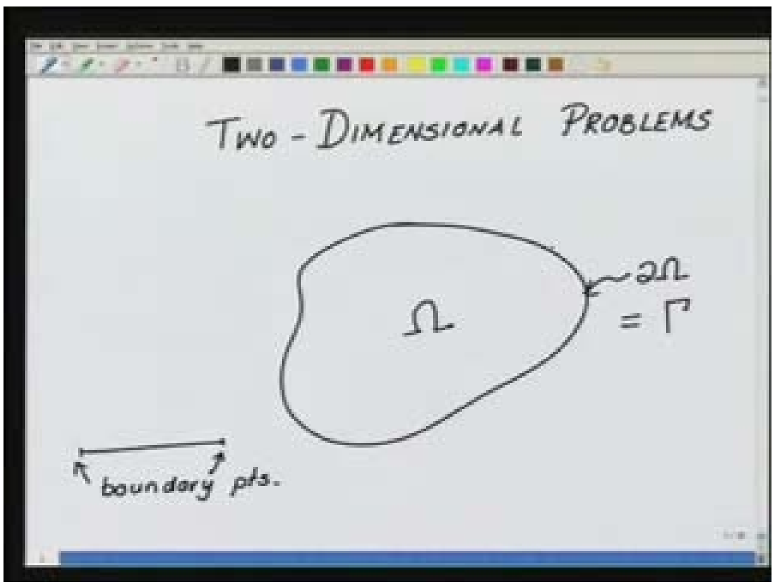
Finite Element Method

Prof C. S. Upadhyay
Department of Mechanical Engineering
Indian Institute of Technology, Kanpur

Module - 7 Lecture - 1

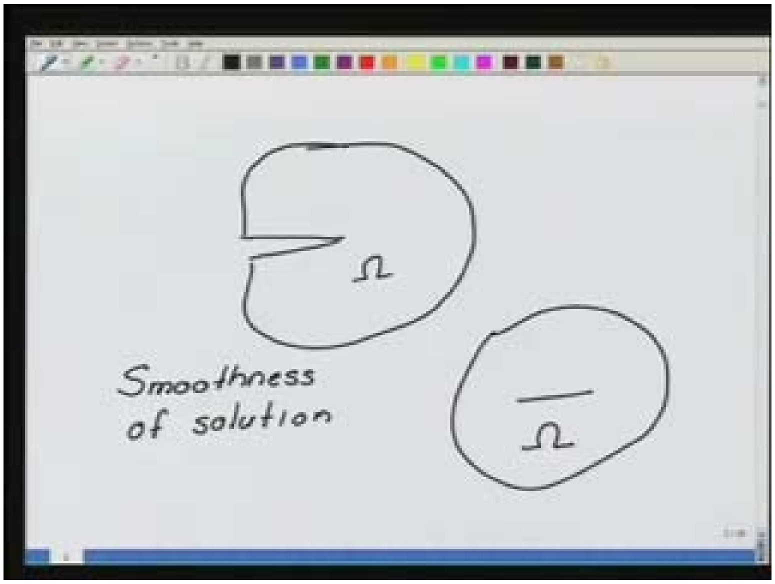
Let us now talk about two-dimensional problems. We have, in the two-dimensional case, a domain which is now given by some area, in the two-dimensional space which we will give the name omega. The domain will have a boundary delta omega which we will call by also the name gamma.

(Refer Slide Time: 00:18)



