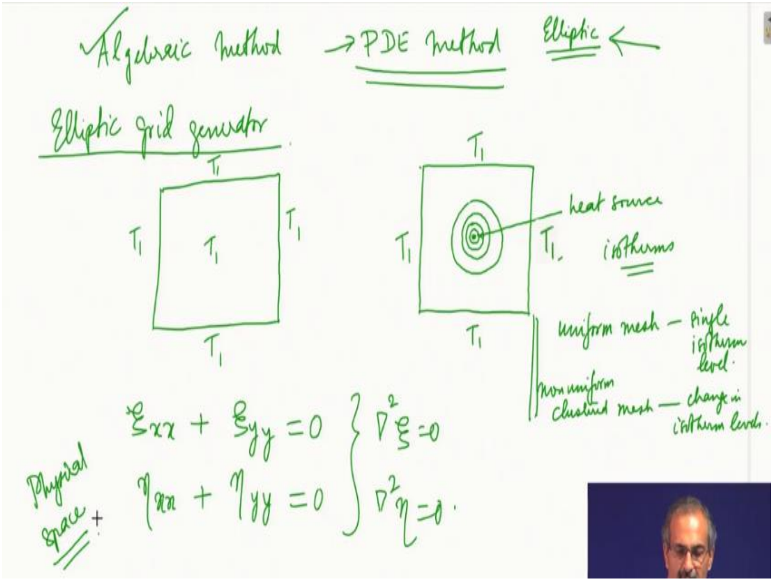
Introduction to CFD
Prof. Arnab Roy
Department of Aerospace Engineering
Indian Institute of Technology - Kharagpur

Lecture – 64
Structured and Unstructured Grid Generation - Continued

We continue our discussion on structured and unstructured grid generation. So in the previous lecture, we had talked about the concept of matrix. Jacobians, how to use them for grid transformation, how to use them for transforming governing partial differential equations. We also have discussed about how to use algebraic methods in order to do grid clustering including logarithmic and hyperbolic functions.

(Refer Slide Time: 01:10)



So you may recall that when we started the detailed discussion on structured grid, we had indicated that there are broadly two methods by which we pursue it. One is the algebraic method. The other is a partial differential equation method. So we have discussed to a good extent on algebraic method. Let us now start discussing about the partial differential equation method.

Now, from our experience on partial differential equations, we already know that they can be categorized into elliptic, parabolic and hyperbolic partial differential equations. When it comes to grid generation, we also have all these equations helping us generate grids, but we are primarily going to confine our discussion on elliptic grid generating methods or elliptic equation-based grid generators. So let us call it elliptic grid generator.

Let us try to first understand the idea behind deploying partial differential equations for grid generation. So let us say we have a square domain and we have set a constant temperature all over this square domain and we are trying to solve the steady state temperature distribution problem in this square domain, which we have discussed at length in the context of elliptic partial differential equations.

So we know from experience that if we try to calculate this problem everywhere we will come up with a solution of T1 in the internal grid points because the boundary conditions are uniform. So this is a trivial problem apparently, but that means that if you were to plot isotherms, everywhere you will be able to show the same level in this problem, but when does that scenario get changed?

