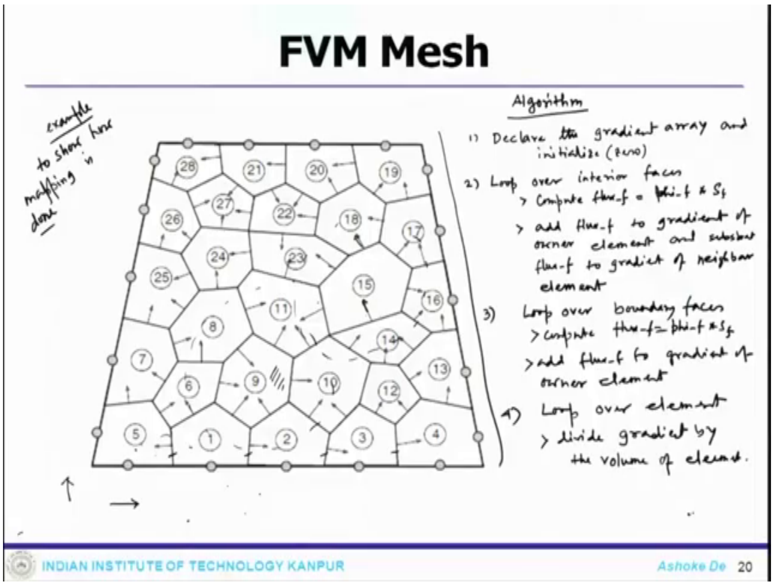
Introduction to Finite Volume Methods-I
Prof. Ashoke De
Department of Aerospace Engineering
Indian Institute of Technology, Kanpur

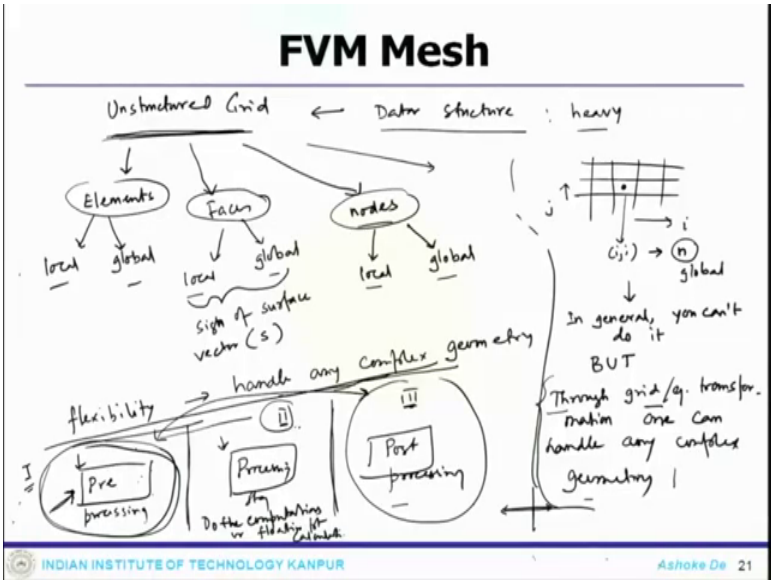
Lecture - 21
Properties of Unstructured Mesh-I

(Refer Slide Time: 00:13)



So, welcome to the lecture of this Finite Volume Method. So, far what you have seen the mesh topology, information, connectivity, data structure regarding both structured and unstructured. And just in the previous lecture we stopped at the calculation of the unstructured algorithm where you have seen that example what we have taken that if you this two-dimensional system, this is the global picture of each elements and the what data structure required and we have discussed the pseudo algorithm also.

(Refer Slide Time: 01:03)



But importantly, so what one should has to keep track of the elements essentially for the unstructured system and unstructured grid. So, you have a huge data structure requirement or data structure handling. So, here you need elements both local indexing and global indexing. Then you have connecting faces where you have a local indexing, you have a global indexing, so which will take care of the sign of surface vector s ok. So, then you need to have the information about nodes. Again there would be local indexing of the node, there would be global indexing of the node and other information.

So, if you see the involvement or the handling of the data structure for a programmer point of view is quite heavy. It is not that straight forward that you have seen for the structure cases because structure cases everything was nicely ordered. So, as you go by i and j loop, you could figure out the i and j element with a global numbering. So, this is your global indexing but unstructured case the complete scenario is different. Here, one has to have elemental connectivity starting from local to global, you have a face connectivity local to global which also includes the sign of the surface vector. Then you have node connectivity, there will be local to global so that 1 to n other mapping can be done.

