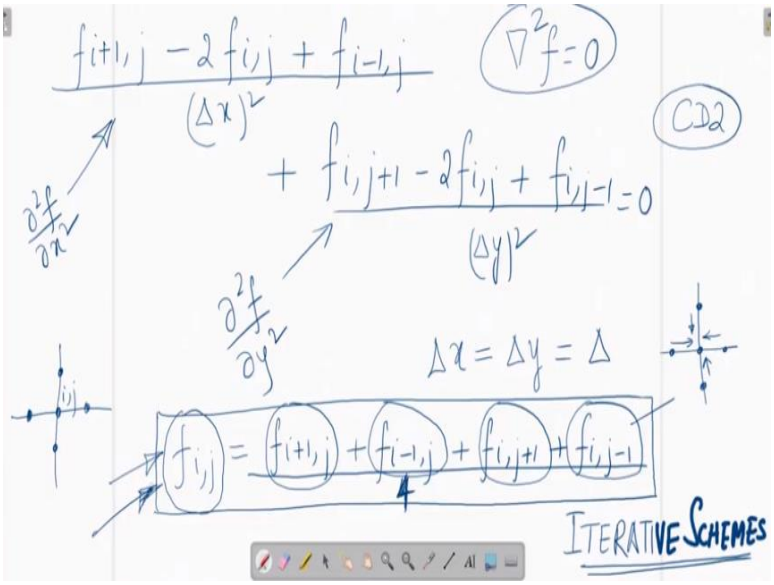
Introduction to CFD
Prof. Arnab Roy
Department of Aerospace Engineering
Indian Institute of Technology – Kharagpur

Module No # 04
Lecture No # 18
Numerical Solution of Steady State Heat Conduction (Elliptic PDE) (Cont'd)

(Refer Slide Time: 00:32)



We continue our discussion on elliptic partial differential equation in the last lecture we had looked at discretization of Laplace equation using the second order central difference scheme. And we had discussed how the Taylor series approximation is to be used when we handle partial derivatives in place of ordinary derivatives. So the strategy remains very similar but we apply it along different directions in these Cartesian directions.

And then we make sure that only one of the coordinates is changing the other remains frozen. So we finally had a CD2 finite difference expression for the second derivative for with respect to x and the second derivative of, f with respect to y. And we further said that if we were considering equal grid spacing along the 2 directions x and y. Then this would give rise to a significantly simplified expression which looks like this.

This looks like a very simple expression but it as a profound physics in the background, which tries to convey that the functional value at the grid point ij would be influenced to an equal extent

by all the neighboring grid points that, surrounds the point ij. And the fact that you have the constant denominative here means that all of those values in the neighborhood are influencing the value at the grid point ij to an equal extent.

It almost look like this that there is an influencing that is going on from all 4 directions to an equal extent which would decide the way the functional value at the grid point ij would evolve. And now if you look at the problem this way that though it actually answers an equilibrium problem could there be a way that we can actually post this problem as an equivalent evolution problem.

That means instead of reaching the equilibrium solution which this governing equation is supposed to give us can be gradually move towards the equilibrium solution in the evolutionary manner. Now this is a very important idea for a kind or class of schemes which we called as iterative schemes. What it essentially means is that you do not know straight away what the equilibrium solution would be that means how the functional value at this grid point ij should approach the equilibrium value.

But what you know is that there could be a way that you could gradually make it move towards the equilibrium value instead of obtaining the value in just 1 step. And if you are following that kind of methodology you would place the problem this way that you would not reach the solution in 1 step. But in several steps and each one of those steps will be mentioned as iterations steps and therefore through a fairly log number of iteration you hope to reach the equilibrium solution finally.