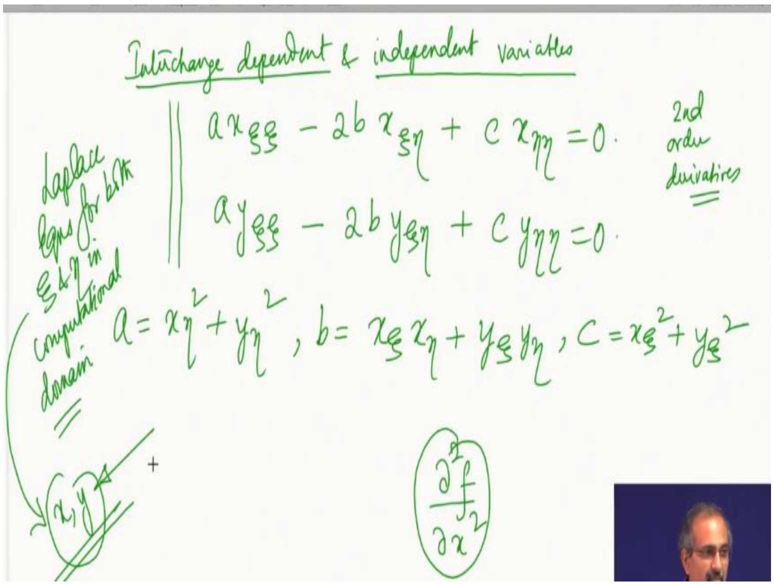
If I am not disturbing the boundary conditions, but I am just incorporating a heat source into the domain, then how would the problem change? Let us say, I have defined a hotspot just at the center of that region. Then I will now see isotherms developing like this around that hotspot and as they spread the levels gradually coarsen out, that means you have very rapid change in levels, very close to where the heat sources and then the changes become more gradual.

So these are isotherms, but they give us a very important clue that in an analogous manners would this incorporation of sources help us in the grid generating sense. That is if we do not have a source, then we have a kind of a uniform mesh scenario corresponding to a single isotherm level and you have a non-uniform clustered mesh which is associated with change in isotherm levels.

So using this analogy, partial differential equations were deployed in the grid generation world and then they were first deployed without any source terms and they were later deployed with source terms and that clearly indicated how the behavior of the grid or the mesh that they produce changed with control coming in in terms of getting the mesh refined in certain desirable regions of the flow or getting the mesh more orthogonal to a certain boundary and so on.

So, all those controls over the grid came later by tuning the source terms in the differential equations. So this is broadly the basis on which the partial differential equations are deployed in the grid generation exercise. So let us propose these two equations, which are Laplace equation in xi and eta. Of course, this is in physical space.

(Refer Slide Time: 06:17)



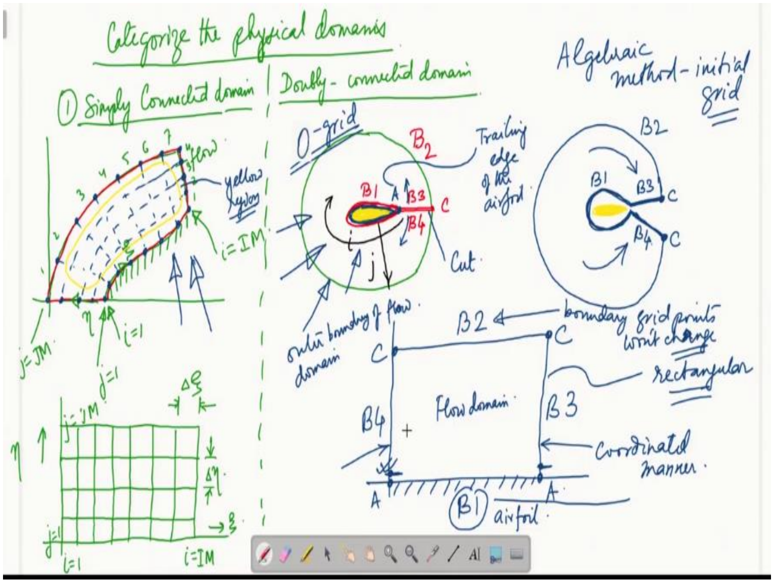
If we interchange the dependent and independent variables, which we have discussed in an earlier lecture on grid generation. Through this process, we will be able to obtain forms of these equations, which are readily computable in the transformed plane that is the xi eta plane where we will have a uniform domain. We are not deriving these equations, but you can certainly do it as a homework exercise for better understanding.

All that you have learned in terms of transformation should be sufficient for you to obtain these equations. Only thing that you have to keep in mind is that this involves second order directives and therefore the competitions will be more involved or rather obtaining the transformation relations will be more involved. So you may recall that we had tried doing that exercise for a term like del to f del x square in the previous lecture.

