Foundation of Scientific Computing

Prof. T. K. Sengupta

Department of Aerospace Engineering

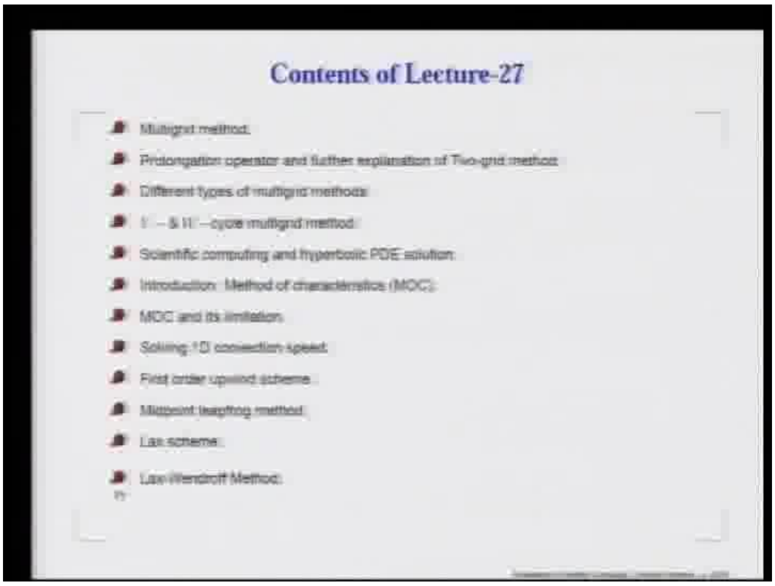
Indian Institute of Technology, Kanpur

Module No. # 01

Lecture No. # 27

We continue our discussion on Multigrid method in this lecture number 27. We see that there are various operations which are not done before. For example, we need to prolong our solution from a coarse grid to fine grid and vice versa; that is where we once again fall back upon the two-grid solution method and try to identify what will work. Once, we have done that, we have understood the basic elements of Multigrid method; we can talk about different types of Multigrid method.

(Refer Slide Time: 00:15)



Contents of Lecture-27

- Multigrid method.
- Prolongation operator and further explanation of Two-grid method.
- Different types of multigrid methods.
- V – & W –cycle multigrid method.
- Scientific computing and hyperbolic PDE solution.
- Introduction: Method of characteristics (MOC).
- MOC and its limitation.
- Solving 1D convection speed.
- First order upwind scheme.
- Midpoint leapfrog method.
- Lax scheme.
- Lax-Wendroff Method.

For example, the cycles that one employs in the Multigrid method in migrating from various levels of grid is given in terms of V cycle or W cycle and this is what we are going to talk about. We note that a Multigrid method, since it works on a host of grids it does put some restriction on the accuracy of the method. There are other methods which will not be able to talk about but, they are of more recent interest. But, this Multigrid method uses the spectral portrait of the iterative process that is why we have discussed it so far.

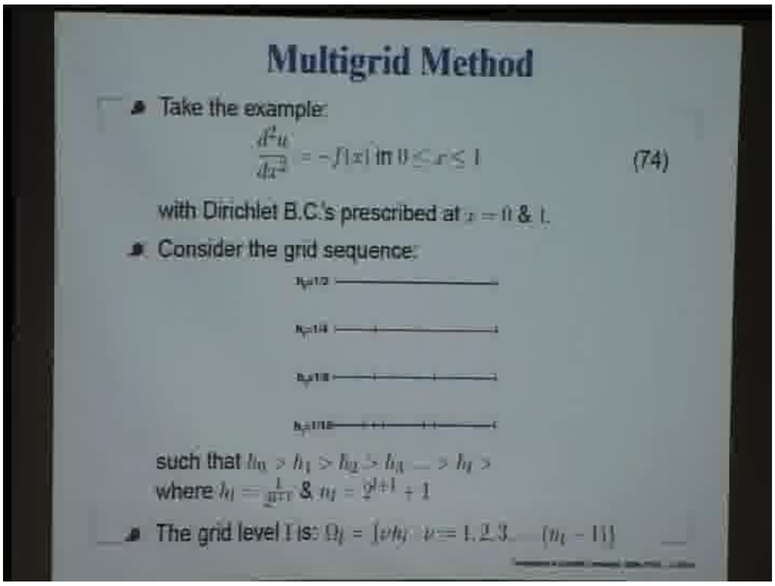
Having concluded briefly about this solution methods of electric equation, we come to considering solution methods for hyperbolic partial differential equations. We note that one of the most accurate methods is the method of characteristics or MOC. Unfortunately MOC has its own limitations which we identify; that it cannot handle mixed problems which are readily amenable to other discrete methods.

That is what we are trying to do and we try to solve the 1D convection equations. We show actually that the most accurate method for this 1D convection equation comes from a first order upwind scheme and then we try to show also that this could be solved by midpoint leap-frog method. This is essentially nothing but, the Richardson method that we have introduced for parabolic equation. We find that though this brings in a spurious mode but, both the modes have neutral stability.

So, there is a case for studying it; in fact, this is what generally used for in visit weather prediction codes. We also talk about other methods of solving hyperbolic equation namely the classic lax scheme and the Lax-Wendroff scheme.

Let us begin, again just a recap of what we started looking at. This is one of the method call the Multigrid method which is quite often used for solving problems. One of the better elements of this method is that you do not have to design it specifically for any particular problem; it is a methodology that can be adapted to any particular electric problems.

(Refer Slide Time: 04:02)



Multigrid Method

- Take the example:

$$\frac{d^2u}{dx^2} = -f(x) \text{ in } 0 \leq x \leq 1 \qquad (74)$$

with Dirichlet B.C.'s prescribed at $x = 0$ & 1.

- Consider the grid sequence:

$h_0=1/2$
$h_1=1/4$
$h_2=1/8$
$h_3=1/16$

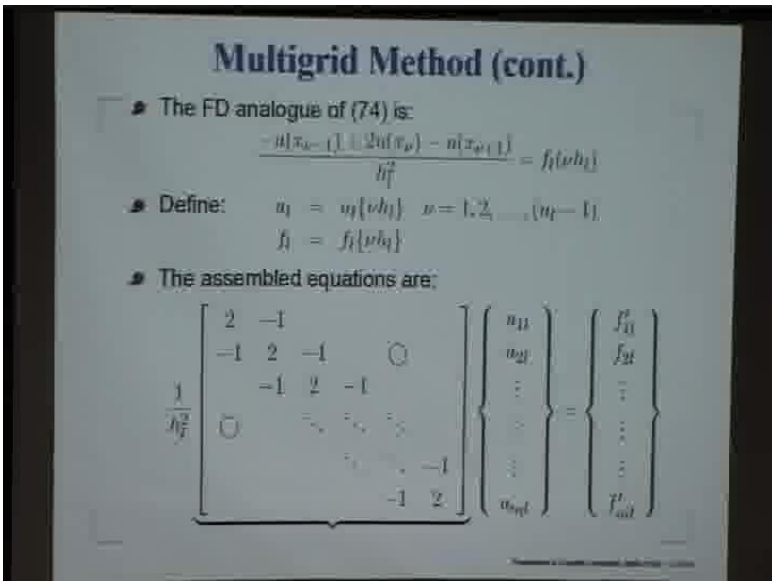
such that $h_0 > h_1 > h_2 > h_3 ... > h_l >$
where $h_l = \frac{1}{2^{l+1}}$ & $n_l = 2^{l+1} + 1$

- The grid level l is: $\Omega_l = \{\nu h_l : \nu = 1, 2, 3, ... (n_l - 1)\}$

We will just go through what we have done yesterday. We started looking at a model problem which admits the exact solution, despite that we noted that we may like to solve this problem in a sequence of grid and which are given by h 0, h 1, h 2, h 3 as the spacing. We have basically doubling the number of points at each level.

So, you find that grid spacing has this kind of a decreasing sequence, where the grid spacing is given in terms of the number of points that we have given by n l; that any particular grid level, we can collate all the points together and call the set as omega l, which will have this sequence of points separated by distance h l.

(Refer Slide Time: 05:01)



Multigrid Method (cont.)

- The FD analogue of (74) is:

$$\frac{-u(x_{\nu-1}) + 2u(x_{\nu}) - u(x_{\nu+1})}{h_l^2} = f_l(\nu h_l)$$

- Define:

$$\begin{aligned} u_l &= u_l(\nu h_l) \quad \nu = 1, 2, \ldots, (n_l - 1) \\ f_l &= f_l(\nu h_l) \end{aligned}$$

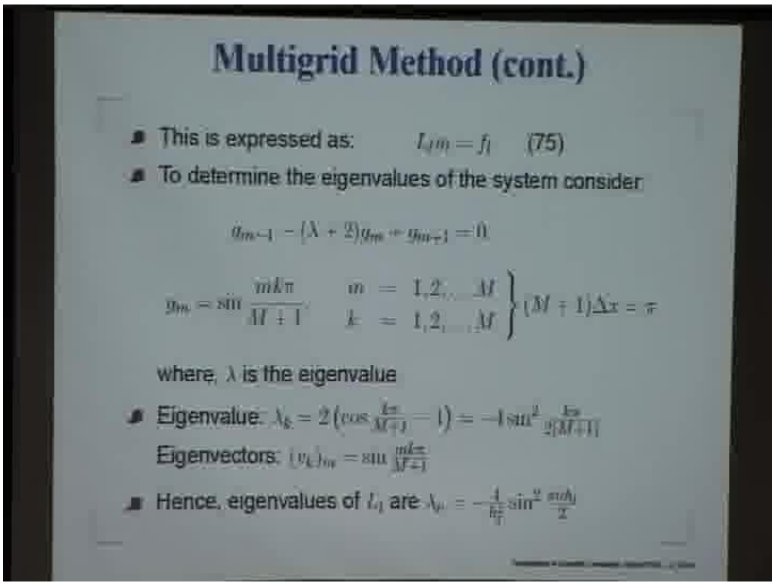
- The assembled equations are:

$$\frac{1}{h_l^2} \underbrace{\begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & \bigcirc & \\ & -1 & 2 & -1 & & \\ \bigcirc & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & -1 \\ & & & & -1 & 2 \end{bmatrix}}_{} \left\{ \begin{array}{c} u_{1l} \\ u_{2l} \\ \vdots \\ \vdots \\ \vdots \\ u_{nl} \end{array} \right\} = \left\{ \begin{array}{c} f'_{1l} \\ f_{2l} \\ \vdots \\ \vdots \\ \vdots \\ f'_{nl} \end{array} \right\}$$

Now, discretizing that primitive equation would lead to a matrix equation that was given here. You can clearly see as you have seen already, that it is basically a tri-diagonal equation; so, it could be solved exactly in a sense numerically. Please note that Dirichlet boundary condition actually alters the first and the last entry of this equation to accommodate boundary condition.

Otherwise, whatever may be the forcing that goes to the right hand side from 0.2 to n minus 1. The first point which we written here f 1 l we have indicated with the prime to indicate that the boundary condition information has been incorporated there.

(Refer Slide Time: 05:54)



Multigrid Method (cont.)

- This is expressed as: $L_l u_l = f_l$ (75)
- To determine the eigenvalues of the system consider

$$g_{m-1} - (\lambda + 2) g_m + g_{m+1} = 0$$

$$g_m = \sin \frac{mk\pi}{M+1} \quad \left.\begin{array}{lcl} m & = & 1, 2, \ldots M \\ k & = & 1, 2, \ldots M \end{array}\right\} (M+1)\Delta x = \pi$$

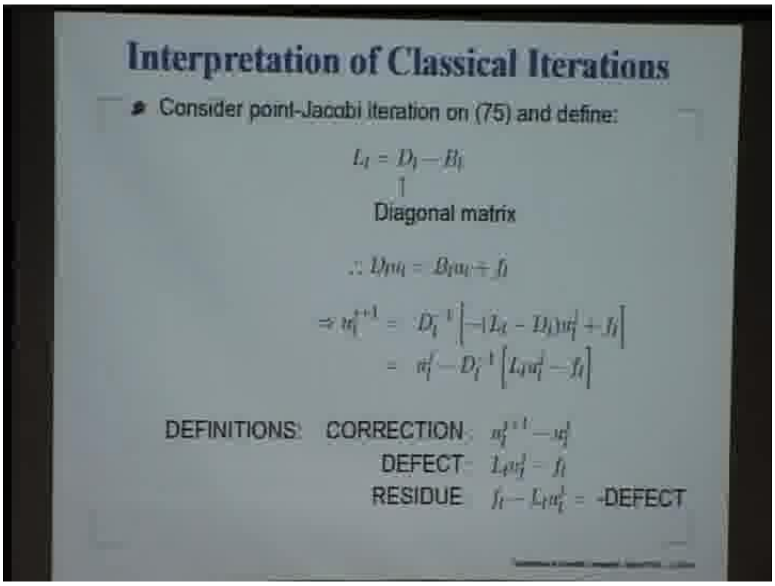
where, λ is the eigenvalue

- Eigenvalue: $\lambda_k = 2\left(\cos\frac{k\pi}{M+1} - 1\right) = -4\sin^2\frac{k\pi}{2(M+1)}$

  Eigenvectors: $(v_k)_m = \sin\frac{mk\pi}{M+1}$
- Hence, eigenvalues of L_l are $\lambda_\nu = -\frac{4}{h_l^2}\sin^2\frac{\pi\nu h_l}{2}$

We started looking at the eigen value and eigen spectrum of this. If we are looking at a periodic problem in a dimension 0 to pi, then we can take the Fourier sine series as the eigen vectors and the eigen values are given in terms of this.

(Refer Slide Time: 06:25)



Interpretation of Classical Iterations

- Consider point-Jacobi iteration on (75) and define:

$$L_l = \underset{\substack{\uparrow \\ \text{Diagonal matrix}}}{D_l} - B_l$$

$$\therefore D_l u_l = B_l u_l + f_l$$

$$\Rightarrow u_l^{j+1} = D_l^{-1}\left[-(L_l - D_l)u_l^j + f_l\right]$$
$$= u_l^j - D_l^{-1}\left[L_l u_l^j - f_l\right]$$

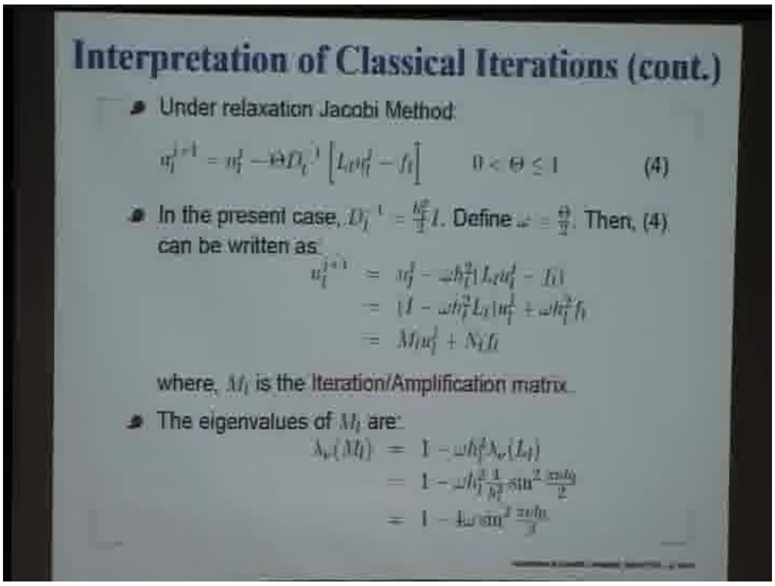
DEFINITIONS: CORRECTION: $u_l^{j+1} - u_l^j$

DEFECT: $L_l u_l^j - f_l$

RESIDUE: $f_l - L_l u_l^j$ = -DEFECT

This is what we get is that, then eigen values will be given by minus 4 by the spacing square times sign square of pi nu h l by 2. We did go through this splitting we showed in terms of that point-Jacobi iteration that the iteration sequence proceeds from u j to u j plus 1 driven by this term which you called as a defect. Defect is nothing but, the discretizing format of that particular iteration level and that is driven by the method. In this case, the Jacobi method pushes it through that diagonal matrix D l inverse.

(Refer Slide Time: 07:00)



Interpretation of Classical Iterations (cont.)

