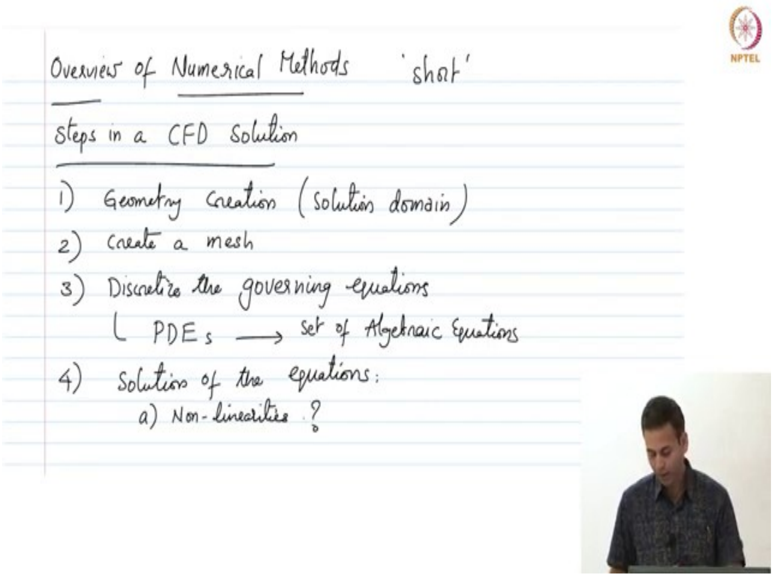
Computational Fluid Dynamics Using Finite Volume Method
Prof. Kameswararao Anupindi
Department of Mechanical Engineering
Indian Institute of Technology, Madras

Lecture - 07
Overview of Numerical Methods: Finite Difference Method

(Refer Slide Time: 00:20)



Welcome to today's lecture. So, we are starting a new chapter today, that is Overview of Numerical Methods. This is our second chapter. So, this is a kind of a short chapter, ok. As the name implies, we are going to see kind of an overview of the entire CFD solution procedure, and the kind of different elements that are involved in it.

So, we will kind of getting an overview, and then, we will go deeper into each of the issues as we progress with the other chapters, ok. So, some of these things might be just for your knowledge purpose and then if you do not understand you are of course, can ask me questions. But some of them I may say we will learn little more as we go into the particular chapters again, ok, alright.

So, what are the steps in a CFD solution? What are the key steps in a CFD solution? Ok. If we set out to do something in CFD. So, the first one is the creation of the geometry, ok, so essentially this is geometry creation. The next one is to create a mesh for the geometry, ok. So, this geometry we also call it as the solution domain, ok. So, essentially create a mesh.

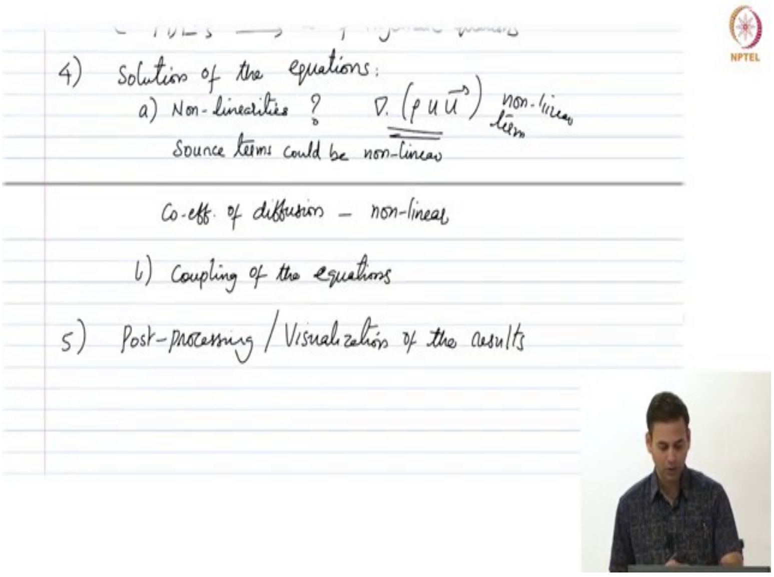
And the third one is the discretize the governing equations. So, in the discretized in the governing equation step we are going to convert our partial differential equations that we have into a set of algebraic equations, ok. Once we have the set of algebraic equations we are going to use a solution technique and going to solve for this, ok.

So, the fourth step is the solution of the equations, ok. So, essentially while solving for these equations we have to account for any non-linearities that may be there. What are the some of the non-linearities that you know that may kind of that we may encounter in the solution process or in the problem? What are the nonlinearities that you that you know of? Where could they come from?

Student: Advection.

Advection. So, if you are talking about, let us say Navier-Stokes equations, the convection term is a has a non-linear operator, right we have del dot rho u u bar, right. So, del dot rho u u bar. So, this is a non-linear term, ok.

(Refer Slide Time: 03:17)



What are the let us say we are not talking about Navier-Stokes, but we are talking about a just a diffusion problem, ok. Could there be some non-linearities in it that you can think of just in a diffusion problem? Or let us say conduction problem, could there be some nonlinearities in the conduction problem? Ok. We are not talking about a non-linear in terms of a PDE, we are talking about, we are not talking about a non-linear PDE we are still talking about a let us say a linear PDE with some non-linear terms. What could that be? Source term, right.

The source terms could be non-linear, right. Instead of just the heat supplied increasing with temperature it could be increasing with a as a function of temperature, right. It is not just t to the 1, it would be t to say t square or t to the power some rational, ok. So, essentially this could be the source terms could be non-linear.

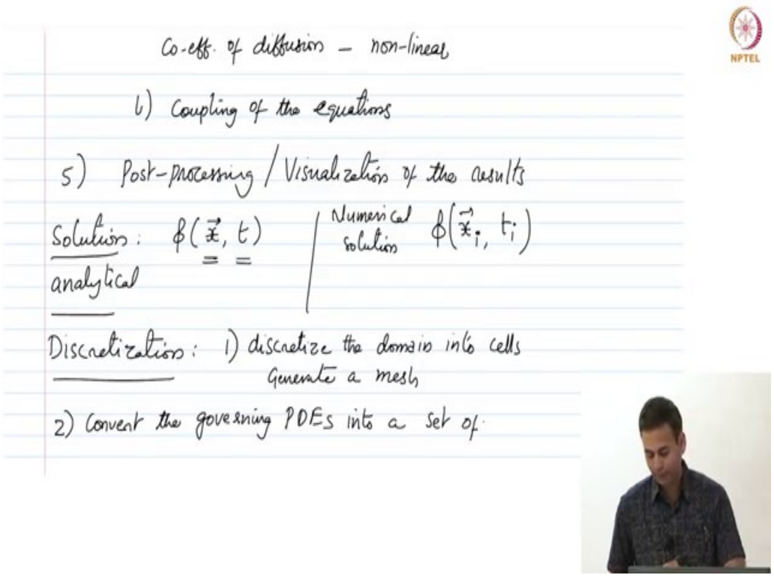
What other things can be non-linear in a simple let us say diffusion problem simple heat conduction problem? I think we kind of had this question before, right. So, what about thermal conductivity, could that be non-linear? Could that be non-linear function? It can be, right k can be a non-linear function of temperature in which case we have to account for non-linearity that we are getting, ok. So, essentially the coefficient of diffusion, right we will call it with a general name that is coefficient of diffusion could be non-linear as well, right, ok.