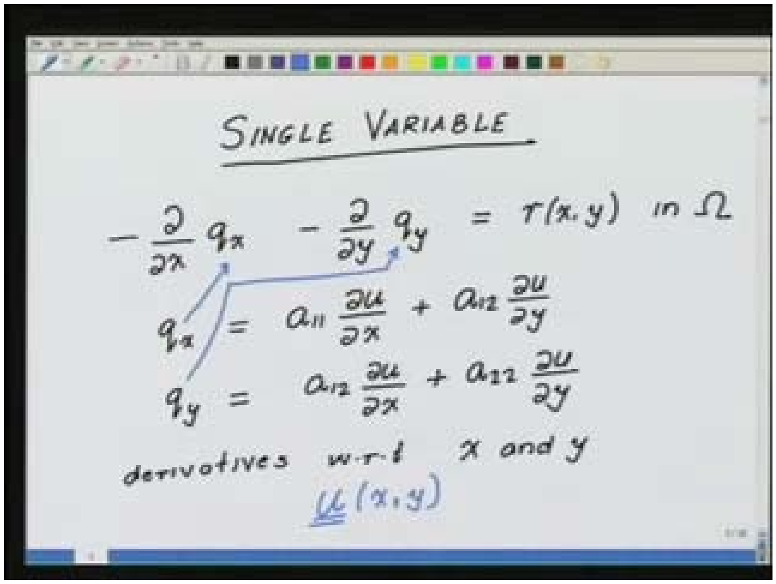
So here, what you see that as compared to the one-dimensional problem which we had earlier where we had the two end boundary points. Here, we have a bounding line or a curve for the domain and as such, the boundary conditions for the problem whichever we should pose on the two-dimensional domain are going to be specified on this bounding curve. So let us now see how this two -dimensional problem is going to be different from the one-dimensional problem. The first thing that we see is that now this bounding curve could have any of many different geometry.

(Refer Slide Time: 01:49)



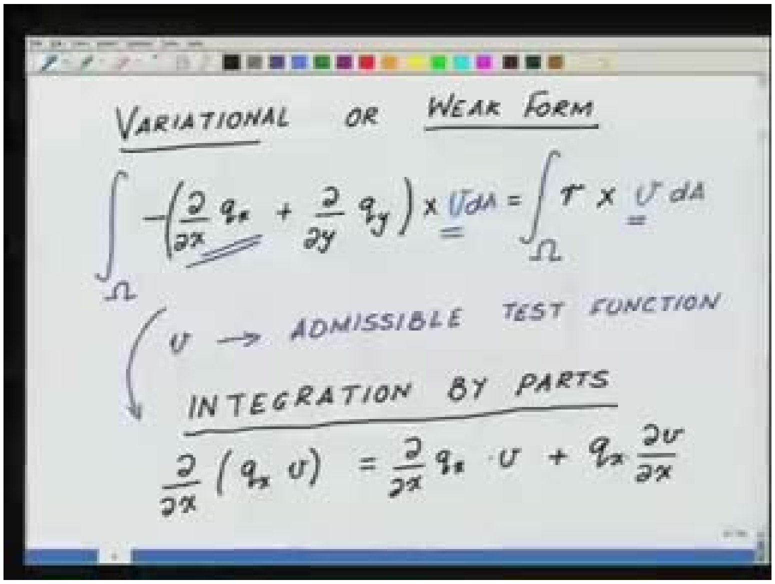
For example, I have a domain where I have a reentrant corner. I have a domain, where I have an internal boundary also that is an inside crack; these are all possible two-dimensional domain. So because of this we expect that the boundary affects the smoothness of the solution to the problem that we are going to pose. This is going to happen for any two-dimensional problem, the domain boundary, the smoothness of the boundary is going to effect the solution; this is not the case in the one-dimensional case. Let us say the simplest possible two-dimensional problems, which is the single variable problem.

(Refer Slide Time: 02:53)



What we do here is let us take the problem given this (Refer Slide Time: 03:23) in omega. If we take this differential equation where q_x is given as a_{11} del u del x plus a_{12} del u del y and q_y is equal to a_{12} del u del x plus a_{22} del u del y. You see that, what we have done now is you have shifted from use of only differential equation in the one-dimensional case, the problems that we had considered in the one-dimensional case, to now partial differential equation, because derivatives with respect to both x and y are involved. So this is another departure from the one-dimensional problem. If I give you this differential equation, this is as you can see a second order differential equation, because, when I plug in the representation of q_x and q_y here in my equation when you see the equation comes as the second order differential equation, in terms of the unknown variable u x y which u is the unknown variable which is now a function of x and y. If we want to use, the approach that we had developed for the one-dimensional problem what is the first thing that we have to do as far as getting to obtain a finite element solution for this problem? So the first thing is getting a variational or weak formulation.

