

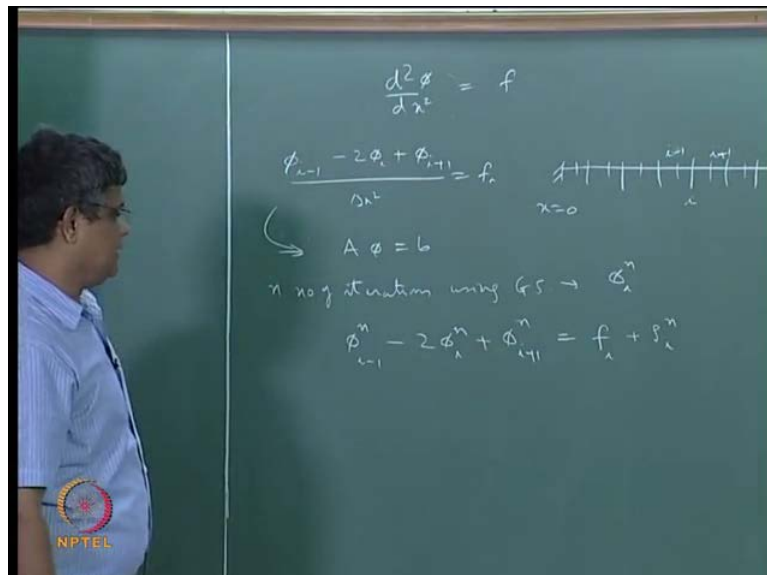
**Computational Fluid Dynamics**  
**Prof. Sreenivas Jayanti**  
**Department of Chemical Engineering**  
**Indian Institute of Technology, Madras**

**Solution of Linear Algebraic Equations**  
**Lecture No. # 30**

**Illustration of Multigrid Method for the Laplace equation**

We have discussed enough about the philosophy of the multigrid approach. Let us try to increase our understanding by taking a very simple example and see how we go about the solution of this. So, we will take our well known familiar example of Laplace equation, which we know is not irrelevant in the context of, for example, the solution for the pressure correction or the pressure equation approach. And we look at the one-dimensional form, and then, we can see how it can be extended to the two-dimensional form.

(Refer Slide Time: 00:47)



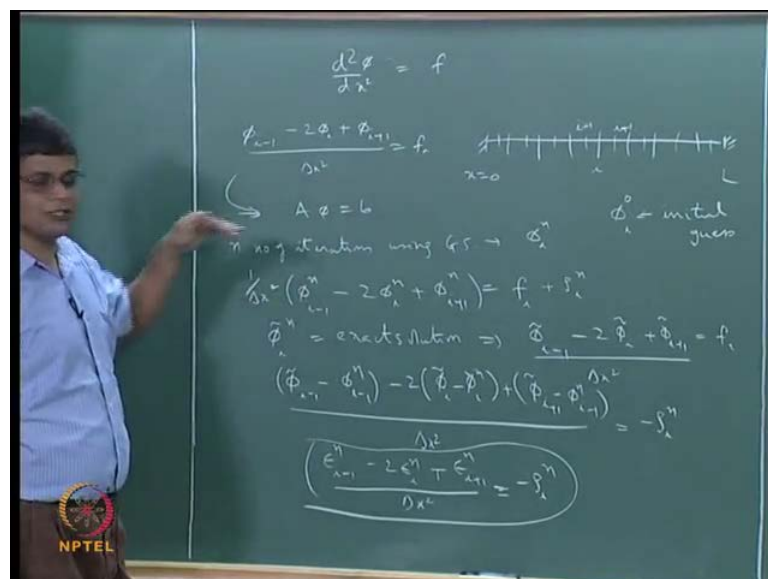
So, we are looking at an equation like  $d^2 \phi / dx^2 = f$ ; this is the equation. It is one dimensional; so, we have an  $x$  domain going from 0 to capital  $L$ , which we break into so many control like that. And we can say that, this is our point  $i$

here, and this is  $\phi_{i+1}$  minus  $\phi_{i-1}$ , and  $\phi_{i+1}$  and so on. And we can make use of central difference of approximation for this and write a discretized form of this as,  $\phi_{i-1} - 2\phi_i + \phi_{i+1} = \Delta x^2 f_i$ . And we can convert this into  $A\phi = b$ , incorporating the appropriate boundary conditions, at  $x = 0$  and  $x = L$ . So, this is the linear algebraic system of equations that we want to solve using the multi-grade approach.

So, let us say that we have run this for  $n$  number of iterations, using the Gauss Seidel method. At the end of this, we have a solution  $\phi_i^n$ , where  $i$  covers all the points upto this. This is not necessarily the exact solution, because there is some residual; and if you were to substitute this here, you would get  $\phi_{i-1}^n - 2\phi_i^n + \phi_{i+1}^n$ , and this would be exactly equal to  $f_i$  plus some residual  $\rho_i^n$ .

