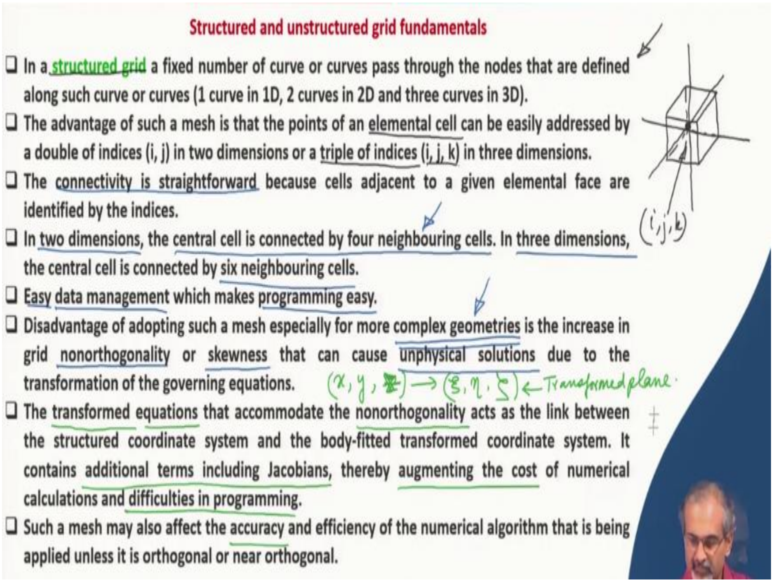
Introduction to CFD
Prof. Arnab Roy
Department of Aerospace Engineering
Indian Institute of Technology - Kharagpur

Lecture – 61
Structured and Unstructured Grid Generation

(Refer Slide Time: 00:30)

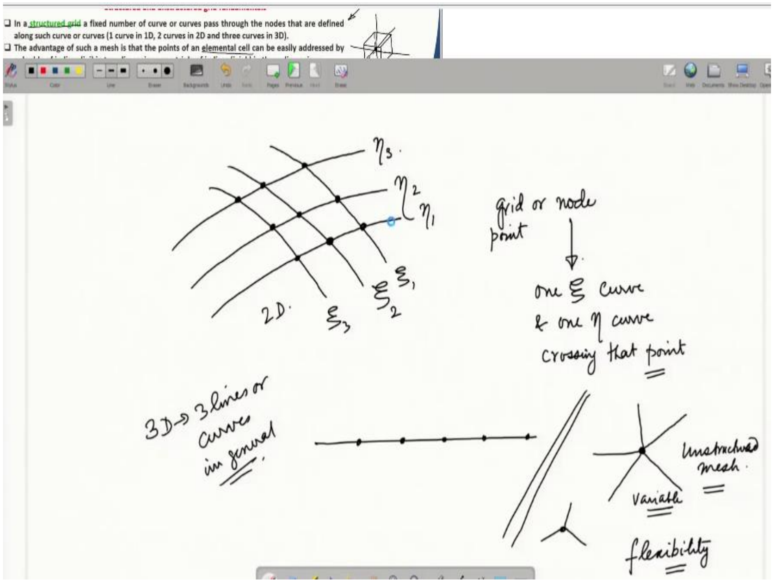


In this lecture, we will begin our discussion on structured and unstructured grid generation. As you all know that all through this course on computational fluid dynamics, we have learned different numerical techniques to discretize governing partial differential equations to impose different boundary conditions and generate solutions. But very often in the practical world, you may actually have to deal with complex geometries which we did not often mention about over here because we wanted to keep things simple.

But complex geometries cannot be avoided in real life. Moreover, in order to do the computational fluid dynamics in a meaningful way, we have to capture those regions of the flow where a lot of physics is actually happening. Large gradients have to be captured or certain complex flow phenomena which are occurring at interfaces have to be captured and so on.