So, essentially we have to account for these non-linearities we will see why do we have to account for them and how do we account for them later, ok. So, that is one thing we have to do. And of course, we have to also consider the coupling of the equations, right.

Let us say if you are solving for more than one equation, you are solving for two components of Navier-Stokes equations. Then, these two the x component and the y component of the Navier-Stokes equations are coupled, right, they are coupled to each other. So, we have to kind of look at the inter equation coupling, between the equations, that we have to kind of consider as part of our solution, alright.

So, once we have a solution we would have to do some kind of a post processing or visualization of the results, right; essentially we are interested in let us say a drag on a particular object or the amount of heat that is transferred and so on. So, all these things we are going to obtain as a post processing step, ok. So, that is kind of a general template for any CFD problem which we would kind of do as part of different assignments we may get, ok, alright.

(Refer Slide Time: 06:10)

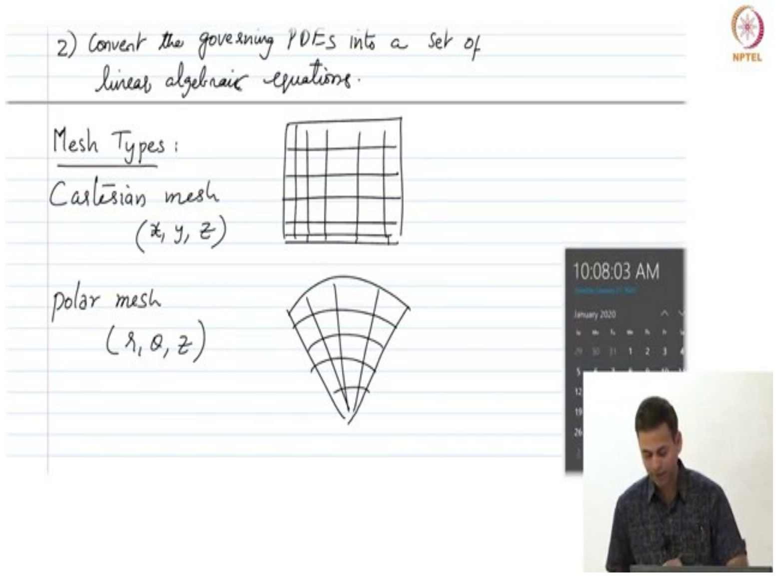


So, coming to the obtaining the solution. So, if we have if we if we kind of solve for a problem, let us say we are solving for a flow field, we were solving for a field phi in on x and t, ok. So, essentially we are looking at solution$\phi(\vec{x}, t)$ of phi in a particular domain x bar that is x, y, z and as a function of time. If we have an analytical solution we are going to obtain phi everywhere at all x bar and t, right essentially this will be a continuous x bar and t, if we if we know if we can find analytical solution to a problem, right. If you want to solve for diffusion equation using analytical method you will get this everywhere at any x you specify, right.

Whereas, if you have a numerical solution, right we understand that we cannot obtain phi everywhere in the domain, rather we can obtain phi only at specific locations in the space, right which we call as the mesh points. So, phi is only obtained at some x i bar, where i we call it as the index, right. We have discretized the mesh with some locations. So, at those locations and for discrete times we could obtain the solution, ok. So, as a result there is a difference between the numerically obtained solution and the analytical solution, ok. That needs to be kind of kept in mind, ok.

Now, I use this term called discretization, right. So, we said discretization. So, there are two ways in which this term can be used, the first one is discretizing the domain, discretize the domain into cells, or essentially generation of the mesh, ok. So, this is known as discretization where you convert a solution domain that you have into discrete cells and in a different context discretization also refer to convert governing PDEs, governing partial differential equations into a set of into a set of linear algebraic equations, ok. So, these two processes are usually referred to as discretization, ok. Depending on the context we would kind of make use of that. So, we looked at the solution process and the discretization.

(Refer Slide Time: 08:52)

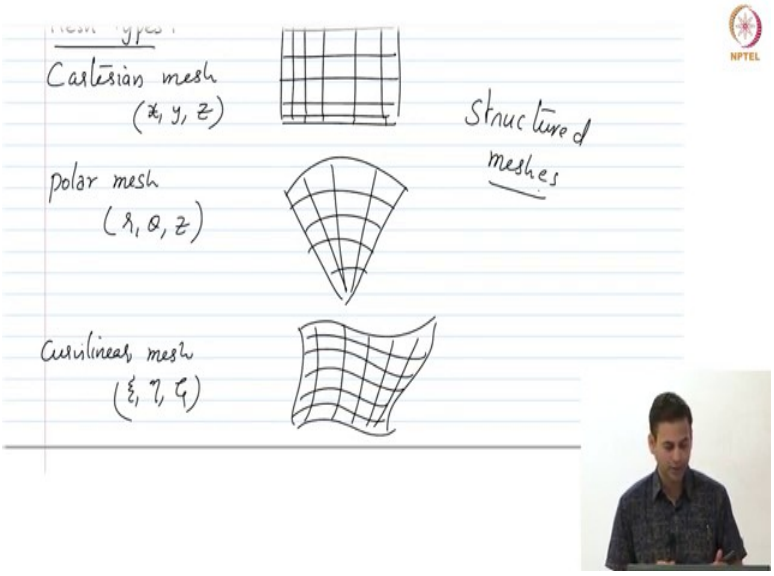


Now, let us look at if we go step by step then the next step is kind of creating the mesh, ok. So, we look at different types of mesh that we get. So, these are the mesh types, ok. Again, as the chapter suggests we are looking at a kind of an overview, ok. So, some of these meshes you might make use of while working on your projects or your assignments, ok, some of them could be useful when you kind of do your research in your as part of your course, ok.

So, there are several mesh types. The first one is known as a Cartesian mesh, ok. So, this is a simple rectangular mesh that we can get. So, we have a solution domain and then we produce some mesh which could be aligned with the Cartesian axis, ok, but it could be of non-uniform size, ok. As a result it could be closer have smaller cells on one side and larger cells on other side. So, this is a Cartesian mesh.

We can also have a circular or a polar mesh, ok. So, this could be similar to the Cartesian which is in x, y, z, you could also have a mesh in r, theta, z. So, this would be another mesh which is let us say if we have some kind of a polar geometry, then we have a mesh like this, ok. So, this is a polar mesh.

