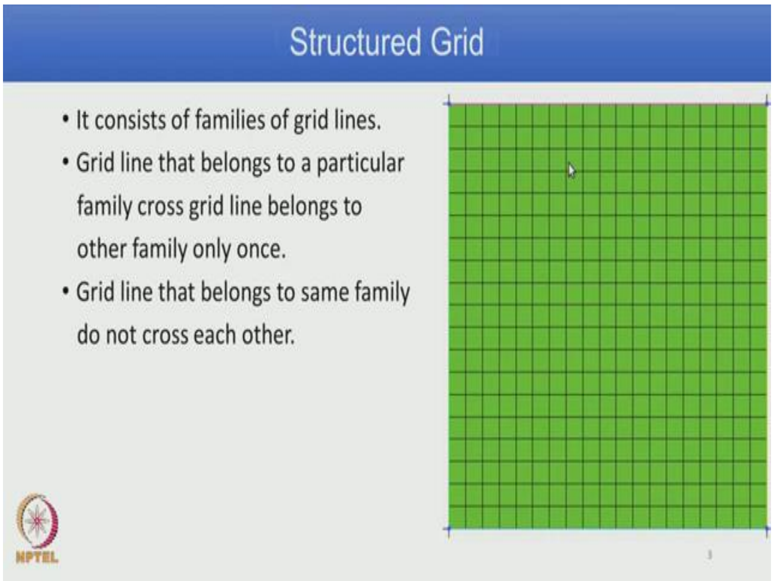
Foundation of Computational Fluid Dynamics
Dr. S. Vengadesan
Department of Applied Mechanics
Indian Institute of Technology, Madras

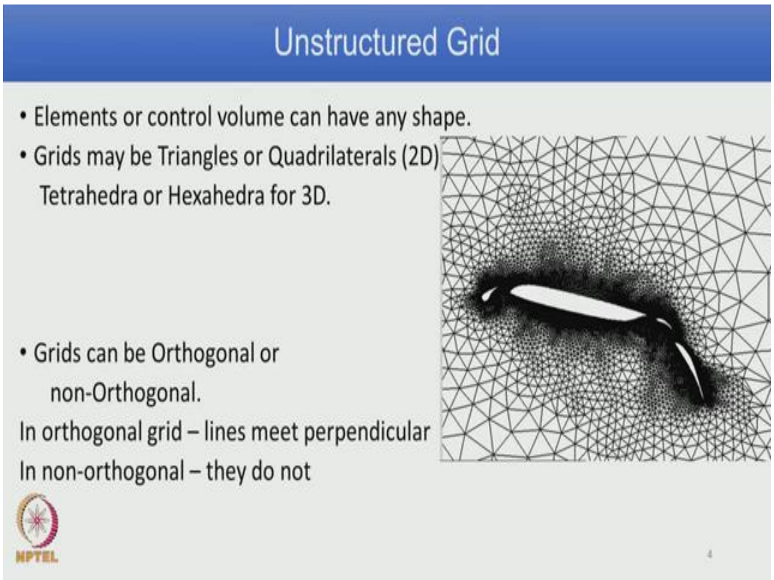
Lecture – 07

(Refer Slide Time: 00:55)