- Under relaxation Jacobi Method:

$$u_l^{j+1} = u_l^j - \Theta D_l^{-1}\left[L_l u_l^j - f_l\right] \qquad 0 < \Theta \leq 1 \qquad (4)$$

- In the present case, $D_l^{-1} = \frac{h_l^2}{2} I$. Define $\omega = \frac{\Theta}{2}$. Then, (4) can be written as

$$\begin{aligned} u_l^{j+1} &= u_l^j - \omega h_l^2 (L_l u_l^j - f_l) \\ &= (I - \omega h_l^2 L_l) u_l^j + \omega h_l^2 f_l \\ &= M_l u_l^j + N_l f_l \end{aligned}$$

where, M_l is the Iteration/Amplification matrix.

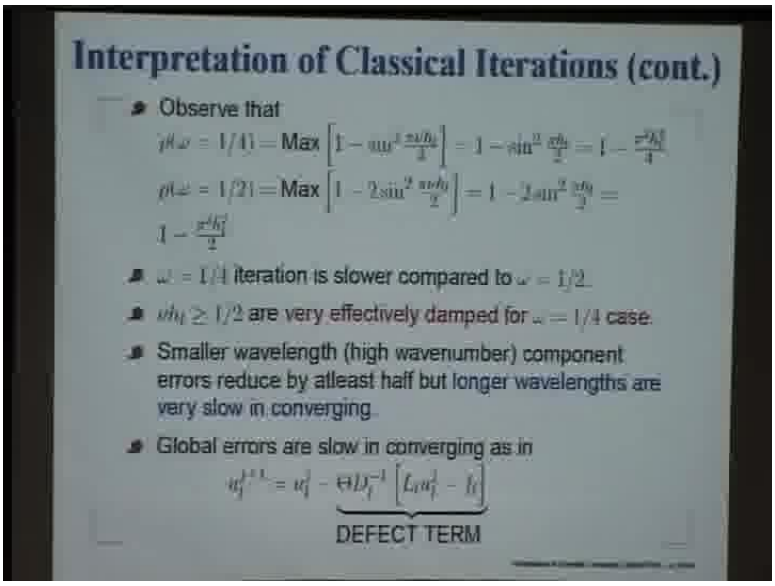
- The eigenvalues of M_l are:

$$\begin{aligned} \lambda_\nu(M_l) &= 1 - \omega h_l^2 \lambda_\nu(L_l) \\ &= 1 - \omega h_l^2 \frac{4}{h_l^2} \sin^2 \frac{\pi \nu h_l}{2} \\ &= 1 - 4\omega \sin^2 \frac{\pi \nu h_l}{2} \end{aligned}$$

Now, we explain what classical iteration does? We looked at under relaxation and Jacobi method. We note that instead of taking the defect times D l inverse you multiply by theta to indicate the kind of under relaxation we are performing. That leads to the usual form that we are familiar with an amplification matrix N working on the previous iterate. To that we add those additional terms coming from the forcing term to arrive at the fresh update of the method that is given here u l of j plus 1.

In the process you actually can note the eigen value of this amplification matrix is given in terms of the equation given in the N, where omega is theta by 2, capital theta is the under relaxation parameter.

(Refer Slide Time: 08:07)



Interpretation of Classical Iterations (cont.)

- Observe that

  $\rho(\omega = 1/4) = \text{Max}\left[1 - \sin^2\frac{\pi\nu h_l}{2}\right] = 1 - \sin^2\frac{\pi h_l}{2} = 1 - \frac{\pi^2 h_l^2}{4}$

  $\rho(\omega = 1/2) = \text{Max}\left[1 - 2\sin^2\frac{\pi\nu h_l}{2}\right] = 1 - 2\sin^2\frac{\pi h_l}{2} =$

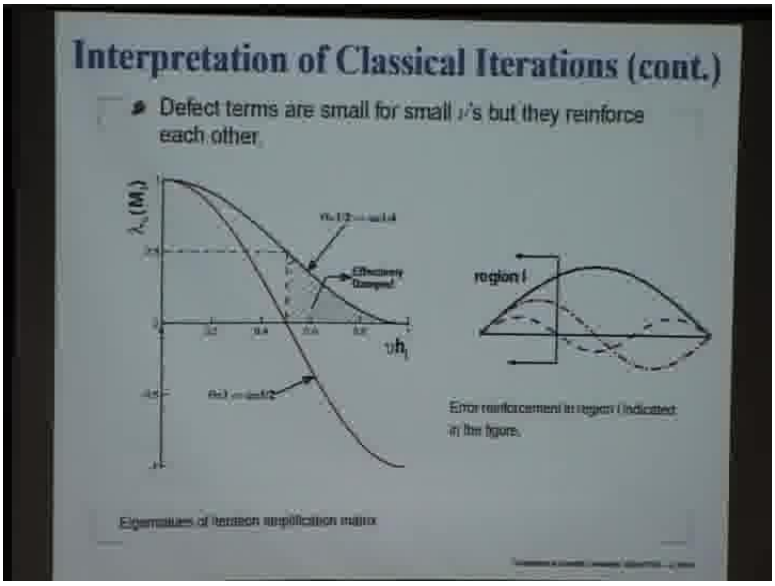
  $1 - \frac{\pi^2 h_l^2}{2}$

- $\omega = 1/4$ iteration is slower compared to $\omega = 1/2$.
- $\nu h_l \geq 1/2$ are very effectively damped for $\omega = 1/4$ case.
- Smaller wavelength (high wavenumber) component errors reduce by atleast half but longer wavelengths are very slow in converging.
- Global errors are slow in converging as in

$$u_l^{j+1} = u_l^j - \underbrace{\Theta D_l^{-1}\left[L_l u_l^j - f_l\right]}_{\text{DEFECT TERM}}$$

Now, we have gone through this and we have shown compared to the spectral radius of the case, where we have a under relaxation with half and a method which is max subjected to any relaxation at all; that is the second line.

(Refer Slide Time: 08:39)



Interpretation of Classical Iterations (cont.)

- Defect terms are small for small ν's but they reinforce each other.

Error reinforcement in region I indicated in the figure.

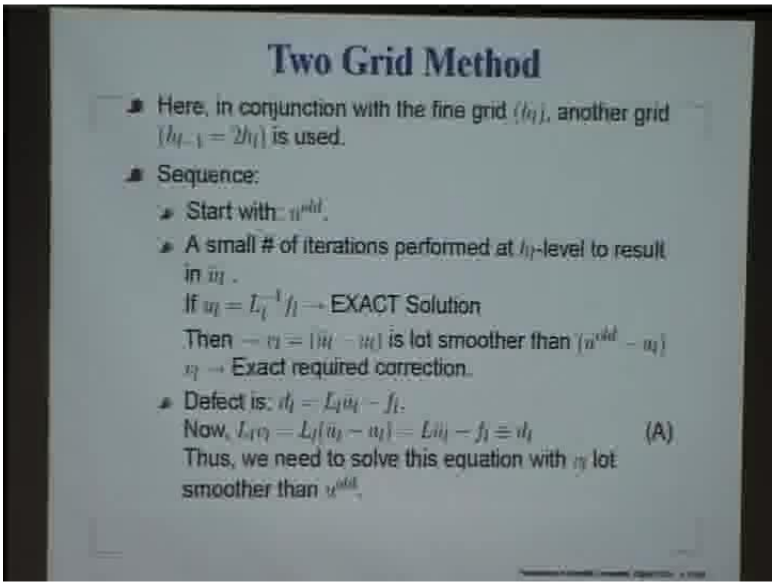
Eigenvalues of iteration amplification matrix

We can see that this basically changes eigen value spectrum quite differentially, that is what we noted in this figure. We noted that when we actually indeed do under relax by choosing theta equal to half, we end up very effectively damping is later half of this wave numbers which are given by nu h l going from 0.5 to 1 whereas, comparatively this as inferior compared to the case of without any relaxation at the large scale.

From this schematic diagram, we reasoned out that at the large scale basically you will find that in some part of the domain all these errors destructively interferes and adds on to the value of the error whereas, in other places it can have some kind of a mutual cancelation.

So I just made a point that normal mode analysis of looking at 1 mode at a time is not a very appropriate one, because after all pure error is composed of all the components simultaneously present, you are going to see the distortion of the error because of the sketch that we are shown here in this to the right of the slide.

(Refer Slide Time: 09:55)



Two Grid Method

- Here, in conjunction with the fine grid (h_l), another grid $(h_{l-1} = 2h_l)$ is used.
- Sequence:
  - Start with: u^{old}.
  - A small # of iterations performed at h_l-level to result in $\bar{u}_l$.
    If $u_l = L_l^{-1} f_l \rightarrow$ EXACT Solution
    Then $-v_l = (\bar{u}_l - u_l)$ is lot smoother than $(u^{old} - u_l)$
    $v_l \rightarrow$ Exact required correction.
  - Defect is: $d_l = L_l \bar{u}_l - f_l$.
    Now, $L_l v_l = L_l(\bar{u}_l - u_l) = L\bar{u}_l - f_l \equiv d_l$ (A)
    Thus, we need to solve this equation with v_l lot smoother than u^{old}.

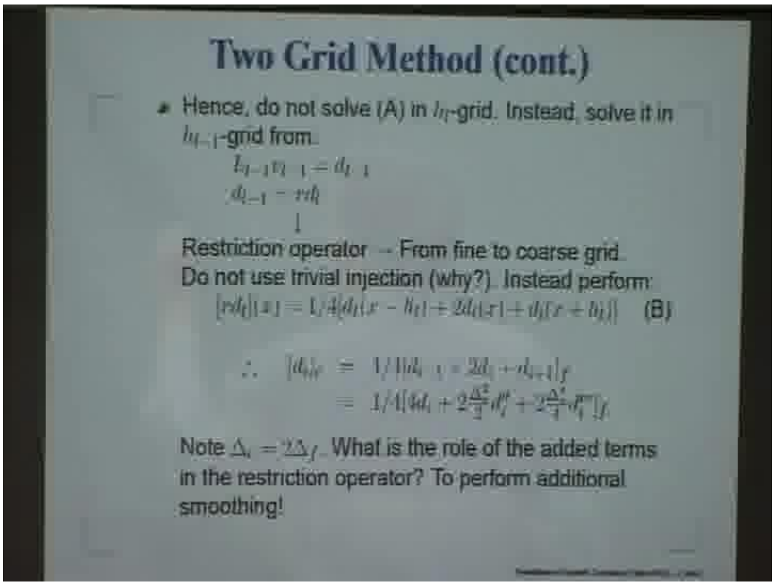
Now having noted this, if we decide to under relax then we can control the high wave numbers very effectively, if we do not do it then we can control it somewhere in between whereas, we have large error at the large scale as well as the small scale. So, that comes us to look at the possibility of using multiple grids. To explain what multiple grids does again we take a sub example of that particular problem where we let say simply only 2 grids; one which is spacing of h l and another which has spacing twice that size.

So what we actually do is we start off with some kind of initial guess u old and then we perform some small number of iterations. Let us say, we are doing some kind of a under relaxation, if we do that what this iteration does is basically, essentially a smoothing operation, because we have seen that it removes the high wave number and high wave numbers give raise to the ((defect)) in the solution.

If I am effectively being able to remove those high wave number components essentially I am basically getting a smoother solution. So, if I look at D of l here which is nothing but, the smooth solution minus the unsmooth solution. Then that quantity would be rather lot smoother then what we began with so, what happens is since the defect is given by what we have at that present level the residue with the minus sign that is what is our defect.

So, what we could do? We could actually try to solve for this exact correction that we require. Our exact solution is u of l and the smoothing gives us u bar of l; so the difference between the two is the exact correction that is required. So, the exact correction is governed by this equation L l into v l. Instead of v l I put that definition down here as u bar minus u and that would give me the defect back. Essentially, you obtain the exact correction we have to solve the same problem but, now the right hand side is replaced by the defect at that level.

(Refer Slide Time: 12:45)



Two Grid Method (cont.)

- Hence, do not solve (A) in h_l-grid. Instead, solve it in h_{l-1}-grid from

$$L_{l-1}v_{l-1} = d_{l-1}$$
$$d_{l-1} = rd_l$$
$$\downarrow$$

Restriction operator – From fine to coarse grid

Do not use trivial injection (why?). Instead perform:

$$[rd_l](x) = 1/4[d_l(x - h_l) + 2d_l(x) + d_l(x + h_l)] \quad \text{(B)}$$

$$\begin{aligned} \therefore \quad [d_i]_c &= 1/4[d_{i-1} + 2d_i + d_{i+1}]_f \\ &= 1/4[4d_i + 2\frac{\Delta_f^2}{2}d_i'' + 2\frac{\Delta_f^4}{4}d_i'''']_f \end{aligned}$$

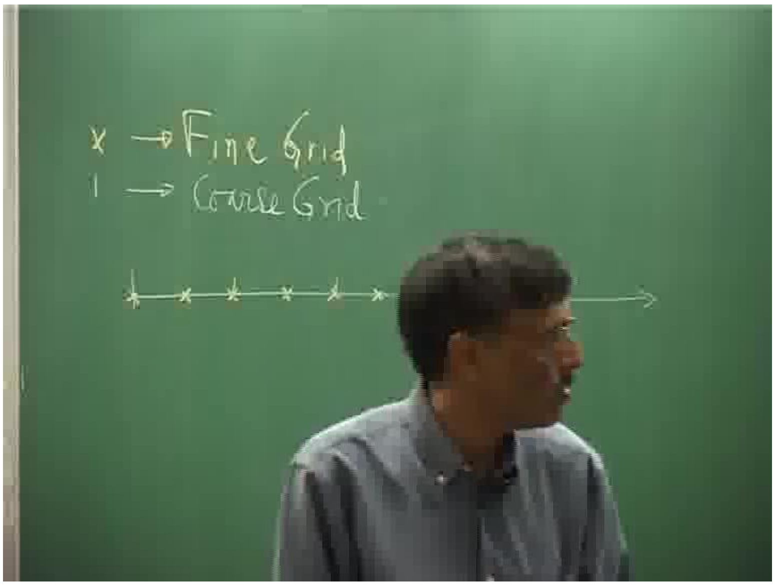
Note $\Delta_c = 2\Delta_f$. What is the role of the added terms in the restriction operator? To perform additional smoothing!

That is what that right hand side of equation A indicates that the correction is driven by the defect at that level. This is what we started discussing towards the end that we started off working with a fine grid, h l grid. Now, what we are going to do? We have exhaustible possible potential benefit at the fine grid, so we migrate to the next grid that h l minus 1 grid. That is where we want to solve for this exact correction.

We are looking for the exact correction v l but, instead of working at h l grid when over to the h l minus 1 grid, because we have exhausted the all positive benefit of h l grid. In solving this equation the operator is known to us and this is what we will be calculating in the coarse grid but, the right hand side is to be obtained.

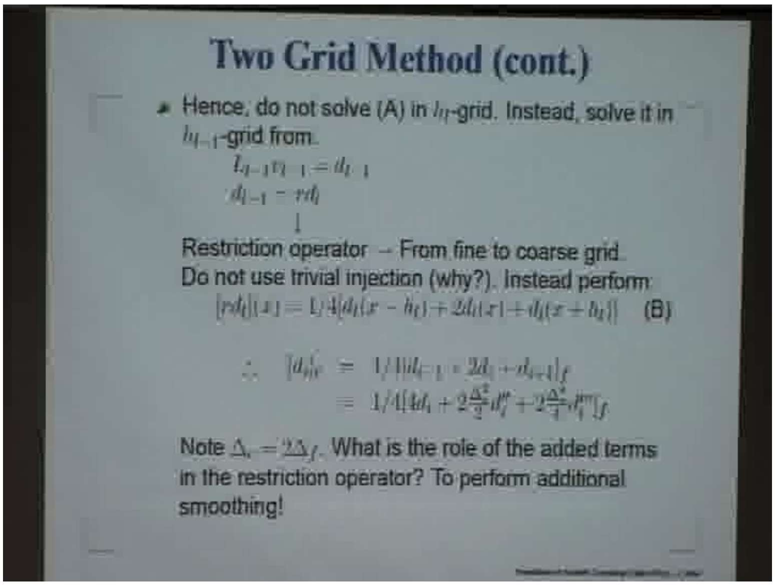
How it is obtain from the fine grid solution? From the fine grid solution, we can estimate d of l and then what we do? We projected to the coarse grid and that projection is done to what is called as a restriction operator. So that essentially means that we are taking the data from the point to the coarse grid.