So for doing that, that means both to cater to complex geometries as well as to capture different physical phenomena which are occurring in the flow field, we need to have a very appropriate mesh or grid and in these next few lectures, we are going to talk about the different aspects of grid generation and we are going to talk about how we do it in both a structured as well as unstructured sense. So if we look at these bullet points, we are first beginning to discuss about structured grid.

(Refer Slide Time: 02:18)



The basic concept of the structured grid is that we have a set of curves let us say in a 2-dimensional plane. We can have 2 curves, let us say one is part of the xi family, another is a part of eta family. So like this curve of the xi family, I can have more of them, so I can name them as xi 1, xi 2, xi 3 and so on. Similarly, the eta family of course can also have different levels.

Now, if you notice that the intersection points over here would end up forming a grid or a mesh. Now if this is the grid or the mesh that you are going to use for a certain flow computation, then the important aspect is that each grid or grid point or node finds one epsilon curve and one eta curve crossing that point, there is no more or there is no less. How about a one-dimensional situation? So we can think about a straight line and then there would be points along that straight line, where each point will be crossed by one line.

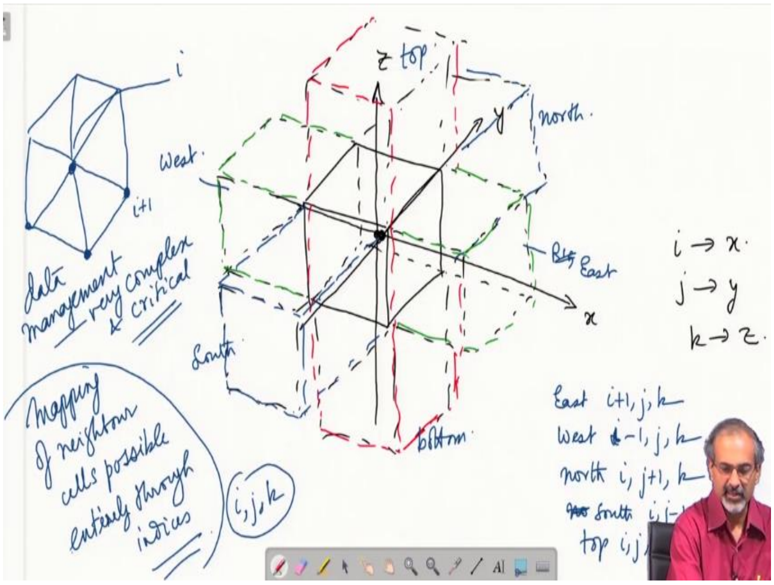
In 2D, each point is being crossed by 2 lines and one from each family. In 3D, it obviously means 3 lines or 3 curves in general. So that is the basic concept of a structured grid. In an unstructured, we will see that if we define a point, there is no restriction on how many lines may meet that point. There may be many. In general, a number of lines which meet at that point is variable, it is not fixed at all.

So if I look at another point in some other part of the grid, I may have fewer or more number of lines meeting that point. This is typically what we see in an unstructured mesh. Of course, that also means that you have a lot more flexibility in terms of locating or defining points in the flow field because you do not have constraints to satisfy that you need to pass so many fixed numbers of curves through that point.

We will discuss more about the unstructured approach later, let us go back to the points once more. So looking at the first bullet point we have discussed to some extent through the small sketches we made just now. Now advantage of that kind of a mesh is that points of an elemental cell, now when we talk about elemental cell, we may even consider it in terms of finite volumes.

So, you can actually have a controllable cell located at the center of that volume, which is the control point surrounded by the elemental cell and it can be easily addressed by double indices in 2D the triple indices in 3D. So here, we have made a cubicle control point and the center of that cell happens to have i, j, k triple index.

(Refer Slide Time: 07:05)



So if you were to draw the layout of such cells in 3-dimensional space, how would it look like? The basic cell, so this kind of sketch would actually address both a finite difference as well as a finite volume scenario because if you just look at this control point, it is like looking at a good point. If you look at the entire cell, it is looking at a finite volume. So if we draw 3 orthogonal lines passing through that point, we mean these as x, y, z directions.