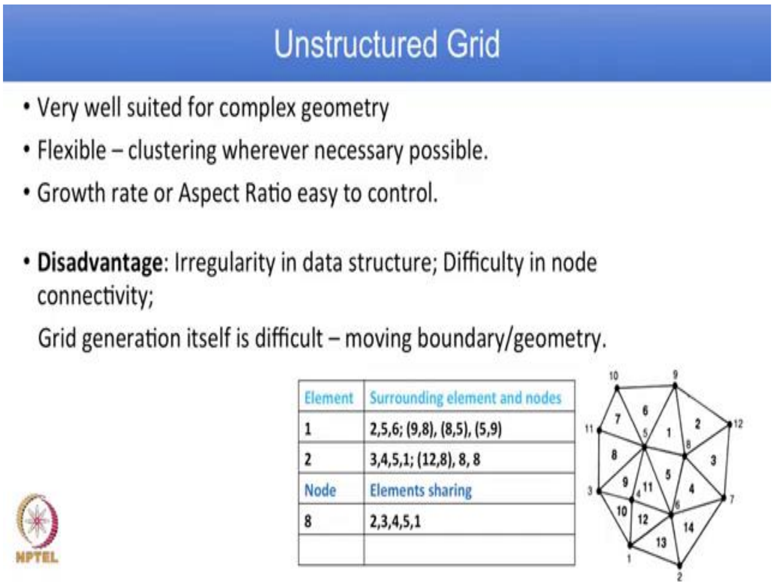
Welcome you all again this course on CFD. Today, we are on module two of week two. Last class, we had seen grid computational domain steps involved in CFD. And today's class, we will go to next category the grid that is unstructured grid then we get into the procedure of finite difference formula. So, in structure grid, we learned the family of grid lines, one family of grid line do not cross same family of grid line. Then grid line that belongs to one particular family cross grid line that belongs to another family only once, and we also learned that the number surrounding nodes for every grid node is the same for the structure grid. We also learned the advantages as well as disadvantages associated with structure grid.

(Refer Slide Time: 01:43)



So, there should be an alternative, which is known as unstructured grid. Here the elements or control volume can have any shape, so there is a definite shape and that shape is same in case of structure grid. Whereas in unstructured grid, you can have any shape, so what is shown here is an example, here a flow passed aerofoil with front portion as well as attachment on the back portion. As you can observe here, you have elements of different shape. It is usually a triangle or quadrilaterals for the case of 2D or tetrahedral or hexahedra in the case of 3D. In general, grids can be orthogonal or non-orthogonal. In the case of orthogonal grid, the lines meet perpendicular; in the case of non-orthogonal, they do not meet at right angles.

(Refer Slide Time: 02:50)



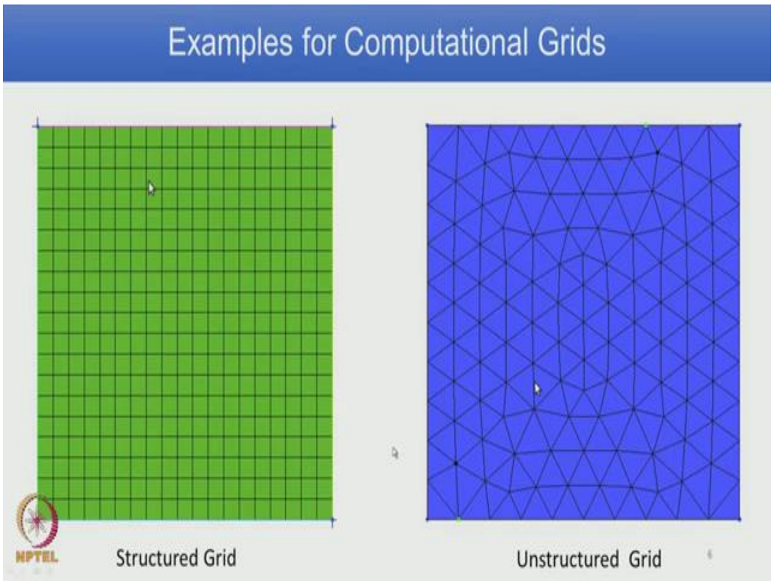
Because of this grid arrangements, it is very well suited for a complex geometry, so as shown before just before, flow pass on aerofoil with attachment in the front as well as the back with the different geometry, second of a complex arrangement. Just one example, structure grid generation is difficult; hence unstructured grid is very well suited. And we also observed there is a flexibility in terms of clustering, so wherever there is a geometry change, or wherever there is a flow change, flow condition change, wherever there is a gradient then you want to have a fine mesh; elsewhere where there is a flow is can have a steady no geometry then you do not handle to have a fine mesh such flexibility is possible. It is also possible to have even within the fine mesh, adjustment of growth rate or aspect ratio. But then you observe that it is show uncomfortable to define exactly the node connectivity.

