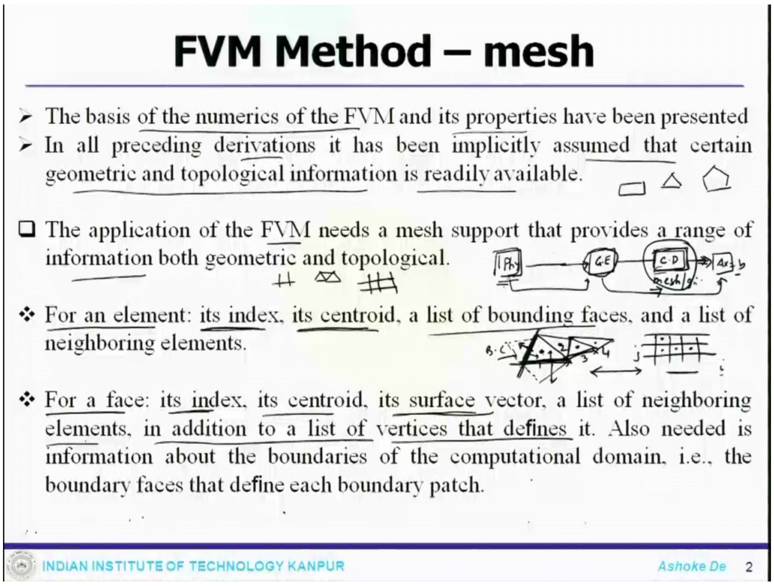
Introduction to Finite Volume Methods-I
Prof. Ashoke De
Department of Aerospace Engineering
Indian Institute of Technology, Kanpur

Lecture – 17
Introduction to Finite Volume Mesh

So, for we have discussed all the basic details, starting from the discretization grid and all these things and how to transform a physical problem through the numerical problem to the algebraic system.

(Refer Slide Time: 00:32)



Now, we will move to the next step; where essentially what we talk is that the as I said the basics of the numerics of the finite volume method is already discussed. And previous lecture we have also discuss the properties like consistency, stability and transportivity all these other properties are also discussed. So, that actually sets the platform for the process of getting a physical problem to the numerical system, where you end up getting a linear system and can be solved through the linear solver.

And, all these derivations whatever we have so far discussed, one of the important assumption which was there; when we have done some derivation we have assumed certain geometric and topological information, which was readily available. Which we mean to say; that means, whenever we have talked about certain derivation of the equations, we have assumed some mesh or underlined element like sometime the rectangular element, sometime that a triangular element, sometimes this pentagonal system, sometime the hexagonal system.

So, these are the different different kind of elements those are assumed to be there inherently and using those information only we have derived those equations or rather discretize those equation, to get the discretize form so that is one of the information. Now, what essentially it means whenever you do a calculation, you have a physical problem in your hand. So, that is your physical problem. Now, physical problem which will be along with your geometrical information or physical problem with your geometric or modeling is that the governing equation that will define the physical problem; then the physical domain was converted to the computational domain or you call it a mesh or grid.

And then you discretize the system get a linear system to get a solution. So, what so, far what we have done we have talked about the physical problem, how to transform the physical problem using the numerical method to a set of linear system. So, we have done all these steps apart from or rather assuming certain underline mesh or the topological information, which was used to get this linear system. Because that is very much important how we calculate the flux, how we calculate the derivative, how the physical domain has been transform to the computational domain. So, all these informations were available.

So, essentially what is require is that to get a simulation done for a physical system. So, we need to transform the physical domain to the computational domain. What essentially it means? You have a physical problem in hand, the physical problem is kind of dictated or governed by the set of governing equations, those governing equations is going to represent the physical problem, now the physical domain must be converted to a computational domain or which we call that which brings all the topological or geometrical information into this computational domain; that means, it include both boundary conditions and all other information.

Then on top of that computational domain the governing equations are discretized using a particular numerical scheme for the current lecture or the purpose, we have been talking about the finite volume. So, using that discretization scheme we finally, leads to a algebra system or the linear solver and the linear solver get you the solution. So, what;

that means, there is a key component what is the mesh and the mesh supports; so, any numerical scheme since we are talking about finite volume.

So, we can say finite volume or but any other numerical scheme whatever one can think about like finite difference or finite elements, it does not matter irrespective of the numerical scheme, there is a underneath or underline support which is required from the mesh. That means, where your physical problem is transform to the computational domain, and it carries all boundary conditions, it carried all the physical information. So, everything should be embedded inside that computational domain.