And then if the i indices move along x, j along y, and k along z, then you can imagine that the next cell to the right would be lying like this, the left like this along y it will be somewhat like this and along z like this. Of course, it is easier to do it with colors. So if you say red is your z direction cells, so this is how they exist. Green is your x direction cells, so this is how they exist, and blue so that gives you a rough structure in 3D.

How would you indicate all of it in terms of indices? So let us say in the x direction this is the right cell or let us say east, west, north, south, top and bottom. So east of course would be mapped like this, west as this, north. That means each of the neighboring cells can be perfectly mapped just through indices. Notice that this would not be possible in an unstructured grid scenario because we said a node point where many lines can meet.

Where are these lines coming from? There are neighboring cells which have their cell faces and these lines are essentially part of those cell faces. So if I have a grid point let us say i, I cannot essentially call this as i+1 because there is ambiguity. One can even call this as i+1 or even this as i+1 because there are multiple directions. There is absolutely nothing fixed or sacrosanct about a certain grid point and its neighboring grid points.

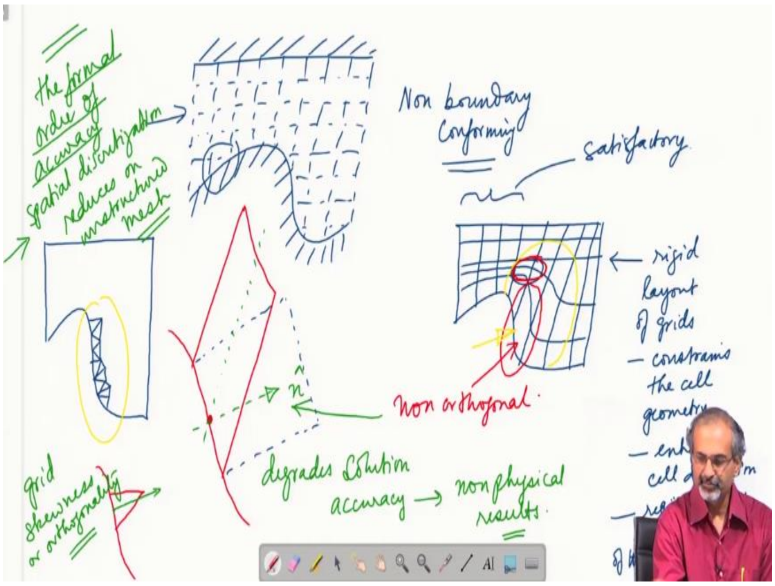
Which means that each and every grid point is an entirely separate entity and therefore you need to know first of all that this good point or node is a part of which all cells and which all cell faces come and meet at this node, which makes the data management very complex and absolutely critical for success of an unstructured grid based flow solver because a lot happens in terms of exchanging information between the grid database and the solver calculations as the competitions go on.

However, as we already saw in a structured grid framework as soon as I specify i, j, k I know exactly which cell it is, which are its neighbors and most often in flow solutions, these are the two important information which are essential and most often complete. So as we said earlier, connectivity information is straightforward. In two dimensions, the central cell is connected by 4 neighbors, in 3D it is 6 neighbors.

We just made a simple sketch and if you think about the 2-dimensional form of that sketch, you can very easily figure out that there are going to be four neighbors instead of six. Easy data management and also mix programming easy. However, the disadvantage primarily comes in adopting such a mesh to complex geometries and this is where the data management, programming, everything is far more complicated in an unstructured grid scenario, the unstructured grids are still much, much more preferred over structured grids in complex geometry situations.

Why is it so? Because in complex geometries it is very, very difficult to maintain grid orthogonality or maintain low skewness and both of these can degrade the accuracy of numerical solutions or even produce on physical solutions, especially when we try to transform the governing equations and solve them in a computational space.