So you have to follow the same approach in order to do it for the derivatives in the context of Laplace equation. Now having said this, we now have defined the Laplace equation in computational domain. So Laplace equations for both xi and eta in computational domain by interchanging the dependent and independent variables. What we have done is we have actually written the differential equations in terms of x and y.

So notice that xi and eta now are the independent variables, earlier they were the dependent variable. So now they have changed role and the dependent variables are x and y and x and y are essentially going to decide on the grid locations in physical plane.

(Refer Slide Time: 09:19)



Now, before we proceed further, we also have to understand certain concepts of physical domains where this exercise can be carried out. So we have to categorize the physical domains where this exercise can be carried out. So we will discuss about three possibilities. One is simply connected. Let us start the discussion with that. So a simply connected domain will look like this. This is just an example, you could have many other examples for this.

Let us say that this is a boundary and this is a region of flow surrounding the boundary and this portion which is shown by red color is your computational domain. Now what we will do is we will define your xi and eta directions in physical plane and also indicate your indices. IM is the maximum index level for I, JM is the maximum index level for J and imagine that your grid lines are kind of deployed like this.

So you have chosen to use the elliptic partial differential equation based grid generator to generate this mesh, but remember that like in solving elliptic equations we need boundary conditions either in Dirichlet or Neumann sense, rather purely Dirichlet or mixed Dirichlet Neumann to have a well-posed boundary condition. So similarly, it has to be done like that here. You have to give the mesh generating program, the initial condition and the boundary condition.

So boundary conditions essentially means the grid points locations that is their x and y coordinates have to be clearly specified all along the boundary and they will remain that way all through the computations. Only thing that will change as computations proceed are the internal grid point locations. That means only these grid point locations will change as computations progress.

This is very important and this is exactly analogous to what you have done for elliptic equations earlier. Having said that we need to see how the things look like in the computational domain. So as we know the computational domain is regular and rectangular, so the mapping how does it take place? It takes place this way. So this is your xi, this is your eta. So this is how the things will get mapped.

So you have 1, 2, 3, 4, 5, 6, 7 intervals along xi. So exactly 7 intervals even here in the computational domain and you have 1, 2, 3, 4 intervals along eta, same thing here in the computational domain, constant delta xi and delta eta. So this is how the system looks like. Now this is a simply connected domain. It will become more clear why we are calling it simply. Let us take it to the next level where we have a doubly connected domain.

Let us say we are solving flow past an airfoil. How do we handle the mesh here? Green is the surface of the airfoil which sits inside or rather for clarity we will make it blue and the grid sits adjacent to it, actually coincident with it, but for the sake of clarity and understanding we are just showing it slightly apart and we would indicate different regions of the grid. Let us call them like this. The outer boundary is B2.

There is a cut in the grid. Upper part of the cut is B3, the lower part of the cut is B4, part of the grid which wraps the body is B1 and where these two cuts meet, we call that as the point C and we will introduce the direction of i and j. So the yellow colored region is a solid region where no flow can enter, it is surrounded by the blue, which is the boundary of the body and you are wrapping the mesh which is given by red color around that body.

And then apparently there is a cut region in the mesh as though we have put the scissors into the flow field and cut it apart over there, but for now the cut just remains that way that it has not fallen apart. Now, if somebody stretches that, how will it look? Let us try to draw it. If that portion is slightly stretched, then it might look like this. That means if somebody stretches it along these two directions, then this cut will open up.

Where is the airfoil, the airfoil is still sitting inside that cut. Let us mark the airfoil in yellow. So how would we indicate the different regions in this diagram? This is how we will indicate them. Apparently, there are two C's now because you have stretched it and taken it apart. This is a B1 which is no longer coincident with the body boundary, so it has now created a gap with the body. Now this picture if you were to represent it in the computational domain, how would you represent it?

