of these coefficients and then you solve for the x 1 x 2 x 3 and again you modify the coefficient values. And again you get a modified update the value of the x minus to x 3 and this you keep on doing till you get convergence by checking the values of x 1 x 2 x 3 in the 2 consecutive iterations ok.

So; that means, we are able to stay within this Gauss Jordan system, but in a iterative manner we are using this system. So, now, we shall go to the another set of solutions which are purely iterative.

(Refer Slide Time: 05:52)



Jacobi Method

Consider a set of equations given as,

$$[A]\{X\} = \{B\}$$

$$[A] = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}, \{X\} = \begin{Bmatrix} x_1 \\ \vdots \\ x_n \end{Bmatrix} \text{ and } \{B\} = \begin{Bmatrix} b_1 \\ \vdots \\ b_n \end{Bmatrix}$$

The solution is given by,

$$x_1^{(k+1)} = \frac{\left(b_1 - \sum_{j=2}^{N} a_{ij} x_j^{(k)}\right)}{a_{11}}$$

For $i = 2$ to N

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

So, here we have the Jacobi method. And this Jacobi method you can see that it is a method which depends on the reorientation or rewriting the given set of equations in terms of each of the unknown variables; that means, if my parent equation is like this ok. Now what we shall do is this we shall be finding the values of the x 1 x 2 up to x n from each of the equations. Now suppose we choose that first equation we shall be choosing to get the value of x 1, the second equation we shall be choosing to get the value of x 2 and so and so forth.

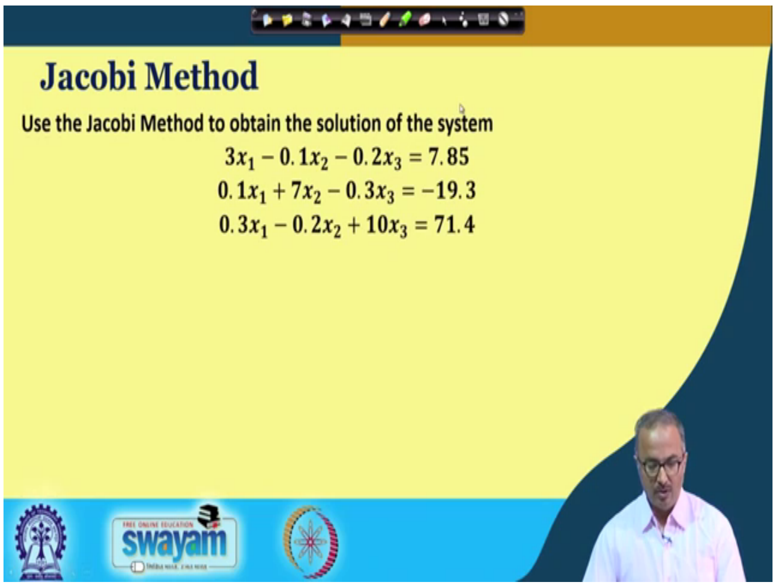
Now, these choice of the equation to get the value of the given unknown is arbitrary and there are some mathematical conditions which are prescribed to get those conditions. So, whatever those things I am not going to those details, but sufficient to say is this, we find these values of this say unknowns from each of the equations. So, for example, we want to find the value of x 1 from the first equation. So, we write like x 1 is equal to b 1 minus all the other terms which appear in the first equation divided by a 1 1 ok. And you can see here that because none of the values are known we have to start with some initial guess values. So, update the value of the xs ok.

So, here you see that we are getting x 1 k plus 1. So, this k tells us the level of the iteration. So, suppose k is 0, k 0 means our initial guess values from that we can update the value of x 1 ok. And now we can see that if we apply this same thing to all the other equations, you find that we can get the value of x i for all the other equations and this i goes from 2 to n. And in this case we find that as long as we are finding the x 1 to x n values for at a given iteration level, we are not changing the values of the x.

So, here you see that we are sticking to x 1 to x n, we are sticking to the same values of x which we guessed earlier ok. So, and then once we have obtained all the xs, then again we start with the second iteration and again we put though all the x values at the second iteration to get the values of x for the third iteration. And before we move on to the next iteration we must check the values of x may at kth level and k plus oneth level and then if we are not getting convergence and the convergence criterion will be decided by the user.

So, there could be one criterion or more than one criteria one criterion. So, that we decided prescribed by the user. So, whatever it is so, we have to keep on checking the iteration then to update the values of x. Now this Jacobi method was found to be less efficient. So, if modification was done and that we got the Gauss Seidel method which I shall be talking about now.