So, this is where you get more and more involved in the programming and the programming becomes quite challenging. But the other side of it or the flip side of it one can think about that the unstructured system this is where you have lot of flexibility. So, the flexibilities in the sense, you can handle any complex geometry ok, have been said that it does not mean that in a structured system you cannot handle any complex geometry.

To do that so in general you cannot do it. But through grid transformation, through grid and equation transformation, one can handle any complex geometry. So, that means, also the structured system can be used, but there has to be through transformation, but transformation means not necessarily your grid, your equations that means the partial differential equations and all this things needs to be transformed in a regular system or the transform plane and solve it and then get back to your solution to the physical flame of your complex geometry.

So, when you do the transformation, so that also involves lot of challenges. Why? Because you need to do the calculation of your Jacobian, you need to do the calculation of your transform derivatives, you need to do the calculation of your cross derivatives and that will come with all other associated terms, so that will make it again bit of challenging.

So, in a nutshell if you think about it no matter what you do whether you are in a unstructured system or in the structured system, each of them has certain pros or cons, that means, some advantage come with some disadvantages. So, but one has to make some tradeoff and do the calculations. But if your goal to achieve a solution for a realistic complex geometry and if you again go back or recall from your first lecture in the introduction where we have talked about all these applications like flow over car, automobile engines, flow over your aircraft, flow over flow through eyes human eyes; all these are complex problem and the problem of that kind of nature needs some sort of a involvement. It is not only solution of some partial differential equation it requires lot.

Lot means, for any CFD calculation, there are three important stages. One is pre processing stage, then processing stage, then post processing stage. And again I would like to reiterate here the time which takes primarily the first block and the third block. If I say this is block 1, this is block 2, this is block 3. The first block and the third block takes time. This is when I am talking let us say one has already developed a numerical code. So, no matter what kind of numerics we have used, we have used some sort of a numerics which is fairly acceptable and using that numeric you have a code.

Now, you are about to get a solution done for a realistic system. For example, let us say flow through your blood vessels. So, there will be three stages one has to follow. One is the pre processing stage this is block 1; one is the post processing stage block 2; third block is the post processing stage. Now, pre and post that means, the block 1 and block 3 consumes most of your a bulk of your time for calculation done. From start to end these two blocks they consume a lot, why?

Pre processing stage actually includes your domain definitions. And if your other words in your domain modeling, physical modeling and all this things, so essentially your physical domain needs to be transform to your computational domain at the pre processing stage that means, it involves grid generation.

So, whether it is a structured or unstructured irrespective of that it involves grid generation. And problem of some realistic nature or complex problem or complex geometry, it actually takes lot of time to generate grid. So, this takes care of your mesh generation number 1. Number 2 also one has to properly assigns the physical boundary condition to the computational boundary condition because your code will only read that information. This processing stage this is where you do the computation or I would say floating point calculation.

So, this part is taken care by computer. And these days you have a high end computers like clusters, parallax possessing system and in a very layman's language for engineer this computers are essentially the bulk, they are doing the bull work for you the kind of instruction you give them they will do that work; that means, they will just do the number crunching and get back the number that you are doing looking for.

And how that through the set of comments and that comments means that is through your programming and that programming finally, builds to your proper numerical code or CFD code. And once you have given those set of instruction this guy or the computer will do the work for you. Then essentially it reads all the information from your grid generation stuff that means, your element information, indexing, connectivity and all this things are done at the pre processing stage, proper assignment of the boundary condition.

Then you move to the once your calculation is done then you need to analyze the data and that is again consume lot of time because at the post processing stage or rather it is a block 3. So, that again consumes lot of time because the data you got you need to analyze and then only you can correlate with your physical problem. So, again this has to be connected with your block 1 that means, whatever physical problem I had in hand and then I have transformed them to a numerical system or numerical problem, solved it and then the results one has got whether that make sense or not.