This is the airfoil B1, on one side B4, on the other B3. This is the outer part of the grid B2. These are C'S. We can call this as A, if you want to call it A, then remember that A must lie somewhere here. That is the trailing edge of the airfoil. So B1 wraps the airfoil. B4 and B3 are boundaries in the fluid domain, which are coincident. B2 is the outermost boundary of the flow domain. This is how the mapping takes place. So all this is of course flow domain.

Again, B4 and B3 essentially are same lines, just for sake of understanding we have shown that split, but in the physical domain they are as though coincident with each other. You are creating a cut so that computationally you can transform it to a domain which looks regular and rectangular like this without which you would not be able to transform it because the body is sitting embedded inside the flow. What is the difference with this situation?

Here, the flow was surrounding the body, the body was not embedded within the flow, but here the body sits inside the flow and that is why the cut had to be created. The moment you do that, you have a so-called doubly connected domain as though one part is coming from this end, another part is coming from this end and getting connected, which means a double connection, which can wrap the grid around the body.

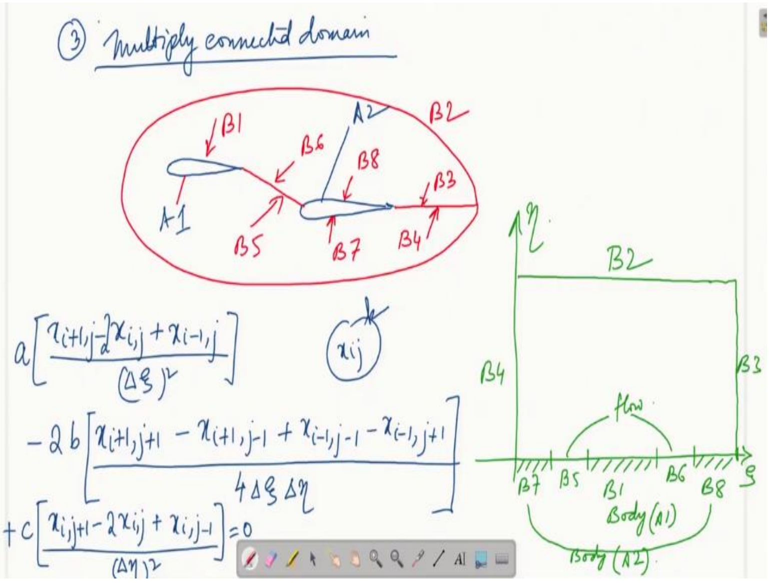
This type of grid incidentally is called as O-grid because it has the shape of O broadly and we have now understood how the mapping goes on. Now, when you deploy the grid initially, you can deploy a grid for this problem or the airfoil problem by using say an algebraic method that is your initial grid. Remember that that initial grid, whatever grid points it defines for the boundaries, one boundary is B1, another boundary is B2.

These boundary grid points would not change, but what can change, the intermediate grid points can change like we had shown in this yellow region. How do they change? As the partial differential equation based grid generator works on those grid points, it iteratively defines their location and therefore they change. Now, remember that B3, B4 these boundaries are also lying within the flow where the grid points can change.

But only thing that you have to keep in mind is that they should change in a coordinated manner. That means the value corresponding to this mesh point and this mesh point, they should give you identical values of x and y in the physical domain. They cannot have separate x and y values because then B3 and B4 boundaries would not be coincident anymore. So that is a very, very important fact.

If you have to change grid point locations at boundaries B3 and B4 you have to do it in a coordinated manner. Again, remember the way i and j indices are moving here, which were indicated by the black arrows, so i wraps from the trailing edge of the airfoil in a clockwise manner surrounding the entire airfoil and finishing the i levels and j radially moves out. So these are important details one have to keep a watch on to understand the process by which we are defining the grid layout.

(Refer Slide Time: 23:09)



Now, if you go one more level, you can have a multiply connected domain. So we have seen simply connected, doubly connected, now it could be a multiply connected domain. So there may be say two bodies in close proximity around which you are wrapping a grid. In that case, you create cuts like this and you create a boundary like this and you cut and you define different regions and then we will see how the entire thing gets mapped in the computational domain.