(Refer Slide Time: 09:13)



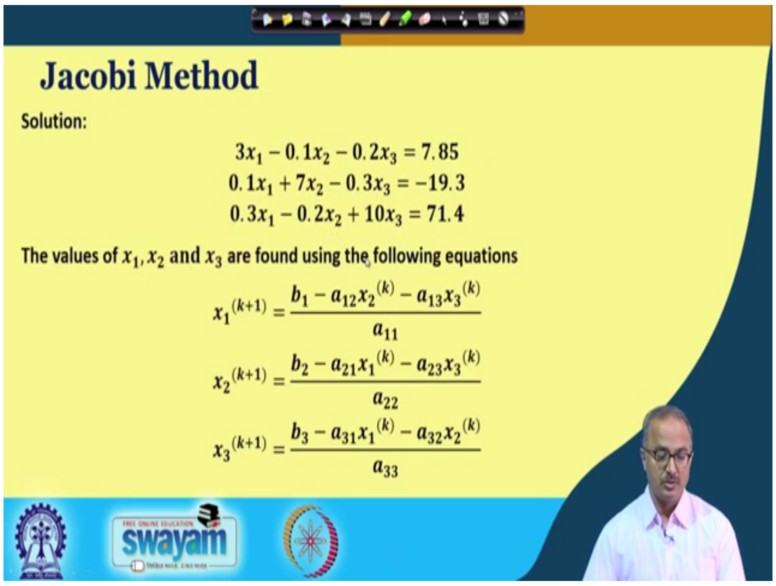
Jacobi Method

Use the Jacobi Method to obtain the solution of the system

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85$$
$$0.1x_1 + 7x_2 - 0.3x_3 = -19.3$$
$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4$$

In this method first let us see the example problem here before we go to the next method. So, here we have the set of equations, these equations are the same as the 1 we have just solved using the Gauss Jordan method.

(Refer Slide Time: 09:28)



Jacobi Method

Solution:

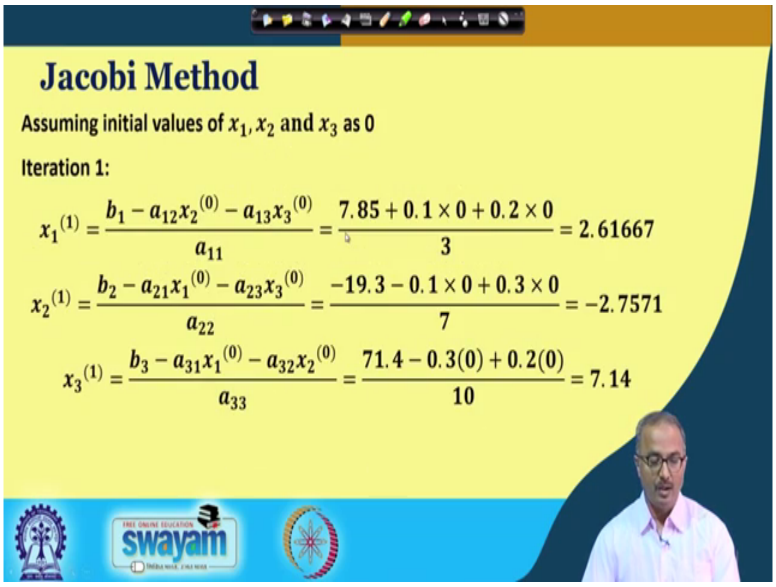
$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85$$
$$0.1x_1 + 7x_2 - 0.3x_3 = -19.3$$
$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4$$

The values of x_1, x_2 and x_3 are found using the following equations

$$x_1^{(k+1)} = \frac{b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)}}{a_{11}}$$
$$x_2^{(k+1)} = \frac{b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)}}{a_{22}}$$
$$x_3^{(k+1)} = \frac{b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)}}{a_{33}}$$

So, here what we do that we are writing this x 1 k plus 1 in terms of x 2 and x 3, and then x 2 k plus 1 in terms of x 1 and x 3 and x 3 from in terms of x 1 x x 2. So, this x 1 obtained from the first equation, x 2 may second equation and x 3 from the third equation.

(Refer Slide Time: 09:47)



Jacobi Method

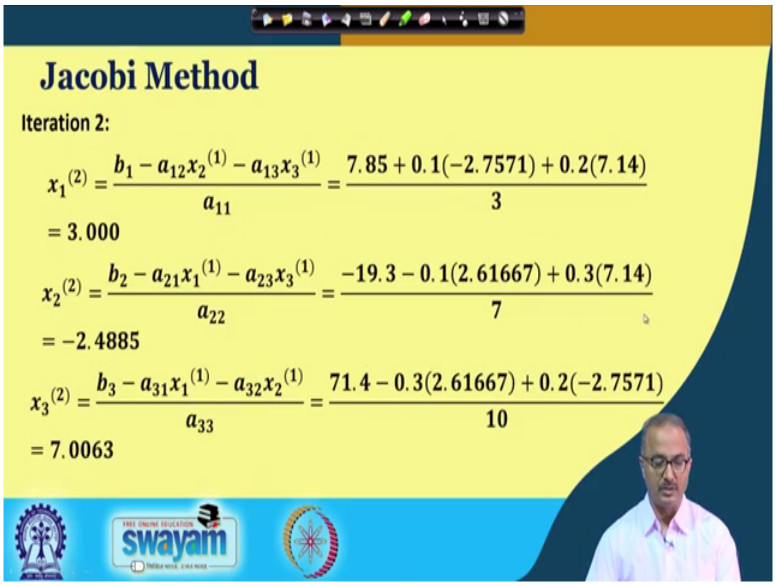
Assuming initial values of x_1, x_2 and x_3 as 0