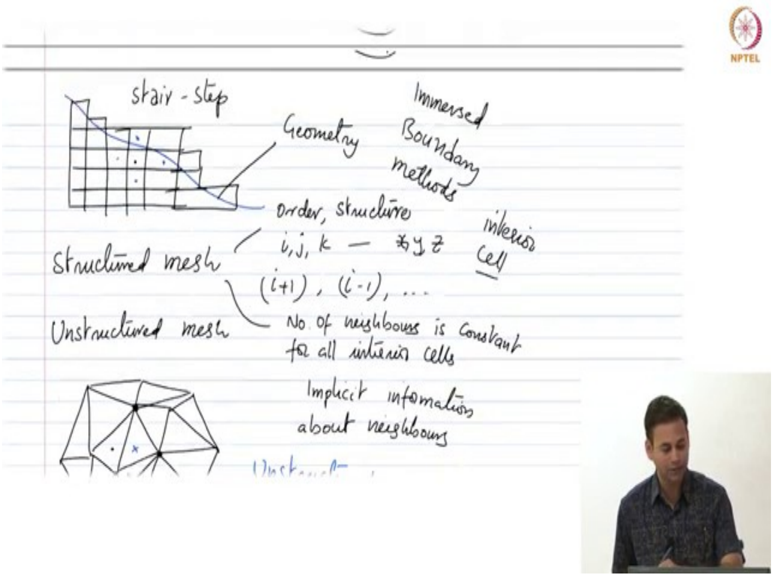
(Refer Slide Time: 10:22)



We can also have a mesh that is curvilinear, ok. Could also be some kind of a curvilinear mesh this could be in some generic coordinate system that is xi, eta, z, ok. So, this could, this may look like something like this. And these are all the cells that we may produce which are all curvilinear, depending on the shape of the domain we are trying to solve a particular equation.

And now one thing that is common between all these things is these all these types that we have listed out are known as what? What meshes are these things? These are all structured meshes, right we call them as structured mesh, ok. Now, we need to of course, understand what is a structured mesh which I have not defined yet, alright. We will define it in a little while, ok.

(Refer Slide Time: 11:28)



Now, you can also of course, have a domain that is still made up of Cartesian cells, which are like this. But we may have let us say use this kind of a Cartesian mesh to somehow resolve a geometry that looks like this, ok. So, our geometry looks like this. We can still make use of a Cartesian mesh like this to resolve this geometry, ok.

So, these are a different class of methods those are called immersed boundary methods, ok. Which we will not discuss much in this course, but this could be a good problem for your project later on, ok. In which case you generate a mesh something that looks like a stair step mesh, which we will use to resolve a geometry that is curvilinear, ok. So, these are also kind of picking speed in the past a decade, alright.

Now, these are all structured meshes. Now, what is a structured mesh? And what is in the same context what is an unstructured mesh? Have you heard of these terms structured and unstructured? You have heard. So, what is the, what is the quality that tells you by looking at it says, this is a structured mesh? Sorry.

Student: Aspect ratio.

Aspect ratio, ok. Not exactly aspect ratio.

Student: Way to index (Refer Time: 13:02).

Way to index, ok. So, there is a particular order, right or a structure, ok. So, there is a particular order or a structure, ok. What else? No, ok. So, essentially there is of course, a way to index, so if you want kind of go on i, j, k to go in 3 different directions that is x, y, z then by going to i plus 1 you go to your one neighbour in the positive x direction. If you go to i minus 1 you go to the another neighbour in the negative x direction and so on, ok. So, there is a particular indexing that you can do.

But more importantly a structured mesh is something where if you take an interior cell, ok, if you take an interior cell the number of neighbours this particular cell has is constant, right is the same wherever you go in the in the entire mesh domain, ok. So, for example, if you take any cell here any interior cell. So, we mean interior meaning away from the boundaries, any interior cell. This guy always has 4 neighbours. We are talking about neighbour in the context of sharing a face, ok. So, essentially this this guy for example, if you take a cell this has 4 neighbours 1, 2, 3, 4 which is sharing a face.

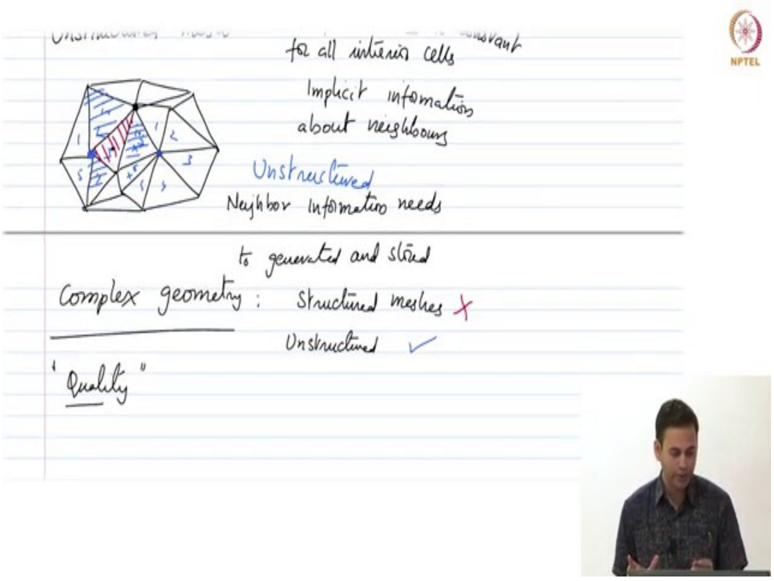
Now, that is the same wherever you go in a structured mesh. Whereas, the number of neighbours need not be the same over the entire mesh in case of an unstructured mesh, ok. So, because that depends on the angles made by the particular cell faces the number of neighbours would not be the same, ok. So, that is kind of a somewhat generic definition. So, whenever there is a structured mesh we know that the number of neighbours is constant, ok, let us say for all interior cells, ok.

Now, of course, there is this advantage of ordering. So, we can always go, we always know that if you increment by one in a particular direction you are going to a neighbouring cell in that direction, does not matter in whether you are in a Cartesian mesh or in a curvilinear mesh. We always know that we can go to our neighbours, right, the neighbouring information is already kind of implicit in nature, right. So, implicit information it is already built in about neighbours, ok, that is already there in structured mesh, ok.

Now, of course, unstructured mesh is does not have these features. So, if I were to draw an unstructured mesh I would draw it like this. So, ok, and so on I have I have used triangles to kind of discretize this particular solution domain. So, this is an unstructured mesh, right, ok. Why do we say that? Of course, because if you use I there is no particular direction if you use any index i, j, k or something we do not know whether i plus 1 we should go here should we go here or should we go here and so on, right. So, there is a not a there is no particular direction which this mesh could be traversed to the neighbours using a particular index.

Now, again if you look at the number of neighbours. For example, for this cell it has let us say or if you take a particular node, ok, not exactly the face neighbours if you take a particular node this node has only has more number of neighbours than let us say this this particular node, ok. Whereas, that was constant in the case of here, ok. I think I used a cell centroid example, I should have used a face or a particular node. So, if I take any node you have only 4 neighbours always, right, that is in a structured mesh.

(Refer Slide Time: 16:57)



Whereas, in an unstructured mesh depending on the angle subtended by these guys. So, here I have 1, 2, 3, 4, 5, 6, 7; whereas, if I take here I have only 1, 2, 3, 4, 5, right depending on the angle subtended by the faces we have different number of neighbours. As a result you do not have a structure in this in this particular type, ok. So, that is the kind of thing.

So, as a result the neighbour information needs to be generated and stored, ok. So, you cannot simply if somebody just gives you an unstructured mesh you have to put in more effort to understand how these neighbours are located, ok. So, we need to know. So, simply by looking at a particular cell I cannot tell which is which cell, ok. For this particular cell I cannot tell i plus 1 is this guy, i minus 1 is this guy and so on, rather I have to keep track of for this particular cell, what are the neighbours. So, I would say for this particular cell we have this one as neighbour, this one as neighbour, this one as neighbour as the face neighbours, such kind of information needs to be explicitly generated and stored, which is not the case in case of structured; where given any cell you can just traverse back and forth anywhere, ok.