Now you see 2 cuts. We are not keeping them apart now because we have already understood the basics. So this is airfoil 1, this is airfoil 2 and then in the computational domain, how does it look like? In the computational domain we have to keep in mind that we have number of regions to mark. So we need to understand B1 and combined B7, B8 are the bodies. So B1 is essentially A1 and this is A2, airfoil 1, airfoil 2 and B5 and this should be B6, these are flow regions.

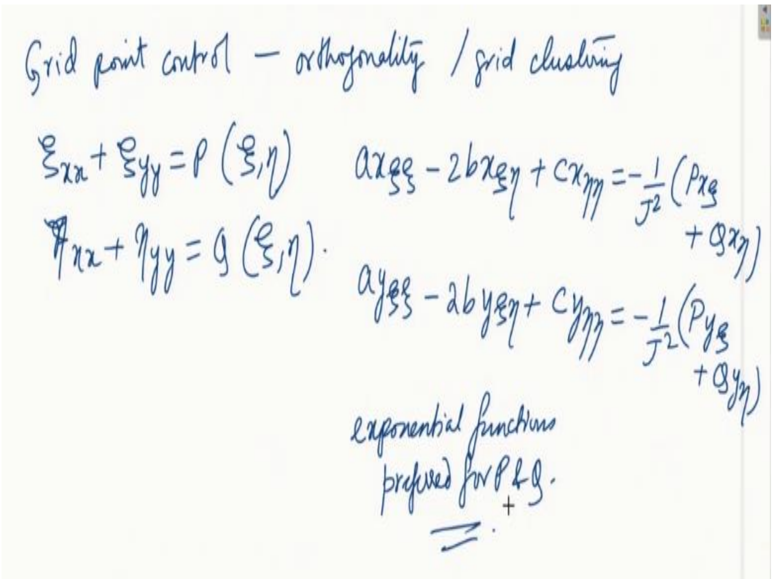
Again, you have B4 and B3, which are flow boundaries and then the outer boundary is represented as B2. So this is a more complex transformation, which we are seeing over here in a multiply connected domain. Now, we were looking at the partial differential equations in computational domain. If we were to discretize them here, one of these equations in x, the equation in x would be discretized like this on any one of these problems simply connected, doubly connected, or multiply connected.

The exercise is very similar as long as you have created the correct transformation between the physical domain and the computational domain, the mapping through these diagrams that we have drawn here, the rest of the exercise is just to get the equations discretized and apply the suitable boundary conditions and as we said that the initial grid gets generated from say an algebraic grid generator and then it is refined by the elliptic grid generator, which is discretized in this manner.

So as you can understand that this equation gives you an evolution equation in xij, which gets iterated. So at any iteration level k, xij is the coordinate for a particular grid point in physical space, but it is getting iterated in computational space. So as it gets iterated in computational space, the physical image, the physical location gets changed gradually. So in physical space, the grid point is gradually moving towards an equilibrium location.

And all grid points are moving this way apart from the boundary grid points and then finally reaching a steady or converged location. So like we have checked for convergence in many other instances in iterative calculations, we do convergence checks based on say RMS values. Till we reach a good amount of convergence, we continue these iterations.

(Refer Slide Time: 28:16)



And then finally if you want to do some kind of say grid point control, either in sense of improving orthogonality or grid clustering, such things are best done not with Laplace equation**,** but in the form of source terms. That means that gives us Poisson equation. So you will get some source terms suitably in both these equations, let us call them as P and Q and then if you are just change the role of independent and dependent variables, you finally would get equations of this form in terms of Jacobians.

And of course, you have to choose these functions P and Q with care so that the desired goal is reached, either in terms of orthogonality or grid clustering the desired goals are reached. Usually, we find that exponential terms in these sources would attract the lines of constant xi and eta close to the desired boundary and also enforce orthogonality. So usually, exponential functions are preferred for P and Q. So with this, we finish our discussion on structured grid. We will discuss on unstructured grid a little more on detail in the next lecture. Thank you.