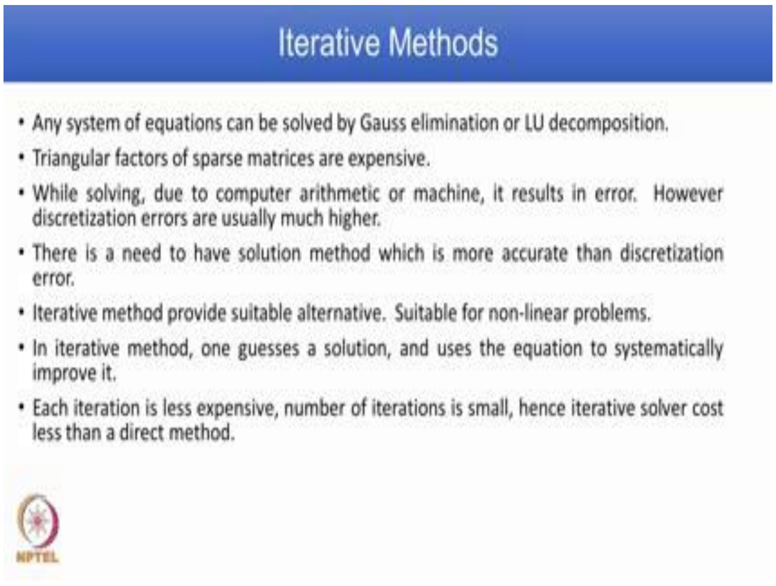
Foundation of Computational Fluid Dynamics

Dr. S. Vengadesan
Department of Applied Mechanics
Indian Institute of Technology, Madras

Lecture – 36

(Refer Slide Time: 00:34)



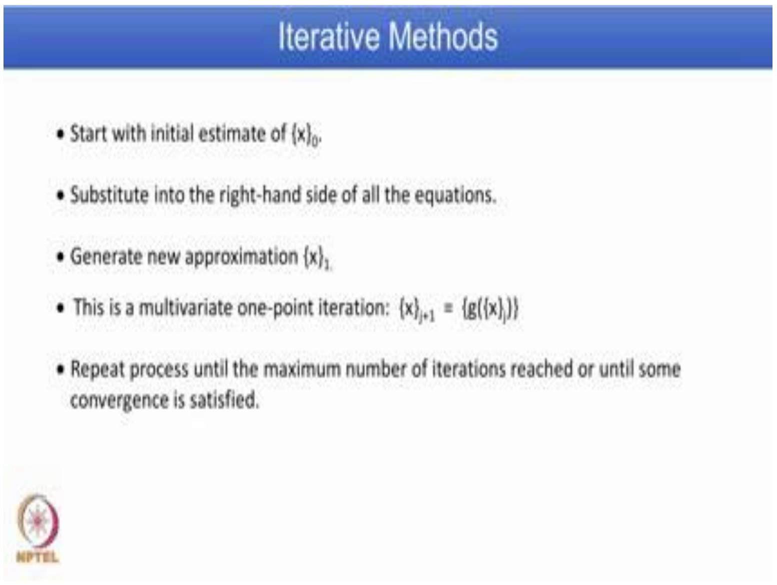
Iterative Methods

- Any system of equations can be solved by Gauss elimination or LU decomposition.
- Triangular factors of sparse matrices are expensive.
- While solving, due to computer arithmetic or machine, it results in error. However discretization errors are usually much higher.
- There is a need to have solution method which is more accurate than discretization error.
- Iterative method provide suitable alternative. Suitable for non-linear problems.
- In iterative method, one guesses a solution, and uses the equation to systematically improve it.
- Each iteration is less expensive, number of iterations is small, hence iterative solver cost less than a direct method.

Welcome again to the course on CFD. Today you are on module four of this week. So far, we have seen direct methods in detail. Today, we are going to see a new procedure what is known as the iterative methods. Any linear system can be solve by Gauss elimination, L U decomposition procedure. The question is if the matrix becomes complicated whether the gauss elimination or L U decomposition procedure is economical or the whether it lead to some erroneous solution in such situation what do we do. The second question whether it is computationally or in terms of time wise economical to do direct methods or is there a way that we can go by iterative way we can get the solution.

There are other issues for example, triangular factors of a sparse matrix. So, sparse matrix is you are not able to identify matrix into some form. It has elements distributed throughout. It has values distributed here and there in the particular coefficient matrix, such matrix is called sparse matrix, and triangular factorization of the sparse matrix is usually very expensive. While solving we have error due to computer arithmetic handling of numbers rounding of errors or there is also error due to discretization. And discretization errors are usually much higher when compared to machine rounding of error. We should have method which is more accurate than discretization error an iterative methods provides a suitable alternative, and it is also good for problems involving non-linear. So, it is suitable for non-linear problems. In iterative methods, one initial makes a guess of the solution then progressively improve and get the accurate solutions. Each iteration is less expensive, number of iterations required happens to be small, hence over all computation time requirement is also less when compared to direct method.