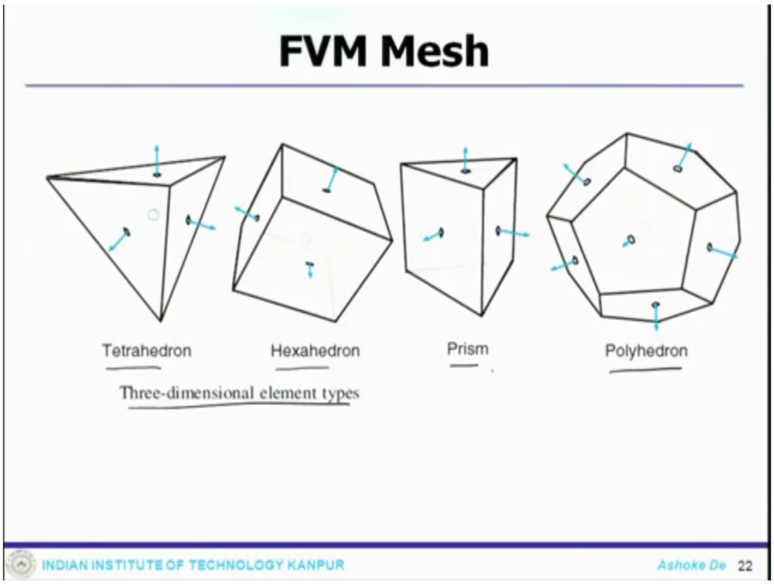
So, the pro processing post processing stage again takes lot of time and so in a I mean in a say I mean in a sense pre and post consumes a lot the computation part compared to them is less. So, this is how one can think about a complex problem can be solved. Now, coming back to our discussion, so when we are talking about grid and all this things this is where we are at the pre processing stage.

And whenever we are talking about the numerics that is something actually not completely processing stage, but it is a back bone of the processing stage. Because the processing stage is the, that means, the you are sending some set of comments through your program and the program does the calculation and through those algebraic calculations. And those algebraic calculation at the set of equations and those equations are coming what we have done through the numerics or numerical approximation and prepared those set of instruction.

And once you convert those set of instruction to the program, then the theoretically the processing is done at this block. So, whenever we are doing those theoretical discussion that is a part of our processing stage or rather the back bone of the processing stage. And those are the calculation should be transform to the programming. And now when you are in the pre processing stage that means, the grid and what we are discussing right now is that structured or unstructured grid, and all this those actually comes under this preprocessing stage. And preprocessing stage takes that kind of effort.

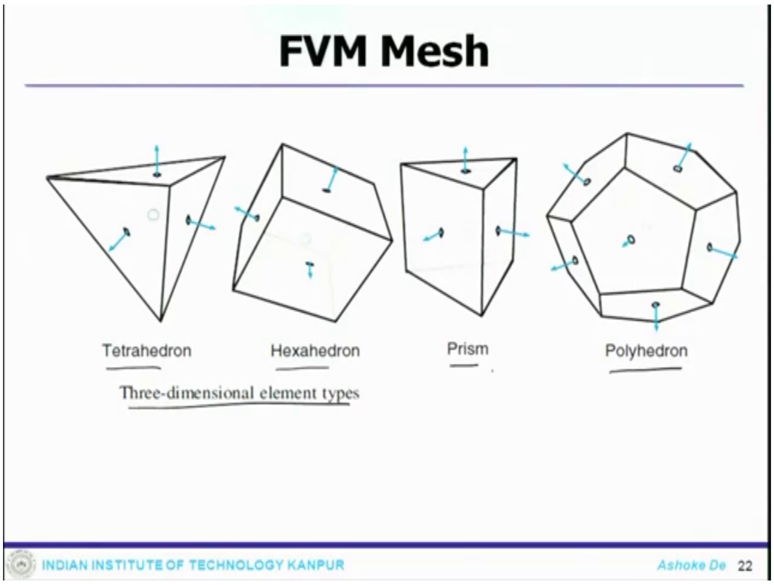
Now, we have discussed on structural system and the unstructured grid system, their connectivity, their information, indexing, data structure all these. So, now, it would be nice to have a look at what kind unstructured elements one can have.