(Refer Slide Time: 14:07)



If you really look at let us say, we have taken this as our point and let us say this is the coarse grid and we take the fine grid by having this color points here. Basically, this is my fine grid solution and this is what we are getting as our coarse grid point. We have 2 sets of points now, what you see is as per the coarse points are concerned, they also belong to the set of the fine grid points.

(Refer Slide Time: 15:01)



Two Grid Method (cont.)

- Hence, do not solve (A) in h_q-grid. Instead, solve it in h_{q-1}-grid from

$$L_{q-1}v_{q-1} = d_{q-1}$$
$$d_{q-1} = rd_q$$

↓

Restriction operator → From fine to coarse grid.
Do not use trivial injection (why?). Instead perform:

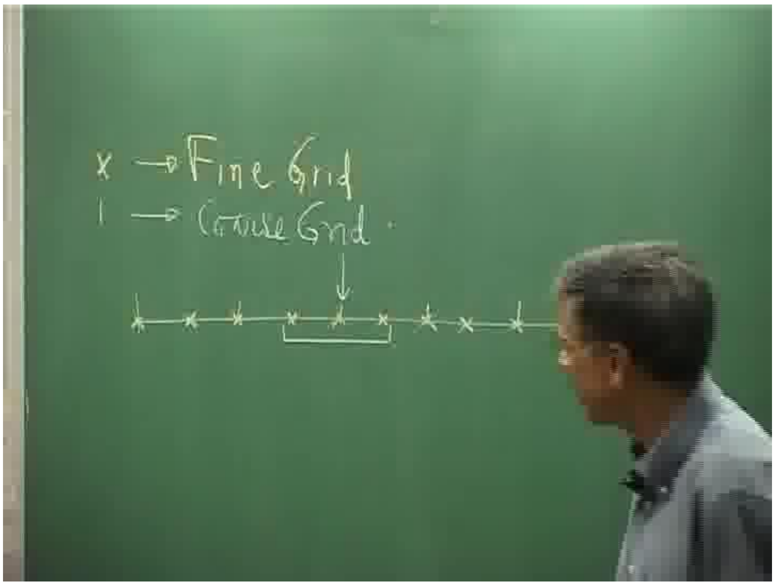
$$[rd_q](x) = 1/4[d_q(x - h_q) + 2d_q(x) + d_q(x + h_q)] \quad \text{(B)}$$

$$\therefore \quad [d_j]_c = 1/4[d_{j-1} + 2d_j + d_{j+1}]_f$$
$$= 1/4[4d_j + 2\tfrac{\Delta_c^2}{2}d_j'' + 2\tfrac{\Delta_c^4}{4}d_j'''']_f$$

Note $\Delta_c = 2\Delta_f$. What is the role of the added terms in the restriction operator? To perform additional smoothing!

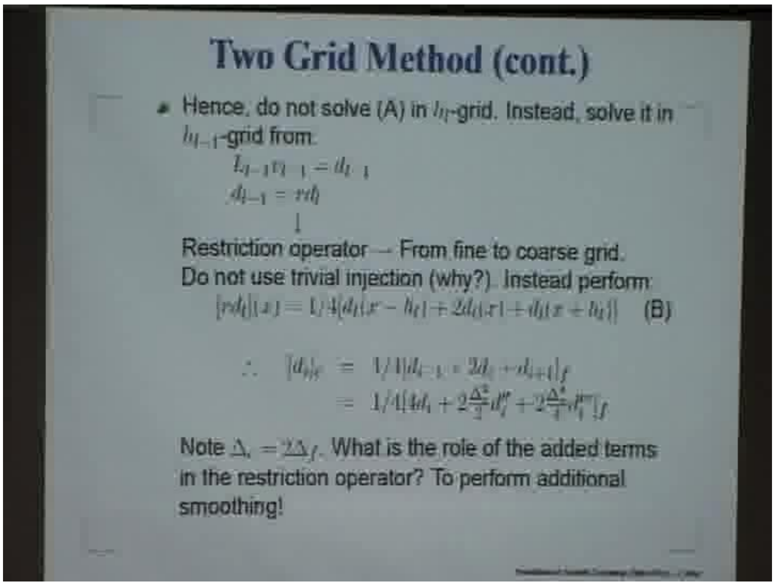
So, we could just simply pick them up but, warning given here that do not use that trivial injection means, do not just simply pick up the points there. Why? The answer is given in equation B what you do instead you actually take the points obtained from this.

(Refer Slide Time: 15:18)



For example, if I want to get the value at this point which is a coarse grid point what I would do is I would involve these 3 points. This is your coarse grid point that you are looking at, now you actually take 2 points from on either side of the point in question from the fine grid.

(Refer Slide Time: 15:39)



Two Grid Method (cont.)

- Hence, do not solve (A) in h_l-grid. Instead, solve it in h_{l-1}-grid from

$$L_{l-1}v_{l-1} = d_{l-1}$$
$$d_{l-1} = rd_l$$
$$\downarrow$$

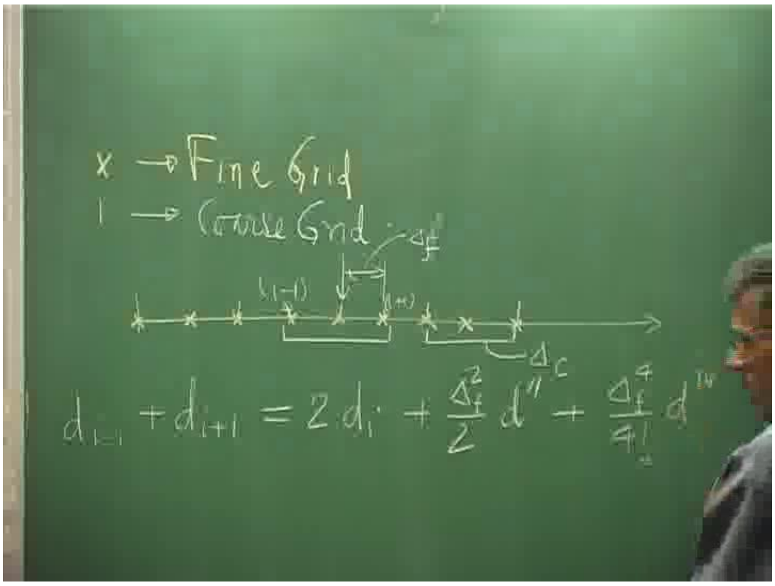
Restriction operator – From fine to coarse grid.
Do not use trivial injection (why?). Instead perform:

$$[rd_l](x) = 1/4[d_l(x-h_l) + 2d_l(x) + d_l(x+h_l)] \quad \text{(B)}$$

$$\therefore \quad [d_i]_c = 1/4[d_{i-1} + 2d_i + d_{i+1}]_f$$
$$= 1/4[4d_i + 2\frac{\Delta_f^2}{2}d_i'' + 2\frac{\Delta_f^4}{4!}d_i''''\,]_f$$

Note $\Delta_c = 2\Delta_f$. What is the role of the added terms in the restriction operator? To perform additional smoothing!

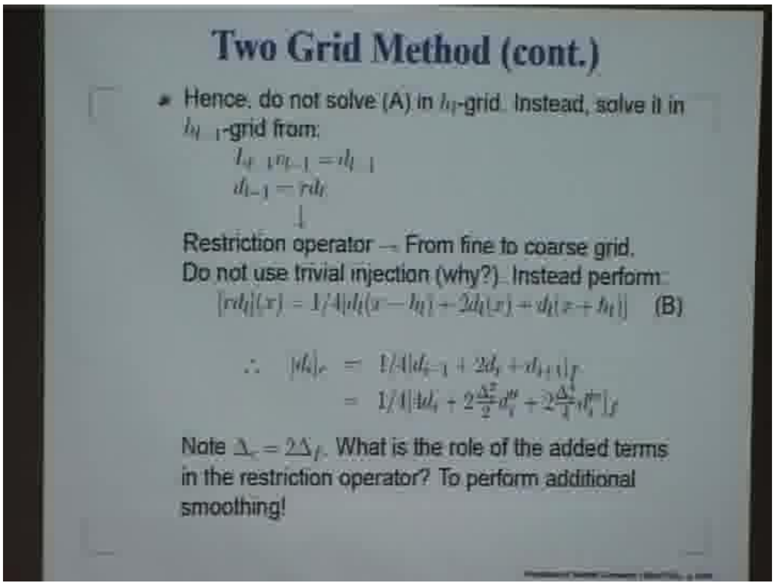
(Refer Slide Time: 16:00)


x → Fine Grid
| → Coarse Grid
Δ_f
$(i-1)$
$(i+1)$
Δ_c
$d_{i-1} + d_{i+1} = 2d_i + \frac{\Delta_f^2}{2}d'' + \frac{\Delta_f^4}{4!}d^{iv}$

Why do we do it? It is very evident if we look at what this convocates; this actually implies that we are taking some kind of a weighted average among these 3 points that is one-fourth times this; this is very easy for you to see that this quantity that we have written there d i minus 1. If I add to that with a Taylor series, you can clearly see that this is going to be 2 d i (( )) odd derivatives they will all fall out, only the even derivatives will survive. If I look at the even derivative are concerned this is our delta f or the spacing in the fine grid whereas, this is our delta c.

What we are finding that (( )) from the fine grid so, we have picked up this (( )) i minus 1 point and this is i plus 1 point and this will actually give us d double prime, then I will have factorial 4 of the fourth derivative and so on so forth.

(Refer Slide Time: 17:25)



Two Grid Method (cont.)

- Hence, do not solve (A) in h_l-grid. Instead, solve it in h_{l-1}-grid from:

$$L_{l-1} v_{l-1} = d_{l-1}$$
$$d_{l-1} = r d_l$$
$$\downarrow$$

Restriction operator – From fine to coarse grid.
Do not use trivial injection (why?). Instead perform:

$$[r d_l](x) = 1/4 [d_l(x - h_l) + 2 d_l(x) + d_l(x + h_l)] \quad \text{(B)}$$

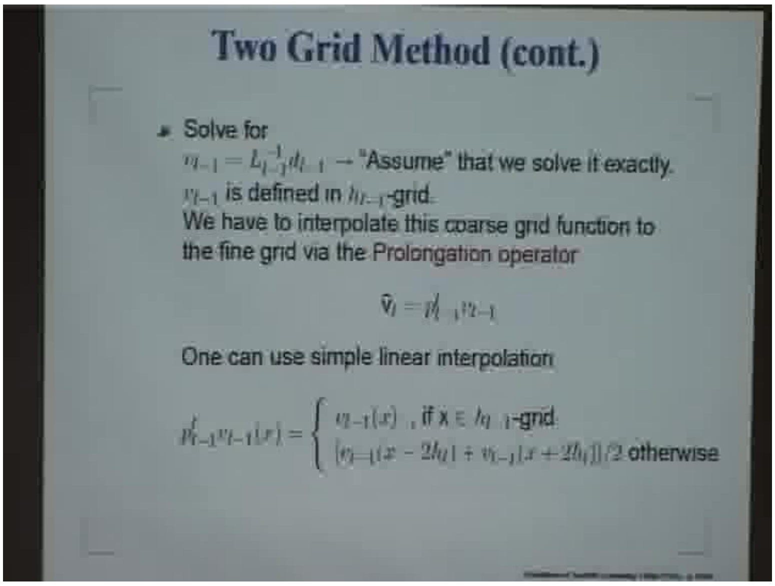
$$\begin{aligned} \therefore \quad [d_i]_c &= 1/4 [d_{i-1} + 2 d_i + d_{i+1}]_f \\ &= 1/4 [4 d_i + 2 \tfrac{\Delta^2}{2} d_i'' + 2 \tfrac{\Delta^4}{4} d_i''''\,]_f \end{aligned}$$

Note $\Delta_c = 2\Delta_f$. What is the role of the added terms in the restriction operator? To perform additional smoothing!

So what you are seeing that this operation that we are calling as restriction operation, essentially gives us nothing but, the so called trivially injected value; that is what would have been if we would have pick the point what we get? We would have actually obtained just this point alone. But, what we are doing by this kind of operation? We are adding up these even derivative terms, so that addition of even derivative is what? It is additional smoothing you are damping out.

If there are any error component at that fine grid level by doing this restriction you are essentially smoothing them down further by this restriction operator. So, we have now understood that even in projecting the solution from the fine grid to the coarse grid we are additionally smoothing the solution.

(Refer Slide Time: 18:29)



Two Grid Method (cont.)

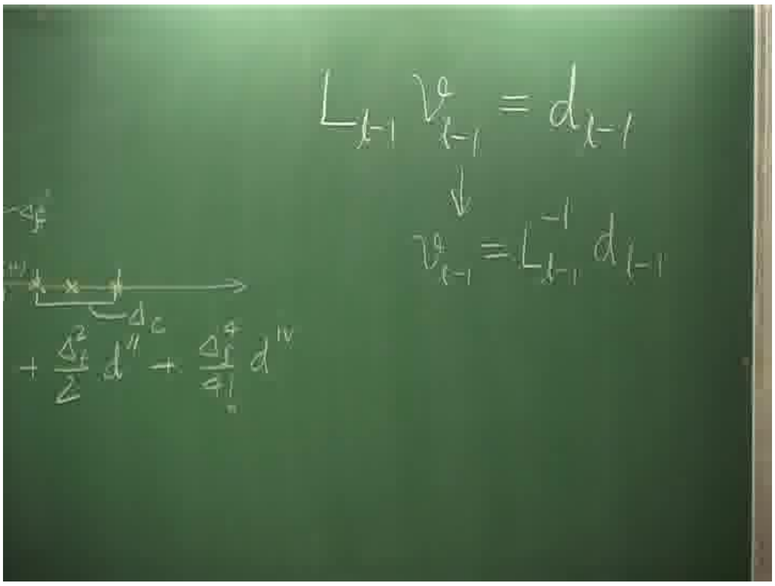
- Solve for
  $v_{l-1} = L_{l-1}^{-1} d_{l-1}$ → "Assume" that we solve it exactly.
  v_{l-1} is defined in h_{l-1}-grid.
  We have to interpolate this coarse grid function to the fine grid via the Prolongation operator

$$\hat{v}_l = p_{l-1}^l v_{l-1}$$

One can use simple linear interpolation

$$p_{l-1}^l v_{l-1}(x) = \begin{cases} v_{l-1}(x) \text{ , if } x \in h_{l-1}\text{-grid} \\ [v_{l-1}(x - 2h_l) + v_{l-1}(x + 2h_l)]/2 \text{ otherwise} \end{cases}$$

(Refer Slide Time: 18:42)



$$L_{l-1} v_{l-1}^c = d_{l-1}$$

$$\downarrow$$

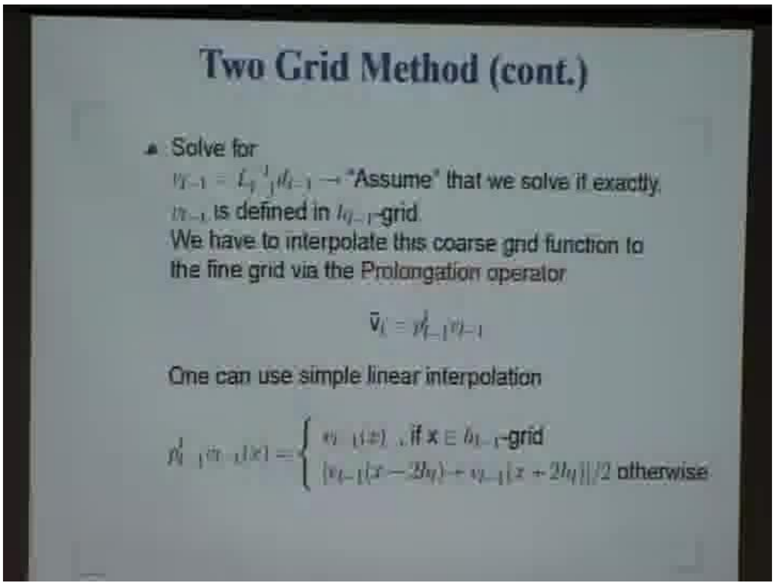
$$v_{l-1}^c = L_{l-1}^{-1} d_{l-1}$$

Now, we are in the coarse grid; the equation that we have is nothing but, the operator at the coarse grid operating on the correction, that is needed at the coarse grid is driven by the restricted defect at the coarse grid, this is essentially we want to do. Let us say we can assume that this is exactly solvable, and then what we are doing essentially is we are inverting the L matrix and operating on the defect at the coarse grid level to get the exact solution. That is what the bullet says assume that we solve it exactly.