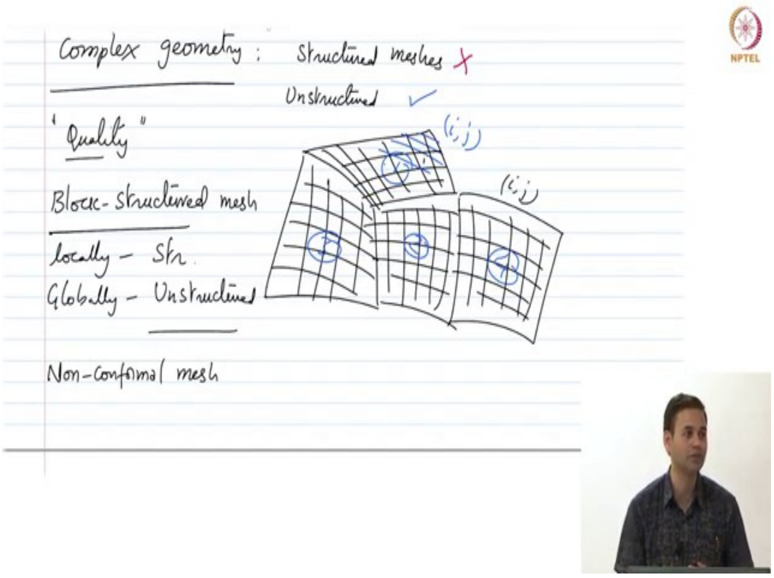
Now, then why do we go for this unstructured meshes if it is so difficult to generate and maintain this data? Like it is an extra effort then what we have for structured. So, why do why do we in the first place go for unstructured meshes? We can make our life simple just by looking at structured meshes, right.

Complex geometry, right. If you have a complex geometry, it is very difficult to mesh these complex geometry with a structured mesh, ok. It is difficult to mesh with structured meshes. So, structured meshes are not good. So, we go for unstructured meshes, ok.

Now, the effort one puts in meshing just a complex geometry could be tremendous, ok. So, for example, people might spend days and days in meshing a particular geometry, ok, let us say an internal combustion engine or a gas turbine chamber and things like that, so you can end up spending few days or a week in just getting a good mesh, ok. Because there is something known as quality which needs to be satisfied by these meshes which will result in obtaining a solution correctly, ok. We are not going into details here. So, we will go through it as and when required, because we are looking at kind of an overview, alright, ok. So, that is what it is.

Now, if we move on there could be also and now that the definitions of structured unstructured is known, I would introduce to you something different that is we call something known as block structured mesh, ok. This is also used in obtaining of solutions. So, block structured mesh.

(Refer Slide Time: 19:44)



Now, as the name implies this is a structured mesh, but comes in blocks, ok. So, let us say I have several blocks. So, this is a block structured. I have some mesh, depending on the geometry I may have several blocks in here, ok, and so on. So, I could have another block here and so on. So, this is a block structured mesh which is basically a structured mesh in each of these blocks, ok. So, 1, 2, 3, 4 these are all structured meshes, but if you look them as a whole it is an unstructured mesh, ok. So, this is basically locally a structured mesh, but globally an unstructured mesh, ok.

So, this is we call it a we understand that this is locally a structured mesh whereas, globally this looks as if it is an unstructured mesh, ok. So, these things are also used depending on the geometry. For example, if you want to mesh a particular aerodynamic shape, or a launching rocket or things like that some of these are used, these kind of meshes.

Now, of course, you also have another type of mesh that is known as a non-conformal mesh, ok. As the name implies a non-conformal mesh is something. Yes, question, Questions. Yeah.

Student: Local and global (Refer Time: 21:36).

What I mean by local and global is if you look at this mesh as a whole, ok, it appears as if it is unstructured, right because given a particular i, i plus 1 or something I can only traverse within this block with that indices, right I cannot traverse here with that. So, I would have a different i, j index range for this and a different i, j index range for this and so on for each of the blocks. So, it is locally for these blocks it is structured, but I do not have a particular index that I can traverse the entire mesh with. So, when you look at it globally like as a whole it is unstructured, right. But if you go to each and every block that is a particular structured mesh, ok. So, that is what I mean to kind of explain that this is a block structured mesh which is locally structured, but globally it is unstructured, ok.

Other questions. Is this clear? Yeah.

Or essentially, you are retaining the benefits of a structured mesh, that is why people use instead of going for a fully unstructured mesh they go for a block structured mesh. For example, let us say you have written a CFD solver or which is exists can only deal with a structured mesh. It can only work if you have i plus 1, i minus 1 and that information, right. In that case, the another way to extend that to handle complex geometry is to have these kind of meshes, these kind of block structured meshes, ok. It may be may not be obvious, but if you think through it, it will be kind of clear, ok.

So, I would suggest you can probably go and kind of check for block structured meshes used in let us say rocket launching applications and things like that, right. If you have some kind of a body fitted mesh then people use these block structured meshes, or any other aerodynamics and things like that. Yeah.

Student: Sir, there also number of number of (Refer Time: 23:20).

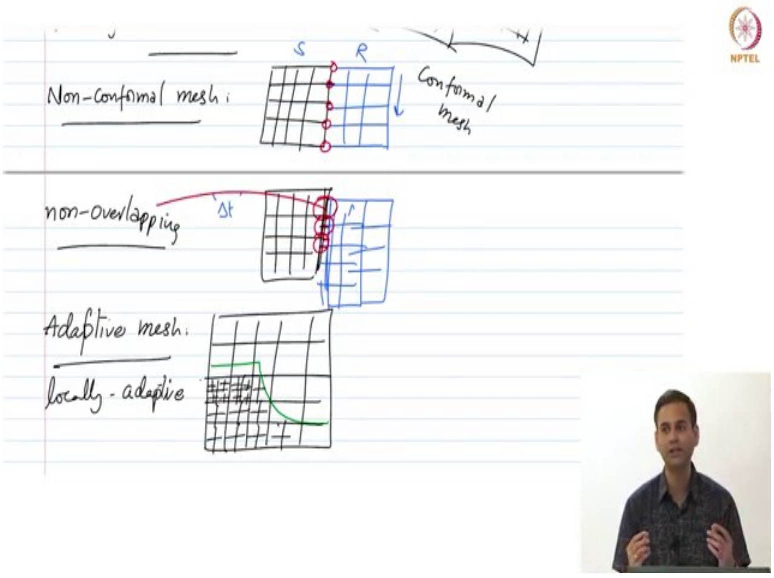
For a.

Student: (Refer Time: 23:23).

Could change, yeah, they could be the same locally and globally or they could be different, ok. But in this particular case I have drawn it is all the same, ok. But the idea here is that there is no single index with which you can traverse the entire mesh with, you still have to treat them as like blocks and then go about it, ok. Other questions? No, ok.