(Refer Slide Time: 14:50)



Let us make a small sketch to understand the issue. So let us say we have a very complex looking domain through which the flow is moving. These are the boundaries which are defining the flow passage. This is the inflow and this is the outflow. Now, if you want to fit a structured grid into this domain, you can imagine that one possible way could be that you try to fit in something like this, but then this is non boundary conforming.

That means the cells are not going to exactly match their faces with the boundary and that can create a lot of computational issues. For example, if you zoom into a cell like this it is half cut, so it does not match with the boundaries and therefore non boundary conforming issue. On the other hand, if you try to conform to that boundary, the mesh may actually try to deform itself. In the process what will happen is the mesh will get extremely distorted in some regions.

You can see that there are cells which are formed of this kind, which are extremely narrow. There are cells which are extremely large and skewed. Moreover, these cells are also non orthogonal to the boundary. What does it mean? That if the boundary is going this way and the cell is looking like this, if this is the cell face which it is sharing with the boundary, if you try to construct the local normal to the surface, the local normal which points outward should be in this direction.

Whereas the cell direction is far away, the cell orientation is far from the normal. This is the instance of non-orthogonal grid. This degrades solution accuracy and also can produce nonphysical results because of huge amounts of errors which are committed when you try to solve the discretization equations in such cells. Therefore, the intended goal is to somehow get these cells orient in a manner like this, exactly normal or near normal.

But the difficulty with structured grids is that because you have a very, very rigid framework within which the grids have to be led, as I already told you that there are xi and eta lines which are intersecting and you cannot distort that topology, that kind of layout, therefore the rigid layout of grids constrains the cell geometry and therefore enhances cell distortion in certain regions, where in regions where the geometry of the flow domain changes rapidly.

So you would have noticed that the grid is not all that bad in a region like this. So this is satisfactory somewhat, but the grid becomes extremely worse in a region like this because of the rapid changing geometry in this part of the domain. So very rapid slope changes of the domain can degrade the grid enormously. In the given scenario, what would an unstructured grid do? That is a question to ask.

The unstructured grid may have filled up this domain in a manner like this and thereby you will not have extremely stressed or skewed mesh at all. You would very often if you zoom into these cells and try to draw them, then you will find that there is a steep slope coming like this, but the cell stands normal to the surface. The local normal to the surface is kind of oriented nearly or exactly with the cell.

Therefore, you could actually address the issue of great skewness or orthogonality better with an unstructured mesh. So they are extremely good at matching up complex geometries or conforming with complex geometry, but then again in general, the formal order of accuracy in terms of spatial discretization reduces on unstructured mesh. There are formal or pretty systematic proofs in order to justify this claim.

We are not doing it here, but we are just stating it as a fact that in spite of this flexibility and the advantage, we again have another penalty. That means there will be degradation in formal order of accuracy when you try to use unstructured mesh. So, there is no free lunch as they say, you have to live with a certain set of advantages and again certain set of disadvantages when you choose a certain grid framework.

So the main point is for your given scenario, you have to make a tradeoff between a number of things, between geometry, between the physical phenomena that you are trying to capture, your available computational resources, the order of accuracy that you are looking at and so many considerations. So put together, it is a final judgment that you try to come up with and of course all this is not in isolation of the CFD which is working as the main workhorse.