Iteration 1:

$$x_1^{(1)} = \frac{b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)}}{a_{11}} = \frac{7.85 + 0.1 \times 0 + 0.2 \times 0}{3} = 2.61667$$
$$x_2^{(1)} = \frac{b_2 - a_{21}x_1^{(0)} - a_{23}x_3^{(0)}}{a_{22}} = \frac{-19.3 - 0.1 \times 0 + 0.3 \times 0}{7} = -2.7571$$
$$x_3^{(1)} = \frac{b_3 - a_{31}x_1^{(0)} - a_{32}x_2^{(0)}}{a_{33}} = \frac{71.4 - 0.3(0) + 0.2(0)}{10} = 7.14$$

So, here you see that we are getting the values of updated values of the x 1 x 2 x 3 by keeping all the values of the xs static at the kth level and this is the how we are getting the updation.

(Refer Slide Time: 10:04)



Jacobi Method

Iteration 2:

$$x_1^{(2)} = \frac{b_1 - a_{12}x_2^{(1)} - a_{13}x_3^{(1)}}{a_{11}} = \frac{7.85 + 0.1(-2.7571) + 0.2(7.14)}{3}$$

$$= 3.000$$

$$x_2^{(2)} = \frac{b_2 - a_{21}x_1^{(1)} - a_{23}x_3^{(1)}}{a_{22}} = \frac{-19.3 - 0.1(2.61667) + 0.3(7.14)}{7}$$

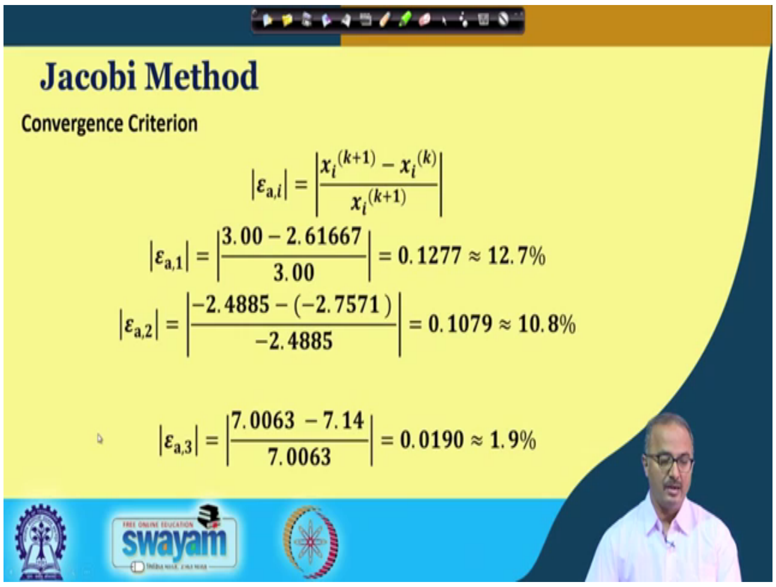
$$= -2.4885$$

$$x_3^{(2)} = \frac{b_3 - a_{31}x_1^{(1)} - a_{32}x_2^{(1)}}{a_{33}} = \frac{71.4 - 0.3(2.61667) + 0.2(-2.7571)}{10}$$

$$= 7.0063$$

And with this updation now we can go for the next iteration and we get these particular values ok.

(Refer Slide Time: 10:09)



Jacobi Method

Convergence Criterion

$$|\varepsilon_{a,i}| = \left|\frac{x_i^{(k+1)} - x_i^{(k)}}{x_i^{(k+1)}}\right|$$

$$|\varepsilon_{a,1}| = \left|\frac{3.00 - 2.61667}{3.00}\right| = 0.1277 \approx 12.7\%$$

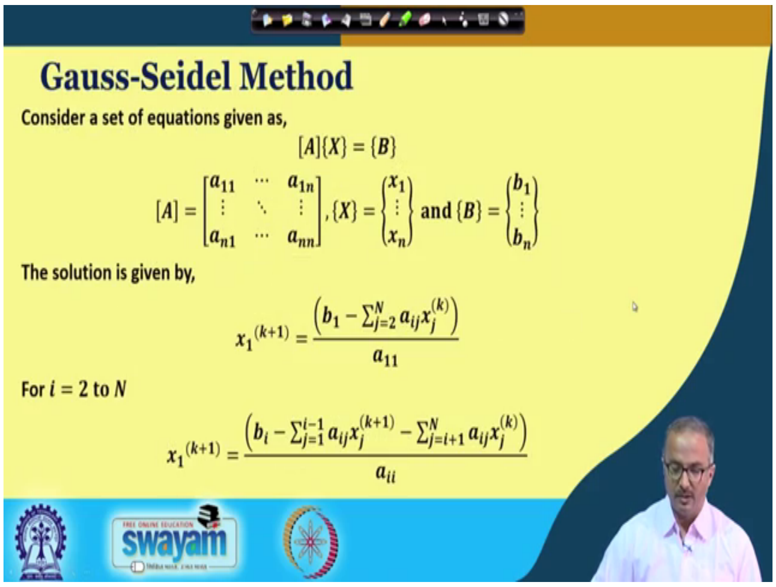
$$|\varepsilon_{a,2}| = \left|\frac{-2.4885 - (-2.7571)}{-2.4885}\right| = 0.1079 \approx 10.8\%$$