If the residual is 0, then we have the exact solution. So, right now what we have is this equation. And the current solution  $\phi_i^n$  at the end of  $n$  number of iterations using Gauss Seidel method, does not satisfy the governing equation exactly - the discretized equation exactly - it leaves the residual  $\rho_i^n$  and that  $\rho_i^n$  is the residual at each point at each node, at the end of  $n$ th iteration. If you carry out another one iteration and then get a new solution at  $n+1$ , the residual will change. If you have a converging solution, the residual will decrease and decrease, with increasing number of  $n$  or the number of iteration.

(Refer Slide Time: 04:06)



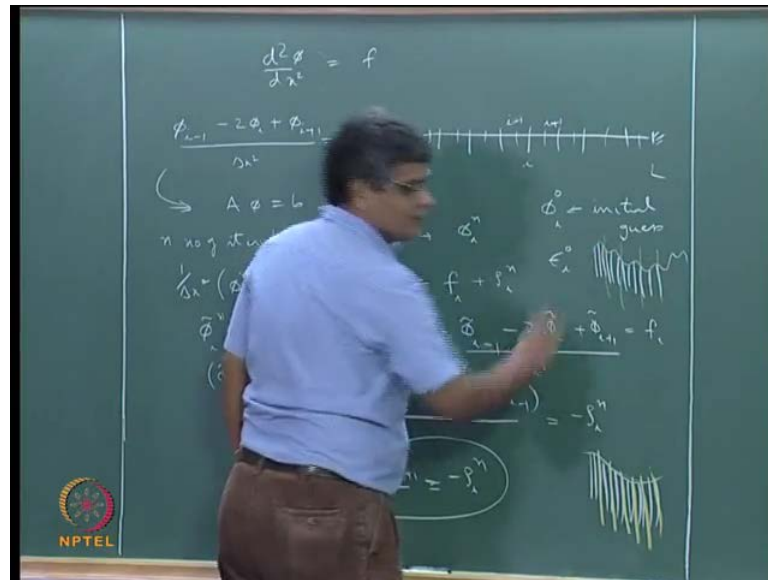
So, if you were to, now let us say that,  $\psi_i$  is the exact solution of this equation of the discretized equation. So, then this means that, we can say that,  $\phi_{i-1} - 2\phi_i + \phi_{i+1}$  is equal to  $f_i$  divided by  $\Delta x^2$  is equal to  $f_i$ , and this is, so that the exact solution obviously satisfies the governing equation.

Now, we can, there is  $1/\Delta x^2$  here. So, we can subtract this from this and that will give us error at  $i$ , we can write it as  $\phi_{i-1} - \phi_{i-1} - 2\phi_i + \phi_{i+1}$  divided by  $\Delta x^2$  is equal to, this  $f_i$  here cancels with this, and we have  $\rho_i$ . And this is nothing but the error at the  $i$ th point. So, this we can say is the error at  $i-1$  point at  $n$ th iteration step, because this exact solution, this is the computed solution at the end of  $n$  iterations. So, this is the error at this nodal value, at this nodal value, at this iteration step; and similarly, this is  $\rho_{i+1}$  representing the error at this nodal value plus error at  $i+1$  divided by  $\Delta x^2$  is  $\rho_{i+1}$ .

So, this is the error at the end of  $n$  iterations using the Gauss Seidel method. And what we are saying is that, at the end of the first iteration or at the beginning, there is some error. At the beginning, you have  $\phi_i^0$ , this is the initial guess; and if you were to substitute this, you would get  $\rho_i^0$ , that is the initial thing and you would have an error distribution at the 0th time step. And as a result of  $n$  number of computations on the error distribution using the Gauss Seidel method, you have smoothed out some of the fine scale variations and then you have come up with smoother distribution of error as given by this, at the end of  $n$  iterations.

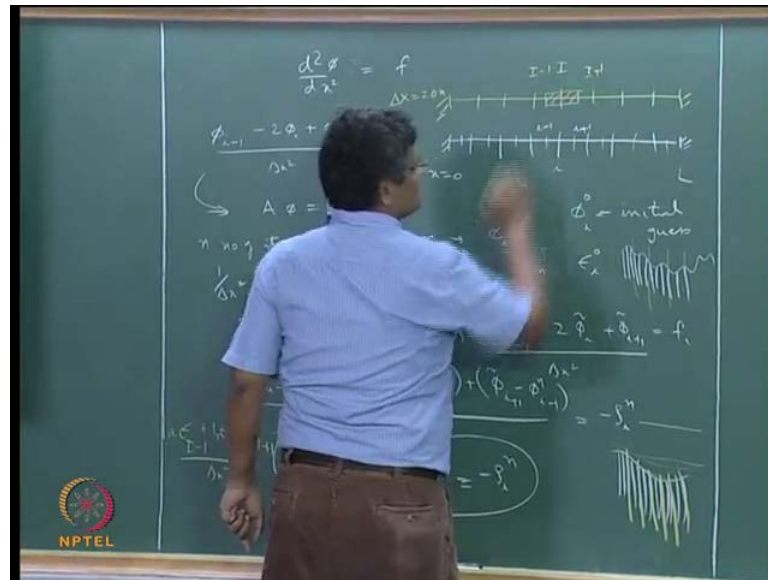
So, now, what you want to do is, that we are claiming that this is smoother than the initial error distribution **and therefore...**