(Refer Slide Time: 02:53)

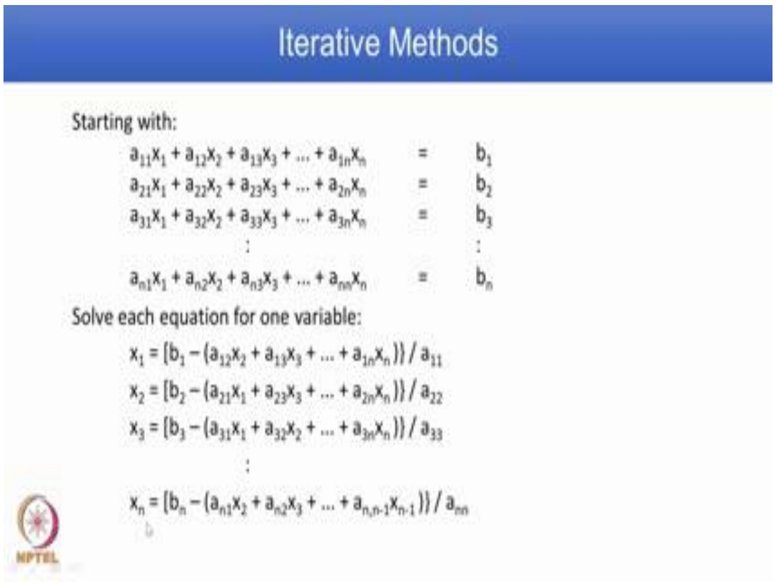


Iterative Methods

- Start with initial estimate of $\{x\}_0$.
- Substitute into the right-hand side of all the equations.
- Generate new approximation $\{x\}_1$
- This is a multivariate one-point iteration: $\{x\}_{j+1} = \{g(\{x\}_j)\}$
- Repeat process until the maximum number of iterations reached or until some convergence is satisfied.

So, we start with initial guess that is x at to zero, and substitute into the right hand side of all the equation generate new approximation value and we call that x upon. By this way we go on repeat and this is the multivariate one point iteration. So, in mathematically x at j plus 1 is equal to x at j by some function g that is way it is called multivariate one point iteration. And repeat this process until you have two conditions either you specify maximum more of iteration or you specify some convergence limit. We learnt to about convergence limit that one such criteria under the convergence limit for example. If that different between present value and old value take the absolute and you specify some limit, if that absolute value of difference is less than the specified value then you can terminate the iteration this is one such convergence criteria. So, you repeat this procedure either specifying convergence criteria or you forcibly say we will to the iterations only for specs number of iteration say 500, 1000 whatever be the error accumulation. Slowly as the calculation proceeds that will also come down.

(Refer Slide Time: 04:25)



Iterative Methods

Starting with:

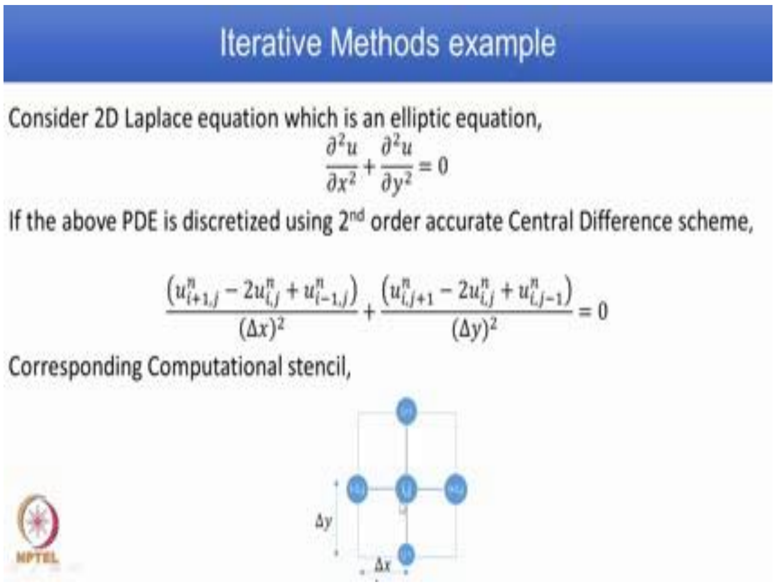
$$
\begin{array}{lcl}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n & = & b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n & = & b_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \dots + a_{3n}x_n & = & b_3 \\
\vdots & & \vdots \\
a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n & = & b_n
\end{array}
$$

Solve each equation for one variable:

$$
\begin{array}{l}
x_1 = \{b_1 - (a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n)\} / a_{11} \\
x_2 = \{b_2 - (a_{21}x_1 + a_{23}x_3 + \dots + a_{2n}x_n)\} / a_{22} \\
x_3 = \{b_3 - (a_{31}x_1 + a_{32}x_2 + \dots + a_{3n}x_n)\} / a_{33} \\
\vdots \\
x_n = \{b_n - (a_{n1}x_2 + a_{n2}x_3 + \dots + a_{n,n-1}x_{n-1})\} / a_{nn}
\end{array}
$$

So, start with the list of linear equations as we did before a 1 1 x 1 plus a 1 to x 2 plus a 1 3 x 3 all the way and then b one on the right side. Similarly for all the rows we get ask row a n 1 x 1 to all the way up to a n n x n equal to b n. Now what we to is start with initial guess value. So, x 1 is an to be determine and all other terms of the equation for brought to the right side as shown here. Now in these values for x 2 x 3 all the way up to x n or from the guess value or from the previous iterated value. So, you get approximately values for x 1. Now we go to x 2 and all the way up to x n.