So, then we talk about something known as non-conformal mesh these as the name implies this is not conformal, ok. So, essentially we do not have a one to one mapping, ok. Something we call it as conformal if you have a one to one mapping, let us say think of an application where we have to use a two different meshes, ok. I have a mesh here, and then I have another mesh which I draw in a different color, ok. So, I would have another mesh.

(Refer Slide Time: 24:03)



As of now I have drawn it as conformal, ok. So, these are now as I am sorry. As of now I would draw with a conformal mesh, ok. So, this is conformal. By conformal I mean that I have these two meshes and these nodes are matching at every location, ok, the kind of end at the same location we call it as let us say a conformal mesh.

Now, let us say if we are solving for a rotor stator interaction, or anything else where one of these geometries would move with time, ok. So, if we have a rotor stator where the stator is held stationary and the rotor kind of starts moving, ok. So, let us say this is a rotor, then this will keep moving with time, right because it is rotating. So, as a result what will happen is this has to go to a new position after certain delta t, ok. So, after a delta t we would have the stator remains at the same location, whereas, the rotor would have moved to a different location. This is the same rotor I have drawn, ok. So, it has moved to a different location.

Now, we can see that the nodes are now not matching, right. They are all non-conforming. As a result we have to introduce extra effort to deal with the flux transfer between these faces, because we know that this particular face here is talking or transferring the flux to this particular face here, but we need to have some special treatment to handle these extra non-overlapping zones, ok, right.

So, as a result we are creating new nodes and so on which needs to be treated explicitly. So, we are not we may not use all these things, but it is kind of good to know that they these kind of meshes exist and in what context they are used for, ok. You also have something known as adaptive meshes, ok.

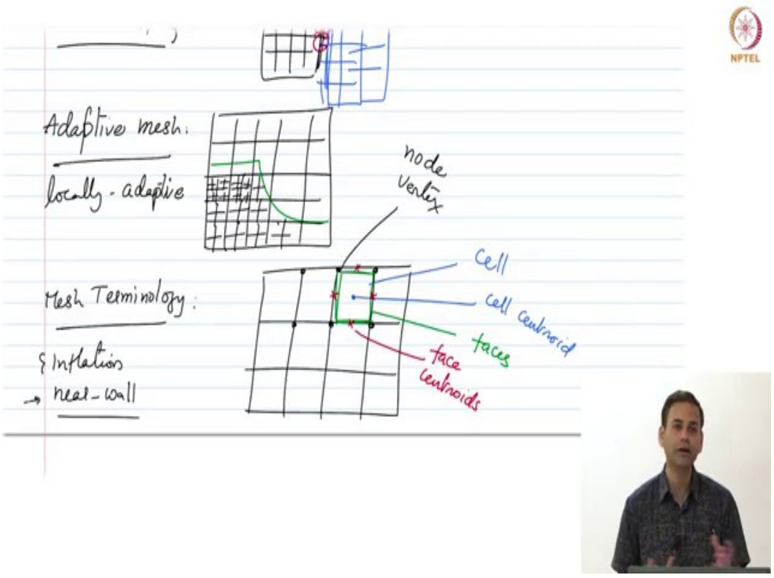
What is an adaptive mesh? Essentially, we let us say have a kind of a mesh that we create. Now, we want to create let us say a finer cells not in the entire domain, but in a particular location in the domain, ok. We do not want a this to be filled with fine cells because if you make smaller and smaller cells the number of cells increases, so we are not interested in that. We know that only in a particular region, for example, let us say in this particular region which is let us say close to the walls, so something we want a mesh to be much smaller the cells to be much smaller than the entire domain.

Then, what we do is ok, we will do something known as a local locally adaptive mesh. What we do is we go here and then start making these cells finer, ok, only these cells whatever we identify we make them finer. And then of course, you can make them much finer and so on, depending on the depending on the context, ok. So, this is a locally adaptive cells or a locally adaptive mesh. This is also used.

Think of it for example, let us say we want to model flow around let us say a rotating you know blades of a helicopter or something then you have this entire domain around the helicopter, but then around the blades you want the cells to be much finer not in the entire domain or a wind turbine blades, right. At these locations you want cells to be much smaller to capture the boundary layer effects and things like that, but not in the entire domain, ok. In such cases you would use a locally adaptive mesh, ok.

Now, you may not be using all of these, but they may be useful at some point, ok, fine. So, we have all these all these different meshes ah, ok. Let us move on.

(Refer Slide Time: 28:20)



So, the terminology here is that, so the mesh terminology. So, as part of this course we will be looking at both structured as well as unstructured meshes and their treatment, ok. So, we will be solving some of these unstructured meshes. But they will be very simple. For example, I will give you an example as an assignment probably you will work with like 4 or 5 cells that are in an unstructured manner, ok.

So, you will be able to write down equations by hand and then solve them with a small program and things like that. Of course, we will not generate a mesh that has like you know 10 million elements, ok, we will not do that. We will work with probably 10 cells or 5 cells and things like that to get to know how it works, ok.

Now, coming to the mesh terminology what we what we call here is that let us say I have a created a mesh, ok. Now, we call this entire thing as a cell, ok, this could be either in 2D or 3D. What I have drawn is a 2D. So, we call this as cell and then we have a cell centroid. So, this is cell centroid, ok. We call all these vertical and horizontal walls or faces, ok. We call these the green ones here as the faces, that make up the cell. So, essentially if you have a cell you have a certain number of faces, ok.

You may need this information as well, you know as you as you go around with an unstructured mesh you need to know which faces make up which cell, or what cell has is made up of what faces and which faces are shared by what cells. These information could be required later on.

Now, we also have something known as these nodes, right which we used in the context of structured or unstructured. These we call it as a node or a vertex, ok. So, we call this as node or a vertex. Then, we also have a face a centroid for the faces, ok. So, these red cross marks are nothing but the face centroids. Now, of course, everything is drawn in 3 2D. If you go to 3D you also have a face centroid, right and a cell centroid and the cells and the faces, ok. So, that is the terminology we will be using, fine. Questions, this part. Yes.

Student: Sir, we know that like (Refer Time: 30:44) Inflation near wall (Refer Time: 30:47).

Right.

Student: (Refer Time: 30:48).

Inflation what? Inflation and?

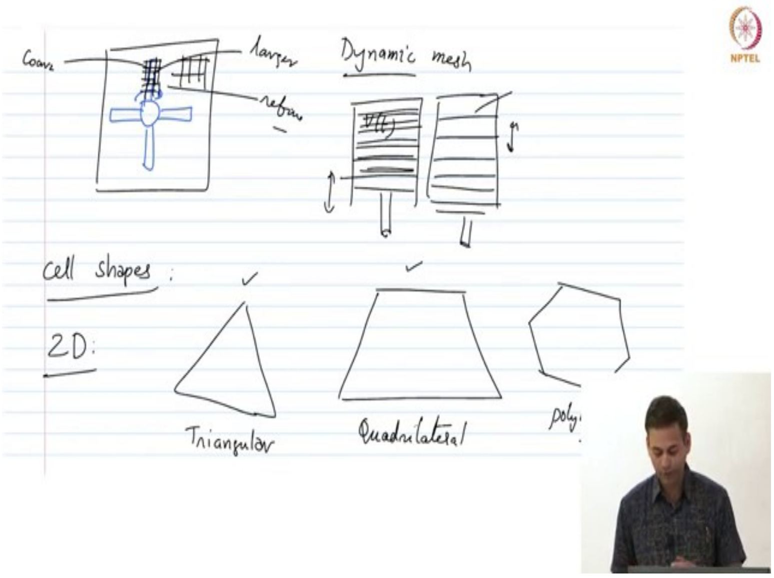
Student: Inflation near wall meshing.