$$|\varepsilon_{a,3}| = \left|\frac{7.0063 - 7.14}{7.0063}\right| = 0.0190 \approx 1.9\%$$

And again we check for the convergence like we are taking the what is the fractional change or the percentage change, we see that how much percentage change is occurring. So, it we are finding that it is giving 12.7 percent for the x 1, then for the x 2 it is given 10.8 percent and for the third one it is giving 1.9 percent.

It shows that by choice of the particulars equation for x 1 x 2 x 3; x 3 is giving the fastest convergence and the x 1 is giving the slowest convergence. Now you can see that if we want to make the rates of convergence for each of the variables faster or of the same magnitude, then we can again change these orders of the equations to obtain or to update the values of x 1 x 2 x 3. So, this is 1 example that in this in this particular choice of x 1 x 2 x 3 expressions we are getting this kind of rate of convergence, but a different choice of expressions for x 1 x 2 x 3 will give us a different rate of convergence. So, these we can carry on and to get.

(Refer Slide Time: 11:20)



Gauss-Seidel Method

Consider a set of equations given as,

$$[A]\{X\} = \{B\}$$

$$[A] = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}, \{X\} = \begin{Bmatrix} x_1 \\ \vdots \\ x_n \end{Bmatrix} \text{ and } \{B\} = \begin{Bmatrix} b_1 \\ \vdots \\ b_n \end{Bmatrix}$$

The solution is given by,

$$x_1^{(k+1)} = \frac{\left(b_1 - \sum_{j=2}^{N} a_{ij} x_j^{(k)}\right)}{a_{11}}$$

For $i = 2$ to N

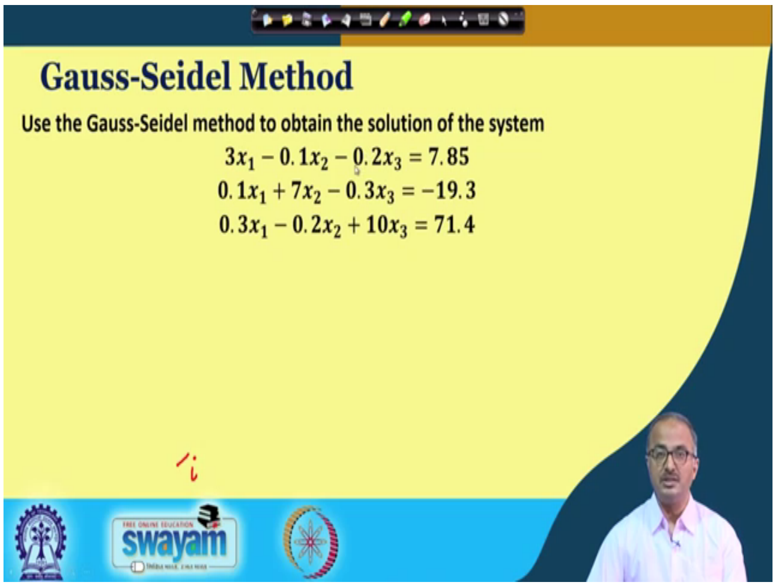
$$x_1^{(k+1)} = \frac{\left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{N} a_{ij} x_j^{(k)}\right)}{a_{ii}}$$

Now let us go to another method which is a improvement of the Gauss Seidel and you see that improvement comes that when we are writing these equations for the first unknown that is x 1, we are using the all the variables which are there for the previous iteration level.

But when we go for the ith value you see this is a ith x then what we are doing that, we are using both the updated values as well as the previous values updated values for those which we have already updated and the previous values for those we are which are yet to be updated. So, we are seeing that for j equal to 1 to i minus 1 that is i if this is i ok. So, these i we are doing the updated values and which are i plus 1 to n we are doing for the previous values ok.

Now, when you are doing this we are finding that the rate of convergence gets speeded up ok. So, this is the advantage we are getting for the Gauss Seidel method.