So for example, if I zoom one area, and such that mesh arrangement, you can see here one is number 1, 2, 3, 1 they are all elements and for each element, there is a node that is defined, so for example, element one has node 9, 8, 5. Element 2 has node 12, 8, 9; similarly for node 3 has three nodes and so on. So, one has to build the table explaining elements and surrounding element nodes. So for example, element 1 has element 2, 5, 6; so node 1 has element 2, 5, 6; and element 1 and 2, shares node 9 and 8 so that is what is shown here the bracket 9 and 8. Similarly, element 5 and element 1 shares two nodes that is 8 and 5 that is what is given here. Similarly, element 6 and element 1 shares two nodes

5 and 9 that is what is given here. So you have to go and build a table connecting node, element, surrounding node and surrounding element.

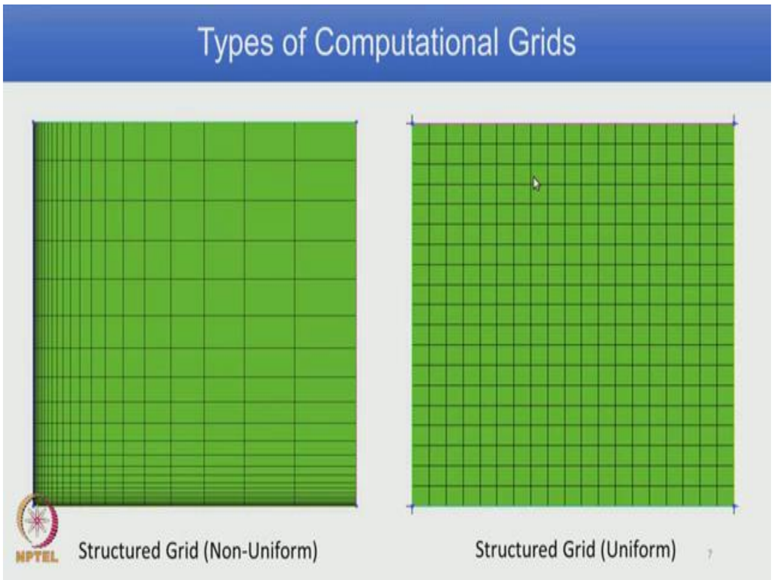
There is also another way of writing, so for example, node 5 is shared by five elements, so in this case, it is 1, 2, 3, 4, 5. So, you can also build a table like this node eight that is this node is shared by so many elements 2, 3, 5, 4, 1; so if I go in the cyclic order 2, 3, 4, 5, 1. So, one has to build a table explaining nodal connectivity and elemental connectivity so such complex situation is there for unstructured grid. when it comes to moving boundary, so when it comes to problem where the geometry is moving, for example, elastic material or the balloon moving or flow in this vibration, this is another example problem in such cases, grid generation and to prove that generated grid at every instant of time, or every movement of the problem requires enormous computational time, and there is a difficulty before going to the solution. So, though unstructured grid looks very good for complex geometry or complex or gradient change problem, flow condition change problem. It also has its own disadvantages. So, obvious question whether you can combine advantages of structured grid and advantages of unstructured grid and get into the form what is known as hybrid kind of a arrangement, yes, it is there. You can have hybrid kind of arrangement.

(Refer Slide Time: 07:31)



So we are going to see some examples of computational grids, this we have already seen structured grid and unstructured grid for the same problem and same domain.

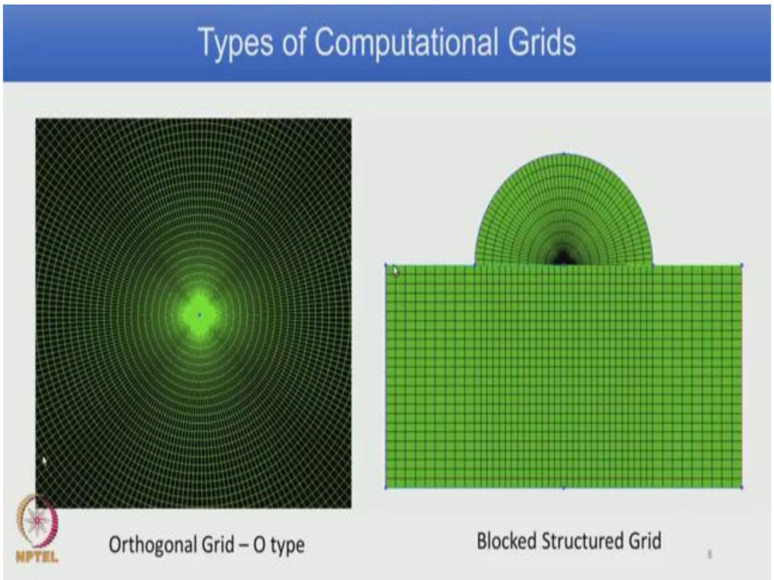
(Refer Slide Time: 07:43)