(Refer Slide Time: 05:24)



Iterative Methods example

Consider 2D Laplace equation which is an elliptic equation,

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

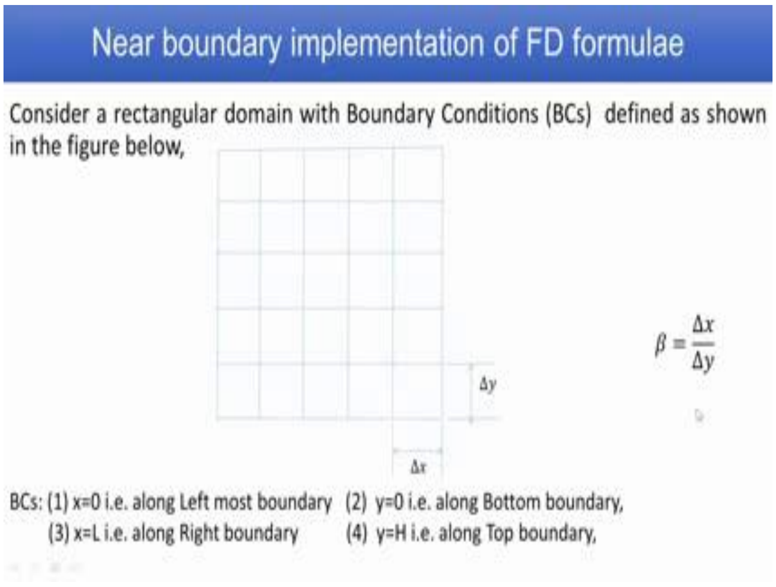
If the above PDE is discretized using 2nd order accurate Central Difference scheme,

$$\frac{\left(u^n_{i+1,j} - 2u^n_{i,j} + u^n_{i-1,j}\right)}{(\Delta x)^2} + \frac{\left(u^n_{i,j+1} - 2u^n_{i,j} + u^n_{i,j-1}\right)}{(\Delta y)^2} = 0$$

Corresponding Computational stencil,

We will explain different iterative procedure by taking one example problem and that is shown here as a 2 D Laplace equation dou square u by dou x square plus dou square u by is equal to zero. And this is an elliptic equation. If that example equation that we have taken is discretized using second order accurate central difference scheme then we get finite differenced form of that equation on that is shown here. So, let me read that term u i plus 1 j n minus 2 u i j n plus u i minus 1 j n divide by delta x square. So, this is the central difference scheme which is apply to the first time in this equation dou square u by dou x square and we use two subscript one i for x second j for y and as usual superscript n is used here for iteration level. We write a similar term for the second derivative in the y direction and that is what is shown here. Now j is given index j plus 1 j and j minus 1. A corresponding computational molecule is shown here. So, i and j is the point of interest, and we consider values from neighbors i minus 1 comma j, i plus 1 comma j in the x direction; similarly the y direction, you have i j plus 1 an j minus 1. So, these four neighboring notes contributes for the solution at node i comma j.

(Refer Slide Time: 07:13)



Near boundary implementation of FD formulae

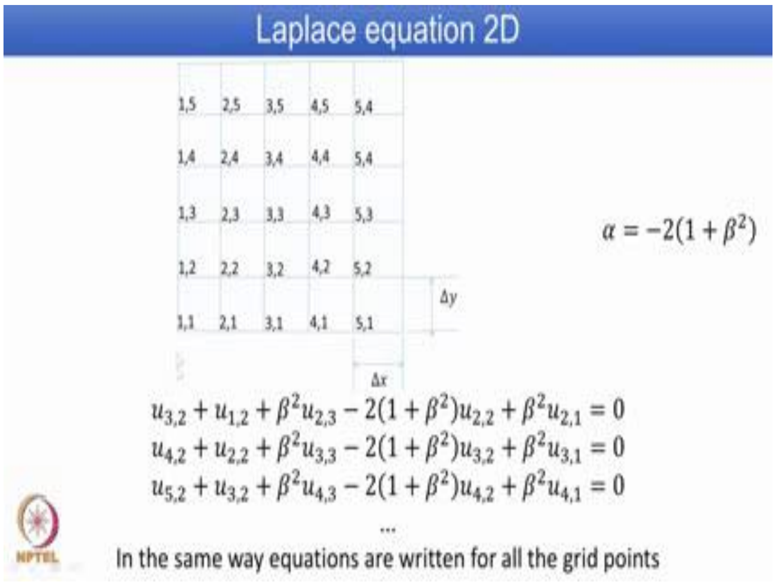
Consider a rectangular domain with Boundary Conditions (BCs) defined as shown in the figure below,

$\beta = \frac{\Delta x}{\Delta y}$

BCs: (1) x=0 i.e. along Left most boundary (2) y=0 i.e. along Bottom boundary,
(3) x=L i.e. along Right boundary (4) y=H i.e. along Top boundary,

For the example problem, the computational domain with the mesh arrangement is shown, delta x and delta y are the respective spatial different between two mesh points in x direction as well as in y direction. And we define a new term beta which is equal to delta x by delta y and we apply boundary condition along x as well as along y.