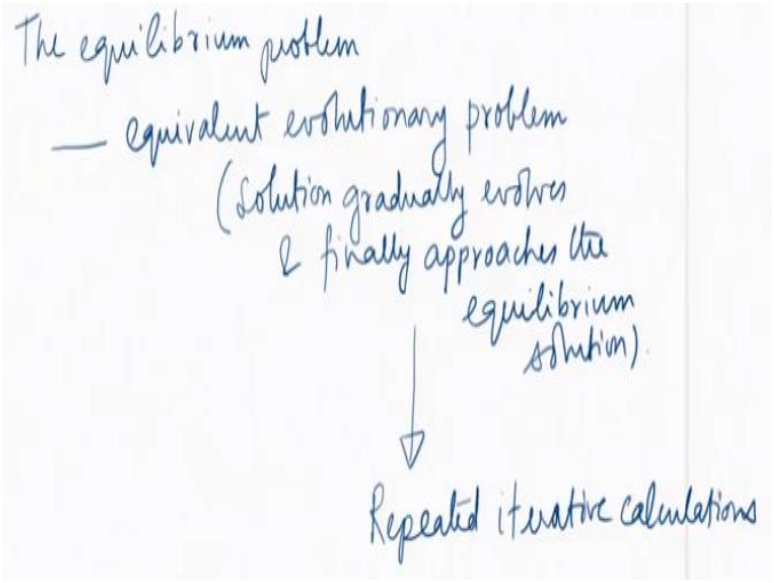
And as you do so the functional value at each grid point ij would evolve gradually from an initial guess value to the final equilibrium value and as it gets updated it gets equally influenced by all the neighboring grid points. And that is precisely what this equation is talking about. So we have now found a deep implication in the equation that stands in front of us. And that gives us the clue for finding a possible solution methodology to reach the equilibrium solution for Laplace equation. And this is through the so called iterative approach.

Now since we are going to deal with each point ij separately through an equation shown as in the previous page we would call this iterative approach as a point based iterative approach. Because

we here, solving for the new value of, f at each and every point ij separately. So there is a separate governing equation for each grid point ij. And we go through the domain sweeping through every possible i and j values so that we reach out to all the points and make them evolve the functional values to evolve at each one of those points separately.

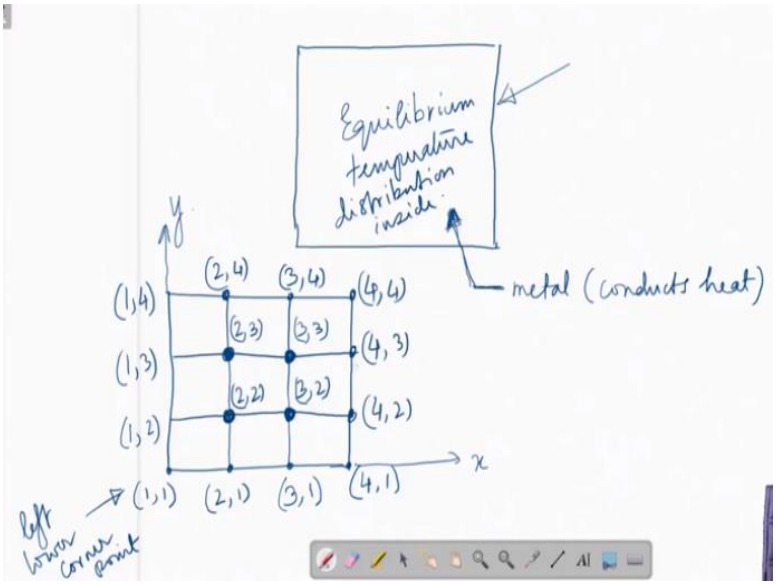
And then finally make them move towards the equilibrium so this is essentially a methodology that we are proposing.

(Refer Slide Time: 06:47)



So we are trying to pose the equilibrium problem as an equivalent evolutionary problem which means the solution gradually evolves and finally approaches the equilibrium solution. And how we do it is through repeated iterative calculations which we will show very soon. How to do them or how to implement them?

(Refer Slide Time: 08:04)



For doing that first let us pose a problem this way that we have a square domain let us say physically this indicates a square path of metal which obviously conducts heat. So if we consider this to be a problem of steady state heat conduction then we want this piece of metal as a heat conductor. So that when we pose the different boundary conditions along the edges of this piece of metal we should get an equilibrium temperature distribution inside.