(Refer Slide Time: 12:29)



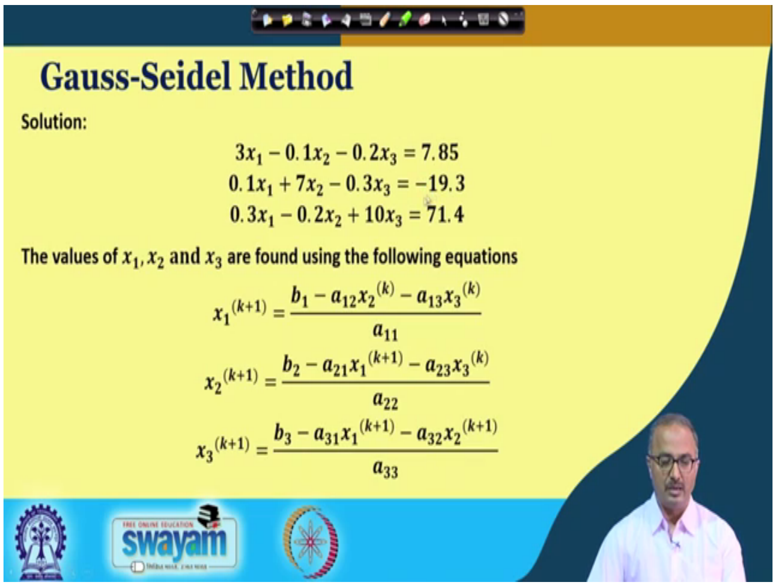
Gauss-Seidel Method

Use the Gauss-Seidel method to obtain the solution of the system

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85$$
$$0.1x_1 + 7x_2 - 0.3x_3 = -19.3$$
$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4$$

Now, this is the same example we are taking for the Gauss Seidel method and we see that.

(Refer Slide Time: 12:34)



Gauss-Seidel Method

Solution:

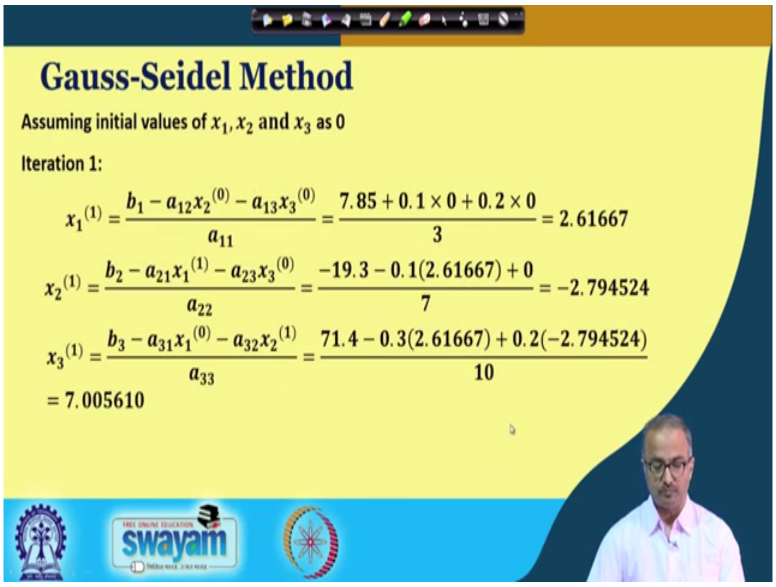
$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85$$
$$0.1x_1 + 7x_2 - 0.3x_3 = -19.3$$
$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4$$

The values of x_1, x_2 and x_3 are found using the following equations

$$x_1^{(k+1)} = \frac{b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)}}{a_{11}}$$
$$x_2^{(k+1)} = \frac{b_2 - a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)}}{a_{22}}$$
$$x_3^{(k+1)} = \frac{b_3 - a_{31}x_1^{(k+1)} - a_{32}x_2^{(k+1)}}{a_{33}}$$

How we are doing it that for x 1 again we take the same first equation to get the value of x 1 and here it is that we are finding the x 1 value. So, once we have got them updated the x 1 value what we are doing? We take this updated value to get the value of x 2. So, we are putting the updated value of x 1, but for x 3 we are taking the same value as in the previous iteration and we update the value of the x 2 and after that for the x 3 we take updated values of both x 1 and x 2.

(Refer Slide Time: 13:06)



Gauss-Seidel Method

Assuming initial values of x_1, x_2 and x_3 as 0

Iteration 1:

$$x_1^{(1)} = \frac{b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)}}{a_{11}} = \frac{7.85 + 0.1 \times 0 + 0.2 \times 0}{3} = 2.61667$$