(Refer Slide Time: 05:34)

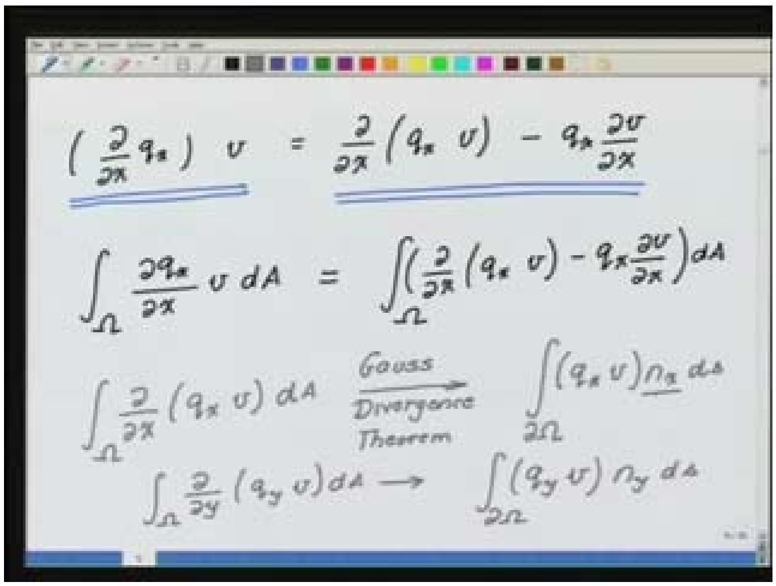


For this, what we had done we will stress on the weak formulation more here. So we take this differential equation which is available to us and multiply the differential equation with a suitable admissible test function. So what do we do?

We have minus del del x of q_x, I will put it like this, plus del del y of q_y is equal to r. So I will multiply both sides with this, at this function v and I am going to integrate this quantity over the domain omega. This is the first step that we had done as for as obtaining the weak formulation for one-dimensional problem. Same thing we are going to do here, where we start from differential equation, multiply it with an admissible test function v and integrate over the area. Remember that v we say is an admissible test function or we can say it is a virtual generalized displacement function or whatever we want to call it. Next step, if I go and see this representation here, because, of this derivative sitting, second derivative of u is sitting in the expression, second derivative with respect to x second derivative with respect to y.

So what we have going to do? We are going to take this expression and do integration by parts. Why are we going to do the integration by part? Because we want to weaken the smoothness requirement on the unknown function u; that is, we want to transfer derivatives from the unknown function u to the test function t. This part we are going to do; so to do the integration by parts what do we do? We know that del del x of q_x v is equal to del del x of q_x into v plus q_x del v del x. Similarly, for the y derivative; that is, del del y of q_y v del equal to del del y of q_y into v plus q_y into del v del y.

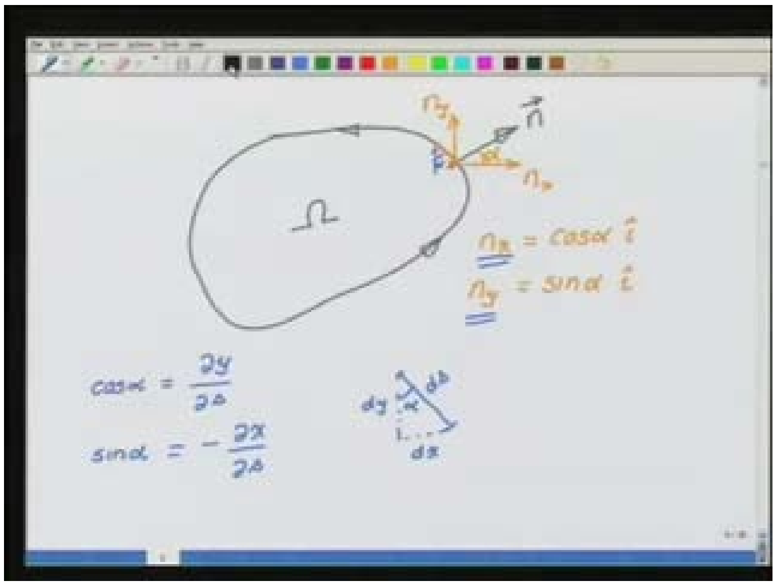
(Refer Slide Time: 09:39)