So, the question is adaptive meshing the same as near wall meshing and inflation, right, ok. So, these are kind of terms used by different software, ok. For example, if you take; so, inflation is essentially same as you want to have different cell sizes as you go around a particular location. Near wall is essentially as you close move closer to a boundary, right, you would like to have the cells smaller and smaller to capture the boundary layers that you have, ok.

Now, adaptive meshing is a more general term, where it need not be closer to any of the outer walls, it could be anywhere else in the domain. For example, let us say we have a we have strong gradients in particular part of the flow, then I would want to have a mesh that is much finer in those regions, ok. For example, near wall mesh is something that you create before you start your simulation, right, you know that, I have these walls, near these walls I would create a very fine mesh and you start the simulation.

Now, if I ask you something for example, to solve for a rotating vertex, ok. So, essentially we have a vertex which is there and then I want you to solve for a or maybe a rotating flow, right, then, you would not know in which locations you would get the strong gradients, ok. So, the location where you have to refine could come out as part of the solution; that means, you need to mesh locally refined mesh incorporate that in as part of the solution is progressing, ok. That could still mean an adaptive mesh.

(Refer Slide Time: 32:37)



For example, take off this, for example, we can have maybe let us say a piston cylinder, ok, or maybe we want to solve for a wind turbine or something, so I have I have blades of this fan, which kind of rotate. Now, at particular instant I want to have the mesh to be very fine in these locations, right, but here I can have a larger mesh, right because this is where there could be large gradients, right.

Now, as the mesh is rotating now you want to course in these, you want to make these bigger whereas, you want to refine these areas, so that could be kind of an adaptive mesh. As same as dynamic mesh. So, this is basically a dynamic mesh. But a dynamic mesh again is again in a term used by a let us say particular software where the mesh is changing with respect to time, ok.

Now, a dynamic mesh could have let us say the volume of the cells is changing with respect to time, but here we are introducing new cells, ok. So, you could say let us say I am talking about a piston cylinder, let us say I want to simulate IC engine. Now, the volume in this is a function of time, right because as the piston is moving up and down we have a dynamic mesh in here.

Now, you may not want to have new cells created in here rather you want to have a particular number of cells that you created, which kind of expand like a spring as the piston comes down, ok. As the piston comes down you want to have these cells expand and contract, but the number of cells could still be the same, ok. In that cases you can call it as a dynamic mesh, ok. It could be any of these things, it could be a essentially it is the cells are changing with respect to time, ok. So, that is a kind of a dynamic mesh. Other questions. Yes.

Student: Sir did the (Refer Time: 34:19).

Right.

Student: (Refer Time: 34:22) of the [vocalized-noise] (Refer Time: 34:24).

That would be.

Student: Yeah. (Refer Time: 34:27) non-conformal (Refer Time: 34:31).

It can as well be non-conformal, we depending on the application. For example, in a particular application if nothing is moving then you can still have as a as a conform mesh. So, when you generate everything will be conformal. But if you have some parts which are let us say moving or sliding then you probably will have a non-conformal mesh as we go with it, with the solution, ok. So, certainly you can have all sorts of complexities with the meshes and the way you solve and things like that, ok. Other questions, ok.

Let us move on then move on to the cell types that we may get, what are the cell shapes. So, in two-dimensions, what are the cells that we can have a in two-dimensions? It does not matter whether it is structured or unstructured. What are the cell shapes we can have?

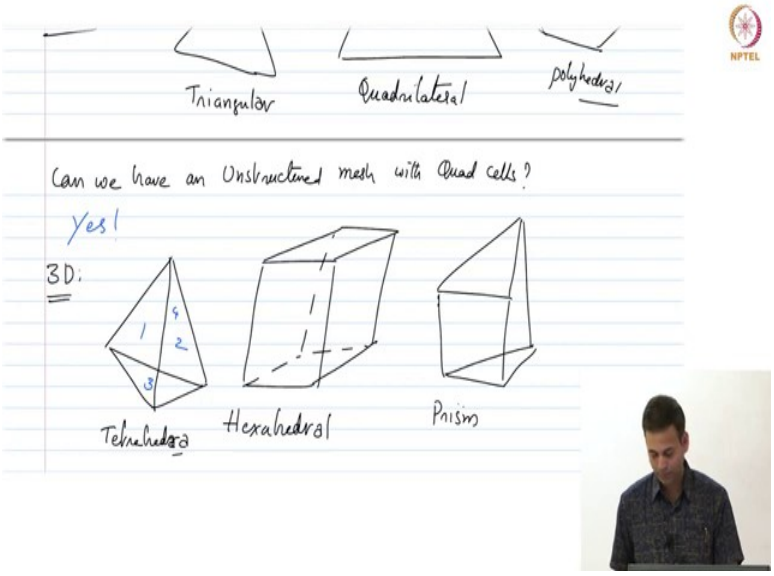
Student: Triangular.

Triangular, ok. So, you can have a triangular cells. What else?

Student: Rectangular.

Quadrilateral. So, more general quadrilateral, ok. Let us call it as a some kind of a quadrilateral mesh. Of course, you can also have something known as polyhedral in which you can have more than 4 you know faces and so on. So, you can also have polyhedral, ok. It can have like some kind of hexagon shape and so on, ok. So, these are the cell shapes that are typically used once are the triangular and the quadrilateral. Of course, the quadrilateral if you are using a let us say a Cartesian mesh or a structured mesh then you would have you can kind of make it a rectangular and things like that, ok.

(Refer Slide Time: 36:21)



Now, can you have an unstructured mesh can you call an unstructured mesh with quadrilateral cells? Can we have an unstructured mesh with quad cells? We can have, right, ok. We can have. The answer is yes, ok. So, you can think about and come back to me later fine.

What about 3-dimensions, what are the cell shapes that we can get?

Student: Tetrahedra.

Tetrahedra, ok. So, we can have tetrahedra. So, that is the tetrahedra is made up of what 4 triangles, essentially 4 triangles. Essentially we have 4 triangles, ok. So, this is this is 1, 2, 3 and then the back face is 4, ok. There are 4 triangles in here. So, this is a tetrahedra, ok. What else?

Student: Hexahedra.

Hexahedra, ok. So, we can have a hexahedra. We can have hexahedra, ok. What else?

Student: Prism.

Prisms, ok. We can have prisms. Now, what I have drawn is a triangular prism, right. A quadrilateral prism is nothing, but a hexahedra, right. You can also have a polyhedral prism as well. So, you can generate prisms on any shape. So, this is a prism. What other things can we have? We have covered tetrahedral, hexahedral, prisms.

Student: Pyramid.