What happens is you are in the coarse grid; you have got the exact correction that was to be obtaining at the coarse grid level that is why the subscript l minus 1 indicates that you are at the coarse grid. Now, what you have to do? You have implied the fine grid for the reason that you want to get the solution at that discretization level, so that is how we started off.

We decide on the finest grid depending on what kind of solutions we are looking for at what level of accuracy we want it? Based on that question we decided on the finest grid. So, our fine grid choice is dictated upon by our requirement of the accuracy.

(Refer Slide Time: 20:07)



Two Grid Method (cont.)

- Solve for
  $v_{l-1} = L_{l-1}^{-1} d_{l-1}$ → "Assume" that we solve it exactly.
  v_{l-1} is defined in h_{l-1}-grid.
  We have to interpolate this coarse grid function to the fine grid via the Prolongation operator

$$\bar{v}_l = p_{l-1}^{l} v_{l-1}$$

One can use simple linear interpolation

$$p_{l-1}^{l}(v_{l-1})(x) = \begin{cases} v_{l-1}(x), & \text{if } x \in h_{l-1}\text{-grid} \\ [v_{l-1}(x-2h_l)+v_{l-1}(x+2h_l)]/2 & \text{otherwise} \end{cases}$$

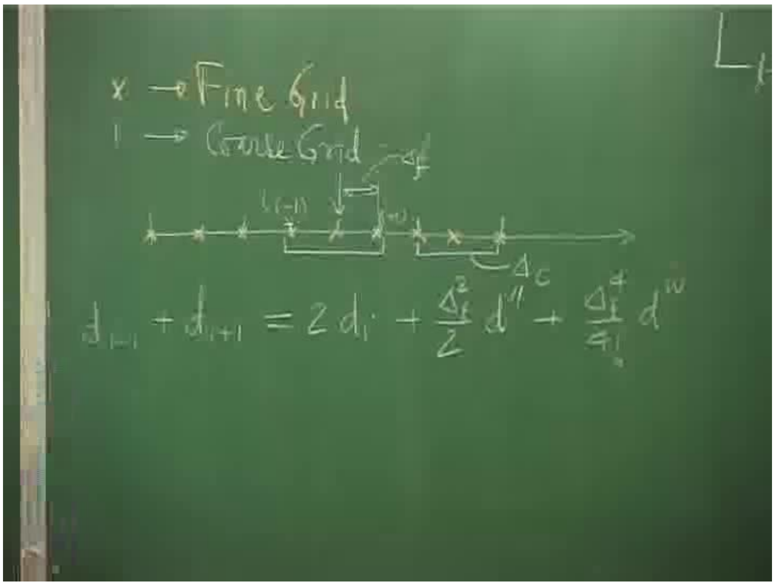
Now, what happens is that I have obtained the so called exact correction at the coarse grid given by v l minus 1; I need to go back to my fine grid level and that is also done by again what is called by a prolongation operator while these are all mouth filling Jorgen's. They are just simply nothing but, what you see in the equation given in the end that we are going from p l minus 1.

Basically, we are going from l minus 1 grid to l grid and this is our exact solution v l minus 1. So, the subscript and superscript indicates the flow direction of data transfer from the coarse grid to the fine grid. The logic is very simple wherever the points are common you pick it up as it is, wherever they are not you do this, that is the last line;

what does it do? Again you are averaging; any averaging operation would involve additional smoothing.

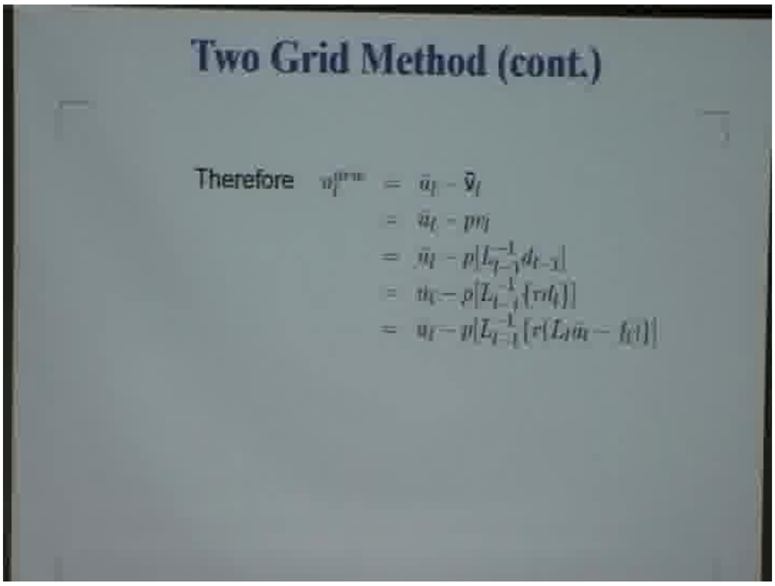
So, you can see I made an observation that computing always meant a (( )) at foam in choosing your smoothing or dissipation and that is what we are seeing. That even here when we are trying to solve this elliptic equation at every step we get an opportunity; we try to smooth out the solution by adding a dissipative mechanism.

(Refer Slide Time: 21:59)


x → Fine Grid
| → Coarse Grid
Δ_f
$(i-1)$
$(i+1)$
Δ_c
$d_{i-1} + d_{i+1} = 2d_i + \frac{\Delta_f^2}{2} d'' + \frac{\Delta_f^4}{4!} d^{iv}$

That is what your last equation implies the points. So, I have point here and here but, I do not have it here. So, what I do? I take this as the average of the 2, more fancy tool of weighing more or less is just simply average, because that is how you will add additional dissipative term (Refer Slide Time: 22:00).

(Refer Slide Time: 22:20)



Two Grid Method (cont.)

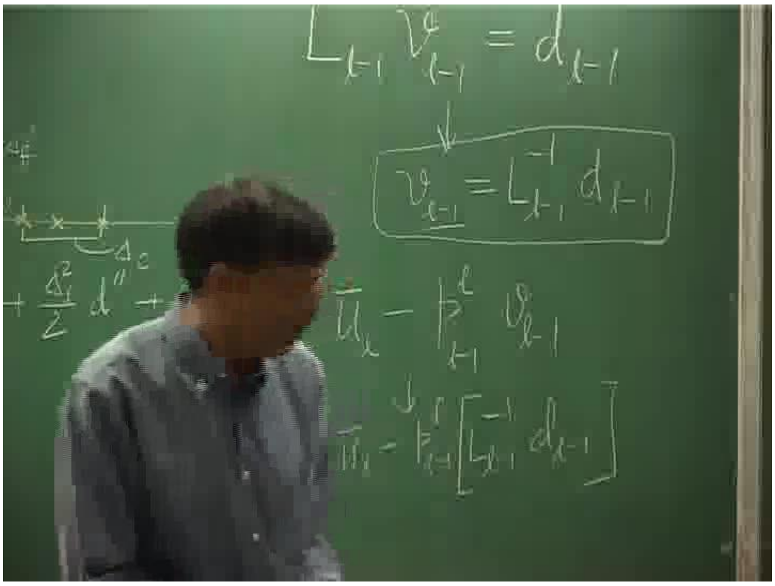
Therefore
$$\begin{aligned} u_l^{new} &= \bar{u}_l - \tilde{v}_l \\ &= \bar{u}_l - p v_l \\ &= \bar{u}_l - p[L_{l-1}^{-1} d_{l-1}] \\ &= \bar{u}_l - p[L_{l-1}^{-1}\{r d_l\}] \\ &= \bar{u}_l - p[L_{l-1}^{-1}[r(L_l \bar{u}_l - f_l)]] \end{aligned}$$

So, you know rotationally then we could write it all down in this way. Let us see what we have done? We started up working with a fine grid and we did few iterations and then we exhausted the benefit there. Then we move to the force grid and then when we came to the coarse grid, we did that restriction operation and then we solve the equation at the coarse grid. Having obtain the solution at the coarse grid we actually again went back to the fine grid and that is our solution that is what we write as the top u nu.

So, u nu is the one such journey going from point to coarse, back to point. What we did was that whatever we had before, you initially we started up with u l some u old and then we did some smoothing operation and that was that u bar given that the first term u bar l. To that we have added exact correction that was to be done was v l but, we do not have that exact v l we have some altered form of it and that altered form we are calling it as v l bar or tilde.

How did we get that; we got by actually the second line would mean that we have got the information at the l minus 1th grid and we projected by prolongation operator that p stand for that.

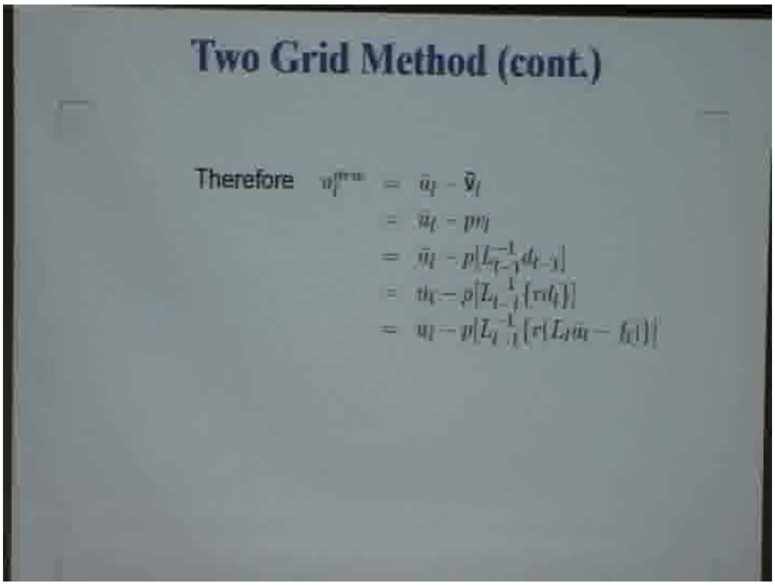
(Refer Slide Time: 24:08)



$L_{l-1} v_{l-1} = d_{l-1}$
$v_{l-1} = L_{l-1}^{-1} d_{l-1}$
$\bar{u}_l - p_{l-1}^{l} v_{l-1}$
$\bar{u}_l - p_{l-1}^{l} [L_{l-1}^{-1} d_{l-1}]$

So, basically that second line should be ideally written like this we have gone from l minus 1 and we have this, so that is the second line that is what we have done (Refer Slide Time: 24:05). Now, how did we get this? This we got in turn like this as I told you here we have gotten that v l minus 1 in this form so, what I would say? I would say that this we have done it on L l minus 1 inverse d l minus 1; that is your the third line.

(Refer Slide Time: 24:47)



Two Grid Method (cont.)

Therefore $u_l^{new} = \bar{u}_l - \tilde{v}_l$

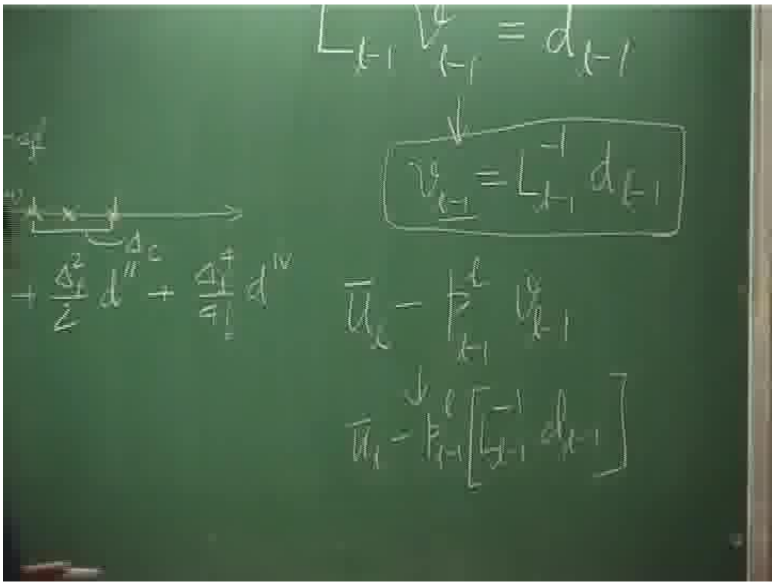
$= \bar{u}_l - p v_l$

$= \bar{u}_l - p[L_{l-1}^{-1} d_{l-1}]$

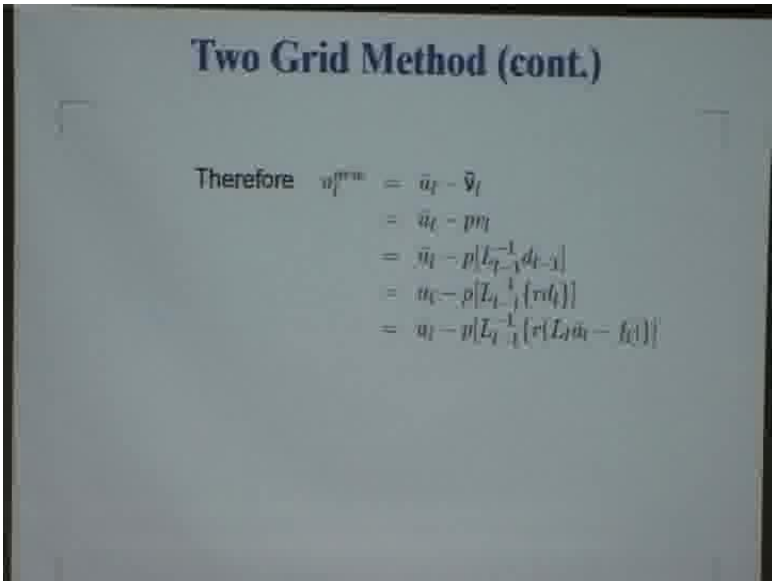
$= \bar{u}_l - p[L_{l-1}^{-1}\{r d_l\}]$

$= \bar{u}_l - p[L_{l-1}^{-1}\{r(L_l \bar{u}_l - f_l)\}]$

(Refer Slide Time: 25:03)


$L_{l-1} v_{l-1}^{l} = d_{l-1}$
$v_{l-1} = L_{l-1}^{-1} d_{l-1}$
$+ \frac{\Delta_1^2}{2} d'' + \frac{\Delta_1^4}{4!} d^{IV}$
Δc
$\bar{u}_l - p_{l-1}^{l} v_{l-1}^{l}$
$\bar{u}_l - p_{l-1}^{l}[L_{l-1}^{-1} d_{l-1}]$

(Refer Slide Time: 25:15)



Two Grid Method (cont.)

Therefore $u_l^{new} = \bar{u}_l - \tilde{v}_l$

$= \bar{u}_l - pv_l$

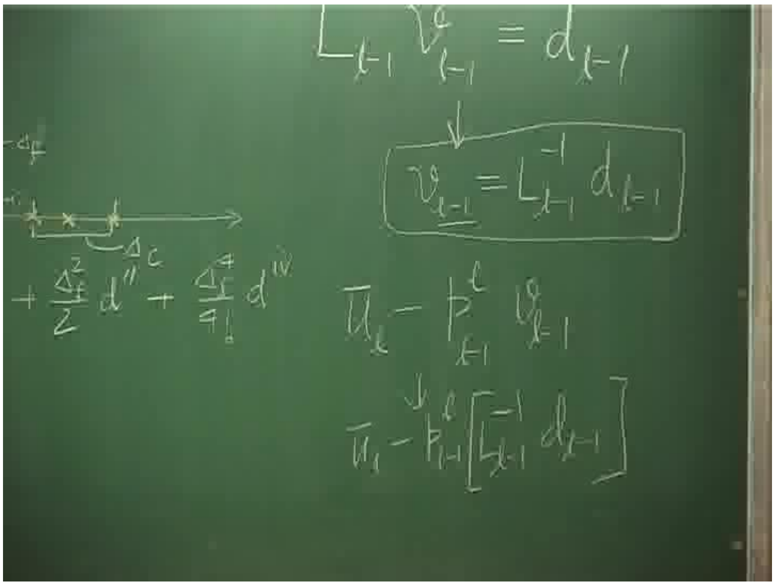
$= \bar{u}_l - p[L_{l-1}^{-1} d_{l-1}]$

$= u_l - p[L_{l-1}^{-1}\{rd_l\}]$

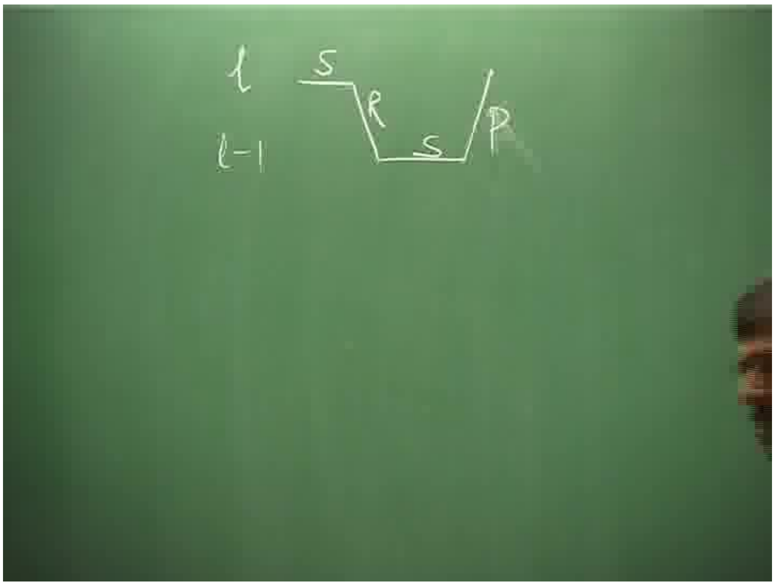
$= u_l - p[L_{l-1}^{-1}\{r(L_l\bar{u}_l - f_l)\}]$

The third line is here which says that you have solved that problem exactly on the coarse grid, then you basically projected it one level up. Now, how did we get this solution? This solution was obtained by calculating the defect at the coarse grid and how did we do it? We actually did it by this restriction operator r times d l that give us d l minus 1.