Then we obtain that del del x of q_x into v is equal to del del x of q_x into v minus q_x del v del x. This is standard calculus that you have done. Now, what I can do is I can replace this expression in my integral with this expression; that is, left hand side is the expression sitting in the integral, I am going to replace this with the expression on the right hand side. I am going to get integral over omega del q_x del x into v d omega or dA, let me write it as dA is equal to integral over omega del del x of q_x into v minus q_x del v del x whole thing into dA. This expression del del x into of q_x into v dA, we can write it using something that you have done in calculus Gauss Divergence theorem.

As this integral is the same, as integral on the boundary also domain of q_x v into n_x $d_{s,}$ this is on the bounding curve. We will see what is this n_x and y? Similarly if I went ahead and did the same job for the other part in the integral, you will get this expression; this is equal to integral over the boundary into n_y ds, where n_x and n_y are component of the unit outward normal on the boundary of the domain. So let us see, what do you mean by this thing.

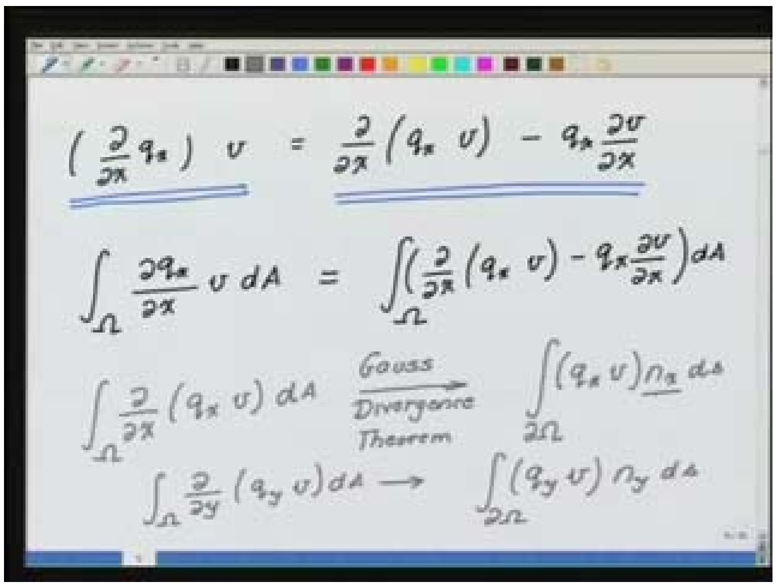
(Refer Slide Time: 13:03)



I come back here, draw my figure; this is my omega what we will say, we will say the boundary to be in a counter clockwise direction. So bounding surface is counter clockwise direction such that the bounded area always lies to the left of the bounding surface. Then we will be talking of the unit outward normal at any point on the boundary. This normal is given as the vector n. This will have component n_x in the x direction; n_y in the y direction and if this makes an angle alpha with the horizontal, then we know n_x is equal to cos alpha i, where i is the unit vector in the x direction.