(Refer Slide Time: 07:43)



Laplace equation 2D

$\alpha = -2(1+\beta^2)$

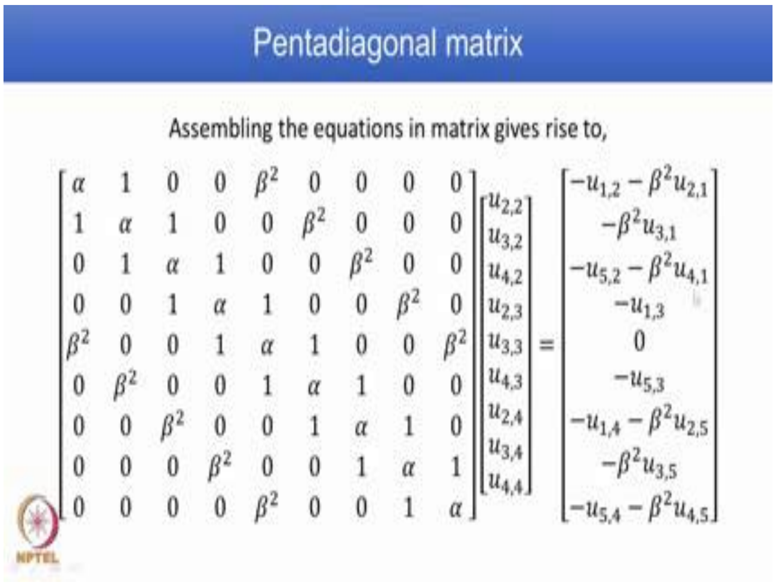
$$u_{3,2} + u_{1,2} + \beta^2 u_{2,3} - 2(1+\beta^2)u_{2,2} + \beta^2 u_{2,1} = 0$$
$$u_{4,2} + u_{2,2} + \beta^2 u_{3,3} - 2(1+\beta^2)u_{3,2} + \beta^2 u_{3,1} = 0$$
$$u_{5,2} + u_{3,2} + \beta^2 u_{4,3} - 2(1+\beta^2)u_{4,2} + \beta^2 u_{4,1} = 0$$
$$\ldots$$

In the same way equations are written for all the grid points

With the specific of nodal points, the same computational domain is repeated here. So, 1 comma 1, 2 comma 1, 3 comma 1, 4 comma 1, 5 comma 1 for example, is along the first row for first in y direction, then we extend this for the remaining nodes and it is marked here. We define another term alpha for equal two minus two beta square now we apply at the discretized equation what we had before what we are the n before that is set second order central difference scheme dou square by dou y square use this definition of alpha and beta for this miss arrangement you are apply for each nodal point.

For example 1 comma 1, 2 comma 1 this may be coming along the boundary are you take a second row. So, 1 comma 2 for again on left side boundary 5 comma 2 is again near the right side boundary. So, we have apply, for example, for the node 2 comma 2 discretized equation then we get the first equation as shown here. So, 2 comma 2 node is influenced by neighbouring notes that 1 comma 2, 3 comma 2, 2 comma 2, 2 comma 3 their respectively i minus 1 j i plus 1 j i j plus 1 un i j minus 1. So, you get corresponding equation written here. We extend this procedure for all the notes. So, it is 2 comma 2 is their 3 comma 2 and 4 comma 2. In the same way equation are written for all the nodal points.

(Refer Slide Time: 09:32)



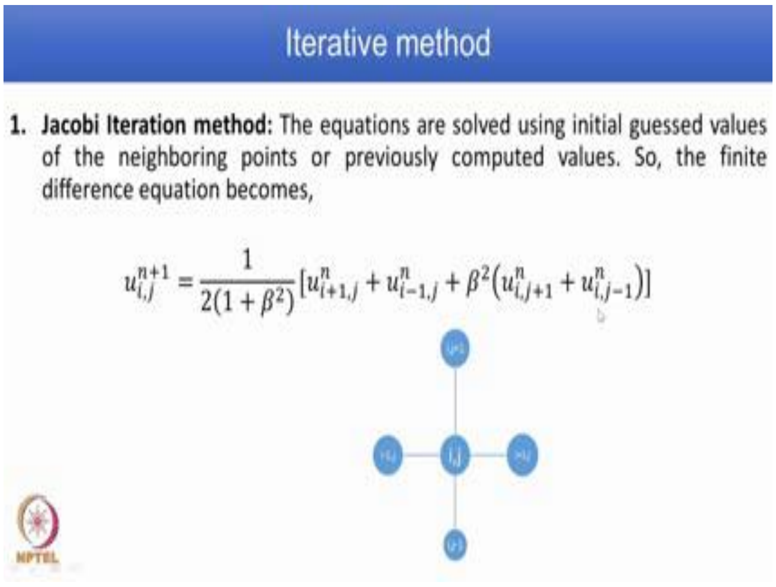
Pentadiagonal matrix

Assembling the equations in matrix gives rise to,

$$\begin{bmatrix} \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 & 0 \\ 1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 & 0 \\ 0 & 1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 & 0 \\ 0 & 0 & 1 & \alpha & 1 & 0 & 0 & \beta^2 & 0 \\ \beta^2 & 0 & 0 & 1 & \alpha & 1 & 0 & 0 & \beta^2 \\ 0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 & 0 & 0 \\ 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 & 0 \\ 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha & 1 \\ 0 & 0 & 0 & 0 & \beta^2 & 0 & 0 & 1 & \alpha \end{bmatrix} \begin{bmatrix} u_{2,2} \\ u_{3,2} \\ u_{4,2} \\ u_{2,3} \\ u_{3,3} \\ u_{4,3} \\ u_{2,4} \\ u_{3,4} \\ u_{4,4} \end{bmatrix} = \begin{bmatrix} -u_{1,2} - \beta^2 u_{2,1} \\ -\beta^2 u_{3,1} \\ -u_{5,2} - \beta^2 u_{4,1} \\ -u_{1,3} \\ 0 \\ -u_{5,3} \\ -u_{1,4} - \beta^2 u_{2,5} \\ -\beta^2 u_{3,5} \\ -u_{5,4} - \beta^2 u_{4,5} \end{bmatrix}$$