(Refer Slide Time: 14:05)



So, one can have so different kind of elements. So, these are all three-dimensional elements. So, similarly we can have two-dimensional elements also ok. So, we will see that. So, this is one of the tetrahedron. And if you see each of this faces, there are center or centroid. And there are surface vector. So, surface vector actually pointing outward, but in calculation as we have seen when we do a actual calculation, they could be inward also. So, this is a example of a tetrahedron; this is an example of a hexahedron; this is an example of a prism element; this is a polyhedral element. Why all this kind of elements would be very handy while generating the grid, because depending on the problem sometimes you use tetra sometimes, you use hexa, sometime you use prism see all different kind of elements one can use.

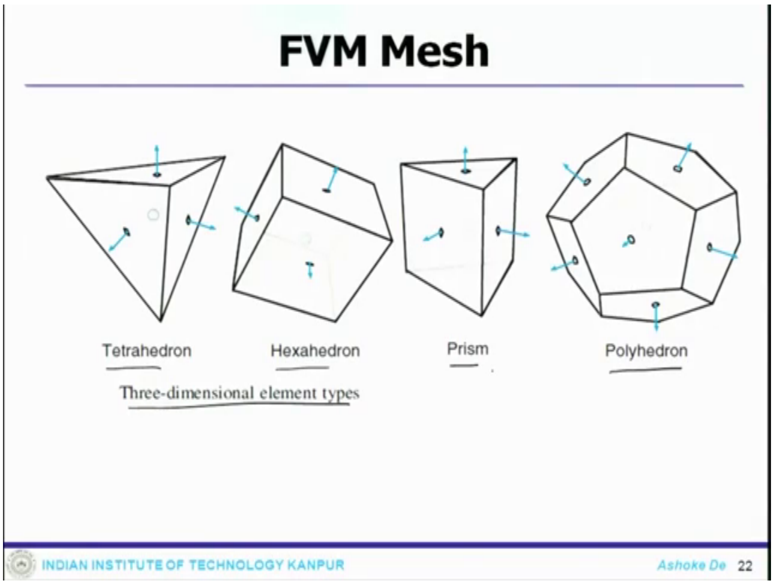
(Refer Slide Time: 15:16)



Now, if you do some sort of a mapping between 3D and 2D, this picture will give you an idea how the three-dimensional element and the two-dimensional element are sort of connected ok. So, this is a three-dimensional element of a hexahedron; this is hexa and 3D we call it hexahedron and when you come down to 2D. So, these are all my 2D elements these are all my so this upper block actually shows all 3D elements, these are some equivalent 2D elements. So, this is quadrilateral. So, that means, when I draw a 2D element, it would be a quadrilateral element.

Now, similarly this guy is a tetrahedron. So, this is a tetra in 3D. So, the tetra when you transform to 2D it actually; so, if you look at carefully the interesting part here is that when you look at that hexa it essentially one particular face of hexa becomes a 2D element. Similarly, one particular face of a tetra becomes a 2D element, it is a triangular element. This is polyhedral. So, this becomes a pentagon or it could be hexa also, hexagon. So, again this is a one particular face of a 3D elements becomes the 2D element, so that how the connectivity between 3D and 2D element are done.

(Refer Slide Time: 16:58)



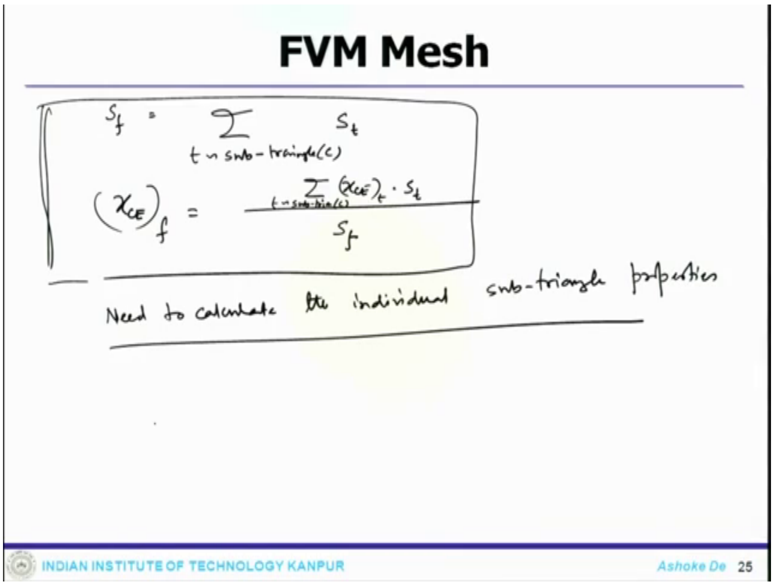
So, now the point comes when you talk about the elements there are certain important information they needs to be handled. One of the important thing is that your surface area ok, sometime volume for 3D, volume of the element surface area, then centroid, all these require to be calculated ok. For any arbitrary element one can do that.