(Refer Slide Time: 07:43)



So, whereas, this variation can be like this, this variation for the same thing will be like that; and since this requires our small delta x here, the same thing will not be showing this much variation. And we will see that, on the same delta x, we have a smooth variation, where therefore we are saying that we do not need such a fine grid, it is sufficient to have may be a grid like this; and using these reduced number of points also, I can write a smooth curve, whereas with this, I were to have grids like this. I will not be able to show variation of this particular thing. So, at the end of n number of iterations, a fine distribution - high frequency distribution - component of the error is reduced, so as to give us to retain only a low frequency error distribution, after n number of iterations of the discretized equation using a smoothing kind of method, for the solution of  $A\phi = b$  like, the Gauss Seidel method or SIF method. Now, what I want to do is, I do not want to do further elimination of the error;

(Refer Slide Time: 09:32)



I would, I ultimately, when I have the exact solution, the error will be 0 and it will not vary with space. I want to go down from here into this, and then finally on to that on to the 0 level. But the point that I would like to say is that, because I have a smooth variation, I can represent this using smaller fewer number of points with increased grid spacing; and then continue further reduction on this smaller number grid points and reach the n distribution, where error at all points is very very low and almost equal to 0. I want to reach that end point quickly, not by doing further computations on the small delta x, but on a larger delta x.

So, what we want to do is, instead of working with further equations on this smaller grid, I want to replace this grid with a bigger grid with a bigger delta x, so that if I take this value here, I drop out this, for example, I can have for every alternate point, I have this. So, this is my capital I on the coarse grid, which corresponds to the small i here, and i plus 1, i minus 1 here on this corresponds to i minus 2, and this is i plus 1 here corresponds i plus 2 here. So, for every two points, I have one grid spacing. So, my delta x delta capital X is twice delta x, just as an example of a coarse grid.

Now, I want to do further error reduction on capital I grid, not on the small i grid. So, I mean, it means that I have to have an error distribution represented like this, in terms of capital I minus 1 and capital i a plus b plus c error i plus 1 divided by delta x square equal to something.

How do I derive this? If I able to derive this equation, then I can work with this equation and reduce the error. In order to do this, I say that the error equation on this coarse grid for a point centered about this covers this domain, and this equation here covers the domain which goes from small a minus half small a plus half like this. So, to this, I need to add half of this domain and half of this domain to get this.

(Refer Slide Time: 12:23)

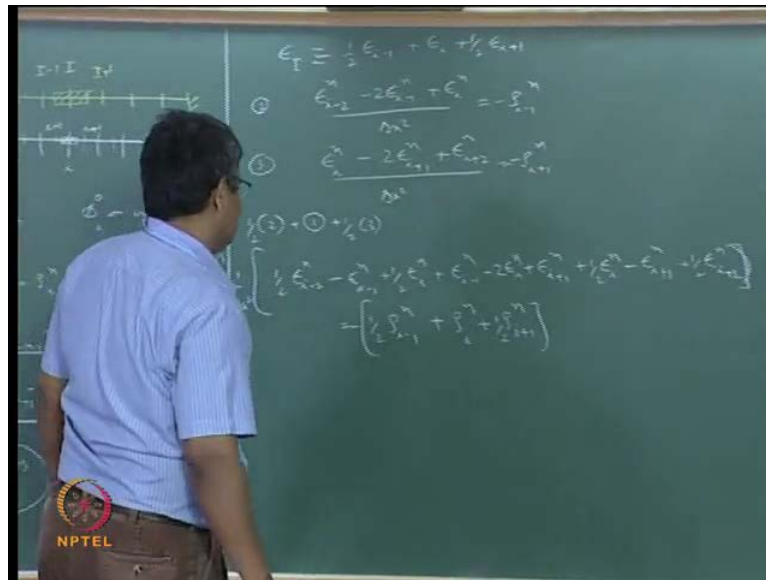
$$E_I = \frac{1}{2} E_{i-1} + E_i + \frac{1}{2} E_{i+1}$$

$$(2) \quad \frac{E_{i-2}^n - 2E_{i-1}^n + E_i^n}{\Delta x^2} = \rho_{i-1}^n$$

$$(3) \quad \frac{E_i^n - 2E_{i+1}^n + E_{i+2}^n}{\Delta x^2} = \rho_{i+1}^n$$

So, I am saying that, I here cover from something like half of error  $i$  minus 1 plus the error equation at  $i$  plus half of, so I am roughly saying this is the equivalence of this; this should give me the overall error value at this point. So, in order to derive an error equation on the coarser grid, I write the error evolution equation at this point, and at this point and this point, and then add them together with this weightage, so as to get overall error here; and so, I take this, let us call this as 1. And equation two will be, the same equation written at  $i$  minus 1, **so that is...** So, this is the error equation giving the residual at  $i$  minus 1. And similarly, if you come to this point here, that will be  $\epsilon_{i-2}^n + \epsilon_{i-1}^n + \epsilon_i^n + \epsilon_{i+1}^n + \epsilon_{i+2}^n$  divided by  $\Delta x^2$  is equal to  $\rho_{i-1}^n$ .