Now, if you put all those equation in the form of matrix an you structure as shown here. Now in this alpha, we already define beta also we have define those alpha and beta or definitions are for convenience. So, that discretized equation and the matrix are return elegantly. If you look at in this matrix then along the diagonal you have alpha and immediately above super diagonal immediately below sub diagonal then there are two zeros entry then you have beta square that is also appearing parlor to the to the diagonal on the upper side as well as on the lower side. So, this is like one form offers for matrix an all the unknown u vector are given then known coefficient value is given on rights side.

(Refer Slide Time: 10:30)



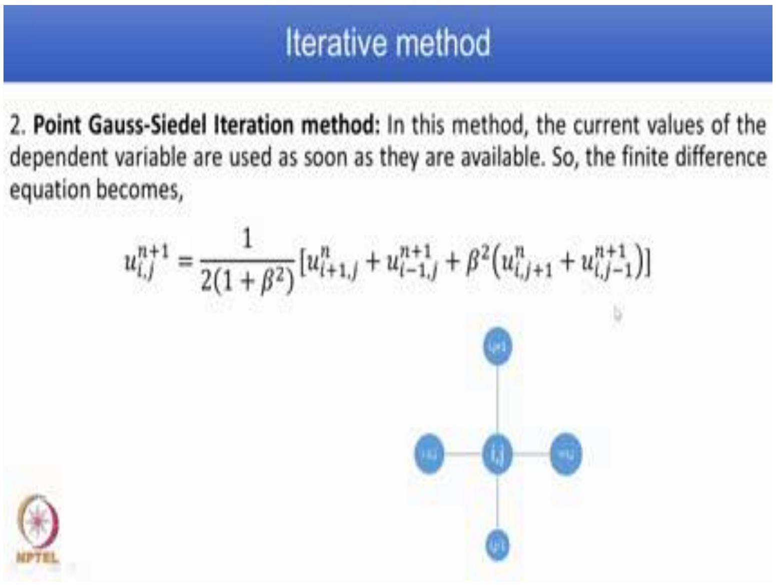
Iterative method

1. **Jacobi Iteration method:** The equations are solved using initial guessed values of the neighboring points or previously computed values. So, the finite difference equation becomes,

$$u_{i,j}^{n+1} = \frac{1}{2(1+\beta^2)}\left[u_{i+1,j}^{n} + u_{i-1,j}^{n} + \beta^2\left(u_{i,j+1}^{n} + u_{i,j-1}^{n}\right)\right]$$

You will explain different iterative procedure for that matrix and that example problems first one is Jacobi iteration method in this the equation are solved using initial guest values of the neighbouring points are from previously computed values. So, that final difference equation becomes as shown here the node of interest that is u i comma j current iteration value superscript is n plus 1. So, you i, j n plus 1 is the one that we determine at the node of interest i, j and all the neighbouring notes are taken to the right side there are either determined from the previous level are known from the guess value and that is again molecule.

(Refer Slide Time: 11:31)



Iterative method

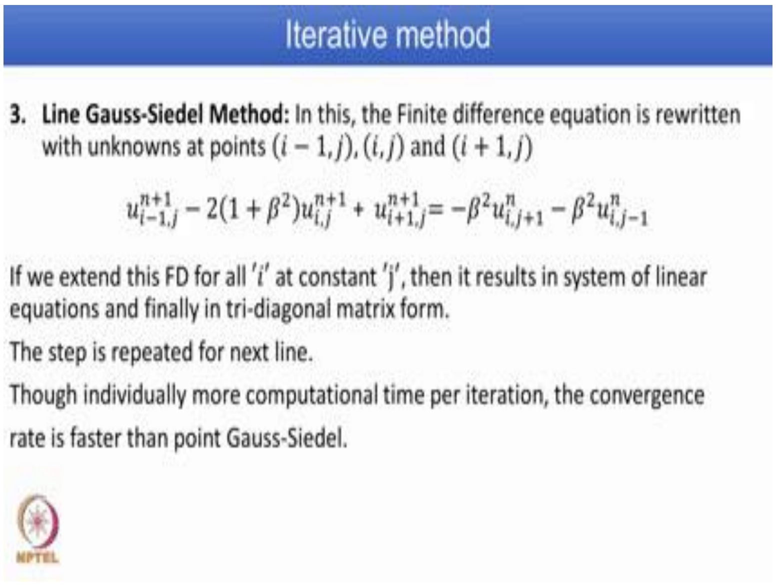
2. **Point Gauss-Siedel Iteration method:** In this method, the current values of the dependent variable are used as soon as they are available. So, the finite difference equation becomes,

$$u_{i,j}^{n+1} = \frac{1}{2(1+\beta^2)}[u_{i+1,j}^{n} + u_{i-1,j}^{n+1} + \beta^2(u_{i,j+1}^{n} + u_{i,j-1}^{n+1})]$$