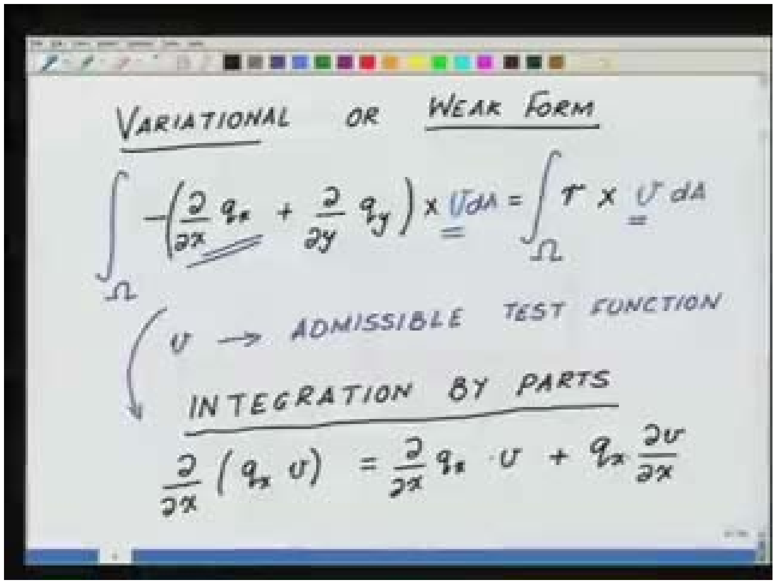
n_y is equal to sin alpha i. It will be useful to write these guys in terms of the local tangent, a small piece taken in the tangential direction about the point of interest. So if I take that, with this is that C, this C in the conjunction direction is the size ds. If you take that here is my d_y and here is my d_x. Note that d_y is positive, d_x is negative. You will see that, this angle is nothing but the angle alpha. So what will happen? Cos alpha is equal to del y by del s, I have written it in terms of the partial and sin alpha is equal to del x del s. This becomes useful from a computational point of view to write this cosine and the sin in terms of the small changes in the length along the curve. We have defined the unit outward normal at any point on the bounding surface any in terms of that we get the two components n_x and n_y.

(Refer Slide Time: 15:57)



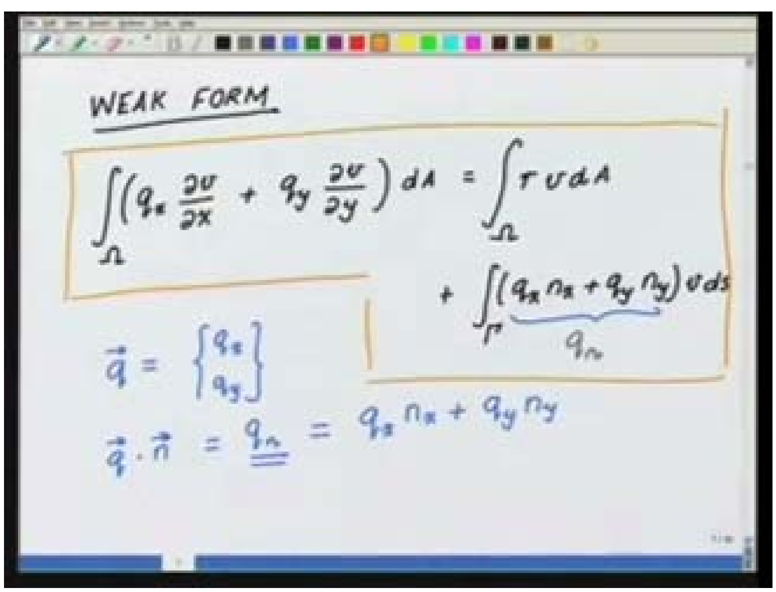
Now in the previous slide if you see that we know what in by n_x, we know what you mean by n_y.

(Refer Slide Time: 16:07)



So let us go back and go up and see that in our initial weighted residual form, that is the first part is called a weighted residual form, why? Because we are taken the differential equation, simply multiplied with the weak function integrated.

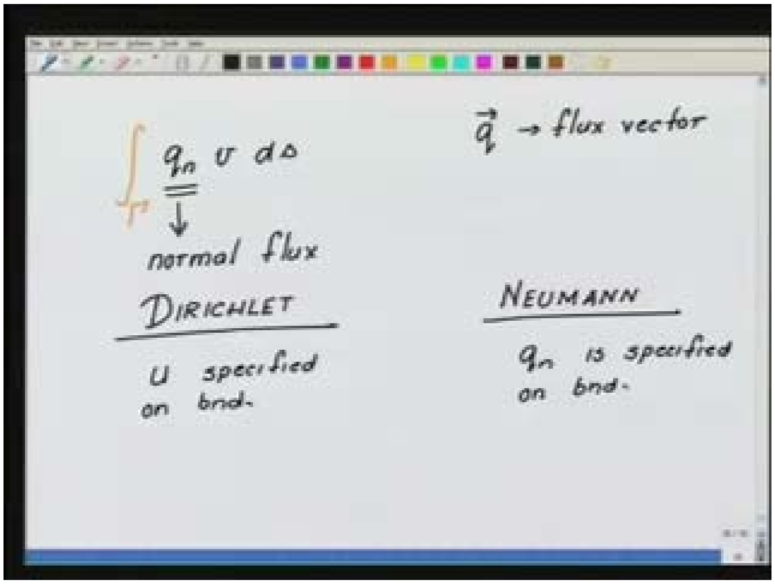
(Refer Slide Time: 16:38)