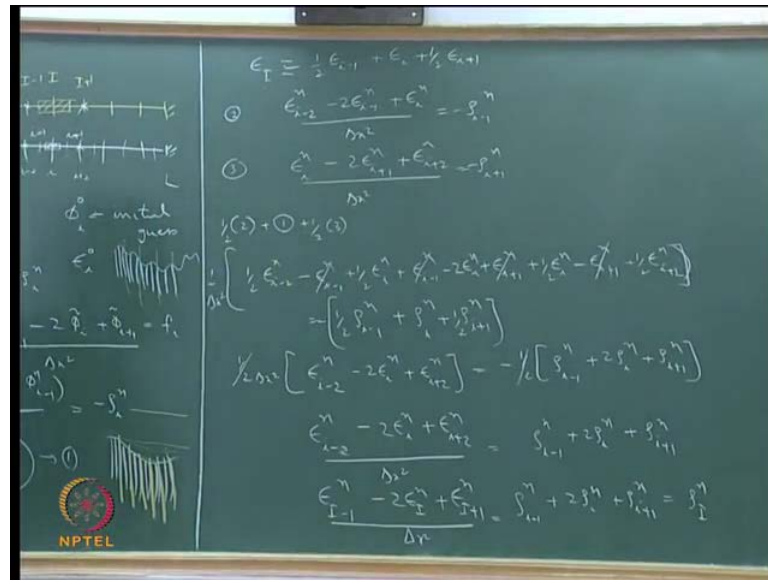
(Refer Slide Time: 14:08)



So, now, I say that, in order to get this one, I take half of 2 plus equation 1 plus half of equation 3. And what will that give me? The delta x square is the same. So, I can say that, this is half of epsilon i minus 2 n, I can say that is 1 by delta x square, minus epsilon i minus 1 n plus half of epsilon i n is what I am getting from this. And the original equation here will contribute, epsilon i minus 1 n minus 2 epsilon i n plus epsilon i plus 1 n and then again half of this plus half epsilon i n minus i minus 1 n i plus one n plus half of epsilon i plus 2 n. This whole thing is equal to half of rho i minus 1 n plus rho i n, we have minus here. So, I have written whatever the error equation I have at point i and at point i minus 1, this is this one, and at point i plus 1, and I have added them together to cover a domain, which is equal to the total domain corresponding to capital I in the coarse grid.

Now, can I do any simplification of this? I have minus i minus 1 epsilon i minus 1 here and this; so, this will cancel out. And I have i plus 1 here and minus of i plus 1. So, this will cancel out; and I have plus half of this and plus half of this, and this is minus 2.

(Refer Slide Time: 16:42)



So, I can write this as, I can now take out half epsilon i minus 2 n is what I have here and here I have 1 is left and then I have this one. So, I can cancel out the halves here. And then get the following expression, minus 2 epsilon i n plus epsilon i plus 2 n divided by delta x square is equal to rho i minus 1 n; hopefully there is no mistake. And what we now see here is, this i minus 2 here, this is i minus 1, this is i minus 2 here, and this is i plus 1, this is i plus 2 here. So, this point corresponds to capital i plus 1, this point here corresponds to capital I minus 1, this point here corresponds to epsilon i. So, I can write this as, epsilon i minus 1 n minus 2 epsilon capital I n plus epsilon i plus 1 n divided by delta x square is equal to rho i minus 1 n plus 2 rho i n plus rho i plus 1 n; these are all in terms of small i's, which I can call as rho capital i n.



(Refer Slide Time: 19:13)

$$\frac{\epsilon_{i-1} - 2\epsilon_i + \epsilon_{i+1}}{\Delta x^2} = \rho_i$$

$\Delta x \rightarrow m \text{ no. of iterations using GS}$

$\epsilon_i^n \xrightarrow{m} \epsilon_i^{n+m}$

$\phi_i^{n+m} = \phi_i^n - \epsilon_i^n$

$\epsilon_{i-1}^{n+1} = \epsilon_{i-1}^n - \epsilon_i^n$

So, now, I have got an equation for the error evolution on the coarse grid, which can be written as,  $\rho_i$ , where  $\rho_i$  is given as  $\epsilon_{i-1} - 2\epsilon_i + \epsilon_{i+1}$ , and this is the equation which is now written for spacing of  $\Delta x$ ; and this is what I am going to solve for  $m$  number of steps iterations using Gauss Seidel method.