So, we have rewritten the form of that discretized equation when you follow Jacobi iteration method the next one is a point Gauss-Seidel iteration method in this method the current values of the dependent variable are used as soon as they are immediately available. So, the discretized form his return for point Gauss-Seidel iteration as shown here though it looks similar you to pay attention to particular term that is u i minus 1 j with the super script n plus 1 an u i j minus 1 with the super script n plus 1. So, in the computational molecule, if you look at i comma j is the point of interest u i minus 1 is on the left side and u i j minus 1 is at the bottom values at these two nodes are known from just immediately completed iteration procedure. So, they can be immediately use behave their available that way the nodal value i comma j is more realistic go be learn from this discretization i comma j value is influenced by neighbouring nodes.

So, if you take very close value then the solution can be obtain faster an in this procedure values at node i minus 1 comma j and i j minus 1 or known from just completed iteration. So, why not use it and that is idea of point Gauss-Seidel iteration method and we are going from point to point that is why the word point is used. So, discretized equation written for point Gauss-Seidel iteration procedure has values immediately found included that is why you is super script n plus 1 for left side node as well as node at the bottom.

(Refer Slide Time: 13:32)



Iterative method

3. **Line Gauss-Siedel Method:** In this, the Finite difference equation is rewritten with unknowns at points $(i-1,j)$, (i,j) and $(i+1,j)$

$$u_{i-1,j}^{n+1} - 2(1+\beta^2)u_{i,j}^{n+1} + u_{i+1,j}^{n+1} = -\beta^2 u_{i,j+1}^{n} - \beta^2 u_{i,j-1}^{n}$$

If we extend this FD for all 'i' at constant 'j', then it results in system of linear equations and finally in tri-diagonal matrix form.

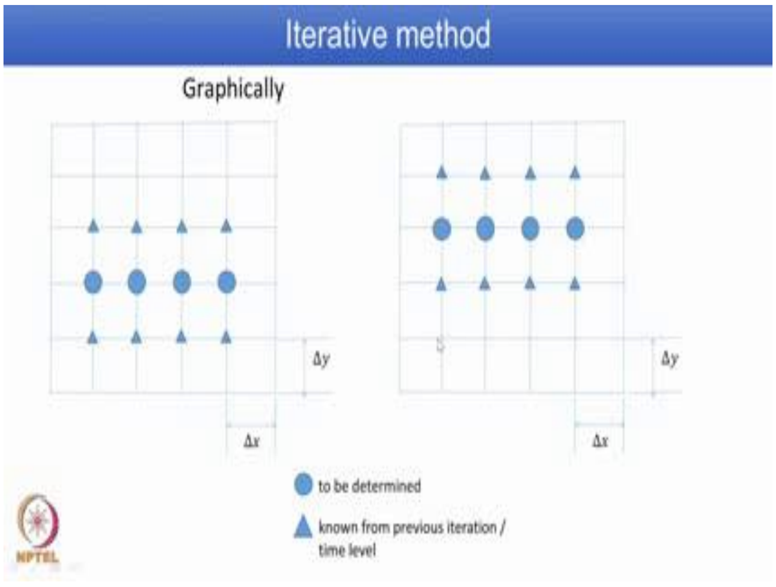
The step is repeated for next line.

Though individually more computational time per iteration, the convergence rate is faster than point Gauss-Siedel.

Next is Line Gauss Seidel method; in this method, the finite difference equation in rewritten with the unknown points at i minus 1 j i comma j and i plus 1 and you can see the equation here. So, i minus 1 j then we define one plus beta square i comma j than you at i plus j n plus 1 equal to values at j plus 1 and j minus 1 here this is a long the line because i minus 1 is on the left side i comma j is a point of interest and i plus 1 is on the right side. So, we identify notes along that particular line and they all have to be inverted together and that is the idea behind Line Gauss-Seidel method.

If we extend this procedure, for all i at particular j, then it will result in what is in system of linear equation and it will have a finally, tridiagonal matrix form, because, if you look at this equation again i comma j is the point of interest on you have one on the left side one the right side, and extend this for all i an j for the particular j then it will result in tri diagonal matrix form and we learned TDMA procedure in the previous class. At this step is repeated for next line and you finally get one full matrix. Though individually it as appears to be taking more computational time per iteration, the convergence rate is faster than that is required for Gauss-Seidel method.