And even in the case of structured grid, we have a uniform mesh in the sense if you take all the vertical lines, all the vertical lines are equally placed. So, if you say del x is the distance between one vertical line and the next immediate vertical line, the same del x is maintained if you go elsewhere. So, if you take this grid line, the next grid line is at a distance of del x from this grid line. Similarly, for the other direction, all the horizontal lines are equally placed, this is what is known as uniform grid. And for the same problem for reasons of geometry change or gradient happens, you do not want to have a uniform mesh, but you can have a fine mesh. So, if you take for example, all the vertical lines, they are very closely placed here, then it is slowly stretched, you get delta x higher, delta x when you far of in this direction. Similarly, if you take all horizontal lines, then it is very finely placed near this bottom, then it is slowly stretched and we get coarse match as you go away from this.

And you get here a different mesh size compared to mesh size in this corner. And elsewhere it is different compared to some other place. So, this is what is known as non-uniform mesh. So, the problem where you do not want have a uniform mesh row in some region then it is possible to have combination of structured mesh. So, in some region you can have a mesh, structured grid still, in some other region, you can have a non-uniform mesh structured grid still. So this combination also one has to decide where to have a fine mesh, where to have a coarse mesh.

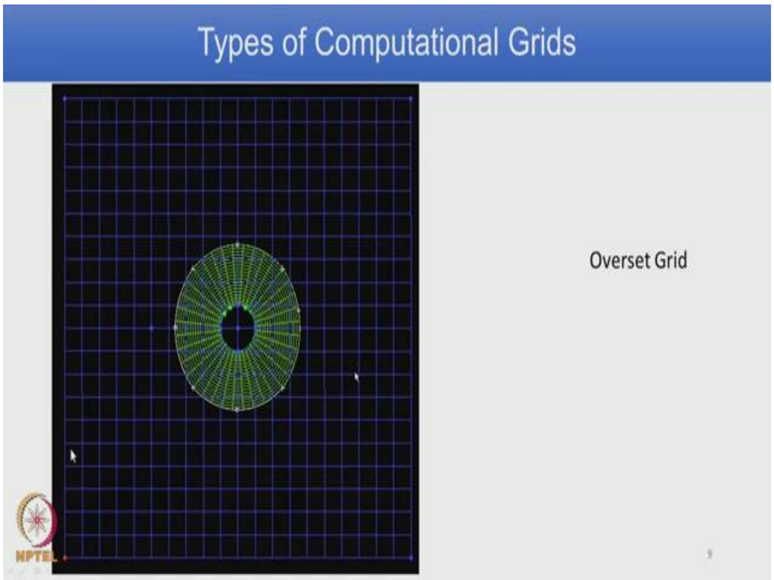
(Refer Slide Time: 09:47)



This another example is again a structured grid, as I mentioned at that time, depending on the final shape of the grid, you can call it as a C grid or O grid. And here is the example for O-grid; the centre that you see here is the geometry, the grid which is generated body fitted the geometry and very close, it is very dense, near geometry is very dense, that is why we are not able to see very clearly the grid line, but as you move away, it is stretched and you get clear. And it is structured grid, so it follows the definition of structured grid. And all the grid lines, they follow the orthogonal condition and you get to see here. And this is what is known as a blocked structured grid; as I mentioned before, you can divide the entire computational domain depending on the nature of the geometry or nature of the flow condition, you can divide the region, and for each region, you can separately generate grid.