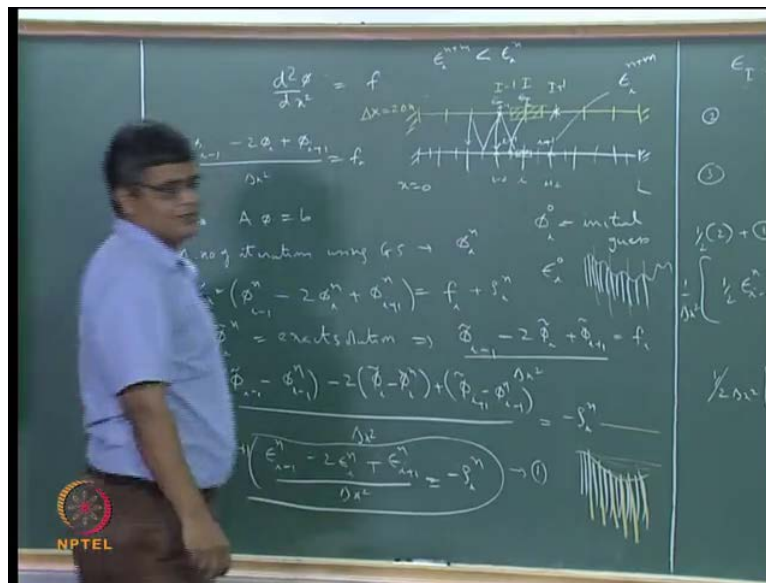
So, at the end of this, I will have  $\epsilon_i$  at  $n + m$ , at the end of  $n$  iterations carried out on the fine grid, and  $m$  iterations carried out on the coarse grid. And because I have carried out  $m$  iterations on the coarse grid, the error would have reduced, **using** because I am using Gauss Seidel method and it is a convergent method and so on. So, the error would have reduced and error would have reduced at a faster rate, because this is a coarse grid. And secondly, it would have, it would have taken me far lesser number of mathematical operations to reduce the error from at the end of  $n$  iterations to at the end of this, because I am working with fewer number of grids. So, therefore, the computation effort is less.

So, to come back to this, I reduce my error at the initial step using  $n$  number of iterations to error at  $\epsilon_i^n$ , and then  $m$  number of iterations on the coarse grid. So, this is done on the fine grid. And then, I have an error distribution here, and using this, I estimate the error on the fine grid at the end of  $n + m$ . And now, I can say that, I have an estimate for error; then, I can say the roughly from this, I can say the value of the function at  $i$  at

the end of  $n$  plus some iterations is  $\phi_i^n$ ; the value that I had here at the end of this things minus the error.

So, this is now, if I have this, then using this value, I can go back to my original equation; and then, reduce the error for some more time on this, reduce the error in  $\phi$  directly and then I can come back on to the coarse grid.

(Refer Slide Time: 23:13)

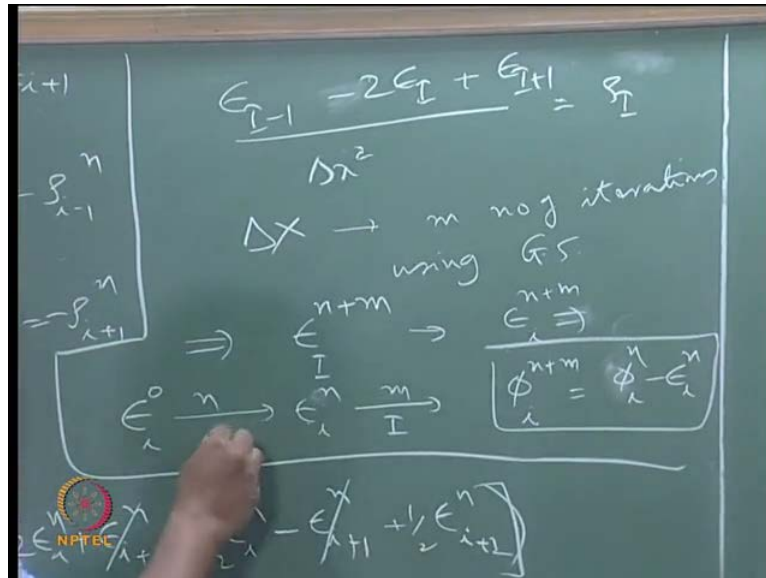


So, essentially what we were trying to do is, that from the original discretized equation, we deduce the residual at the end of  $n$  iterations at on each of the fine grid points. And using this residual at each of the fine grid points, we find out that error evolution equation on the coarse grid. And with this, we work for further  $n$  number of iterations and then reduce the error to some extent. Now, from this, so now, **we have error at...** So, this we have error at  $i$  error at  $i$  minus 1 and so on.