(Refer Slide Time: 15:24)

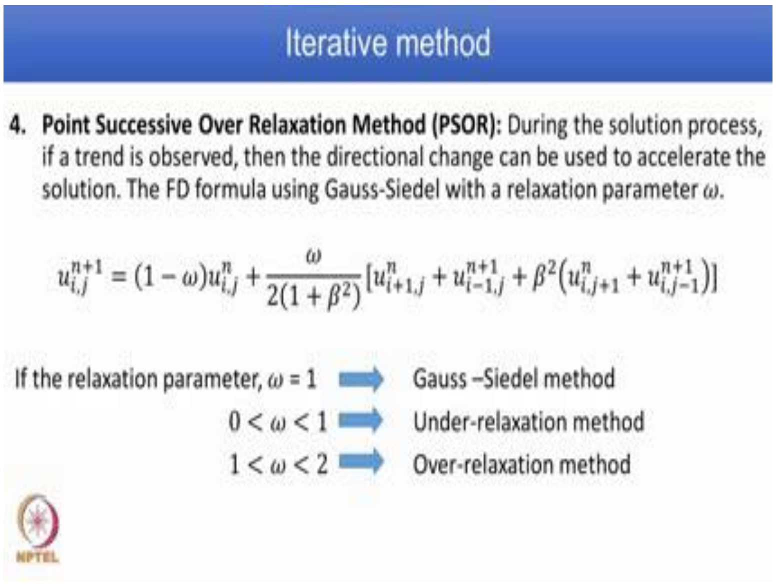


Iterative method

Graphically

Graphically the line iteration procedure is shown. So, initially you have values along this particular j node not known previously iterative values as shown by the triangle or from the previous known previous time level or previous iterative values as shown again with a triangle. So, along this particular j and u setup equation, when you solve equation for the particular j then this nodes values are coming to known and they are used for the next level as shown here. So, now for the same line j it becomes triangle that we need is a known value this triangle becomes unknown value along the same j and then we have previously iterated value just above.

(Refer Slide Time: 16:32)



Iterative method

4. **Point Successive Over Relaxation Method (PSOR):** During the solution process, if a trend is observed, then the directional change can be used to accelerate the solution. The FD formula using Gauss-Siedel with a relaxation parameter ω.

$$u_{i,j}^{n+1} = (1-\omega)u_{i,j}^{n} + \frac{\omega}{2(1+\beta^2)}[u_{i+1,j}^{n} + u_{i-1,j}^{n+1} + \beta^2(u_{i,j+1}^{n} + u_{i,j-1}^{n+1})]$$

If the relaxation parameter, $\omega = 1$ ➡ Gauss –Siedel method

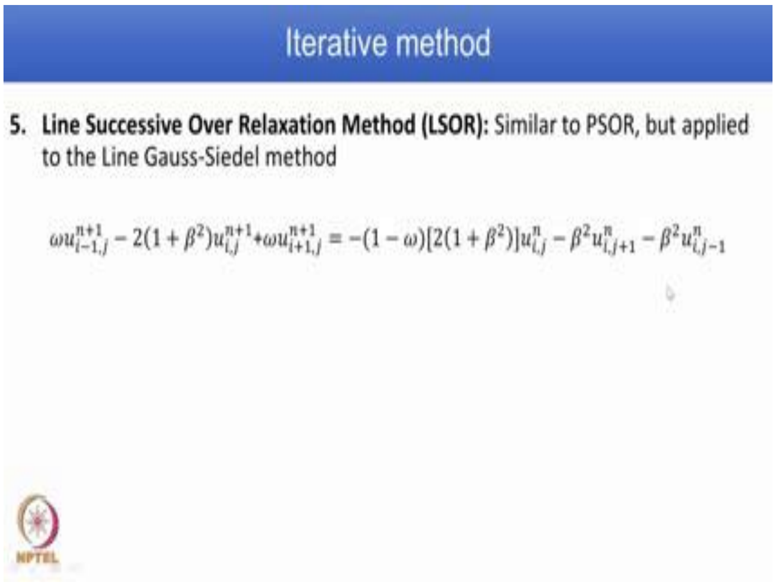
$0 < \omega < 1$ ➡ Under-relaxation method

$1 < \omega < 2$ ➡ Over-relaxation method

So, this way you repeat this procedure and you get complete solution next we see a procedure called point successive over relaxation method, otherwise PSOR. So, we know the solution is progressing and it is likely to reach the solution in the same direction, and this is what is known as the directional change. So if trend is observed why not we accelerate and that is idea and that is implemented by a procedure called over relaxation method. So, u i comma j n plus 1 is to be determine we now introduced factor called omega an one minus omega. So, this is kind of fraction. So, omega as one then it the contribution of u i comma j n is actually o. If omega is less than 1 is say for example, 0.2 then becomes 0.8, the 1 minus omega becomes 0.8.

So, we take 80 percent weightage of previous level value for the same node i comma j that is a meaning of the term one minus omega u i comma j n and the remaining terms. Again we introduced omega for taking wait age of neighbouring notes if the relaxation parameter omega is equal to one then it will result in Gauss-Seidel method which we have just now seen. If the omega has value between zero and one the method is called under relaxation method if it is above 1 then the method is called over relaxation method. Now what is shown here is only a sample example equation, you can write this for auxiliary equation as well. And you need to define omega differently for different equation different variable will behave differently when you iterate.