If I do integration by parts, put in all the things that we have obtained, then I will get the final weak form as: integral over omega of q_x del v del x plus q_y del v del y dA is equal to integral over omega r v dA plus integral over the boundary gamma or delta omega you will have q_x n_x plus q y n_y whole thing into v ds. Now, if you look at this quantity q_x q_i form the component of the vector q. The vector q dotted with the normal vector n will be equal to the normal component of q. That is the component of q in the direction of the normal which we call by q_n; this will be equal to q_x n_x plus q_y n_y. So we can replace what we have written there by q_n. Now you see that this is our final weak form for the second order differential equation in terms of one variable that we have taken.

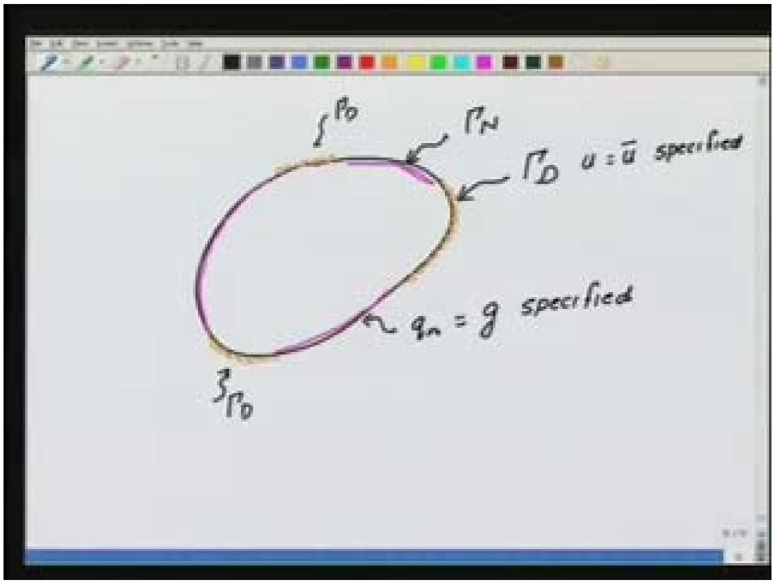
Next, what had we done earlier? Next is we would like to identify in the boundary condition that can be applied on the boundary of the domain gamma; the boundary gamma of the domain omega.

(Refer Slide Time: 19:08)



From the boundary condition, you see that on the right hand side I have the integral over gamma of q_n v ds. This is called the normal in a generic way normal flux where q is called the flux vector. So as we talked about earlier, here either you can give what is the value of q_n on the boundary gamma, apart from the boundary gamma or we specify what is the unknown variable u on the boundary. We have the boundary conditions which we will quite different name as we written earlier Dirichlet and here is Neumann. Dirichlet will [ ] have an analogy with the one d k that u is specified on boundary. Neumann means q_n is specified on the boundary. Now, the Neumann is subscribed on the full boundary; just like in the one d case at one point I can have one boundary point, I can have one type of boundary condition, on the other I can have different type of boundary conditions; in this case we can take the domain omega.