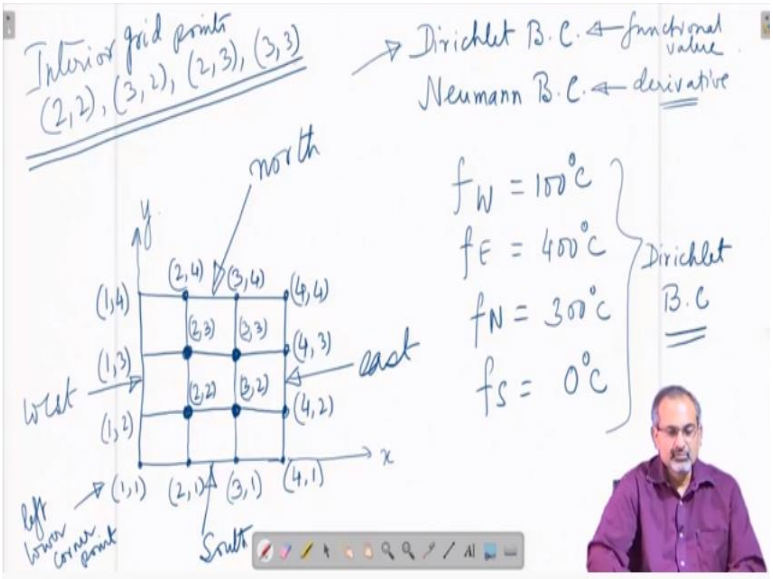
So; for solving such a problem let us put in some values typically so that we can work out an example problem and try to understand how this iterative method works. To begin with we will define these grid points we define this lower most left lower corner point as the grid point 1, 1 you are free to even name it as 0, 0 if you feel more comfortable to do it that way. Because that; seems to be similar to the origin in a Cartesian system so we are more familiar in calling that point as 0, 0.

So you can happily do that as well so there is no essentially harden and fast rule about it so here you notice that we have formed a mesh of points in such a manner that you have 4 points lying along the x direction covering the domain. And again 4 points along the y direction covering the domain so, we are naming these points as 1, 2 1, 3 and 1, 4. So as we move along y the second index changes as we move along x the first index changes.

And now we can likewise name all the remaining grid points so, we will mark these inner points which are not boundary points slightly darkly. Because those are the points where we would

actually need to solve the equations? Or rather the equation the governing Laplace equation while; the boundary grid points would actually have the values defined in some way. So let us see how we put these values at the boundaries.

(Refer Slide Time: 11:32)



So for putting them we could actually make use of 2 different kinds of boundary condition which we discussed earlier. Dirichlet boundary conditions or Neumann boundary conditions and we said earlier that we could choose to have the Dirichlet boundary condition all over the domain boundary. Or we could have a mix of Dirichlet and Neumann but we usually do not use Neumann boundary condition all along the domain because of the ill-posedness of such a boundary condition.

So in the first example that we solve here you like to define Dirichlet boundary conditions all along the boundary condition so let us see how about doing it. For doing that we; will name this boundary as the west boundary this one as the north boundary now this as the east and this as the south. So let us say that we continue to represent temperature by f. And the f on the west boundary let us give it some numerical value let us say that is equal to 100 degree centigrade f on the east boundary.

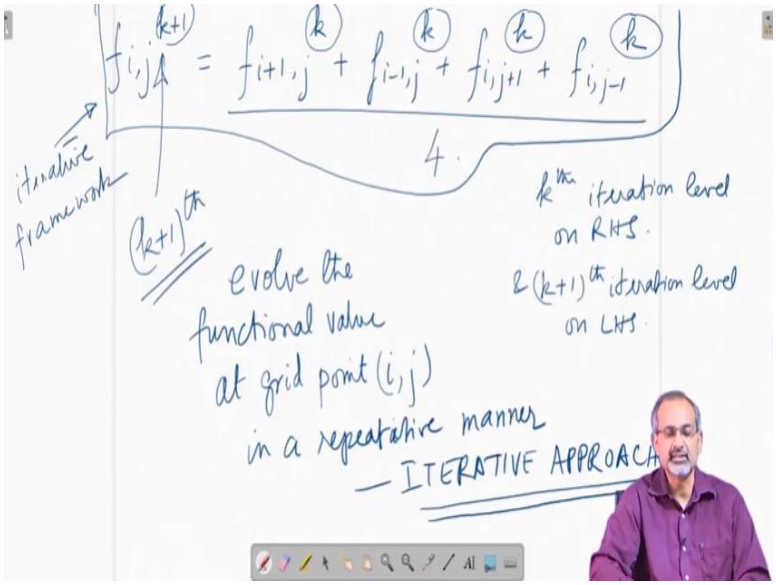
Let us say it is 400 degree centigrade on the north we have 300 degree centigrade and on the south we have 0 degree centigrade. So these are the boundary condition that we have on the 4 boundaries and as is evidence this is Dirichlet boundary condition because you are defining the

value of the function f directly at the boundary. You are not defining it in terms of the derivative so if it was a derivative you would have talked about the Neumann boundary condition.