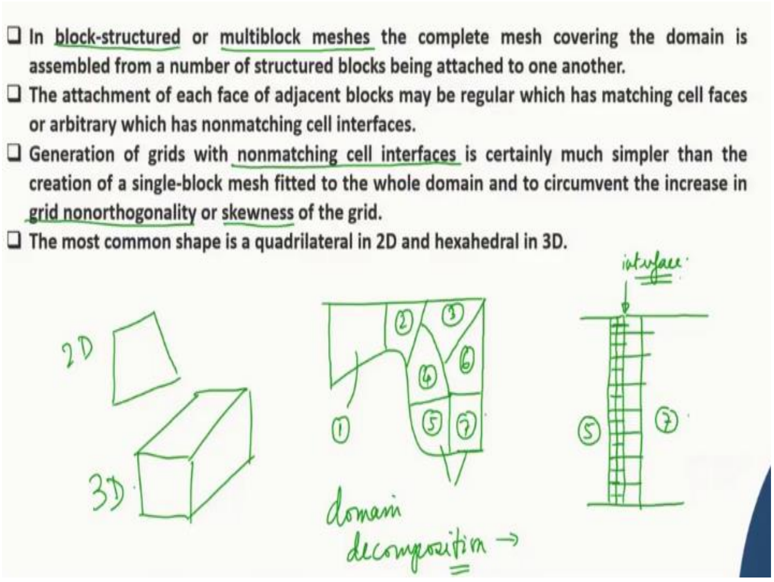
That means there are a set of partial differential equations which you are discretizing and you are solving on the grid, so grid is just your tool. So there are a whole lot of things which are happening making grid as the tool to help you solve for the problem. Many a times in computational fluid dynamic simulations, we use so called transformed equations. So most often in our course we have talked about equations existing in the physical plane and that too in terms of Cartesian coordinates.

But it is not always possible to solve problems in physical space due to different reasons. One reason is because you are handling complex geometries and at the same time you are using structured grid. We can show that it is easier to compute such problems in a transformed plane rather than in the physical plane itself. So then let us say we have taken the problem to a new plane in xi, eta, zeta. As we do that, we are basically transforming the coordinates.

And we are also going to show that the equations get transformed, and then this is being done because you want to address the issue of nonorthogonality because if you want to solve it in a physical plane, you are seeing that the grids are becoming nonorthogonal, but if you take it to computational plane, the same grid actually acts like an orthogonal grid. So this is a very interesting transformation and therefore in the transformed plane because you can see the whole problem in an orthogonal framework.

You can use the usual discretization techniques that we have learned in Cartesian coordinate systems, which is an orthogonal system. But as you do this, you have to incorporate additional terms including the Jacobins, so that augments the cost and also adds on to the difficulties in programming and also to accuracy issues. So cost, difficulty in programming and accuracy all these are issues which have to be addressed when you try to transform the equations and solve them in a transformed plane.

(Refer Slide Time: 25:54)



There is another very interesting strategy that we very often use in structured grid based solutions, and this is basically a kind of strategy which addresses the issue of how can structured grid be suitably used on complex geometries. So there is a strategy called block-structured or multiblock measures which can be used in such a situation. So going back to the same problem that we were discussing earlier, so in that problem, a multiblock strategy may mean something like this.

You may have several blocks which individually function as structured grids. So you have block 1, 2, 3, 4, 5, 6, 7 and if you are looking at the interfaces of these blocks, there can be two situations. One is that let us say if you are looking at 5 and 7 and you are just looking at the interface, how does the mesh look like at the interface? There are two possibilities that the mesh exactly matches on the two sides.

So this is the side of the fifth block and this is the seventh block and the interface essentially lies in between. This is the interface. So there are two issues. One is that the interface is the region through which the blocks will communicate in terms of data, also in terms of connectivity because you are running the indices i and j, specific to each block, it may not be a global indexing.

Again, there may be issues that the grid actually does not exactly match across interfaces. There may be a fine grid on one side and a coarse grid on another side. There are number of possibilities which can actually occur across blocks, but nevertheless, there are strategies by means of which you can actually make them communicate very effectively in spite of these inhomogenities across blocks through the block interfaces and then you have the advantage of still retaining the structured nature of the grid, at least block wise.

Also, this is a strategy which may help in so-called domain decomposition. That means you have decomposed the domain into number of blocks and may help in distributing the calculations of each one of these domains across several compute nodes, which may be a very effective tool in parallel computing. So there could be matching or non-matching cell interfaces as we said.

But by and large the grid nonorthogonality and skewness can be best sorted this way if you still want to address the problem through a structured mesh and then most often in structured meshes we have chord or quadrilateral type of control volumes in 2D or hexahedral kind of control volumes in 3D. So with this, we end this lecture. We will discuss more about grid generation in the subsequent lectures. Thank you.