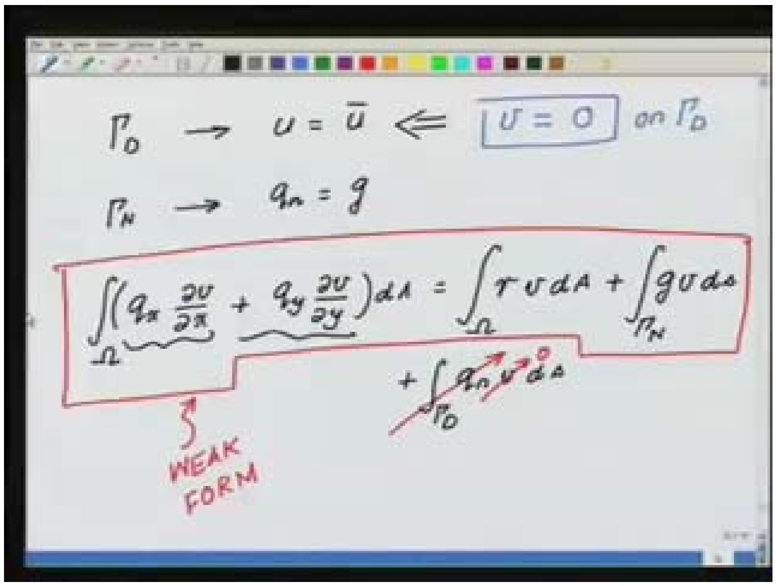
(Refer Slide Time: 21:05)



In this, I can have part of the boundary with Dirichlet boundary conditions; I can have this part, I can have this part, and I can have these parts. The remaining part of the boundary I can have Neumann boundary condition. That is in the remaining part I specify Neumann conditions. The part of the boundary where Dirichlet conditions are specified I am going to call it by gamma$_D$. On the part of the boundary where Neumann conditions are specified I am going to call it by gamma$_N$.

So this is gamma$_D$, this is also part of gamma$_D$, essentially we mean the collection by gamma$_D$ of all the parts where the Dirichlet boundary condition is specified or u is specified. u is equal to u bar specified. On this part I will have q_n is equal to g.

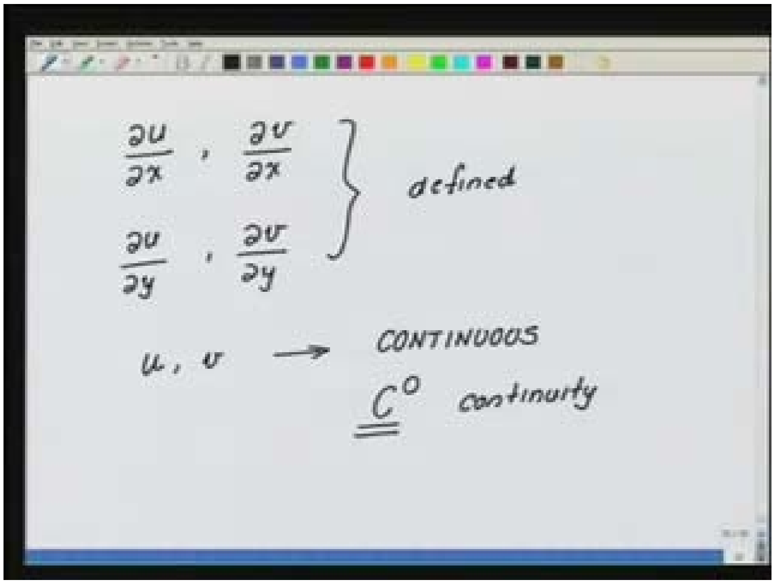
(Refer Slide Time: 22:37)



The next question is that on the part of the boundary gamma$_D$, u is equal to u bar which is u bar is a given function; it could be 0, it could be something else, whatever is specified by the user as in input data. So on this boundary, the v has to be constrained. So v will be constrained to the 0 on this boundary. Just like in the one d k. On the Neumann boundary we have… if I go back to my weak form I will get the following expression: plus q_y del u del y dA is equal to integral over omega r v dA plus. Now we will have the integral over gamma$_N$; on gamma$_N$ q_n is given a g, so this will be gv ds plus integral over gamma$_D$ where I will have q_n v dA, but on gamma$_D$ I know v is equal to 0, so this whole integral drops out. So the weak form essentially is given in terms of this expression. Remember here we have not touched the finite element method at all. All we have done is followed the procedure of weighted residual development and using integration by parts, we have obtained the weak form for the problem. In getting the weak form, we found that we now have the two possible boundary conditions that can be specified at any point on the boundary. Nothing else, either the displacement or the normal flux can be specified; no other quantity can be specified on the boundary. This looks very similar to the development that we had made in the one-dimensional problem.