(Refer Slide Time: 18:37)



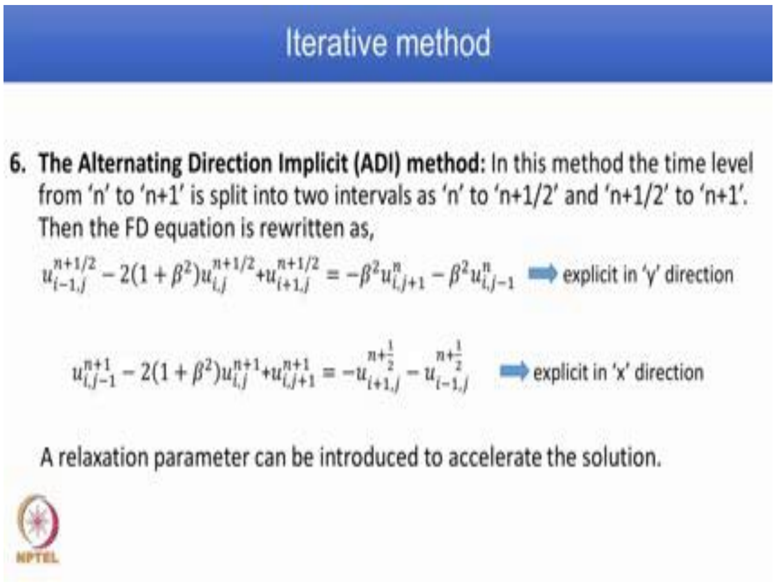
Iterative method

5. **Line Successive Over Relaxation Method (LSOR):** Similar to PSOR, but applied to the Line Gauss-Siedel method

$$\omega u_{i-1,j}^{n+1} - 2(1+\beta^2)u_{i,j}^{n+1} + \omega u_{i+1,j}^{n+1} = -(1-\omega)[2(1+\beta^2)]u_{i,j}^{n} - \beta^2 u_{i,j+1}^{n} - \beta^2 u_{i,j-1}^{n}$$

Next is a line successive over relaxation method is the same as p s o r only thing it is implemented for the line as a whole. So, we have omega i minus 1 j n plus 1 then for i plus 1 value extra and similarly for neighbouring nodes contribution.

(Refer Slide Time: 19:02)



Iterative method

6. **The Alternating Direction Implicit (ADI) method:** In this method the time level from 'n' to 'n+1' is split into two intervals as 'n' to 'n+1/2' and 'n+1/2' to 'n+1'. Then the FD equation is rewritten as,

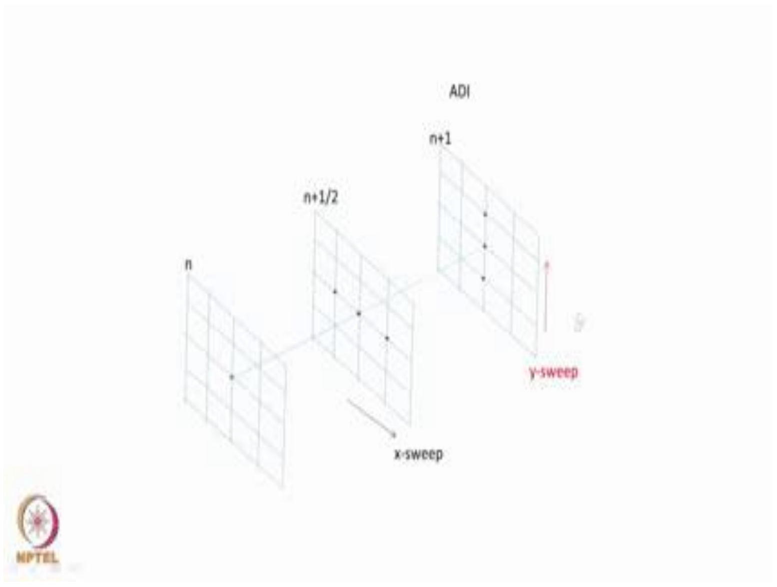
$$u_{i-1,j}^{n+1/2} - 2(1+\beta^2)u_{i,j}^{n+1/2} + u_{i+1,j}^{n+1/2} = -\beta^2 u_{i,j+1}^{n} - \beta^2 u_{i,j-1}^{n} \Rightarrow \text{explicit in 'y' direction}$$

$$u_{i,j-1}^{n+1} - 2(1+\beta^2)u_{i,j}^{n+1} + u_{i,j+1}^{n+1} = -u_{i+1,j}^{n+\frac{1}{2}} - u_{i-1,j}^{n+\frac{1}{2}} \Rightarrow \text{explicit in 'x' direction}$$

A relaxation parameter can be introduced to accelerate the solution.

There is another procedure what is known as the alternating direction implicit ADI method. In this method, we go from length to n plus 1 level in two steps that is n to n plus half and then n plus half to n plus 1. Now every time one set is given in implicit another set is given in explicit. The first step for example, it is implicit in x-direction and explicit in y-direction, accordingly you can rewrite the discretized equation. So, we get final discretized equation as shown here. The next stage n plus half value is known and it is explicit in x-direction. So, it is taken to other side and implicit in y direction that is taken on the left side. And you already learned this is usually applied for Penta diagonal matrix, and by applying this ADI procedure it gets reduced in to tridiagonal matrix and it is possible to apply TDMA procedure to get solution. It is also possible to introduced a relaxation parameter as we did in SOR procedure and accelerate the solution.

(Refer Slide Time: 20:33)



Now, ADI method is explain graphically as shown here. So, value at n, n plus half, you do to sweep in x direction and n plus 1 you to sweep in y direction. So, in this class, we have seen in detail, particularly iterative methods to explain different procedure, we took an example problem that is Laplace equation; we discretized by second order scheme, and we explain different iterative procedure for that example problem. We understood advantages. We also learned method of reaching faster by a procedure called over relaxation or under relaxation.

Thank you.