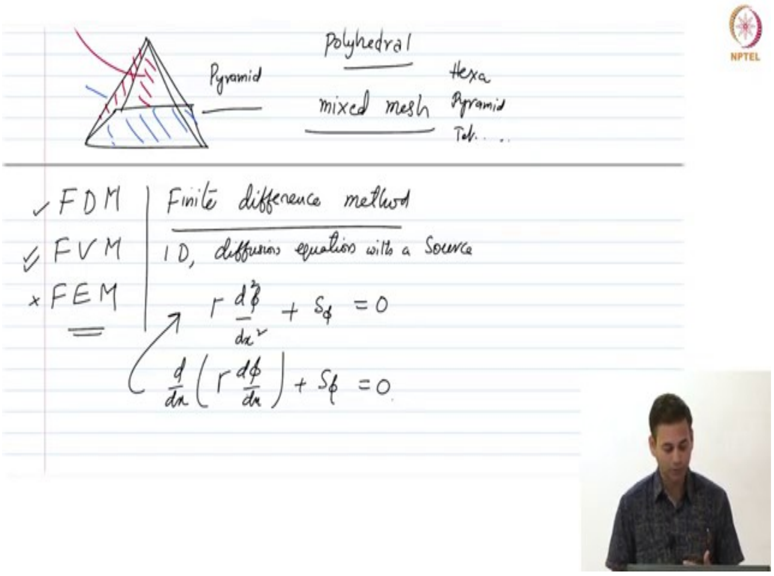
Sorry.

Student: Pyramid.

We can have pyramid, ok. We can have a pyramid. How does a pyramid look like? What will be the difference between tetrahedra and a pyramid?

Student: Quadrilateral.

(Refer Slide Time: 38:26)



Yeah. It is based on a quadrilateral, right, essentially we have a quad here, ok. So, we can have a pyramid, that is the pyramid, ok. So, essentially these are the typical cells. You can also have polyhedral in 3D, which we are not drawing. Of course, all these mesh shapes reminds such that we can also have a single mesh with different cell types, right.

One part of the domain we may have hexahedral then, so we may start off with let us say close to the walls. So, we may start off with hexahedral cells, and as we move away from the wall we may put on pyramids on top of these hexahedrals, right by matching this face with these faces, right. We can have pyramid sitting on these hexahedrals.

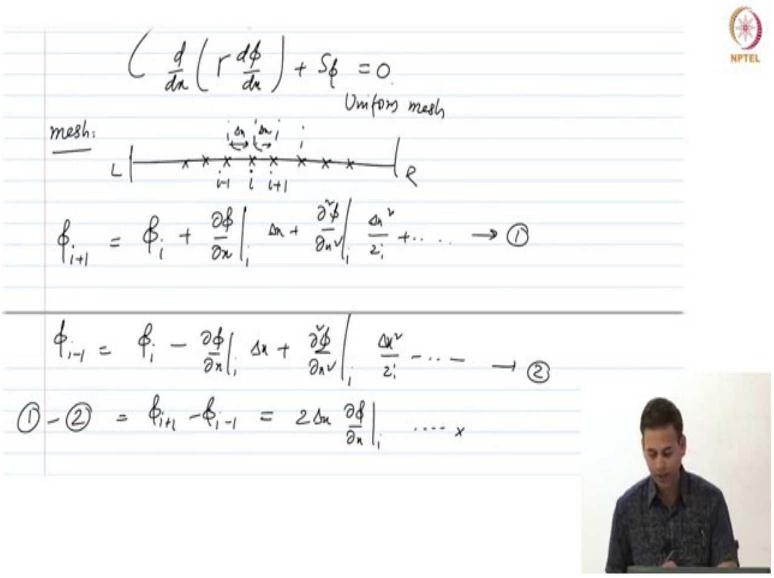
Now, once we have a layer of pyramids then the triangular faces that these pyramids have, right can be attached to the tetrahedra, right and build tetrahedral, and then and so on, right. You can have a mesh like that which is actually a kind of a mixed mesh, right, where you can generate hexahedra, then pyramids, a layer of pyramids and followed by tetrahedra and so on, right. Of course you can also involve prisms and things like that. So, these can also be built, depending on the application, ok, fine. So, we can also have a mixed mesh that is what you take away, fine, ok. Questions still now, these parts? No, ok.

Let us move on then. Move on to an overview of the; so, we have covered the geometry mesh creation type of the cells and so on the next step is the discretization, ok. This discretization of the equations. So, we have of course, several methods. We will be looking at the finite difference method and finite volume method as part of the course. Of course, there are another set of methods known as finite element methods which we are not looking at in the overview, because that kind of there are like several formulations of finite element methods. We will not look at it as part of the course, but we will look at these two to start with and we will stick to finite volume method for the rest of the course, ok.

So, coming to the finite difference method. Let us consider a one-dimension diffusion equation, 1D diffusion equation with a source term, ok. So, let me write it a $\Gamma \frac{d^2\phi}{dx^2} + s_\phi = 0$ ok. S phi is S sub phi, S sub phi equals 0, ok. This basically I have assumed that the gamma is constant, but gamma need not be constant. So, the general equation would be $\frac{d\phi}{dx}\left(\Gamma \frac{d\phi}{dx}\right) + s_\phi = 0$ ok. So, that is the general equation. I kind of rewrote as gamma d square by dx square thinking that gamma is a constant, ok. It is how my conductivity of the diffusion constant is a constant.

Now, we have this equation. So, first thing is to this is a 1D problem. So, the first thing is to essentially generate a mesh, ok.

(Refer Slide Time: 42:11)



So, we have, it is basically 1D. So, we have a set of points we would identify which go from, let us say this is one boundary. So, this is one boundary left boundary and then this is the right boundary and then we have discretized it with several nodes, ok. These are several nodes, and we would identify one particular node we call it as i, the next one we call it as i plus 1, we call this as i minus 1, ok. Let us stick to something like a structured mesh for now, for both of them, we will introduce the concepts of unstructured mesh later on.

Now, in a finite difference method what you do? Essentially you start off with Taylor series expansion of the neighbouring points about the particular point you are interested in, ok. So, essentially we will expand what is d phi dx or I would expand this point, I would like to expand phi at i plus 1, in terms of phi i, ok. This in terms of phi i would be what using a Taylor series expansion? Let us say these are all of uniform distance. They are all separated by a distance of delta x, ok. This is a uniform mesh we want to write express everything in terms about the point i, ok. So, this would be $\phi_{i+1} = \phi_i + \frac{\partial \phi}{\partial x}\bigg| i \, \Delta_x + \frac{\partial^2 \phi}{\partial x^2}\bigg| i$

Student: Dou.

Student: Delta x.

ok. What about phi i minus 1? This would be in terms of phi i would be phi i.