So, in order to find the error on the coarse on the fine grid here, we do an interpolation of these two, to get the error here; this error is brought here, the error between these two is used to compute the error here and this error is going here. So, by interpolation from the coarse grid, we can get the error on the fine grid at the end of  $n$  plus  $m$  iterations. And we can say that, error at the end of  $n$  plus 1  $m$  iterations is less than the error at the end of  $n$  iterations, because this is Gauss Seidel method and it a convergent thing; and we have probably gotten into that asymptotic convergence kind of thing. So this, So, we have

carried out some reduction of error by doing m number of computations on the coarse grid, and then once we have this error, we can get a revised estimate for the phi value and then go back into this equation, and then we can do further calculations.

(Refer Slide Time: 25:15)



So, to summarize here, at the end of this, we have the error at each of the coarse grid points. From this, we get the error at each of the fine grid points by interpolation; and from this, we estimate the value of phi i here in this way; this is already computed and this is the error; therefore, we get the thing.

(Refer Slide Time: 26:00)

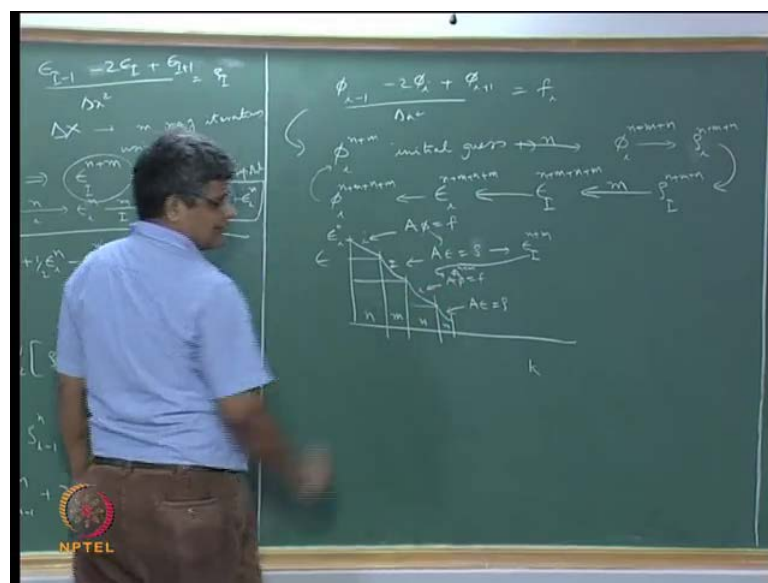


And then, we start again computations on the... So, this is the equation that we have. Now, we start this with an initial guess of  $\phi_i^{n+1}$  and then we carry out another  $m$  number of iterations to get  $\phi_i$  at  $n + m$  plus  $m$  number.

So, from this thing, we get an equation for the error on the fine grid like this. So, essentially, we compute the residual at the end of  $n + m + n$  number of iterations. Using this residual on the coarse grid on the fine grid, we evaluate the residual on the coarse grid, without doing any further iterative calculations; and doing further  $m$  iterations on the coarse grid, we get error distribution on the coarse grid at the end of  $n + m + n + m$  iterations. So, by interpolation, you bring it down to error on the fine grid at the end of  $n + m$  plus; and from this, we go back, we get the estimated value of  $\phi_i$  at  $n + m + n + m$ ; using this as the initial guess, we go back to this.

So, we are doing looping, starting with an initial guess, we get after  $n$  number of iterations, we get residual on the fine grid. And from this residual, we in a way we compute the residual on the coarse grid; and using this residual on the coarse grid, we try to minimize the error to reduce the error on the coarse grid using  $m$  - using  $m$  number of iterations. Using this error on the coarse grid, we by interpolation, we get the error on the fine grid; and using this fine grid, we update the value of the  $\phi$ , that we have computed, and using this, like this, we can go through.

(Refer Slide Time: 28:30)