(Refer Slide Time: 25:40)


$L_{l-1} v^{l}_{l-1} = d_{l-1}$
$\boxed{v_{l-1} = L^{-1}_{l-1} d_{l-1}}$
Δc
$+ \frac{\Delta_f^2}{2} d'' + \frac{\Delta_f^4}{4!} d^{iv}$
$\bar{u}_l - P^{l}_{l-1} v_{l-1}$
$\bar{u}_l - P^{l}_{l-1}\left[L^{-1}_{l-1} d_{l-1}\right]$

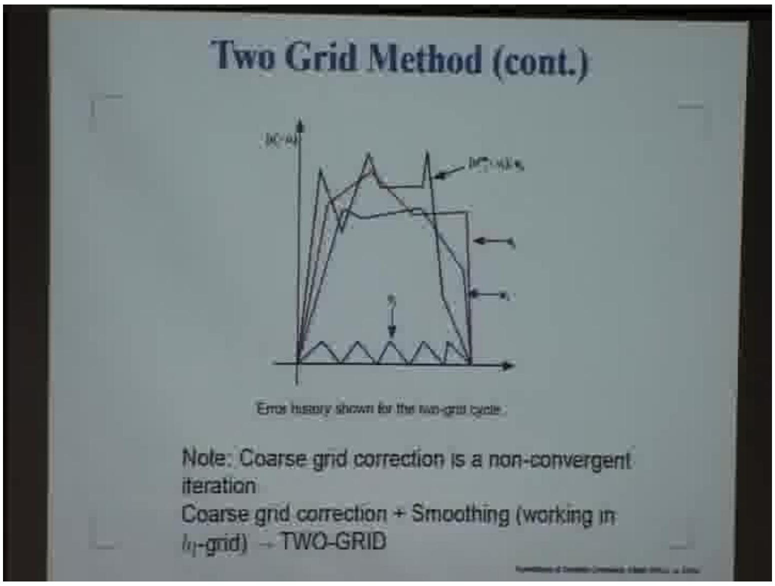
(Refer Slide Time: 25:54)



Basically that step and d l itself is nothing but, L l operating on u l bar minus f l; this is whatever may be the defect at that level of iteration. So, this is basically a very small journey that we just seen that we start off with u old then we do few smoothing operation write it as that smoothing. Then what we did? This is I have say l th level we have and then what we did is we brought it down to the next coarse grid level.

So from here to here is basically getting here d l to d l minus 1 and then you have solved L l minus 1, d l minus 1 that is against smoothing operation. So, this operation I will call it as restriction. So, first is a smoothing followed by restriction, then we have again at the coarse grid we have done some smoothing operation. We are here now then, what we did was we just simply went back to the fine grid for prolongation.

(Refer Slide Time: 27:05)



Two Grid Method (cont.)

Error history shown for the two-grid cycle

Note: Coarse grid correction is a non-convergent iteration

Coarse grid correction + Smoothing (working in h_1-grid) → TWO-GRID

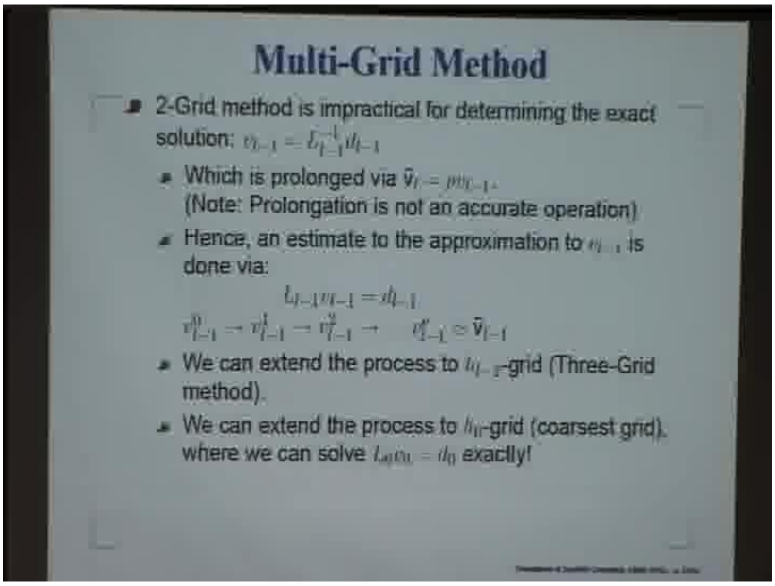
This is kind of a unit process in Multigrid; this is how you go from one grid level to the next. This opens up many possibilities of performing that I will just explain some of the popular once, but before I do that let me tell you what we did achieve? Let us say in this schematic what we show is this top curve that black curve is the one that we started up. Let us say we plotted on the error versus x let us say error history at different level.

What we did was we started up with that kind of a badly behaved error, then we did some kind of a smoothing, so that is given by this red line. Then we did few more iterations that restriction and then solving the coarse grid. After few more steps of smoothing we get this kind of an error.

So we end up doing this again and again in a different way. Basically, we got to understand that at the coarse grid level if we keep on doing it, it will never converge it is not a convergent iteration, for the same reason that we explain with the point Gauss-Seidel method because eventually the g becomes greater than 1.

So, what happens is we need to keep on working between this coarse grid and the fine grid because the reason I told you that we do not really work in a normal mode scenario, we have to see when all the errors are together. That is what happens at the every level of iteration we are actually redistributing the error of among different components.

(Refer Slide Time: 28:47)



Multi-Grid Method

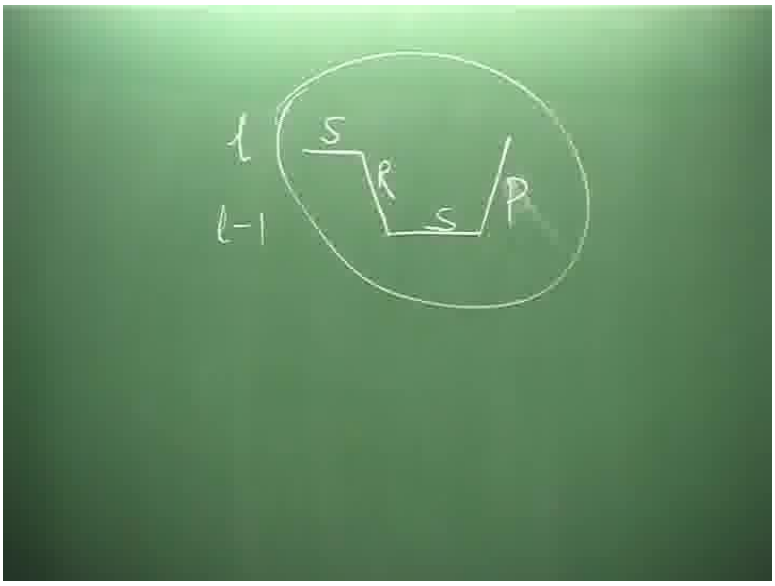
- 2-Grid method is impractical for determining the exact solution: $v_{l-1} = L_{l-1}^{-1} d_{l-1}$
  - Which is prolonged via $\hat{v}_l = p v_{l-1}$.
    (Note: Prolongation is not an accurate operation)
  - Hence, an estimate to the approximation to v_{l-1} is done via:

$$L_{l-1} v_{l-1} = d_{l-1}$$
$$v_{l-1}^{0} \rightarrow v_{l-1}^{1} \rightarrow v_{l-1}^{2} \rightarrow \quad v_{l-1}^{\nu} \simeq \hat{v}_{l-1}$$

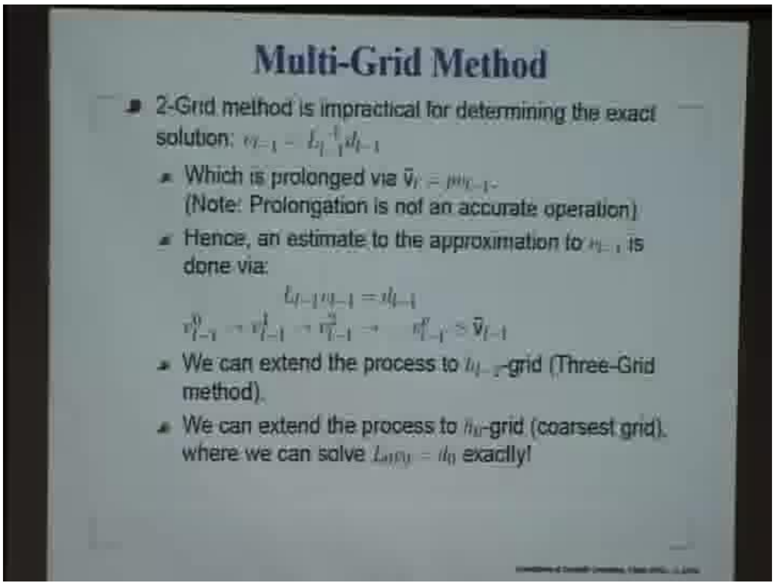
  - We can extend the process to h_{l-2}-grid (Three-Grid method).
  - We can extend the process to h_0-grid (coarsest grid), where we can solve $L_0 v_0 = d_0$ exactly!

That is what it is suggestive that if we only employed two grids it is going to be rather a difficult task. So, what people keep doing is that you may have gotten the exact solution at the fine grid level here, but then the moment you project it back into the fine grid. So, from the coarse grid value if I project it, prolong it back to the fine grid that itself will add on to some error, because it is again some numerical operation. You saw that what we do? We either pick it up or we take average so, the moment we do any numerical operation we are actually incurring error.

(Refer Slide Time: 29:35)



(Refer Slide Time: 29:52)



Multi-Grid Method

- 2-Grid method is impractical for determining the exact solution: $v_{l-1} = L_{l-1}^{-1} d_{l-1}$
  - Which is prolonged via $\bar{v}_l = p v_{l-1}$.
    (Note: Prolongation is not an accurate operation)
  - Hence, an estimate to the approximation to v_{l-1} is done via:

$$L_{l-1} v_{l-1} = d_{l-1}$$

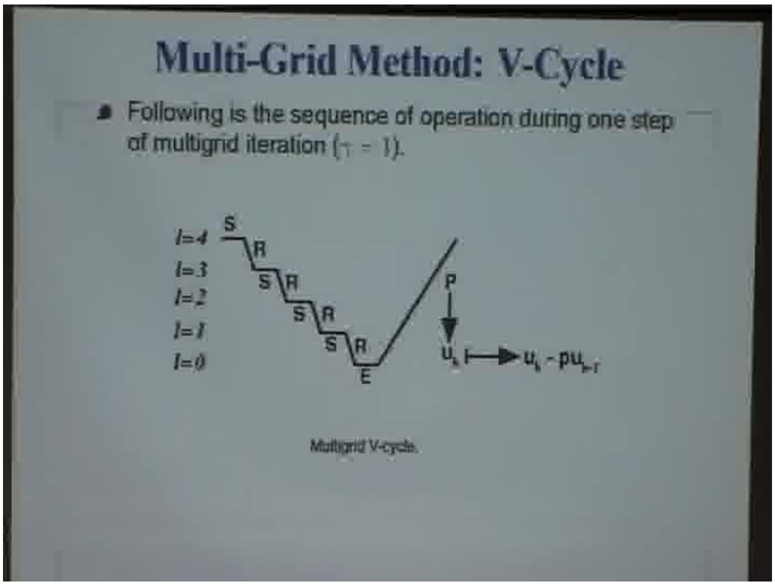
$$v_{l-1}^{0} \rightarrow v_{l-1}^{1} \rightarrow v_{l-1}^{2} \rightarrow \ldots v_{l-1}^{\nu} \approx \bar{v}_{l-1}$$

  - We can extend the process to h_{l-2}-grid (Three-Grid method).
  - We can extend the process to h_0-grid (coarsest grid), where we can solve $L_0 v_0 = d_0$ exactly!

You know what happens is that this kind of a single block of operation would not be really adequate. So, what you usually end up doing is basically keep going from fine grid to a coarser grid to finer even a coarser. That is what you do till you come to a level where you actually I have exhausted all possibilities and you can really solve it exactly, because we said that coarse grid iteration is a non-convergent process.

So, we could go and stop at that level where we can solve it exactly because we do not have to depend on iteration that is what we are talking about here. Let us say at the 0 th level, we have the exact solution available. Then we can keep on prolonging it back to the next finer grid then the next finer grid and all the way to where we have begun.

(Refer Slide Time: 30:46)



Multi-Grid Method: V-Cycle

- Following is the sequence of operation during one step of multigrid iteration ($\gamma = 1$).

Multigrid V-cycle.

If we extend the process from h l to h l minus 1 to h l minus 2, then we have so called 3 grid methods, actually this is the way that we use in Multi-Grid method. Let me show you one such Multi-Grid cycle that has been employed quite often and this is called a V-cycle.

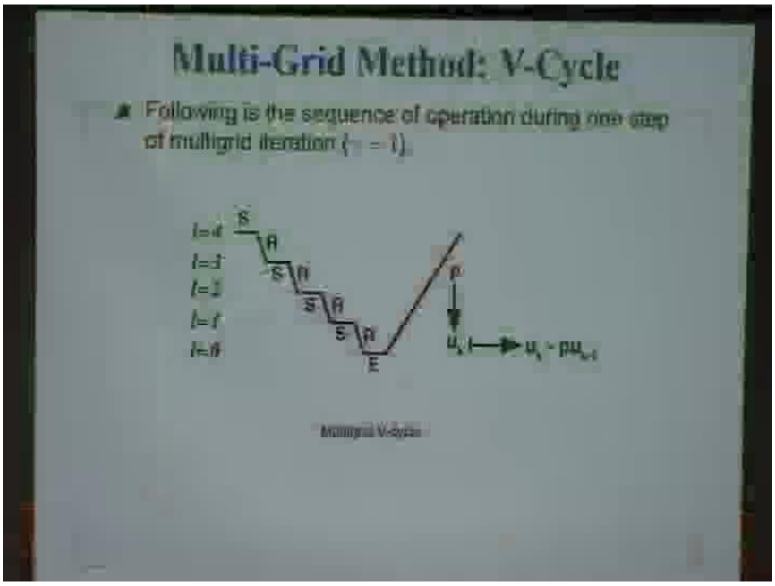
What I have shown you here is basically a 5 level of grid, you start with l equal to 4 that is your finest grid then, you perform few smoothing operation and this smoothing operation could be anything that you choose. We have not talked about what kind of thing you can do? You can do a Jacobi, you can do a Gauss-Seidel, you can do any rather fancy method but, the essence of Multi-Grid method is that it is smoothing operation should be as simple as possible.

So, most of the time what you end up doing is something like your Jacobi method and you could add to that some kind of a under relaxation or over relaxation. That is what your S is? S implies that smoothing operation of any particular method that we choose. Then what you do? You restrict that to go the next grid level, which you call as l equal to 3. There we perform few levels of smoothing operation, then again we restrict it and go to l equal to 2 level and performs some smoothing there followed by restriction then again smoothing, restriction.