Now, when you talk about that that essentially talk about the mesh; so, any numerical scheme just like finite volume, it needs a support from the mesh what is called the mesh support. And the mesh once you talk about the mesh, it could be structured, it could be unstructured so, it could be multiple ways one can generate the mesh. But for a particular element let us say if you talk about this particular element rectangular element, then the information which are very much essential or need.

That for an element so, when you talk about the finite volume process, you have this cells or the elements. Now, for these elements these are the cell centre and you have some information there. For a particular element what is very much important is that its index. So, the information about is index. So, when you talk about the global system, as we see later instance when you talk about the global systems so, that index of this is very important to keep that.

So, the cell number could be 1, 2, 3 these are the index of that particular element. And if it is structured element or continuous elements, regular element then its much easier to keep a track of that for example, I can have I j locations or in three dimension and i j k location. So, one of the important or key information that is required to when somebody generates a mesh or one generates a mesh and feed that mesh to your numerical code for getting a solution done, is its index. Now, index which will keep track of the elements; second point is the key point is that its centroid. So, the centroid means the cell centers.

So, if it is rectangular these are going to be the cell center if it is triangular these are going to be the cell centers of the centroid. So, the information of its centroid is another important property that also has to be tacked upon. Then certain faces like some faces may be associated with the bounding or boundary faces.

So, a list of bounding faces are very much essential because this bounding faces are going to import or invoke the boundary condition to the system. So, the face or the element or the associating surface which are associated via physical boundary then that also has to be tacked upon. Then the last, but not the least is that the surroundings elements; that means, one particular cell or element is surrounded by multiple elements. So, the indexing or the lists of those elements are also very much essential to keep track off.

Now, these things are going to make a huge impact in any computation, because these are what is going to actually be handling your data structure. Keeping track of your element numbers their indexing, their connectivity and everything. Now if you compare these two side by side one case it is a unstructured triangular element other case its a structured rectangle element, the things are much simpler and easier rather in this particular case. Because here if you keep track of i j direction then you can easily track of the element you can track of the coordinate and the other things.

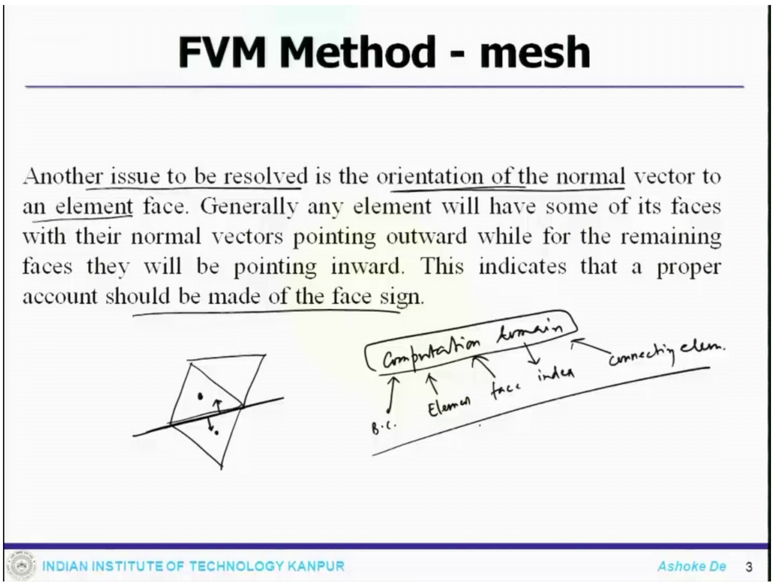
Now, that essentially what information one should have from the element? Now when you come down to a face if you talk about a face, let us say we talk about a face this is the face then again that is index. So, any particular element let us say for this particular example you have a triangular element. So, it must have three faces. So, keeping track of those index are also very important. So, one has to keep track of these faces associated with this neighbors, because this face if you think about it is kind of a common face for the neighboring element. So, that information also needs to be I mean kept into the data base. So, that while calculating the normal or the face fluxes that can be used, then against the its center or the centroid.

So, the face also have some centroid and getting that centroid value is also very very important. Then surface vector so; that means, the direction of this normal vector at this face. So, any face that you have the surface vector is going to dictate your direction of the flux. So, it could be inward it could be outward so, that surface vector information is also very important, when it comes down to the face. Then the face which is connected with the neighboring elements that means, the neighboring elements sharing a common face. So, one has to keep a track of that. So, that information also is required while calculating the flux. Because once this face is common one case it will be going outbound, other case it could be going inbound to a particular cell.