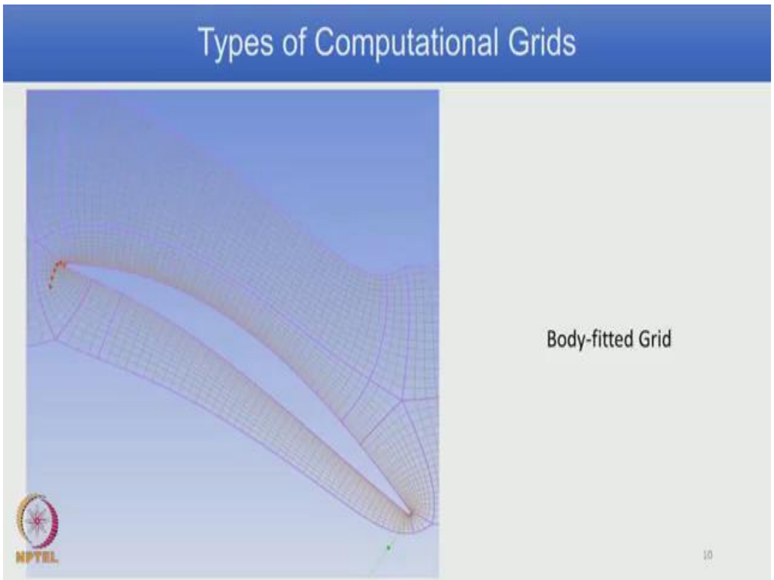
So, here is the example, you see bottom, there is a rectangular kind of a domain; and there is a doom, which is circular kind of a domain, and here you have a uniform mesh in the rectangular type of a domain. For one part of the domain, there is a uniform mesh then there is another type of uniform mesh then third type of uniform mesh here. And then on the top – doom, you have again structured grid, but it following a body fitted, some kind of a C shape grid. So, what you have to be careful is the interface between one type of grid to the another type of grid and that is need to be taken care of while solving. So, it is possible to generate mesh region wise, zone wise, different type of structured grid or combination of structured grid and unstructured grid.

(Refer Slide Time: 12:02)



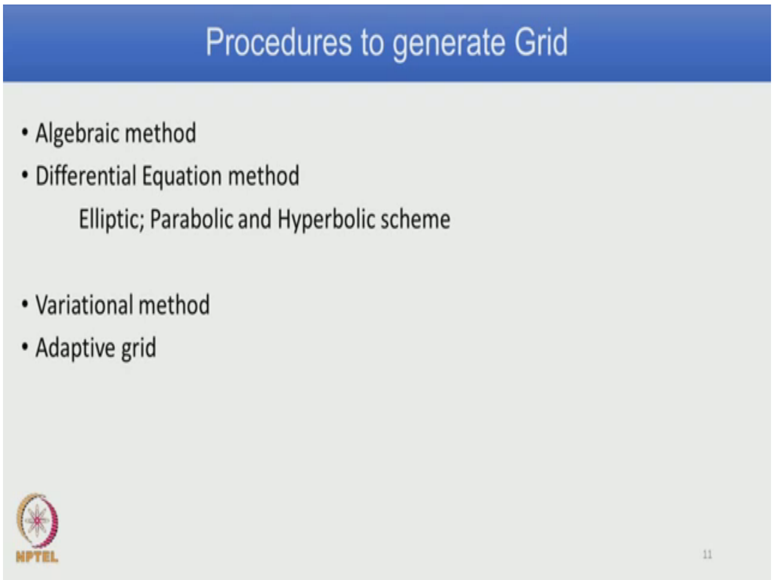
One more example is overset grid, so as you can see here, the background blue color is the uniform mesh and then there is a body fitted O type grid that is embedded over the background mesh this is what is known as an overset grid.