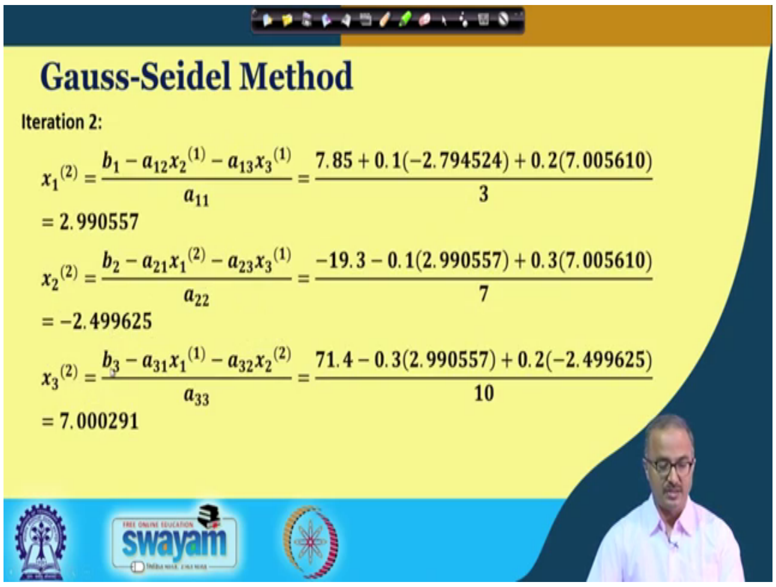
$$x_2^{(1)} = \frac{b_2 - a_{21}x_1^{(1)} - a_{23}x_3^{(0)}}{a_{22}} = \frac{-19.3 - 0.1(2.61667) + 0}{7} = -2.794524$$

$$x_3^{(1)} = \frac{b_3 - a_{31}x_1^{(0)} - a_{32}x_2^{(1)}}{a_{33}} = \frac{71.4 - 0.3(2.61667) + 0.2(-2.794524)}{10}$$

$$= 7.005610$$

So, here we have that we are getting the x 1 x 2 x 3 values for after the first iteration.

(Refer Slide Time: 13:13)



Gauss-Seidel Method

Iteration 2:

$$x_1^{(2)} = \frac{b_1 - a_{12}x_2^{(1)} - a_{13}x_3^{(1)}}{a_{11}} = \frac{7.85 + 0.1(-2.794524) + 0.2(7.005610)}{3}$$

$$= 2.990557$$

$$x_2^{(2)} = \frac{b_2 - a_{21}x_1^{(2)} - a_{23}x_3^{(1)}}{a_{22}} = \frac{-19.3 - 0.1(2.990557) + 0.3(7.005610)}{7}$$

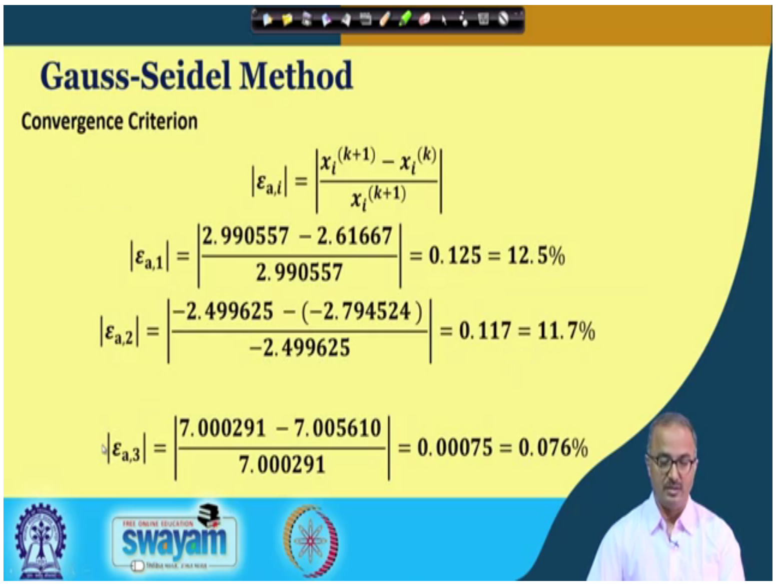
$$= -2.499625$$

$$x_3^{(2)} = \frac{b_3 - a_{31}x_1^{(1)} - a_{32}x_2^{(2)}}{a_{33}} = \frac{71.4 - 0.3(2.990557) + 0.2(-2.499625)}{10}$$

$$= 7.000291$$

And these are the values of x 1 x 2 x 3 for the of the second iteration.

(Refer Slide Time: 13:19)



Gauss-Seidel Method

Convergence Criterion

$$|\varepsilon_{a,i}| = \left|\frac{x_i^{(k+1)} - x_i^{(k)}}{x_i^{(k+1)}}\right|$$

$$|\varepsilon_{a,1}| = \left|\frac{2.990557 - 2.61667}{2.990557}\right| = 0.125 = 12.5\%$$

$$|\varepsilon_{a,2}| = \left|\frac{-2.499625 - (-2.794524)}{-2.499625}\right| = 0.117 = 11.7\%$$