(Refer Slide Time: 32:21)

At l equal to 0 let us say we have arrived at a situation, where we could directly invert the matrix directly it could be simply as simple as that. Let us say this is my domain, I could just simply take a point here driven by the boundary condition only unknown is at 1 point. So that is what we meant by the exact solution or at the most let us say may be you could take this kind of a point so, you will have 9 unknowns then also you can invert an 9 by 9 matrix it is not at all difficult.

(Refer Slide Time: 32:57)



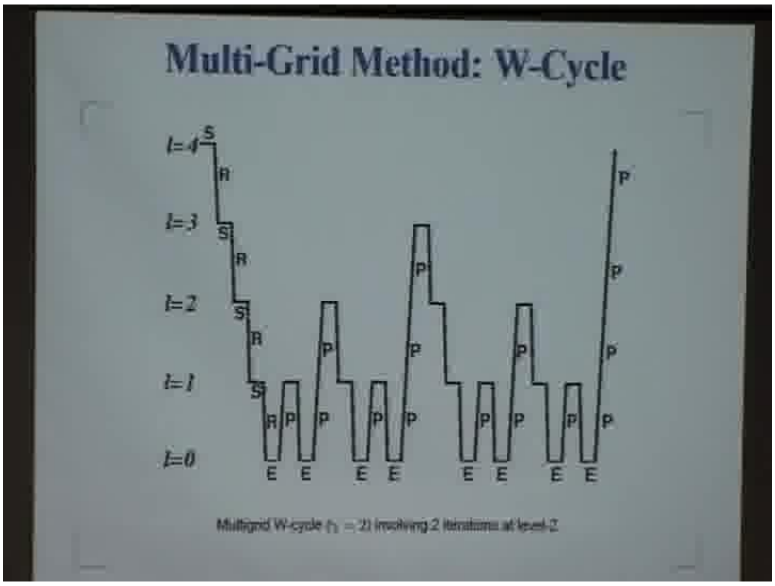
Multi-Grid Method: V-Cycle

- Following is the sequence of operation during one step of multigrid iteration ($\gamma = 1$).

Multigrid V-cycle

You do not have to do iteration; you can just directly invert the matrix and solve it that is what we are referring to at l equal to 0 level that is what it is. So, once you have that then again you start your journey back to your finest possible grid via all these prolongation that is what is a V-cycle 1 unit. So, you may have to do it again till you convert.

(Refer Slide Time: 33:37)



Multi-Grid Method: W-Cycle

Multigrid W-cycle ($\gamma = 2$) involving 2 iterations at level-2

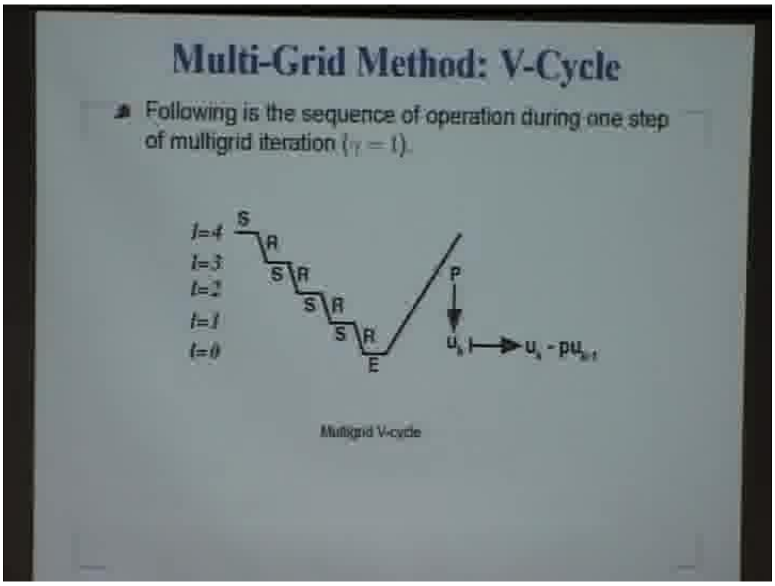
So, this is a V-cycle operation, it is very simple. You can see it just appears like the letter V and it is easiest code, but these are some fancy method this is what is called as a W-cycle and the W-cycle implies that it has got something like a shape of a W here it is a basically a distorted W.

What we have done again? We have let us say started from l equal to 4, then do some smoothing followed by restriction, smoothing restriction and all the way we come down to the 0 th level, where we have the exact solution then we prolong it back to the next finer grid l equal to 1.

Once, we reach there again we do some smoothing operation there. Then again we come back to the finest grid and this process keeps on happening again and again and then essential idea is that how you design the cycle depends on how many iterations you are doing at some particular level. So, here we have defined the iteration by saying that we will do 2 iterations at level 2.

So, 2 iterations at level 2 in 1 cycle this is what? It means we have 2 suffixes here. At l equal to 2 we visit twice again once, you do it coming down from the fine grid but, that is not counted, but this is what we are counting or you could even say that this is something like you are at l equal to 3 level we are doing it once more.

(Refer Slide Time: 35:28)



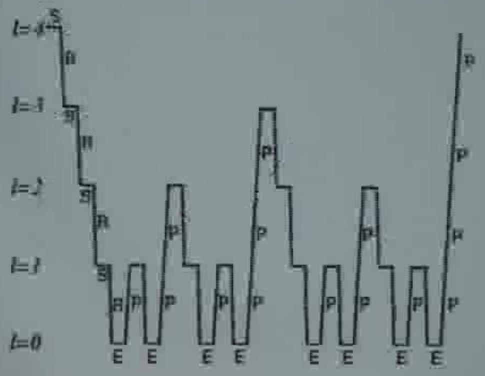
Multi-Grid Method: V-Cycle

- Following is the sequence of operation during one step of multigrid iteration ($\gamma = 1$).

Multigrid V-cycle

(Refer Slide Time: 35:36)

Multi-Grid Method: W-Cycle



Multigrid W-cycle (γ = 2) involving 2 iterations at level-2

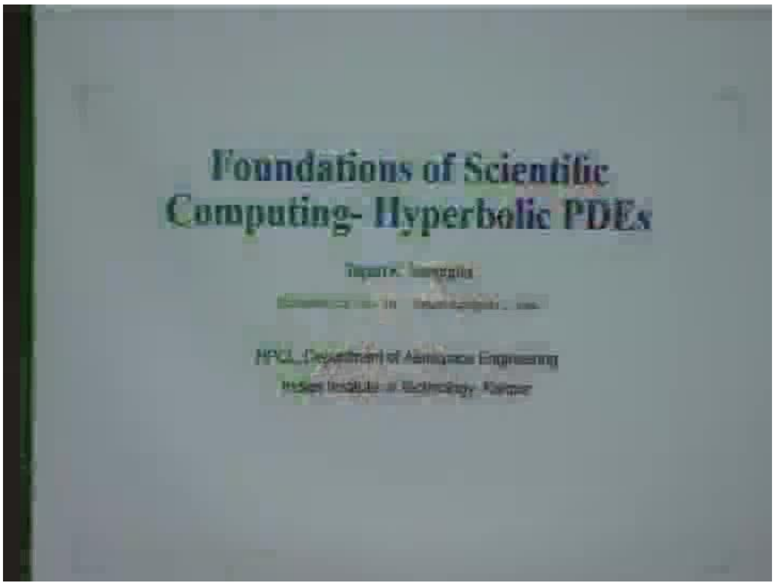
You can see that you can design such W-cycle this is one example of two iterations at level 2. I think one could similarly define other W-cycles I could have just simply removed this. I could have done just simply 1 at l equal to 3 1 and then what we could have done is we could have removed this path of the cycle. So, we could have come here and from there we could have gone to l equal to 3 again we could form and done this bit and then that would have been a perfect W; here it is looking like this.

Suppose, I have been able to communicate to you that so far when it comes to elliptic PDE although it does show that the solution is dictated by the boundary condition, but the way we solve it we actually involuntarily include some kind of a time variation. The moment you do that you have basically looking at how error propagates in the domain as a function of space and time.

These actually like it or not wave like, the waves that we have been talking about and that is the reason that you could understand why we could explain the ADI or Multi-Grid method in terms of various harmonic components of waves. In fact if you actually do this sort of calculation, I can just simply suggest that you take a Laplace’s equation and we solve it.

What you would notice at every level of your calculation if you identify where your maximum error is? You will see which iteration there would be some this maximum is actually propagating from one of the boundary and going towards the other boundary.

(Refer Slide Time: 37:47)



Foundations of Scientific Computing- Hyperbolic PDEs

Tapan K. Sengupta

[illegible]

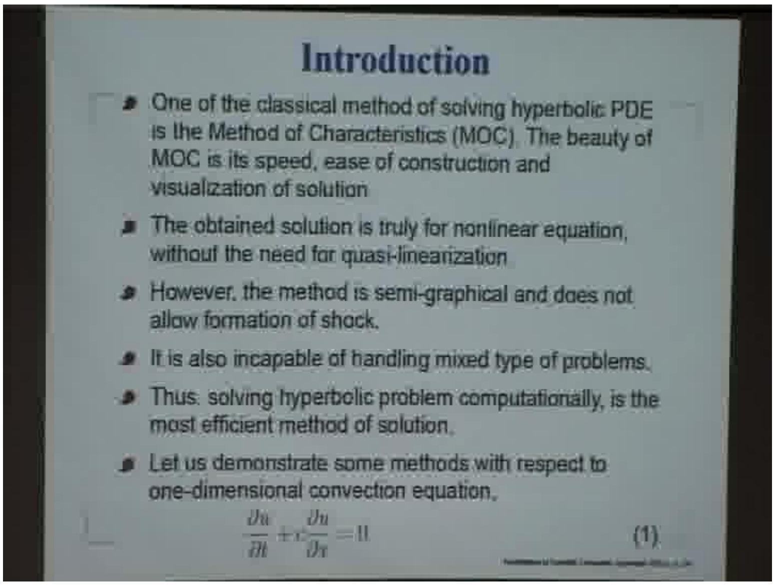
HPCL, Department of Aerospace Engineering

Indian Institute of Technology, Kanpur

So, you can literally see these errors are propagating like waves and this is not a matter of imagination, you can actually do it. You can see it would be a very nice exercise for one to see that you can see again a case where we go from so called time independent problem to a time dependent scenario. We track and control the error that is what it amounts to; error propagation is central to any scientific computing you cannot escape that fact.

Now, let us look at this case, we have been taking a look at quite on and off that is why I will not spent too much of time may be whatever is needed today and little bit of time in the next class, we are looking at hyperbolic PDEs.

(Refer Slide Time: 38:26)



Introduction

- One of the classical method of solving hyperbolic PDE is the Method of Characteristics (MOC). The beauty of MOC is its speed, ease of construction and visualization of solution
- The obtained solution is truly for nonlinear equation, without the need for quasi-linearization
- However, the method is semi-graphical and does not allow formation of shock.
- It is also incapable of handling mixed type of problems.
- Thus, solving hyperbolic problem computationally, is the most efficient method of solution.
- Let us demonstrate some methods with respect to one-dimensional convection equation.

$$\frac{\partial u}{\partial t}+c\frac{\partial u}{\partial x}=0 \qquad (1)$$

Here, we have characteristics which are real and we would like to basically make some introductory comments. One of the classical methods of solving hyperbolic PDE is what is called as method of characteristics. So, there actually we track the solution along this characteristics and it is a very elegant method, because of its speed and because of the ease with which you can really construct the solution. It is also provides you a glimpse of the way the solution actually propagates.

This obtained solution can be look for a non-linear equation without the need for quasi-linearization. This is rather important, because so far we have seen that whenever we encounter any class of equations if they happen to be non-linear, we have to sort of quasi-linearize like what we have seen also even when we are doing a time advancement of solutions; we actually probably lag one part of the solution at the earlier time level.

The same thing can happen when you have non-linear term you can take this product term, take one of the term at the previous low level of solution times. Another part which is at the current level, where you are looking for that kind of problem does not actually arise for method of characteristic. But, this is the problem side is that it is a semi graphical method you have to draw it and you keep following this characteristics. Whenever this characteristic actually of the same family tries to merge together then what you have is called the shock.

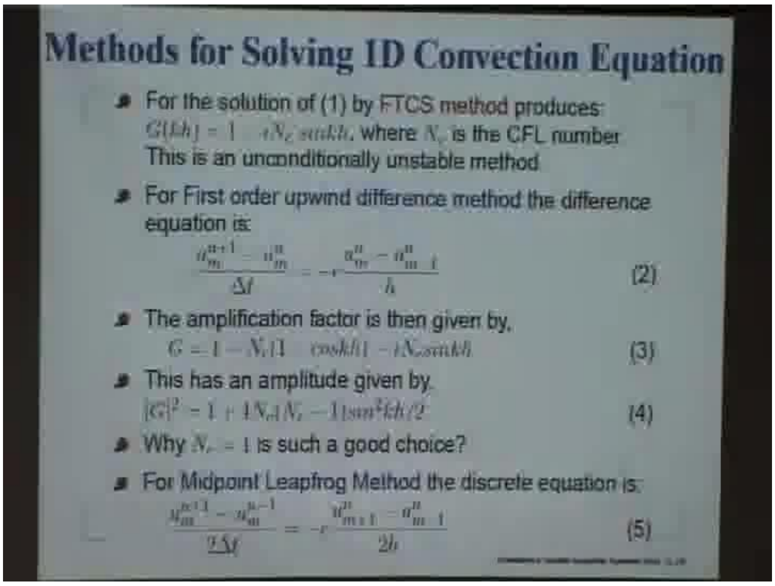
So that is the genuine problem especially in problems of fluid mechanics. Other non-linear dynamics you actually can see this formation of shock is a real phenomenon; so, it cannot handle those kinds of problems. Another issue is that in many problems that we encounter in real life, they are not strictly governed by one equation type in the whole domain that we have already seen from our exercise.

When we looked at the boundary layer that inside the boundary layer we have one characteristics of the problem, outside the boundary layer we have different type of problem, this is what is called as mixed type of problems. Whenever you have mixed type of problems where solution types changes from one to other in different part of the domain, you cannot handle those things rather easily at all with the method of characteristics.

It actually took about 40-50 years for people to really handle such issue in a clean manner. When you have a domain and flow problems especially let us say this was a major issue in aerospace engineering, where we have been trying to look at say solving flow pass triple aircraft. Then we found that some part you get the shock informing, some part is just simply almost like a uniform flow. So that was a major challenge and it took almost 70-80 years for people to realize how to get around.

So mixed type of problem is a very tough problem, which was only solved in early 1970s. Now, we can do something much better than what we did at that time. Basically, when it comes to any real life problem, it is always better that you do not call back upon this classical method of characteristics. You try to solve it computationally there are quite a few efficient methods of solution.

(Refer Slide Time: 42:54)



Methods for Solving 1D Convection Equation

- For the solution of (1) by FTCS method produces: $G(kh) = 1 - iN_c \sin kh$, where N_c is the CFL number This is an unconditionally unstable method
- For First order upwind difference method the difference equation is:

$$\frac{u_m^{n+1} - u_m^n}{\Delta t} = -c\frac{u_m^n - u_{m-1}^n}{h} \qquad (2)$$

- The amplification factor is then given by,

$$G = 1 - N_c(1 - \cos kh) - iN_c \sin kh \qquad (3)$$

- This has an amplitude given by

$$|G|^2 = 1 + 4N_c(N_c - 1)\sin^2 kh/2 \qquad (4)$$

- Why $N_c = 1$ is such a good choice?
- For Midpoint Leapfrog Method the discrete equation is:

$$\frac{u_m^{n+1} - u_m^{n-1}}{2\Delta t} = -c\frac{u_{m+1}^n - u_{m-1}^n}{2h} \qquad (5)$$

One of the simplest examples of hyperbolic equation is this 1D convection equation that we have seen. Let me just simply go through and tell you we already know that FTCS method, this is the forward in time central space method. We find that G amplification factor is given by this, where N c is basically the CFL number which you have familiar with; we are not talking about it because this is a conditionally unstable method, it is going to be of any use at all.

Since, the solution is propagating from left to right with the speed of C; there is a possibility that you could actually adopt an upwind scheme. A first order upwind scheme as given by equation 2 is physically consistent because your information is propagating from left to right.

Now that is what we have done if you look the spatial derivative on the right hand side that we are seeing that del u del x come has been written as u m minus u m minus 1 by h. This is something we must realize that please do not be ((suave)) by this opinion of the order and if it is higher or lower it does not matter what is more important that you follow the physics.