Here you talk about the functional value itself so having imposed this boundary condition you would like to figure out how this iterative technique would work in order to find a solution to Laplace equation within this domain. So when we say within this domain what we essentially mean is we will look at the interior grid points so we try to seek a solution at the interior grid points which of the interior grid points these are the ones.

Because they are not occurring on any of the boundaries we need to remember a few things in order to pose the problem as an iterative problem.

(Refer Slide Time: 14:48)



Let us first place the discrete form of the governing equation assuming equal grid spacing along the 2 directions in an iterative frame work and try to explain what that means. So we have written now we have written this discrete equation in an iterative framework where we have introduced a super script here associated with each one of the functional values that we are using in this equation. Only thing is that we have the functional values at the kth iteration level on right hand side and k + 1th iteration level on left hand side.

Now what does this mean? It means that if you have the functional values available at the neighboring grid points. You can update your value based on their values and therefore the latest

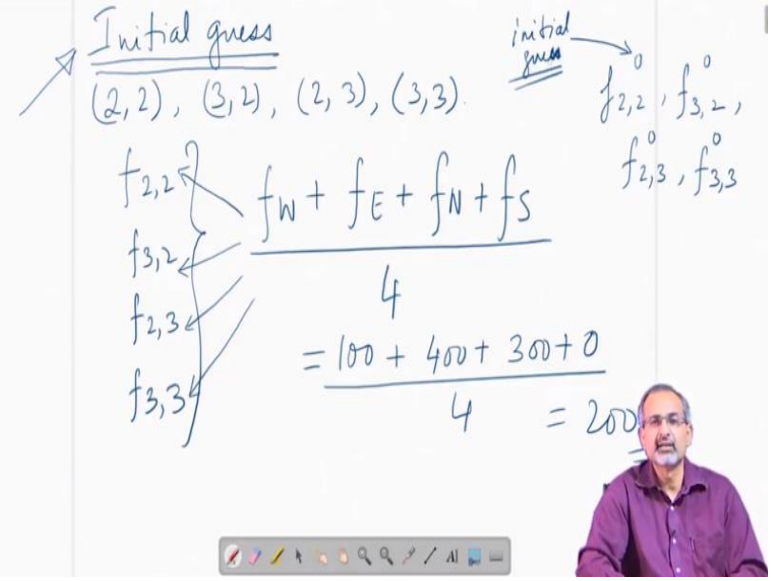
available values which we use on the right hand side from the neighbors are at the certain iteration level which you are calling as kth. And because they are going to be used update your value to the next iteration level therefore you are going to move to the next iteration level which is k + 1th iteration level.

So this kind of perspective will help you to evolve the functional value at grid point i, j in a repetitive manner which we are calling as the iterative approach. So what we essentially mean by the iterative approach is that you do repeated calculation by means of which you keep updating the values of the function at a certain point like i, j based on the latest available values which are coming from the neighboring grid points. And that is precisely what this equation is going to do for you.

And as you keep applying it to different grid points you are essentially changing the values of i and j. And therefore the moment you change the value of i and j you are sweeping through all the interior grid points and therefore updating their values or evolving their values continuously as you move from 1 iteration, level to the other. So this is the essence of the approach now as you can understand to start this calculation we need to begin somewhere.

That means this kth iteration level values which we are indicating on the right hand side of this equation as to come from somewhere. Where will it come from when you start the whole process of iterations?

(Refer Slide Time: 18:52)



For doing that we say we need an initial guess for example in this problem when we try to obtain the values of the function at equilibrium for this 4 interior grid points. We would probably have to go through several steps of iterations but the iterations have to start somewhere. And that somewhere is essentially coming from an initial guess how do we make an initial guess of the function at these grid points what could be the possible strategy of doing that?