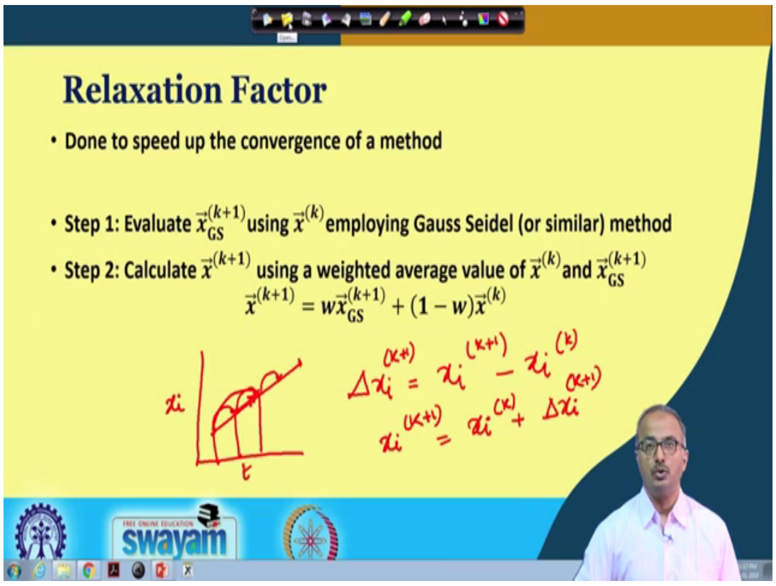
$$|\varepsilon_{a,3}| = \left|\frac{7.000291 - 7.005610}{7.000291}\right| = 0.00075 = 0.076\%$$

Now, you can see that in this case that this rate of convergence of the x 3 is the largest and all of them if you go on at repeating this thing we will find that x 1, x 2 will also go in at a faster convergence rate then in case of the Jacobi. Now what happened that because the Gauss Seidel method was showing a better convergence characteristics than the Jacobi method. So, and it was also easy to implement in the computer. So, we found that the Jacobi method was almost abundant and all this Gauss Seidel became the popular method.

But what happen later that when the parallel computing came this Gauss Seidel method was seemed to be not that efficient because this is a sequential method, you cannot get x 3 or x 4, unless you have the value of x 1 x 2 ok.

So, its sequential method , but when parallel computing came into picture then what we found that by if we use Jacobi method, then each of these equations can be given to each of the CPUs and you can get the values of all x 1 x 2 up to x n together ok. So, due to this parallel computing this Jacobi method again got revived. So, you are going from a sequential calculation to a parallel computation ok. So, that is why you find at present day when if you are using go for parallelization and if you are parallel computers then you will like to go for the Jacobi kind of method and if you have a sequential thing then you go for the Gauss Seidel kind of method.

(Refer Slide Time: 15:08)



Relaxation Factor

- Done to speed up the convergence of a method
- Step 1: Evaluate $\vec{x}_{GS}^{(k+1)}$ using $\vec{x}^{(k)}$ employing Gauss Seidel (or similar) method
- Step 2: Calculate $\vec{x}^{(k+1)}$ using a weighted average value of $\vec{x}^{(k)}$ and $\vec{x}_{GS}^{(k+1)}$

$$\vec{x}^{(k+1)} = w\vec{x}_{GS}^{(k+1)} + (1-w)\vec{x}^{(k)}$$

$$\Delta x_i^{(k+1)} = x_i^{(k+1)} - x_i^{(k)}$$

$$x_i^{(k+1)} = x_i^{(k)} + \Delta x_i^{(k+1)}$$

Now, after we have learnt a some of these equations of matrix methods to solve the equations, another important thing is the relaxation factor. Now the concept of relaxational factor comes in order to speed up the convergence. Now you see that many a times that if we know the direction of my solution if you can find out especially if we are going for a linear solution.

So, if we know the direction of our solution, then we should be able to take a larger step size ok; that means we can be more adventurous in going leaping towards the exact solution. So, whatever updation we are getting that we can still update further arbitrarily to go reach the solution. And what we define that we always take the difference between the values of the previous and the present time step.

So, this is the kind of error you can see. So, if I look at the error basically all these methods are depending on the error. We are putting some error for and this error is taken as the value of the x I at k plus 1th level and value of the x i at the kth level. Now you will see that when once we do this now; that means, x i k plus 1 is equal to x i k plus this error which we are finding at the k plus 1th iteration level. Now if we know the direction of our solution suppose I put this in the graphical form suppose we are solving for the xi versus t and suppose this is the solution we know we have to reach here from here.

So, suppose by our step size we are doing from here to here we are reaching 1 by 1 ok, but now we know that because it is almost linear what we can do, that suppose even if we go to this point we can take a push this point of further on this side and that we do by taking some multiplying factor so, that I can push this solution further.

So, that I can after this same iteration I can go directly to this instead of going from this to this ok. So, this is the principle. So, we are using some arbitrary factor what we call the relaxation factor, which is kind of pushing my solution towards the actual solution ok. So, these kind of leaping it is take a jump from the exact value which we have got from the equations ok.

Now, this is the case when we are we are talking about the linear solution. But this thing cannot be applied if we are going for a non-linear problem there you find that if there is slight changes here and there it can cause a sudden we can move away from the solution. So, whatever delta x or the corrections we are getting at a given iteration level we may find that, applying that whole correction may take us away from the solution or cause divergence.