In this case, if you take this first order of upwind scheme that we have written here in equation 2, you get the amplification factor that is given in equation 3. What you notice that you have a real part and the real part seems to indicate that G is less than 1, but at the same time you have an imaginary part and what you can do is you can calculate the modulus of it; if you do you get equation 4.

If you recall when we talk about that error propagation equation, we specifically noted that if we have a method where G is equal to 1 that actually is a low error method, because G not equal to 1 actually gives rise to error component to that term if you recall that we had l n of modulus of G. If you can have G is equal to 1, then you are really and see a very interesting thing, now that you have done some time dependent calculation; you can really see that here you could take N c equal to 1 and you get fantastic result.

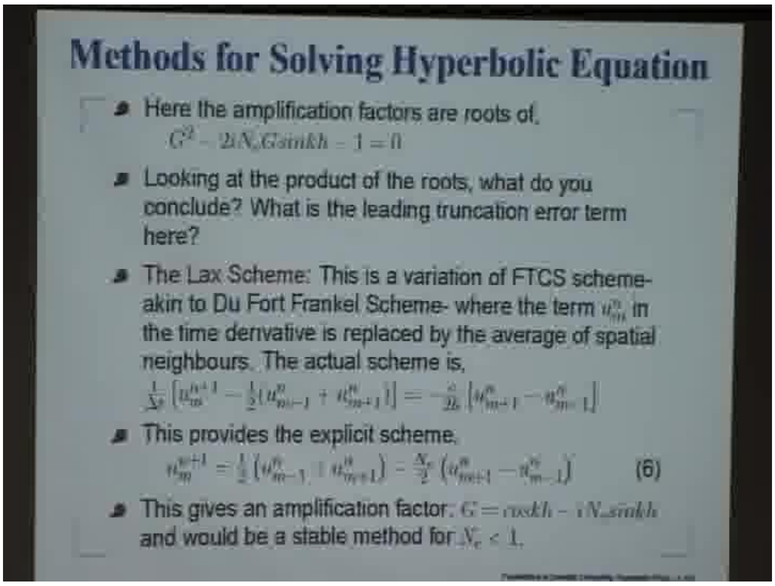
N c equal to 1 is not a very trivial thing when I do not know some of you may have taken N c as low as 10 to the power minus 2, minus 3, minus 4. We do some real life calculations now with N c as low as 10 to the power minus 5, minus 6. So, compare to that this N c equal to 1 is a good choice, question is why? Mathematically you can see mod G equal to 1, but physically you can also reason it out by looking at it.

The way what you are doing that every time step delta t your solution should go what distance c into delta t. If c into delta t is also happens to be your delta h, then you are actually going from a right point to another right point. So that happens to be what if c delta t equal to delta h that N c is equal to 1. You can see that although it is a first order method this happens to be the most correct method in fact, it is going to be the exact method for this equation.

Suppose, somebody tries to convince you, you know this is not so good, you are using a first order method, it is not necessary because if you take a higher order method you will actually incur error. So that is what is the situation, if you follow the physic then just simply in a simple minded manner the mathematics you actually gain, so following physics is so important.

Now, the last line that we have here talks about a method which is the so called second order accurate method both for space and time discretization. This is what is called as midpoint Leapfrog method and I am referring to it because in almost all the weather forecasting code those have been use which does not invoke. Let us say this is the spam’s their, this happens to be a kind of a statement of faith everybody uses this, why? That is what we will be talking about now and somewhat later.

(Refer Slide Time: 47:55)



Methods for Solving Hyperbolic Equation

- Here the amplification factors are roots of,
  $G^2 - 2iN_c G \sin kh - 1 = 0$
- Looking at the product of the roots, what do you conclude? What is the leading truncation error term here?
- The Lax Scheme: This is a variation of FTCS scheme- akin to Du Fort Frankel Scheme- where the term u_m^n in the time derivative is replaced by the average of spatial neighbours. The actual scheme is,
  $\frac{1}{\Delta t}\left[u_m^{n+1} - \frac{1}{2}(u_{m-1}^n + u_{m+1}^n)\right] = -\frac{a}{2h}\left[u_{m+1}^n - u_{m-1}^n\right]$
- This provides the explicit scheme,
  $$u_m^{n+1} = \frac{1}{2}\left(u_{m-1}^n + u_{m+1}^n\right) - \frac{N_c}{2}\left(u_{m+1}^n - u_{m-1}^n\right) \qquad (6)$$
- This gives an amplification factor: $G = \cos kh - iN_c \sin kh$ and would be a stable method for $N_c < 1$.

This is the discrete equation that you are seeing an equation 5 and then you can construct the governing equation for the numerical amplification factor. Now, you will be tempted to conclude like what we concluded while discussing the parabolic equation that G 1 into G 2 would be equal to minus 1. We say like if one is stable, the other will be unstable but, please do not disquiet by that because if you do what you are going to do is you will be drawing a wrong conclusion. In this case it so happens at both the modes is 1, so the product is 1.

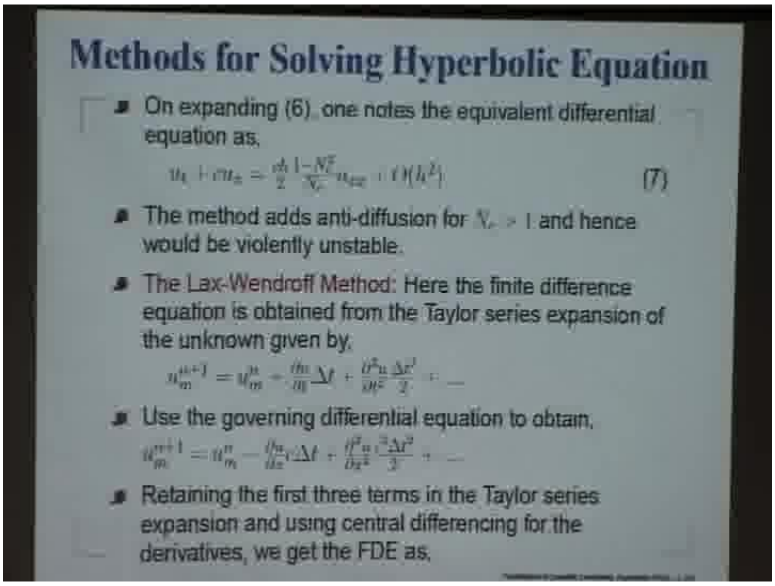
It is one of the very interesting methods, that is why it is used in all weather prediction codes. Even today people do use leapfrog method the once that use this kind of time discretization. So, I ask questions I answered the first, you can look at the second question, you look at the leading truncation error term and figure out what is happening?

Then there are other methods suggested, this method is due to eta lax is basically a variation of the forward in time and centering space scheme similar to like what we did in Du Fort Frankel scheme. What we do here in the time derivative term here, you can see there this half term that is appearing here, this term is nothing but, actually u n at m (Refer Slide Time: 49:48). I have just taken it as average of 2 neighbors, this is exactly what was done in Du Fort Frankel method 2 and that is what was proposed by lax.

If you do that this gives you an explicit scheme, which is given by equation 6 and work out the amplification factor that would be cosine of k h minus i N c time k h. Once again you can see here, if you choose N c equal to 1 you will get mod G equal to 1. So that was what really prompted lax to suggest this method that you could actually end up with a neutrally stable method for N c equal to 1. Not only that, you would also get G less than 1, so called stable method if you restrict your N c to less than 1.

So N c above 1 is forbidden you cannot take N c greater than 1 then you would be having numerical instability. If you keep N c below 1, then you have so called stable method and N c equal to 1 will give you a neutrally stable method.

(Refer Slide Time: 51:06)



Methods for Solving Hyperbolic Equation

- On expanding (6), one notes the equivalent differential equation as,

$$u_t + c u_x = \frac{ch}{2}\frac{1-N_c^2}{N_c} u_{xx} + O(h^2) \quad (7)$$

- The method adds anti-diffusion for $N_c > 1$ and hence would be violently unstable.
- The Lax-Wendroff Method: Here the finite difference equation is obtained from the Taylor series expansion of the unknown given by,

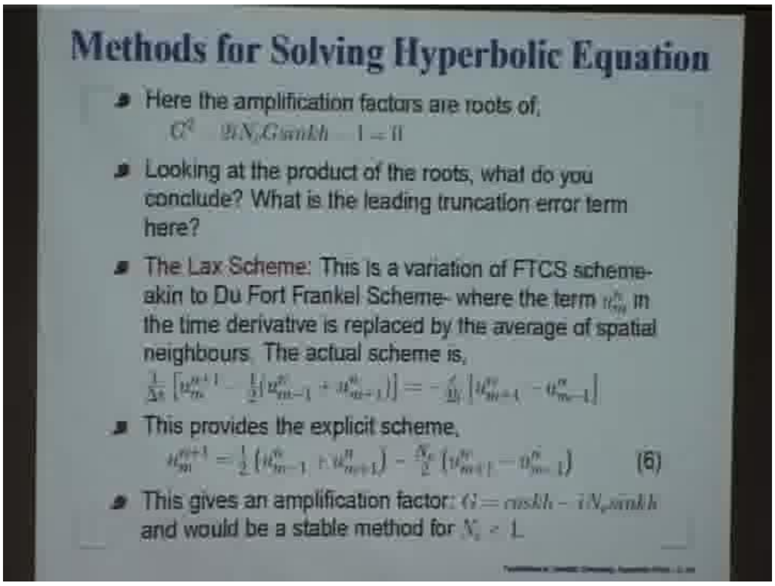
$$u_m^{n+1} = u_m^n + \frac{\partial u}{\partial t}\Delta t + \frac{\partial^2 u}{\partial t^2}\frac{\Delta t^2}{2} + \dots$$

- Use the governing differential equation to obtain,

$$u_m^{n+1} = u_m^n - \frac{\partial u}{\partial x} c\Delta t + \frac{\partial^2 u}{\partial x^2}\frac{c^2\Delta t^2}{2} + \dots$$

- Retaining the first three terms in the Taylor series expansion and using central differencing for the derivatives, we get the FDE as,

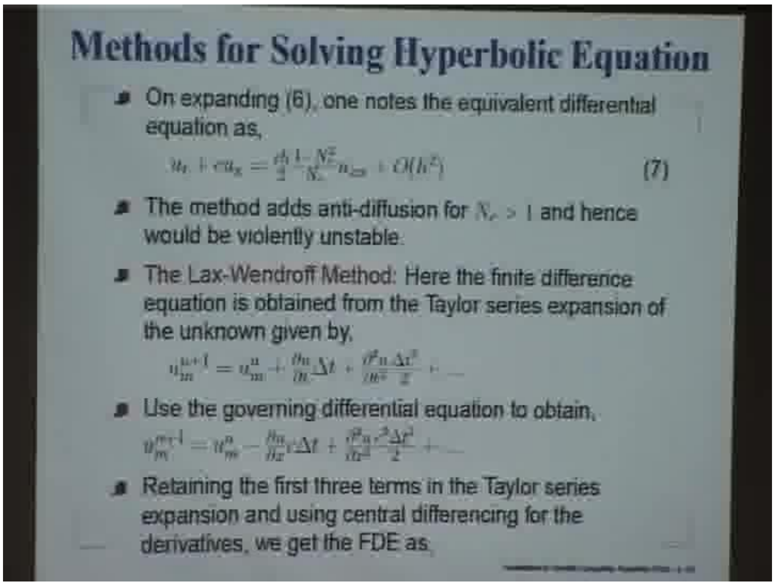
(Refer Slide Time: 51:14)



Methods for Solving Hyperbolic Equation

- Here the amplification factors are roots of,
  $G^2 - 2iN_cG\sin kh - 1 = 0$
- Looking at the product of the roots, what do you conclude? What is the leading truncation error term here?
- The Lax Scheme: This is a variation of FTCS scheme- akin to Du Fort Frankel Scheme- where the term u_m^n in the time derivative is replaced by the average of spatial neighbours. The actual scheme is,
  $\frac{1}{\Delta t}\left[u_m^{n+1} - \frac{1}{2}(u_{m-1}^n + u_{m+1}^n)\right] = -\frac{c}{2h}\left[u_{m+1}^n - u_{m-1}^n\right]$
- This provides the explicit scheme,
  $u_m^{n+1} = \frac{1}{2}(u_{m-1}^n + u_{m+1}^n) - \frac{N_c}{2}(u_{m+1}^n - u_{m-1}^n)$ (6)
- This gives an amplification factor: $G = \cos kh - iN_c\sin kh$ and would be a stable method for $N_c < 1$.

(Refer Slide Time: 51:19)



Methods for Solving Hyperbolic Equation

- On expanding (6), one notes the equivalent differential equation as,
  $u_t + cu_x = \frac{ch}{2}\frac{1-N_c^2}{N_c}u_{xx} + O(h^2)$ (7)
- The method adds anti-diffusion for $N_c > 1$ and hence would be violently unstable.
- The Lax-Wendroff Method: Here the finite difference equation is obtained from the Taylor series expansion of the unknown given by,
  $u_m^{n+1} = u_m^n + \frac{\partial u}{\partial t}\Delta t + \frac{\partial^2 u}{\partial t^2}\frac{\Delta t^2}{2} + \ldots$
- Use the governing differential equation to obtain,
  $u_m^{n+1} = u_m^n - \frac{\partial u}{\partial x}c\Delta t + \frac{\partial^2 u}{\partial x^2}\frac{c^2\Delta t^2}{2} + \ldots$
- Retaining the first three terms in the Taylor series expansion and using central differencing for the derivatives, we get the FDE as,

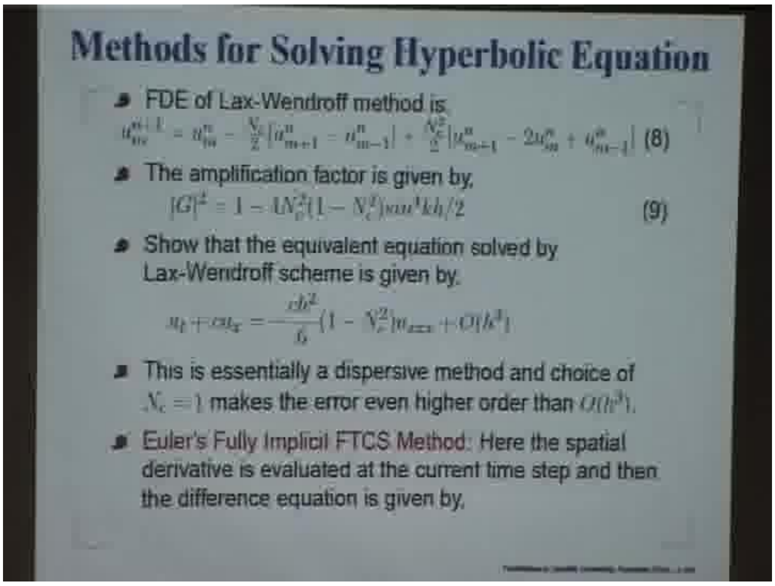
So what you could do? You could basically look at the previous equation 6 and write out what is the equivalent equation that you are solving; that happens to be this. So, you ended up solving this equation u t plus c u x equal to 0 but, you add up this term. You can see that this added term actually add that is special because that is what you accept.

That is the story why it is stable for N c less than 1 but, if you take N c equal to 1 you are not adding anything because 1 minus N c square will take it to 0 contributions from the right hand side. Then you will be consistently what you have started with and you can also see why N c greater than 1 gives raise to anti-diffusion that is why it is violently unstable.

Some out of a different proof you have suggested in this Lax-Wendroff method that we discuss next. What we actually do is we have the Taylor series expansion of the load l value at the advance time level in terms of the load l value at the previous time level and various time derivatives. Now, suppose you choose a differential equation like the 1D convection equation that the del u del t I could write in terms of minus c del u del x and so on so forth.

So what Lax-Wendroff suggested that look I mean FTCS blows how blows up, you do not stop at 2 terms. So, you take the additional term because that will have this stabilizing in terms of u x, x term coming from the third. So what you do? You retain these 3 terms in the Taylor series expansion and use the central difference for all of it because at the formulation level you have added the dissipation.