Well one simple strategy could be that you obtain the values of the function at the 4 boundary values from the 4 boundaries that you have. And then just take an arithmetic average of the functional values of the boundaries and set that value at each one of this grid points. Which; means that in our case this would be 100 + 400 + 300 + 0 divided by 4 which give you a value of 200. So it is essentially 200 degree centigrade which would be the initial guess for the 4 interior grid points that you have chosen.

So that is where from your functional values at all these grid points will get defined at the zeroth iteration level. So let us put that as super script 0 here so what we mean by 0 is? It is essentially the initial guess that we are talking about we have not iterated anything but we have just guessed a value for each one of this interior points to being the calculations. Note that we never touch the boundary conditions because there are predefined.

And additionally as the iterative calculations go on the boundary values will never be touched when you have defined them in a Dirichlet sense. However if you are putting a Neumann

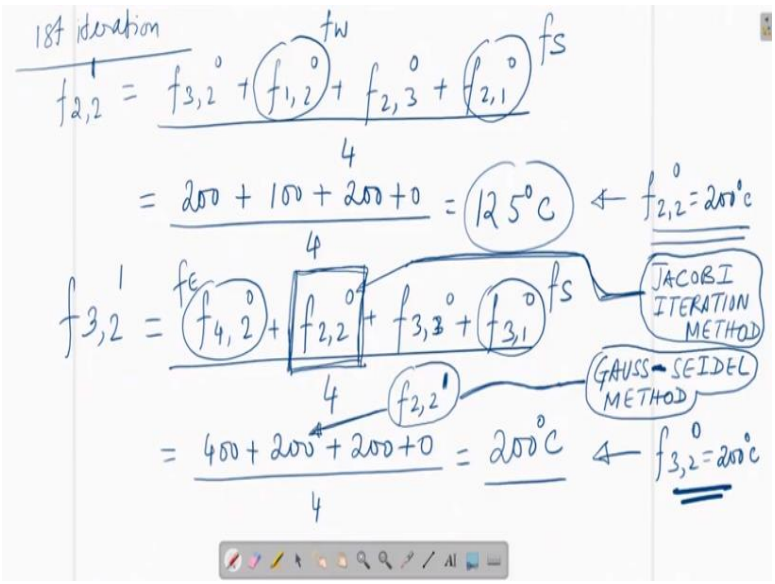
boundary conditions somewhere in the domain then even the boundary condition at that particular boundary where Neumann boundary condition has been imposed will evolve iteratively. Like it is going to evolve in the inner grid points the values will also get evolved at a boundary where you have imposed Neumann boundary condition.

So there is a difference between how boundary condition values have to be updated in the Neumann case. So you do not touch the boundary conditions when you are having a Dirichlet based problem but you have to update the boundary condition at the Neumann boundary. Whenever, at least one part of the boundary is defined through a Neumann boundary condition. Now let us begin the iterative calculations now that we have fair idea about the basic approach that we are going to follow.

As we do it we also need to note that the way we have imposed the boundary conditions you have gradient of the temperature along both the directions. So if you look at the east and west boundary temperatures you have a higher temperature on the east boundary compared to what you have on the west boundary. So you have a positive temperature gradient along the positive x direction and again the north boundary as a higher temperature then the south boundary. So you also have a positive temperature gradient along the positive y direction.

So you can imagine that as you sweep through those grid points 2, 2, to, 3, 3 in a diagonal one you can expect that at the point 2, 2 you will see a comparatively low temperature. While at the point 3, 3 you are going to probably see much higher temperatures. Because of the approximate of correspondingly lower or higher temperature boundary conditions for these 2 points respectively.

(Refer Slide Time: 23:28)



So we will start the first iteration this way so for the point 2, 2 you put a superscript 1 which means the first iteration and we put it this way. And we need to realize that this value comes from the west boundary and this one from the south boundary while the other values come from the initial guess. So this gives you an, updated value of the function at the point 2, 2 remember that the original guess for 2, 2 was 200 degree centigrade which has already changed to 125 degree centigrade.