(Refer Slide Time: 12:25)



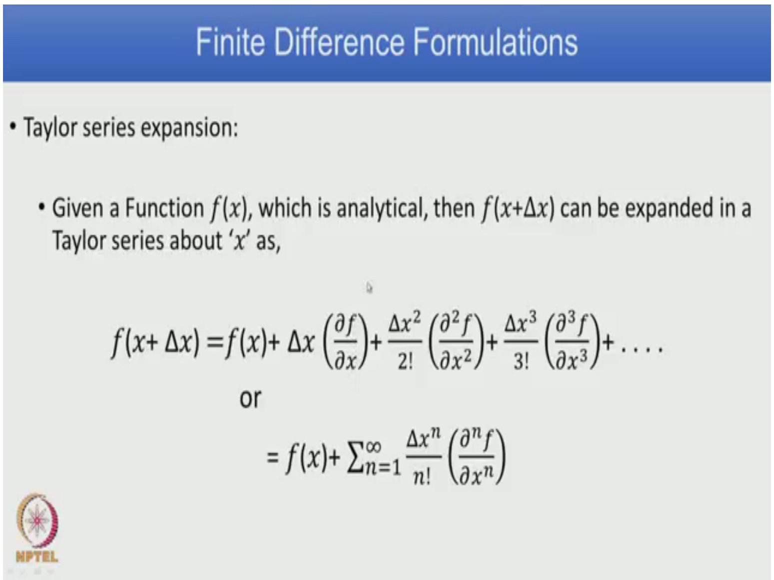
Then this is again example for body-fitted grid, and this is geometry, which represent aerofoil and you have a grid one family of line following closely the geometry and another family of line that is running normal or emerging from the surface, and it follows the orthogonal condition, so wherever the other family of line changes its gradient then this line, another family line changes to satisfy normality condition or orthogonality conditions.

(Refer Slide Time: 13:12)



So, we have seen different examples for different types of grid. There is separate procedure available to generate a grid itself. We are going to the details of the procedure, I am just going to mention the procedure one is known as algebraic method, where you just solve algebraic equation to get grid. Then there is a differential equation procedure and you can recognize it has either elliptic type or parabolic type or hyperbolic scheme available. Then there is also a variational method, and there is also a procedure called adaptive grid. So, in the case of geometry changing it shape as a function of time, or as a solution progresses then you want to have grid adapting to the changes, so that is also possible and that is called adaptive grid. To understand more about grid generation, you need to go to the separate detailed procedure that is not covered in this course.

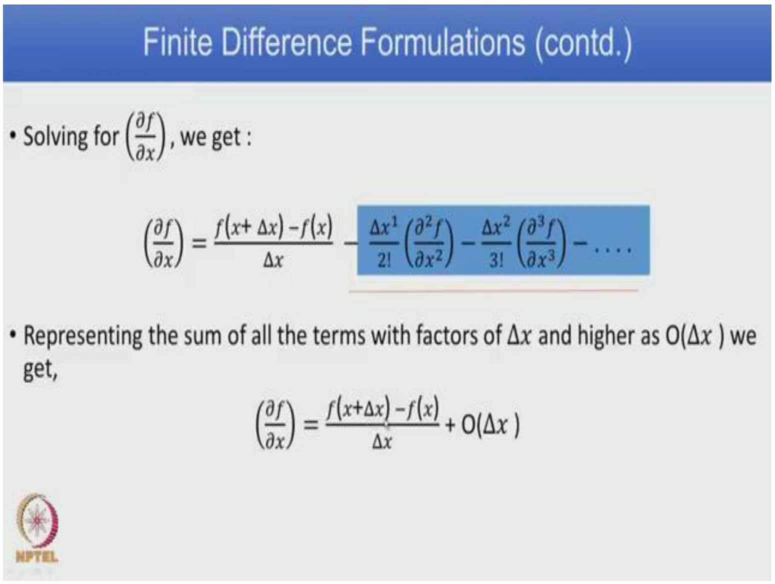
(Refer Slide Time: 14:52)



So, next important topic is how to discretize the governing equation. So to know or to do that one has to know different discretization procedure available. A foundation for all this is finite difference formulation. And in the case of difference formulation, it is basically derived from Taylor series expansion of a function. So, if you are given a function f of x, the function can be expanded, say in this case, it is expanded with respect to delta x in positive direction. And it is given in this formula, which is f of x plus delta x equal to f of x plus delta x into dou f by dou x plus delta x square by 2 factorial dou square f by dou x square plus delta x cube by 3 factorial dou cube f by dou x cube plus and it goes on with many other higher order derivative terms. We have written emitted to only third derivative.