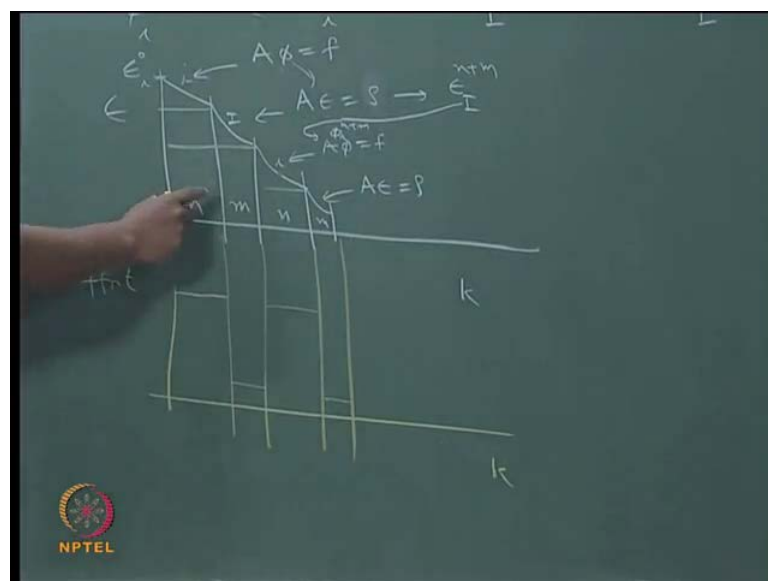


So, if you look at so called error as a function of  $k$  the number of iterations, you have, this is the initial error, let us say like this; part of this is reduced on  $k$  iterations carried out on the  $i$  th grid. And then, we have another  $m$  iterations, in which the error reduces on the coarse grid; and then, it reduces another  $n$   $m$  like this on the fine grid; coarse grid,  $m$  number of iterations like this. So, part of this error reduction is done on the fine grid and part of it on the coarse grid; again, fine grid, coarse grid like this.

This reduction is done by solving a  $\phi$  equal to  $f$ ; this reduction is carried out by solving  $A$  error equal to residual  $\rho$ . So, this again this is done using  $A \phi$  equal to  $f$ , this is using  $A \epsilon$  equal to  $\rho$ . So, from this, we estimate the error and then we get from here error at the end of  $n$  plus  $m$  iterations on the coarse grid. So, this we give as the input to get  $\phi$   $i$   $n$  plus  $m$ , which shall be the initial guess for this and then we carry out further number of things to reduce this.

So, the computations done on the coarse grid serve as refinement of the initial guess, for computations done on the fine grid. So, this is how that the error reduction takes place on the multigrid approach; part of it is on the fine grid and part of it on the coarse grid. What is done on the fine grid is the solution for  $\phi$ ; what is done on the coarse grid is solution for the error. So, this error is used to guesstimate the value of  $\phi$  for the fine grid calculations.

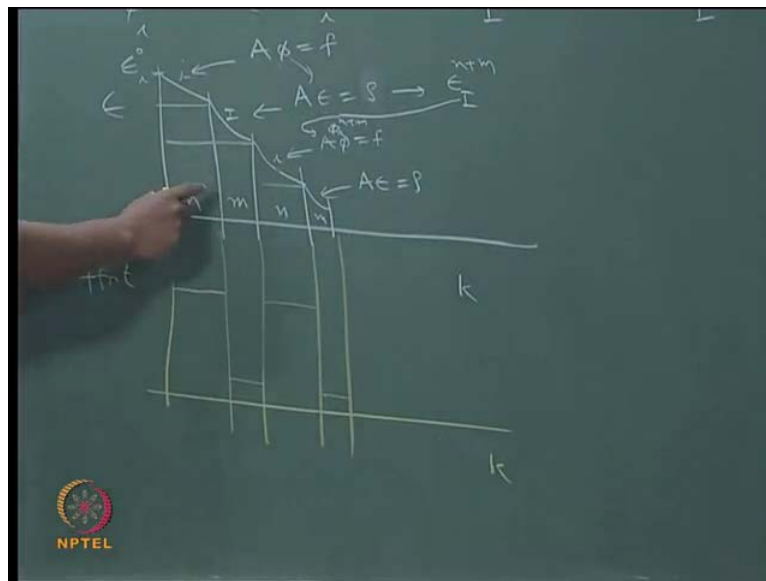
(Refer Slide Time: 31:07)



So, if you look at the computational effort  $k$ , so during the same  $n$  number of iterations and  $m$  like this, the computational effort for this will be high, and the computational effort for this part will be low; again, high here, low here. So, the overall computational effort, this is so low that it probably would not count; And so, in carrying out this  $n$  plus  $m$  plus  $n$  plus  $m$  iterations, the computational effort is not fully at this level; it is only the part of the time at this level and rest of the time, it is at low level.

So, the overall computational effort to carry out  $2n$  plus  $2m$  number of iterations is significantly less than the computational effort required to carry out the full solution, that is the whole number of iterations on the same fine grid. So, in an actual case, instead of doing so many iterations on this and so few iterations on this, you may actually do it like this.

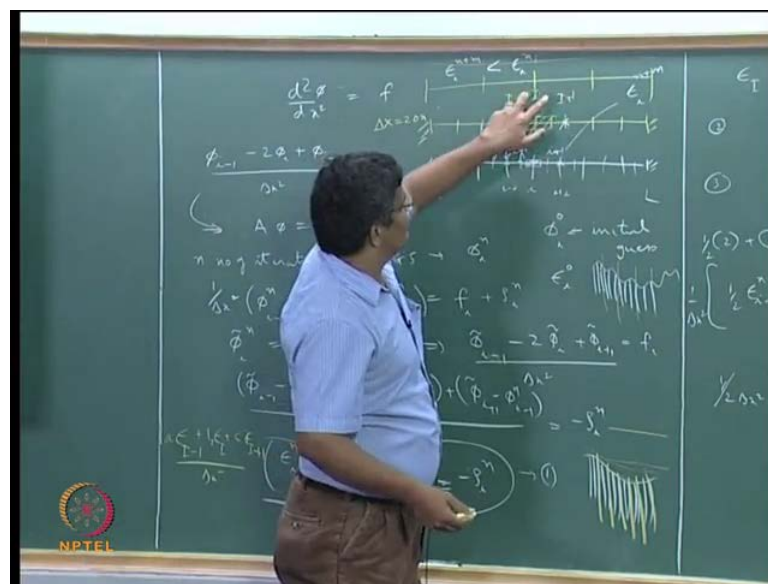
(Refer Slide Time: 32:23)