Which as you can understand is happening because of the proximity of lower boundary temperatures coming from the west and the south boundaries which influence the point 2, 2. If you go to point 3, 2 then the calculation look like this. Again here you have the east boundary temperatures and the south boundary temperatures which are influencing the calculations. Incidentally it gives you back the same temperature as was the initial guess.

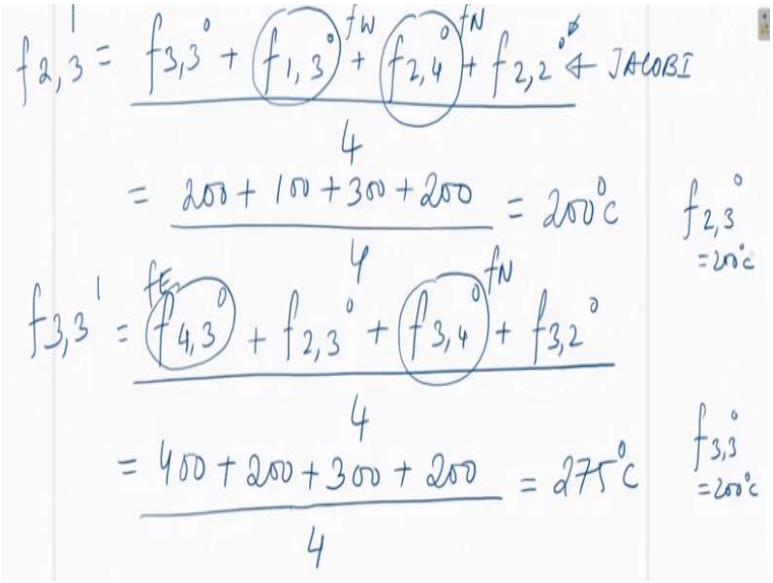
So effectively there is no change here let us look at the remaining points but before we do that we would like to point out 1 thing that is that you have noticed an interesting thing happening here that we have a value f 2, 2 coming from the zeroth level iteration. That means we have actually used the initial guess here to substitute the value 200 over here. Now one may question this calculation by saying that why used the previous iteration value here.

When you already have f 2, 2 available from this latest iteration level which we are calling as f 2, 2 at first iteration level which comes up as a superscript. So incidentally this gives rise to 2

methodologies of iterative calculations if you are always using the older iteration level irrespective of whether the functional value as already been updated or iterated at that grid points or not.

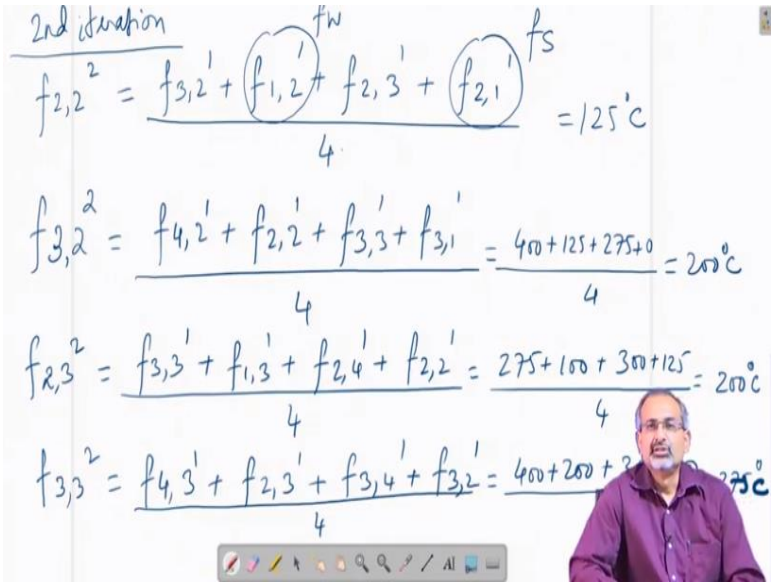
Then you are actually using a class of iteration methods which are called as the Jacobi iteration. While, if you were to use the latest available value then you are essentially using an iteration technique which is called as the Gauss Seidel iteration method. Will later learn that Gauss Seidel method can be applied point by point or can also be applied at line by line. Having said that let us move on to the remaining 2 interior gird point calculation to complete the first iteration level calculations.