(Refer Slide Time: 53:22)



Methods for Solving Hyperbolic Equation

- FDE of Lax-Wendroff method is

$$u_m^{n+1} = u_m^n - \frac{N_c}{2}[u_{m+1}^n - u_{m-1}^n] + \frac{N_c^2}{2}[u_{m+1}^n - 2u_m^n + u_{m-1}^n] \quad (8)$$

- The amplification factor is given by,

$$|G|^2 = 1 - 4N_c^2(1 - N_c^2)\sin^4 kh/2 \quad (9)$$

- Show that the equivalent equation solved by Lax-Wendroff scheme is given by,

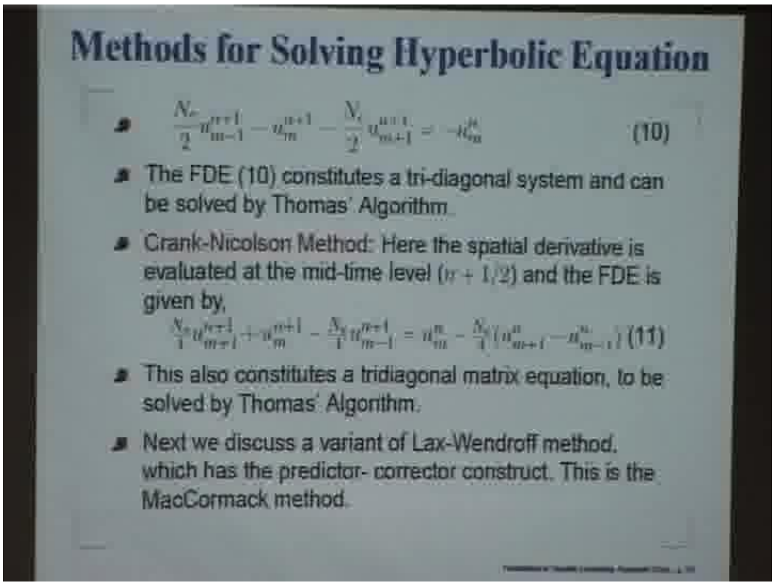
$$u_t + cu_x = -\frac{ch^2}{6}(1 - N_c^2)u_{xxx} + O(h^3)$$

- This is essentially a dispersive method and choice of $N_c = 1$ makes the error even higher order than $O(h^3)$.
- Euler's Fully Implicit FTCS Method: Here the spatial derivative is evaluated at the current time step and then the difference equation is given by,

When you are discretizing you do not have to do extra, you do it central difference that is the whole idea of Lax-Wendroff method. The finite difference equation for this Lax-Wendroff method looks like this. You look at the amplification factor given by equation 9 and the interesting bet is that if you try to find out what is the equivalent equation that you are solving in Lax-Wendroff scheme and that is interesting.

Because you can notice that the added term is basically a dispersive term because added term is the third derivative of u with respect to x, it is not the second derivative. That is why we make the observation that this is an essentially dispersive method and once again choice of N c equal to 1 makes the error higher order. So, there will not be having this term will go up to the leading error would be of order h cube.

(Refer Slide Time: 54:32)

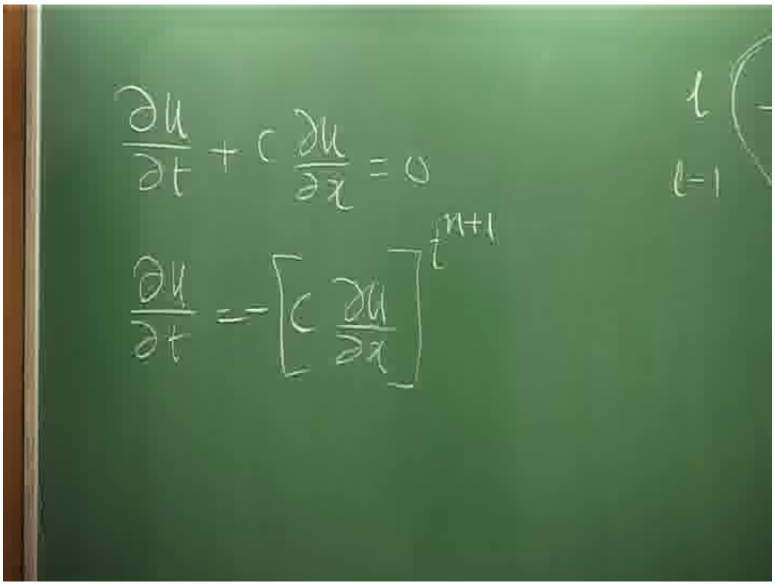


Methods for Solving Hyperbolic Equation

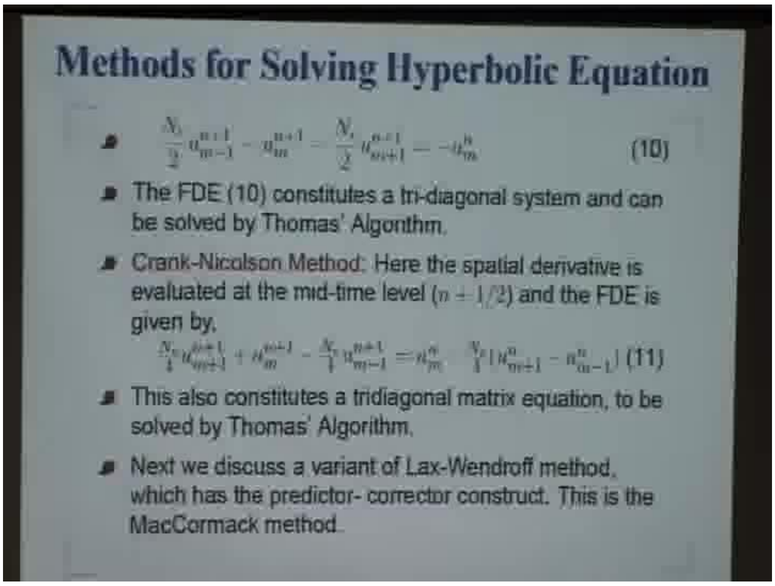
- $$\frac{N_c}{2}u_{m-1}^{n+1} - u_m^{n+1} - \frac{N_c}{2}u_{m+1}^{n+1} = -u_m^n \qquad (10)$$
- The FDE (10) constitutes a tri-diagonal system and can be solved by Thomas' Algorithm.
- Crank-Nicolson Method: Here the spatial derivative is evaluated at the mid-time level $(n + 1/2)$ and the FDE is given by,
  $$\frac{N_c}{4}u_{m+1}^{n+1} + u_m^{n+1} - \frac{N_c}{4}u_{m-1}^{n+1} = u_m^n - \frac{N_c}{4}(u_{m+1}^n - u_{m-1}^n) \qquad (11)$$
- This also constitutes a tridiagonal matrix equation, to be solved by Thomas' Algorithm.
- Next we discuss a variant of Lax-Wendroff method, which has the predictor- corrector construct. This is the MacCormack method.

Suppose these are the some of the explicit method so, the way that we have talked about implicit method here also we can do the same. The Euler's fully implicit FTCS method is given by this so, what we have done here? We have actually written down this equation usually what we do?

(Refer Slide Time: 54:50)


$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$
$\frac{\partial u}{\partial t} = -\left[c\frac{\partial u}{\partial x}\right]^{n+1}$
l
$l-1$

(Refer Slide Time: 55:22)

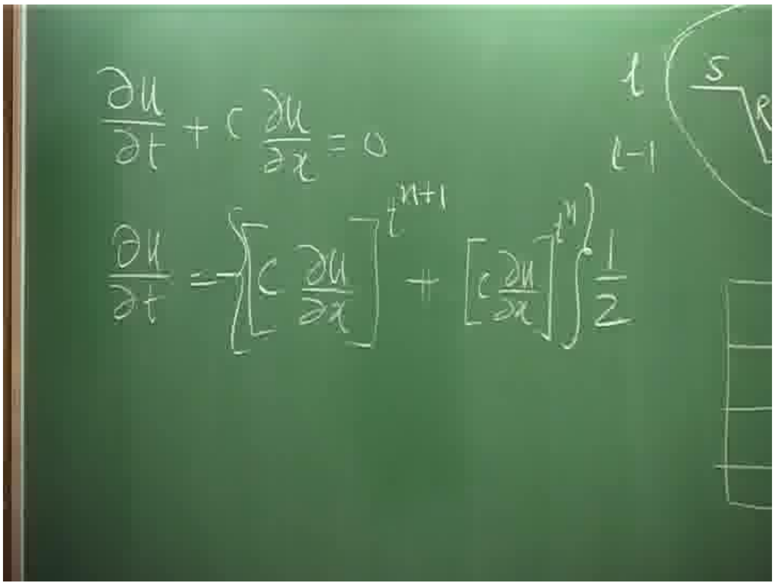


Methods for Solving Hyperbolic Equation

- $\frac{N_c}{2}u_{m-1}^{n+1} - u_m^{n+1} - \frac{N_c}{2}u_{m+1}^{n+1} = -u_m^n$ (10)
- The FDE (10) constitutes a tri-diagonal system and can be solved by Thomas' Algorithm.
- Crank-Nicolson Method: Here the spatial derivative is evaluated at the mid-time level ($n+1/2$) and the FDE is given by,
  $\frac{N_c}{4}u_{m+1}^{n+1} + u_m^{n+1} - \frac{N_c}{4}u_{m-1}^{n+1} = u_m^n - \frac{N_c}{4}(u_{m+1}^n - u_{m-1}^n)$ (11)
- This also constitutes a tridiagonal matrix equation, to be solved by Thomas' Algorithm.
- Next we discuss a variant of Lax-Wendroff method, which has the predictor- corrector construct. This is the MacCormack method.

In the explicit method we did that this we did it at t n, but in the implicit method what you do? You do it also at n plus 1. Then when you discretizing it and you write down the finite difference equation you get equation 10 that is what you get. For this equation it is rather simple because you can clearly see that it will give rise to a tri-diagonal matrix (Refer Slide Time: 55:05). The diagonal term is minus 1 and half diagonal term, sub diagonal term, is plus N c by 2, super diagonal term is minus N c by 2 and you can actually use the Thomas algorithm for this.

(Refer Slide Time: 56:00)

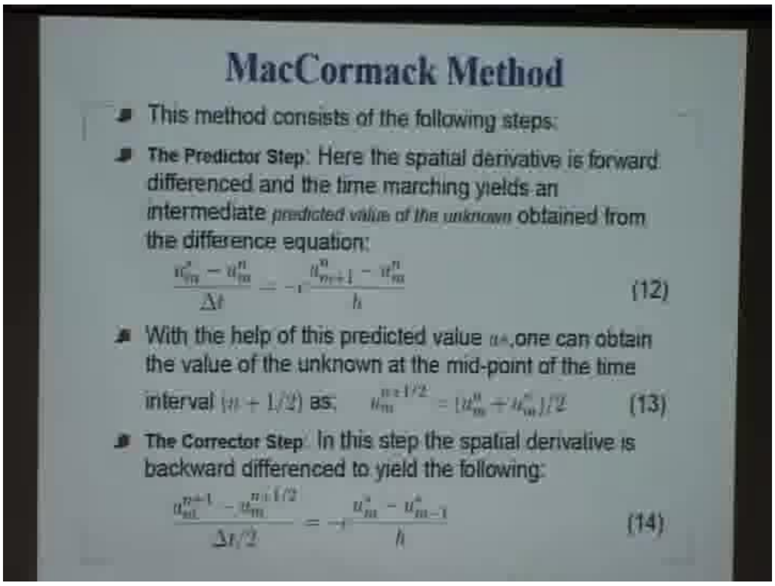


$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0$$

$$\frac{\partial u}{\partial t} = -\left\{\left[c\frac{\partial u}{\partial x}\right]_i^{n+1} + \left[c\frac{\partial u}{\partial x}\right]_i^{n}\right\}\frac{1}{2}$$

Now, people have also done all kinds of other things; one of it is basically doing a Crank-Nicolson kind of well, we can do that. So, this we do it like this multiplied by half. We can also do that average value of the quantity at n th level and n plus 1 th level and that would give raise to finite difference equation of the kind given in equation 11 here. This also is a tridiagonal matrix equation, this also can solve by Thomas algorithm; this is what has been the sort of a scheme of solving hyperbolic equation. However, in late 1960s, MacCormack proposed some method which is based on the predicted character frame work and this is what is done.

(Refer Slide Time: 56:55)



MacCormack Method

- This method consists of the following steps:
- **The Predictor Step**: Here the spatial derivative is forward differenced and the time marching yields an intermediate *predicted value of the unknown* obtained from the difference equation:

$$\frac{u^*_m - u^n_m}{\Delta t} = -c\frac{u^n_{m+1} - u^n_m}{h} \qquad (12)$$

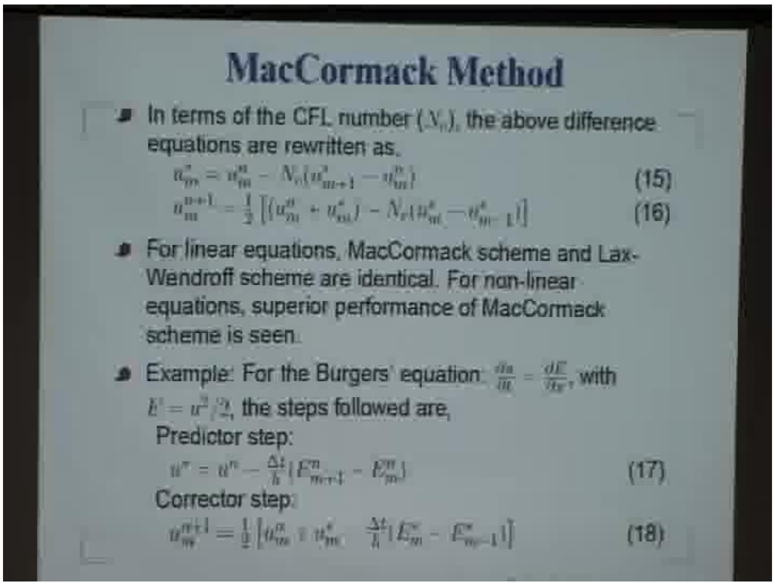
- With the help of this predicted value $u*$,one can obtain the value of the unknown at the mid-point of the time interval $(n + 1/2)$ as: $u^{n+1/2}_m = (u^n_m + u^*_m)/2$ (13)
- **The Corrector Step**: In this step the spatial derivative is backward differenced to yield the following:

$$\frac{u^{n+1}_m - u^{n+1/2}_m}{\Delta t/2} = -c\frac{u^*_m - u^*_{m-1}}{h} \qquad (14)$$

So, you have 2 steps and you have a predictor step, where you actually take the spatial derivative in a forward difference form. If you perform time matching in that sequence that you get a predicted value of the unknown let us say for this equation it will give you u star; u star is the predicted value.

Now, with the help of this predicted value u star you can actually go over to the new level u n plus 1, where the predicted values have been used on the right hand side. In the second part of the corrector step, you are going from n plus half level to n plus 1. So, this n plus half value is a kind of an average value between what is your starting get value and the predicted value averaged.

(Refer Slide Time: 58:04)



MacCormack Method

- In terms of the CFL number (N_c), the above difference equations are rewritten as,

$$u^*_m = u^n_m - N_c(u^n_{m+1} - u^n_m) \quad (15)$$

$$u^{n+1}_m = \frac{1}{2}\left[(u^n_m + u^*_m) - N_c(u^*_m - u^*_{m-1})\right] \quad (16)$$

- For linear equations, MacCormack scheme and Lax-Wendroff scheme are identical. For non-linear equations, superior performance of MacCormack scheme is seen.
- Example: For the Burgers' equation $\frac{\partial u}{\partial t} = \frac{\partial E}{\partial x}$, with $E = u^2/2$, the steps followed are,

  Predictor step:

$$u^* = u^n - \frac{\Delta t}{h}(E^n_{m+1} - E^n_m) \quad (17)$$

  Corrector step:

$$u^{n+1}_m = \frac{1}{2}\left[u^n_m + u^*_m - \frac{\Delta t}{h}(E^*_m - E^*_{m-1})\right] \quad (18)$$

This is in the Mac Cormack scheme where you do it in 2 steps and in terms of the CFL number you could write down the algorithm in terms of equation 15 and 16. It is easy to show that if I have a linear equation that this Mac Cormack scheme and Lax-Wendroff scheme; they are basically identical. It is only for non-linear equation that you derive more benefit from Mac Cormack.

I think I will stop here. I will put this up; it is there only; the example that is left, you can read it up there and these matters will be up on the site by this evening.