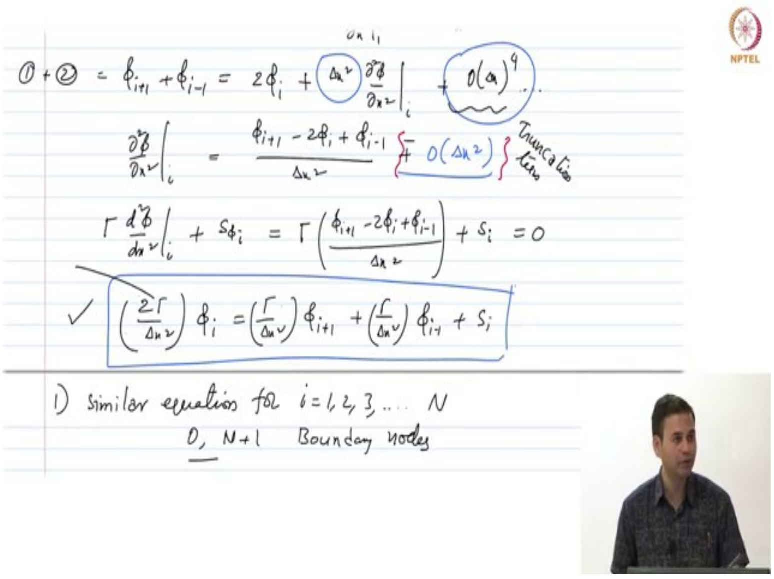
Student: Minus.

Minus, ok, $\phi_{i+1}=\phi_i-\frac{\partial\phi}{\partial x}\bigg|i\,\Delta_x+\frac{\partial^2\phi}{\partial x^2}\bigg|i\,\frac{\Delta x^2}{2}$ factorial and so on, and minus and so on, ok. So, we have these two.

Now, I would like to let us say subtract one equation from the other, or say this is equation 1, this is 2, what would be 1 minus 2? $\phi_{i+1}-\phi_{i-1}=2\,\Delta x\,\frac{\partial\phi}{\partial x}\,i$Student: 2 delta x.

We are kind of obtained an equation for partial phi partial x, right, but we do not want this at the moment, right because what we want is a an equation for a discrete equation for second derivative, ok. So, I want the second derivative.

(Refer Slide Time: 45:01)



Let me try adding these two equations, ok. What if we if I add 1 plus 2? That would be $\phi_{i+1}-\phi_{i-1}=2\,\phi_i+\Delta x^2\,\frac{\partial^2\phi}{\partial x^2}\,i+0\,(\Delta x)^4$phi i plus 1 plus phi i minus 1 equals how much? 2 phi i, right, plus.

Student: Delta x (Refer Time: 45:15).

Delta x square.

Student: Dou square.

And then what will be the next element? So, I have written plus minus, so it will be the fourth order, right that will be and so on, ok. So, from here can I obtain an equation for $\frac{\partial^2 \phi}{\partial x^2} i = \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2} \left(\Delta x^2 \right)$ What would be the leftover term? What would be the order of that?

Student: Delta x square.

Delta x square because we have now divided this delta x power 4 with order delta x square. So, we will be left with order delta x square, ok, which we are we will not consider these terms, ok. So, we call this as the truncation term or truncation error which is of the order delta x square. Question. Sorry.

Student: Will be negative.

Will be negative, that is fine, whether it is positive or negative it is ok. So, this will be minus, ok, this is a minus, fine, ok.

Now, we got something. So, what we got? And we got an expression for dou square phi by dou x square, can I plug this back into the. So, I want to now discretize our governing equation which is gamma d square phi by dx square at location i, ok. So, this is what? This should be $\Gamma \frac{\partial^2 \phi}{\partial x^2} i + s \phi_i = \Gamma \left(\frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2} \right) + s_i = 0$ also needs to be evaluated i, right. This is what I want to calculate. So, this will be gamma times. I would substitute for square plus I would write this S phi i as S i, ok. That is source term evaluate x i, ok weplug in what is the x coordinate at that location and calculate that.

So, if I rearrange this I would get $\left(\frac{2\Gamma}{\Delta x^2} \right) \phi_i = \left(\frac{\Gamma}{\Delta x^2} \right) \phi_{i+1} = \left(\frac{\Gamma}{\Delta x^2} \right) \phi_{i-1} + s_i$ I take this to the right hand side. Say the, what remains on the left hand side is.

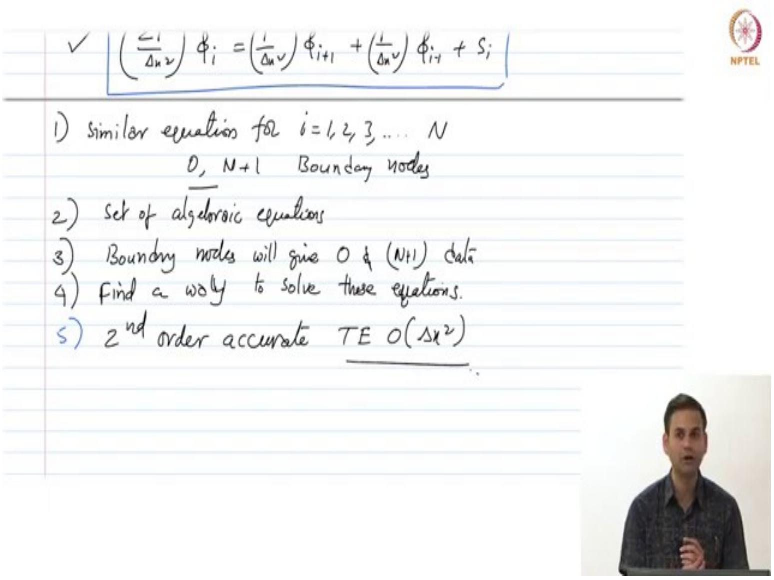
Now, I can write a similar equation for every node that we have, right for every i, i plus 1, i minus 1 I can write a similar equation. So, we can generate a similar equation for i equals 1, 2, 3 and so on to all the way N. Let us say if we have N nodes and where we call 0 and N plus

1 as our boundary you know, the boundary nodes that fall on the boundary. So, we can generate N such equations.

Now, what is the what is this equation? We have started off with a differential equation. Now, what is the equation that we have arrived at? This is an algebraic equation, right. This is basically algebraic equation. We know that we assume that the gamma is known, ok. So, this is known, delta x is known depending on the discretization. So, this entire coefficients are known.

So, we have an equation in terms of some a times phi i equals some b times phi i plus 1 plus c times phi i minus 1, and we know the source term, right we know how much energy is generated because of let us say temperature or something. So, S is known. Essentially, you get one such equation per every node, ok. Now, as that means, what we get is we get N such equations for all the N nodes, ok.

(Refer Slide Time: 49:14)



So, essentially we obtained a set of algebraic equations. Now, the boundary nodes will give the will give the information about the boundaries, right$0, N+1$ will be given through the boundaries, ok. So, the boundary nodes will give 0 and N plus 1 data. Of course, we need to find a way to solve all these equations, find a way to solve these equations, ok, fine. So, that is about a about finite difference method, ok. We have kind of discretized and got these equations.

Now, I want you to kind of remember what we got here, ok. So, we got something like 2 gamma by delta x square and so on. Of course, we did not do any integration on a particular cell and things like that, right. So, we just took the equation as it is, introduced Taylor series expansion, changed the terms and substituted, and then arrived at something like this. This is of course, second order accurate. We call it second order accurate; because the truncation error is of the order delta x square, ok. So, this is second order accurate because of the truncation error being order delta x square, fine, ok.

I think this part, all of you know; you might have done a course on finite difference method or a numerical methods, so that is already known. We will proceed to finite volume method in the next class, and then highlight the difference between finite difference and finite volume and we will move from there, ok.