(Refer Slide Time: 28:00)



So we are left with f 2, 3 so this comes from the west boundary this comes from the north boundary. And again remember that we are using the Jacobi method and therefore we consider the zeroth level value. This gives you 200 degree centigrade again which is incidentally the same as f 2, 3, 0. Then we have f 3, 3 this comes from east boundary condition this comes from the north boundary condition. So this gives you 275 degree centigrade of course the initial guess was 200 so this as increased and now we have the values at the first iteration level.

(Refer Slide Time: 30:14)

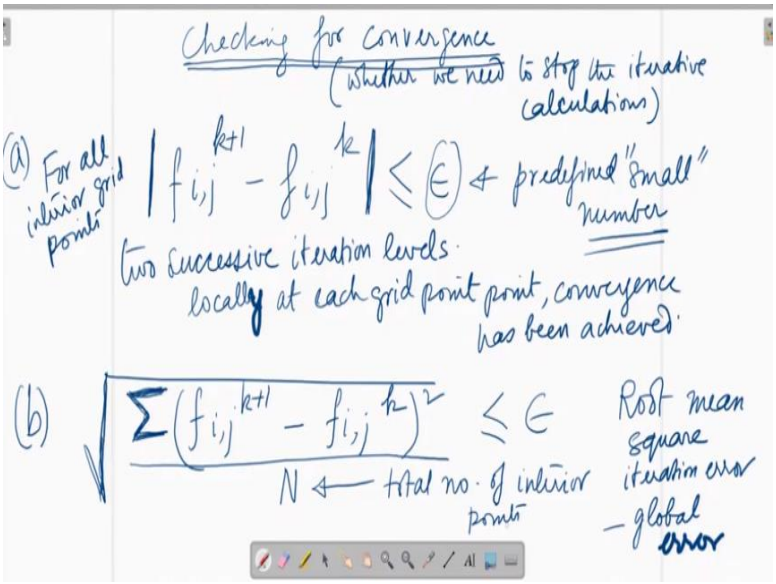


It would be very interesting to check what the values would turn out to be when you go to the second iteration. And now that we know the process pretty well we can calculate the values at our convenience by substituting all the relevant boundary values as well as the first iteration level values of the concerned interior points. And remember that whenever we are invoking values from the boundaries they do not change irrespective of whatever iteration level.

Because they are invariant and so when you plug in all the values you will find that this comes to 125 degree centigrade which is the same as the previous iteration this remains as 200. So incidentally what we notice is that the second iteration level calculations give us the same values as what we had produced in the first iteration. That means the values are already converged by the time we had done the first iteration.

Now this is happening primarily because we have a very few grid points in the interior part of the domain and incidentally the convergence has been achieved very rapidly. But this may not in general be the case so we need to check for convergence very often in problems which are of in bigger scale. And there could be essentially 2 different methods by which we check convergence.

(Refer Slide Time: 33:20)



That means whether we need to stop the iterations. So this actually answers the question that whether we need to stop the iterative calculation because otherwise. Let us say if you were doing it through a computer code, the computer code would go into an infinite do loop just repetitively updating the functional values and never able to come out of the loop because you do not have a proper condition which checks for the convergence and allows the computer program to terminate.

So how do you check for it they were 2 possibilities you could actually look for values of the function at 2 successive iteration levels. So we are looking at 2 successive iteration levels and we are checking for all interior grid points we are checking what is; the value of the function? The difference of the functional value between these 2 successive iteration levels the modulus of that and ensuring whether it is smaller than predefined small number.

Now this small number is rather difficult to state in general for all kinds of problems so we have to carefully in choosing this small number. So it is neither too small or again not trevally large so that we have the iterations terminating without too strong restrictive conditions for convergence or 2 weeks of convergence criteria. And this essentially ensures that locally at each grid point convergence has been achieved

Another possible way of assessing convergence could be through a route means square estimate of the iterative errors. So again we take the difference between the functional values we square

them up and we sum over all the interior points if the total number of interior points happens to be N then we divide it by that number. And we again check whether this remain smaller than a predefined small number.

So this is an assessment of the so called route means square iteration error and this is inner way a measure of the global error because you are taking the errors from all the points and we are collectively defining an error matrix so that we can collectively represent the errors through a certain value. So this would be called as the route means square estimate of the error we will discuss more on this in the next lecture thank you.