So, one common face can have a two different surface vectors, which can be associated with two different cell elements. Now in addition to all of these the vertices that defines it. So that means, top of all these information one has to keep track of this vertices and why they are important? Because the vertices are going to get you back the coordinate of the element, and once you get the coordinate of the element that can be used for calculation of the area, and for three dimensional system or three dimensional element it could be used to calculate the volume, and these vertices are also going to be used sometimes for calculating the location.

So, all these information which are required they go into the computational domain; so, the boundary faces that defines each boundary patch and all these things. So that means, one element must have certain information which is to be tacked upon the faces must have some information, which must be tacked upon. So, all this information can be tacked and then one can be used that in a data structure.

(Refer Slide Time: 13:23)

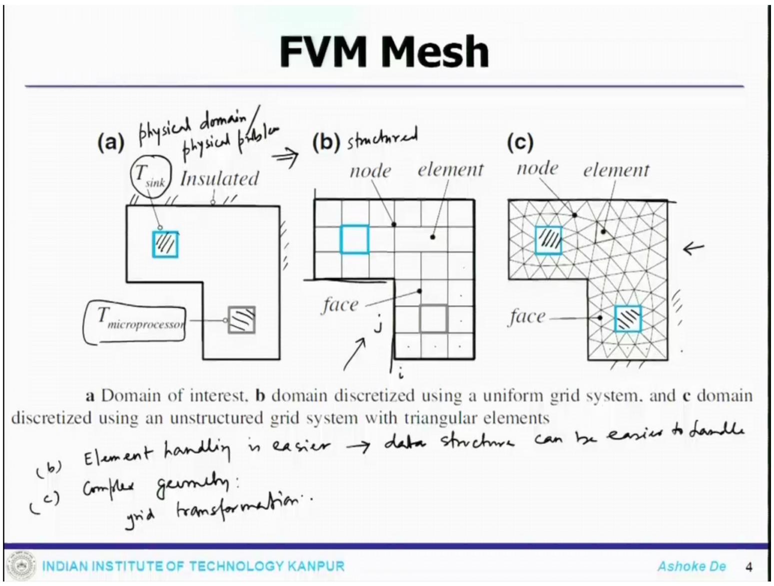


So, one more issue which needs to be resolved is very important is the orientation of the normal vector. As I said if there are some elements, which are connected with each other then there are common faces this is a common face. Now, this particular element the surface vector could be this side, but when I consider this face for this element then the vector could be the side. So, that is why the orientation of the normal vector to an element face. So, that information is very very important to keep in the database.

So, that while calculating the flux, the flux calculation can be done accurately; because as I said some element will have some faces towards outer some pointing towards inward, while the complete some would give the direction. So, one has to account for that. So, essentially that is done through the sign of that vector. So, all this essentially the important properties of this mesh which has to go in your computational domain or rather the computational domain must have this information.

Boundary condition, element, face, indexes, connecting elements all this and so and so on all this should be embedded in a computational domain, and that has the data structure which should keep track of all this information. Now once you essentially go to the mesh finite volume mesh how it looks like.

(Refer Slide Time: 15:31)



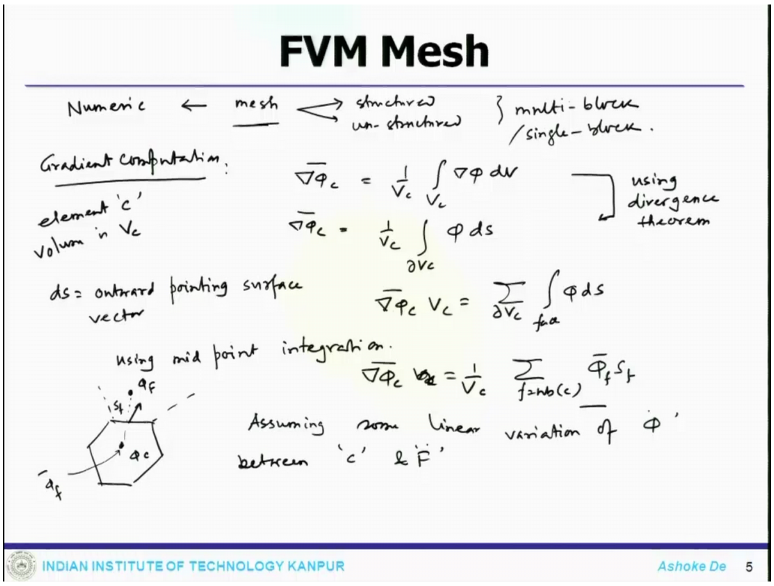
So, if you go back to this particular problem that we started off and every now and then we bring back this particular example to see; so, this is our physical problem of interest. So, all these faces are essentially insulated. So; that means, there is no heat flux going out and this is the patch or the zone where temperature is boundary condition has been given, which is acted as a sink and this is the microprocessor which is actually the heat source term or heat generator that uses that and rest of the domains are how the temperature is going to spread or distribute over the domain. So, this is our essentially the physical domain or rather the physical problem of interest ok. And once you have that then the next step is to convert this physical thing to the computational domain.