So, typically for any polygon, the geometric center is so the geometric center geometric center of a polygon is let us say x g which will be nothing but x g, y g, z g if it is in 3D. Now, we will that will only coincides with the centroid; otherwise not that means, centroid of the polygon let us say x CE which is again x CE, y CE, z CE. So, it will coincide with the polygon only for some very special cases ok, so that could be the I mean one immediate example could be the triangular. For the triangle, if you look at a triangle the centroid and the geometric centroid will coincide, so that is the beauty of a triangle ok.

So, now, how do you calculate the geometric center? So, the geometric center you can calculate by 1 by k summation of i 1 to k by x i. So, this is where the k is the number of points forming the polygon. For example, if you look at this pentagon 1, 2, 3, 4, 5, so here k would be 4. So, the geometric center, again if you look at this you can have multiple triangles sitting there. So, this is the one which we have connected with line; and you have multiple sub triangle and the sub triangle will have a centroid. So, there is a polygon centroid, and the centroid of the polygon, and these are the sub center of these things.

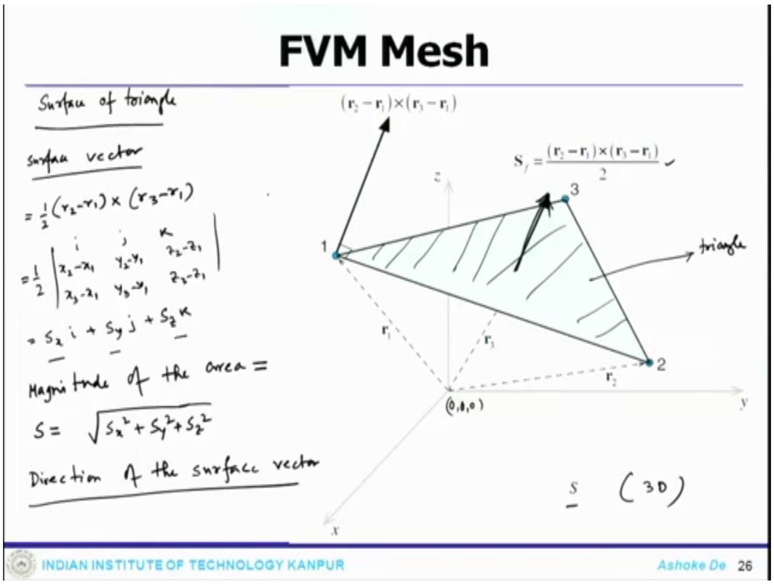
Now, for these particular smaller triangle or the sub triangle, you can calculate both the geometric center and the centroid ok. And then one should do that while rather doing that for computing the centroid centroid, for computing the centroid of the polygon, you some sort of a area weighted calculation area weighted centroid calculation ok, calculation for each sub triangle and summed over the polygon. So, that will get you back the centroid of the polygon ok.

(Refer Slide Time: 22:02)



So, if you read down the mathematics, how it would be looking like? Your S f is summation of this is sub triangle C. S t and x CE for this would be summation of x CE t dot S t divided by S f, and this goes sub triangle C ok. So, this is mathematically how you can estimate the centroid of the polygon.

(Refer Slide Time: 23:00)