So, in that case we have to be more restrictive ok. So, instead of taking the whole jump to that particular point what we shall do? We shall take a small jump; that means, whatever corrections is being offered by solving the equations we shall take less than that correction. So, we are getting more conservative this is something like this, suppose you are walking in a particulars room with almost blindfolded and you know that the room has many furniture scattered here and there.

Now, before you have been blindfolded, you know that perhaps that you have seen the place where you have to reach destination, but once your blindfolded you do not know where the all the furnitures are scattered, then what you will do you will be bit more cautious. So, even if you know the direction of your solution, you do not want to jump around too much here and there because you know you are going to heat some kind some furniture and you may not be able to reach that. So, you will take smaller steps and slowly and slowly you will move towards the destination and once you know that you have you are reaching their destination, then you can take a faster step size so, that you can reach the destination quickly ok.

But on the other hand suppose you have not no barriers on your path, you know that if you then what you will do the decide that you can take a longer step or faster step to reach your destination. So, that is the way we decide that when we need to be conservative or when we can be more adventurous or border to choose this particular relaxational factor.

So, you see that in this example, we if we take the standard Gauss Seidel method to give us the solution. It need not be Gauss Seidel it can be any other method it can be gauss elimination or any other method. So, you see that when we find that this kind of things are there so, we first evaluate the solution from the Gauss Seidel method and we this is the solution we have for the previous iteration level ok.

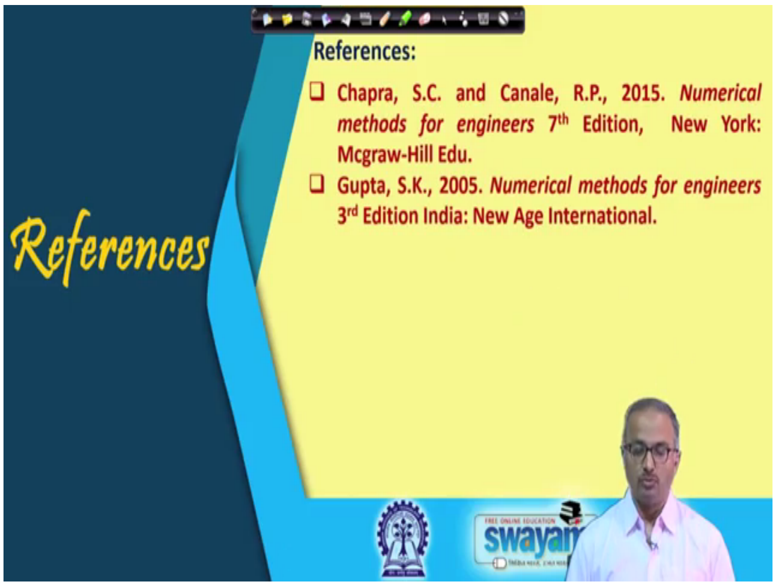
Now, what we do that, to make the exact value of the particular unknown what we do that instead of the Gauss Seidel what we take a relaxational factor of w we take the w factor. We take here and with w factor we put with this Gauss Seidel method and we take with the value we have obtained in our earlier iteration, now you see that when w is equal to 0; what it means is this we are not taking action on the present value. So, this whole weightage is being given to the previous value and when we take this w equal to 1; that means, that we are not giving in any weightage to the previous value. We are putting whole weightage to the 1 we have obtained from the particular solution with any given method.

But if we take the w value between 0 and 1 that means, we are giving weightage to both the present value we have obtained and the value we have taken from the previous iteration ok. Now you see that depending on the value of the w we can have over relaxation or under relaxation, over relaxation means generally if we take the correction factor more than 1, then we say it is over relaxation ok. And if we take the correction factor less than 1, then we say we are going for under relaxation. Over relaxation is generally prescribed if the problem is going towards linear problem and we go for under relaxation if the problem is more non-linear ok.

So, depending ah so, this is the way we are able to kind of artificially we try to ensure convergence of the particular set of equations ok. So, this is the significance of the relaxation factor the choice of the relaxation factor is arbitrary and you need to do some kind of experimentations while you are solving the equations, to arrive at the right value of the iteration relaxation factor. And what also happens is this as you gone on iterating, you will find that you may have to tune this relaxation factor to get the convergence.

One more thing sometimes researchers do is this, they find the best solution kind of by optimization of the relaxation factor. So, you see that how much error you are getting in between the previous iteration the present iteration and you try to minimize this error by optimizing the value of the relaxation factor. So, it the choice of this factor may be arbitrarily or maybe done through some optimization depending on the problem at hand. So, that is how you see that this particular factor assumes a great importance in or the solution, numerical solution of these equations.

(Refer Slide Time: 24:05)



References

References:

- Chapra, S.C. and Canale, R.P., 2015. *Numerical methods for engineers* 7th Edition, New York: Mcgraw-Hill Edu.
- Gupta, S.K., 2005. *Numerical methods for engineers* 3rd Edition India: New Age International.

So, more on this you can find in these references.

Thank you.