So, we can put it this as some residual reduction. So, you may have the same effort; you have for a short time, you are working on fine grid and then you work for a longer time on this. And then again, short time on this and then longer time on this, short time on this, longer time on this. And in the process, the residual is decreasing something like this here and then like this, and so on, on a log-log plot, it would be better. But you can see that some of the residual is being reduced during this time, and some more is being reduced this time and some more here and so on, so that the residual reduction is carried out, part of the time on the fine grid with high computational effort and part of the time

on this. So, that the overall computation time to reduce residual from 1 level to say 10 to the power of minus 4 of the initial level, will be a fraction of the only part of that fraction, is part of that overall computational effort is done on this; therefore, the overall computational effort which is the area under this curve is not this whole thing, which would be the effort that is required, if you were to use a single grid, but it is only this much. So, the area under this thing is much less than the area under this whole thing, and that is why the overall computational effort decreases.

(Refer Slide Time: 34:43)



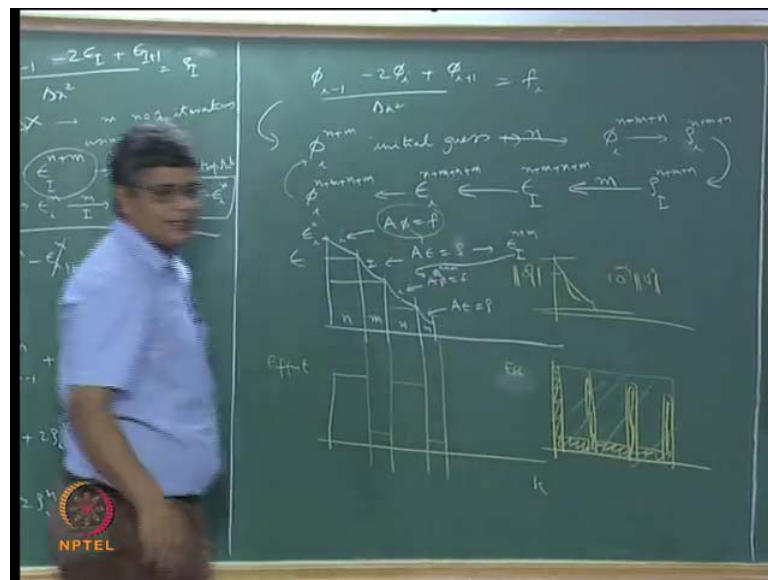
So, this is with two grids, you can have multi-grids, you can go from this to even coarser grid. Now, you have this one and then this one like this; and then, you can even have even coarser grid corresponding to this. So, on each of this, as you go from here to here and here to here, you are trying to take the residual here and defining an error equation with the residual corresponding to this, which is estimated from the residual computed here and so on. So, only in the coarsest in the finest grid, you solve  $A \phi = b$ ; and in all subsequent grids, you solve for the error in terms of the residual which is computed from the earlier thing.

So, the solution and multigrid approach, it uses different equations on different grids; on the finest grid is where you get the actual solution, and on the coarser grids, you solve only for error in terms of the residuals, which are already computed. So, you go from, there are certain interpolation and restriction operations that are done here, that is, in

order to, from the computed value of residual on the fine grid, you have to get the value of the residual at the coarse grid, so that you can work with this.

So, there is some interpolation or extrapolation that is to be done here. And from here, you get the error, and by interpolation, you can get this. So, the overall effort to get the solution is partly for the solution of  $A\phi = b$  on the fine grid, partly for the solution of  $A\varepsilon = \rho$  on the coarse grid, and partly for estimation of residual on the coarse grid and partly for interpolation of error on the fine grid **from**. So, the overall computational effort is considered to be, is typically found to be a small fraction of the computational effort required, if you were only working with the finest grid.

(Refer Slide Time: 37:07)



So, the area under this shaded thing is a small fraction of the total area, and that is a gain in computational effort that we are gaining in the multi-grid approach. And if we are very systematic and go through lots of grids, then we can really achieve significant reductions in the overall computational effort. So, this is the basic idea of the multi-grid approach, which results in almost a linear like linear scaling of the computational effort with respect to getting the solution.