Now, there are different ways one can do, simplest way since it is a rectangular surface. So, one can generate this kind of as its shown here one can generate this kind of rectangular elements, which is a structured element. And the beauty of this is that all these faces are properly accounted for and one can keep track of these elements very nicely through their i j locations and this faces are all going to be the boundary faces. Or other way is that I mean one can generate some sort of a unstructured mesh. So, unstructured means they are not regular so, there could be some triangular element. But it is essentially going to spread over the domain and there will be faces where the elements are also going to sit there.

And this particular patch is going to be the boundary condition, sink boundary condition this patch is going to be the source boundary conditions and other sides are all essentially insulated boundary condition. So, one can generate mesh like that. Now if you generate this kind of a structured mesh like shown in here this structured mesh, there are certain advantages. Advantages is that handling of the element like for case b, the element handling is easier or rather the data structure can be easier to handle ok.

But the other case because since here things are in ordered fashion, but in the other case that is not the case. So, here things are not in ordered. So, you have to keep track of all the indices, you have to keep track of the elements, you have to keep track of the vertices and all the connecting elements. But sometimes for a complicated geometry or complex geometry, the structured greeding is not some preferred people do prefer some kind of a unstructured greeding, but or some sort of a grid transformation. So, all these are different ways one can actually handle a complex geometry in a different fashion.

(Refer Slide Time: 20:05)

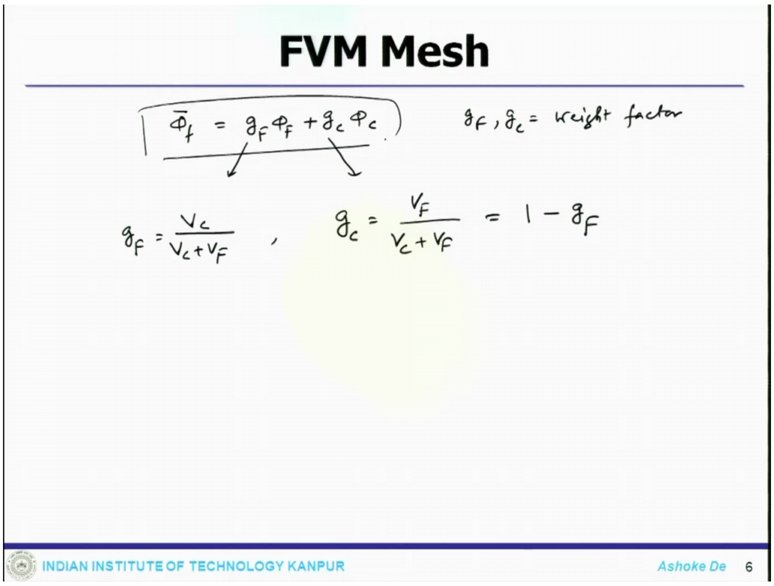


So, now so, once you do that now we can so, for your numeric you have a underlined system which is called the mesh. Now mesh could be of any type it could be structured, it could be unstructured, it could be multi block, it could be single block, it all depends on the complexity of the geometry what kind of mesh you require, but essentially the underlined system is that mesh. So, once you get the mesh your numerics is going to act upon and let us say for the for a given a mesh, you can always calculate your gradient and gradient computation you can look at it what you do.

For a given system is in the green gauss theorem for a particular element let us say you calculate the gradient, here you talk about element c. So, the volume of the element is V c. So, you calculate the gradient like that. So, it becomes the is in the divergence theorem. So, from here to here using divergence theorem that is purely vector calculus and you get 1 by V c del V c phi d s. So, the volume integral is actually transformed to the surface integral, where ds is outward pointing surface vector ok. For a given set of system one can write this into V c equals to summation of over face phi d s. So, one can write that.