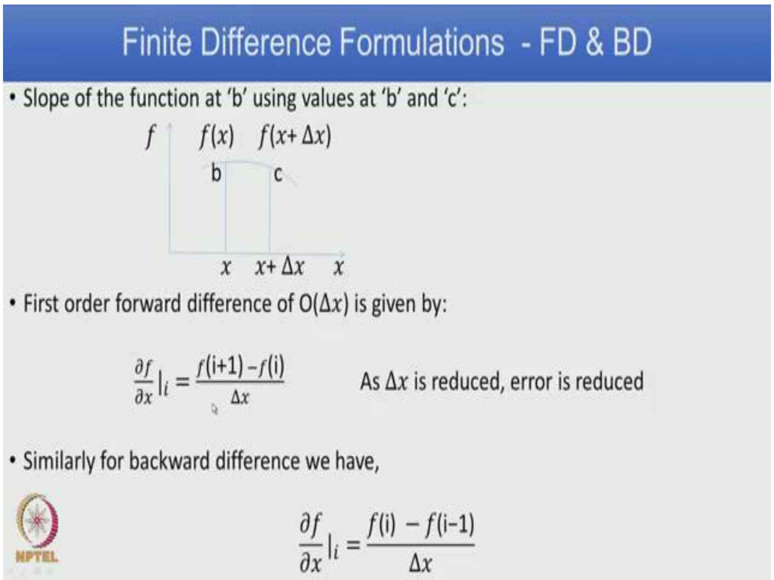
You can also recognize that this follows some sequence, hence it can be rewritten in this form as f of x plus summation going n going from one to infinity delta x to the power of n b n factorial dou n f by dou x n. So you can cross check by substituting value for n in this and see whether you can recover this equation expanded form of that equation. as you can observe partial derivative terms are appearing here, so for example, this term, which is dou f by dou x is the first derivative, second derivative and third derivative. So, if you are interested to get only the first derivative then you limit the expansion here and then get formula.

(Refer Slide Time: 16:51)



So, we will see that here. So, if you solve for dou f by dou x that formula is written here and all other terms are taken to the other side. So, representing the sum of all the terms with the factors of x and higher than we get dou f by dou x of f of x plus delta x minus f of x by del x and remaining term, which is shown in this highlighted portion. They are all not consider and del x one is the leading term and that is of order del x, so this formula which is written here for the first derivative taking only the first two terms and not considering other higher order terms such a formula is called first order finite difference scheme with order of accuracy delta x. this you got expansion going forward in the direction, in other words, f of x is expanded with f of x plus delta x and plus delta x is forward in the direction.

(Refer Slide Time: 18:24)



Basically, what is delta f, dou f by dou x, it is actually the derivative. So if you represent the function with respect to x, and this is the function then b is where you are trying to evaluate the function, and c is where you are trying to get expansion, so it is f of x plus delta x. So function is expanded in forward direction. So, you show it here f of x and f of x plus delta x. So, dou f by dou x is actually the derivative slope at this point x, and you can also represent in terms of i notation, so x is i there is a point of interest, so f of x plus delta x becomes i plus 1, so f of delta x becomes f of i plus 1, f of x becomes f of i and you get formula like this. And this is forward difference formula for the first derivative. And you can observe that as you increase as you decrease the delta x, c becomes closer and closer and you get better approximate of the first derivative. You can follow a similar procedure to get what is known as a backward difference formula for the first derivative, and that is shown here as f of i minus f of i minus 1 by delta x. So, this is forward difference and this is backward difference.

So in this class, we have seen unstructured grid, examples for unstructured grid, advantages and disadvantages, how to combine different unstructured grid or unstructured grid with structured grid or different structured grid depending on different region in a problem. Then we started building finite difference formula for the first derivative.

Thank you.