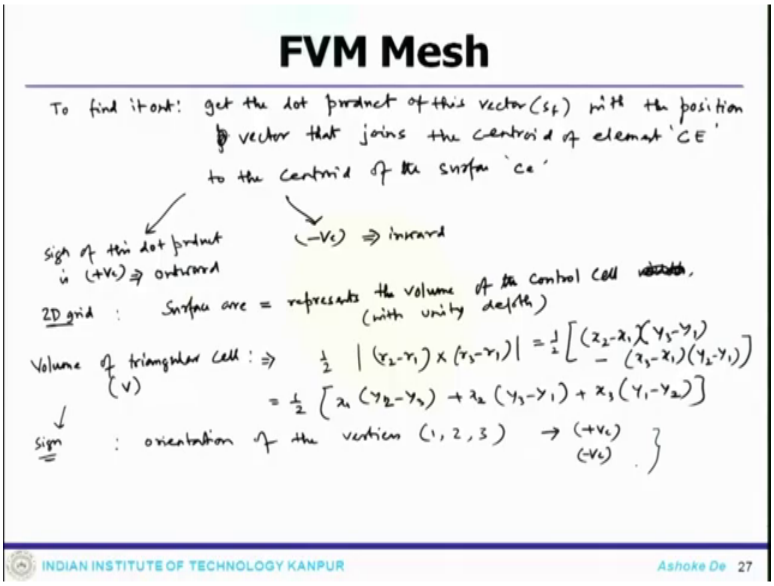
So, now you have a now if you have a triangle, you can again calculate the surface. The reason being when you look at the centroid of the polygon, so you actually need to calculate the individual sub triangle properties and then you go over all the or summed over all the triangle, and then get back the geometric center. So, if you come back to this of a individual triangle, then this is how you define that. This in the coordinate system x, y, z coordinate system this is your triangle; it has three vertices, 1, 2, 3. And the and the from the center of the referring reference plane the radius vectors are r 1, r 2, r 3, which are the standard notation.

So, once you say that then actually you have some you can calculate the normal vectors at 1, this should be the normal vector, and this is the surface vector. And how you can calculate the surface vector, surface vector you can calculate like half of r 2 minus r 1 cross r 3 minus r 1 ok, so that is a cross product. So, this is how you calculate the surface and the surface vector would be pointing like this. So, essentially this is my triangular though it is shown in this x y z frame work, but it is essentially this is a two-dimensional system or two-dimensional element. This is the triangular face top of that the thing would be pointing outward normal and this is how you calculate the surface vector.

And if U write the down in matrix form this is nothing but x 2 minus x 1, this is x 3 minus x 1, this will be y 2 minus y 1, y 3 minus y 1, z 2 minus z 1, z 3 minus z 1. So, in a more generic sense this is nothing but S x i S y j plus S z k. So, there are three components in each direction. So, essentially we are talking about the surface vector. So, surface vector being an vector quantity it has three different component. If one of them are 0, it will be. So, this essentially the can be and the what would be the magnitude of the area. So, the magnitude of the area equal to S, this is the surface area which is nothing but square root of S x square plus S y square plus S z square.

Now, the point important point here is that to note the direction of the surface vector. So far while doing these calculation, we have no where taking into account whether the surface vector is going to be outward or inward.

(Refer Slide Time: 27:51)



Now, to find out that one has to so find out that to find it out first to get the get the dot product dot product of these vector that means, the S f surface vector with the position vector. So, that can give me. So, the position vector which or that joins the centroid of element CE to the centroid of the surface C small e ok.

So, once you do that, if the sign of this dot product is positive that means, this will be outward directed or outward pointed; if the sign of this is negative that means it will be inwardly pointed. So, the one end you get the calculation of the surface vector; and once you get the calculation of the surface vector you find out the direction of the surface vector. So, these are the two important things one has to have. And if you just recall from your unstructured grid system you need to have both the calculation of vector and their dot product.

Now, for a 2D system, if you consider a 2D grid ok, the area or the surface area, this is or something which surface area which will actually represents the volume of the control cell with volume of the control cell with uniform depth or unity depth with unity depth that is how a 2D triangle can be looked at. It represents the volume of that whether unit depth out of plane essentially in the third direction.

So, you can find out the volume of a triangular cell that you can calculate as this is V, V is half of r 2 minus r 1 r 3 minus r 1 the magnitude which is essentially x 2 minus x 1 into y 3 minus y 1 minus x 3 minus x 1 into y 2 minus y 1. So, if I write it is x 1 y 2 minus y 3 plus x 2 y 3 minus y 1 plus x 3 y 1 minus y 2 ok.

Now, again the sign concern will be issue and that will depend on the orientation of the vertices that is 1, 2 and 3. If they are oriented in a counter clock wise direction this will written a positive sign; if they are oriented in a and a clockwise direction this would be a negative sign. So, the sign convention will also be an issue. So, we will stop here today. And we will take from here in the follow up lectures.