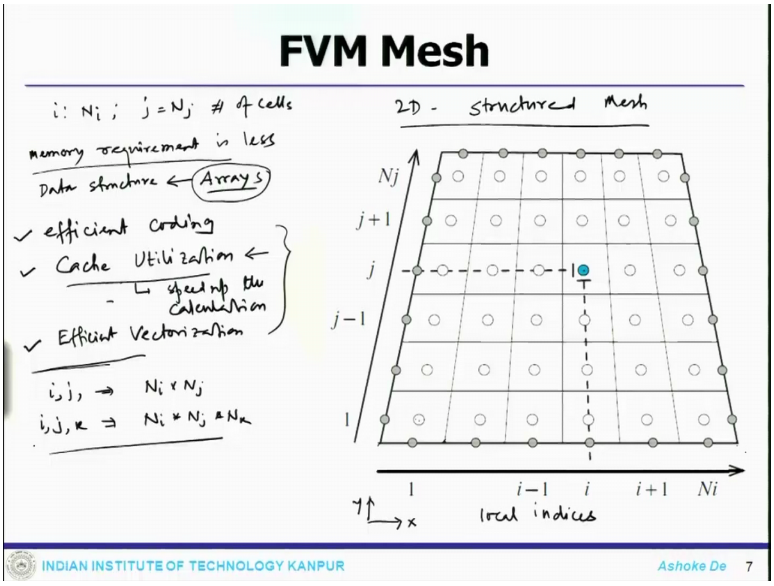
Now, using some midpoint integration, one can write that this guy del phi c equals to 1 by V c summation of n by c phi f s f. So, or alternatively what you can do. So, essentially here the direction of a s f is required, then only one can compute this one precisely. So, once you have this element faces, you can actually have a surface like this and you have an element phi c, here is that element F and this is the connecting one this is the vector. So, here it is phi f then s f. So, one can actually linearize the system or assuming some linear variation of phi. So, between two elements between element C and F these are neighboring elements one can write that phi f equals to g F phi F plus g c phi c ok.

(Refer Slide Time: 25:02)



Now, that requires some calculation of this weight factors, these are g F or g c are the weight factor. So, one simple way to calculate that is that g F would be V c divided by. So, that sort of it and g c is V F divided by V c plus V F excuse me which is nothing, but 1 minus g F. So, that kind of simple linearization loop can be applied to calculate the phi f and one can do that. Now, if you look at a particular system of grid let us say you look at this grid.

(Refer Slide Time: 26:13)



So, this is your straight way structured mesh 2D structured mesh and what you can see the domain is divided in both the direction, this is i direction this is j direction. So, it goes essentially x and y direction and the surface is bounded. So, you can have 1 2 3 different elements they are nicely ordered; and in the j direction it goes from 1 2 3 like that. For a particular j and i you can always find out that particular cell element and the number of cells actually in i direction its N i and j direction its N j.

So, that is a number of cell ok. So, once you have this kind of structured system this has certain advantages let us say one of the biggest advantage is that the memory requirement memory requirement is less. So, the handling of data structure is much easier because only thing you need to keep track of these elements in a contiguous memory. Because one thing you need to note here whenever you define these elements or divide the computational domain in this N i N j, you need to define your arrays or data structure which will be essentially lot of arrays or arrays of variables.

So, once you define your dimension the size of the arrays are also going to be that size, which will store these variable from the cell center and it also needs to keep track of the coordinates of the cells and the connectivity amongs the neighboring cells. So, having that requirement is less, it is also very efficient or rather coding point view one can think about its a efficient coding can be achieved with this kind of structured grid. So, that is very one of the good or biggest advantage of this kind of a structured mesh.

Secondly since you can have the cache utilization which can be proper so; that means, these are all related to the hardware ok. So, the hardware whatever you are using or the computation is done on those hardwares, their the cache utilization would be very efficient. If that is efficient then obviously, your calculation is going to speed up. So, this is going to speed up the calculation or simulation time. So, that is a biggest advantage.

Now, thirdly one can think about the also efficient vectorisation. So, these are all some properties which are associated with your coding; that means, how good your programming is going to be and how efficient your coding, how efficient your algorithm or code that uses the memory, that uses the cache, that uses the vectorisation so that your code can run faster and faster. This is one of the strong requirement of any numerical code that how efficient your code is ok.

And so, that is where this structure mesh becomes quite handy and if you look at this numbering 1 to i to N i and N j these are essentially you call it a local indices. So, these are local indices which actually tacked with that thing. Now once you go from i j direction the total number of cell would be N i into N j if it is i j k then it would be N i into N j into N k. So, that is how you keep track of the arrays.

So, we will stop here today and we will take from here in the follow up lectures.