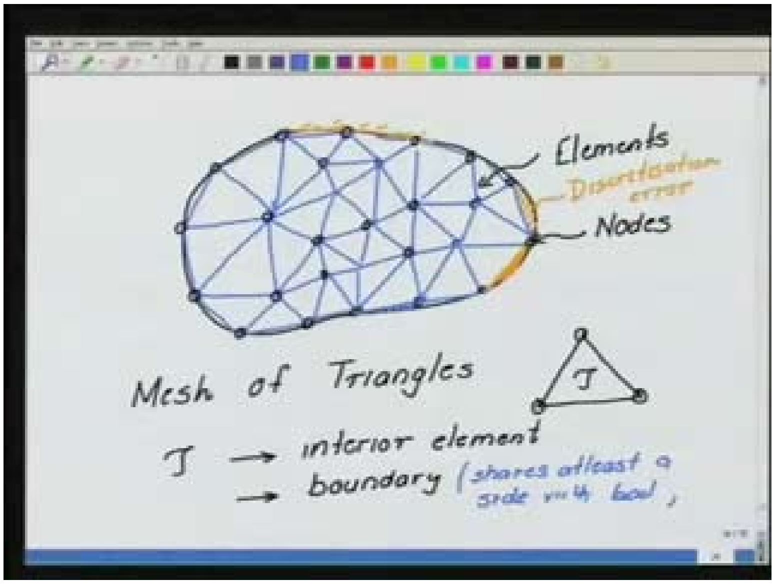
Next is a question that we have the weak form, then what? After we have the weak form comes the question of as far as the construction of an approximation is concerned, where should my approximating function for u lie? Similarly, where should v lie? So what kind of smoothness requirements we have to satisfy. If I go back to the weak form we have, we want this expression, this whole integral, to be finite, all the integrals to be finite. If I want the integral of the left had side to be finite, the integral of the right hand side to be finite, what is the smoothness requirement that I have to have on u and v?

(Refer Slide Time: 26:16)



So the answer is, again the same thing that we had, we want for them to be finite del u del x, del v del x, del u del y and del v del y should all be defined. That is again in the domain my u and v have to be…. the u as well as v are continuous in the domain and we want only the derivative, the first derivative - of each function to be defined. What we need is again what we call a C zero continuity; this is what we need. Then comes the question of how to go about constructing a solution to the problem.

(Refer Slide Time: 27:31)



So what we do is look at our domain. In this domain, how did we go about solving the problem in the one-d case? If made a mesh; with respect to the mesh we defined this so-called local basics functions, basis functions with local support. Here also we have to make a mesh. So what kind of mesh is possible? Generally, what we do is we take a mesh of triangle of quadrilaterals. I am drawing some mesh of triangles here as an illustration; so I will cover my whole domain with this mesh of triangles. I am drawing a very gool mesh, so that I am able to show this connects. What I have done is I made a mesh to cover my domain omega; mesh of triangles.

Here I have mesh of triangles. In the one-d case we had mesh of the sub intervals or the small segment. Here we have made a mesh of triangles. Now how do I define the extremities of a triangle? A triangle here will be defined in terms of three end points of the triangle. So the end point of the triangle. Now, what did call the end points of an element in the one-d case? We call those as nodes. The triangles are our elements and the end points of the elements or collection of the end points of the elements are called the nodes. So these are our nodes.

There are a few differences with one-dimension case, that we should see right away. First thing, we see that observe the following thing that if I take this size. So, there is discrepancy in capturing the actual boundary of the domain, with the elements that we have taken. If you look here, it is really bad. So if I take a mesh is represents a different domain; that is it is not able to exactly sit on the boundary of the domain. So in the two-dimensional problem, which is not the case in the one -dimensional problem, we could exactly mesh the line that we had.

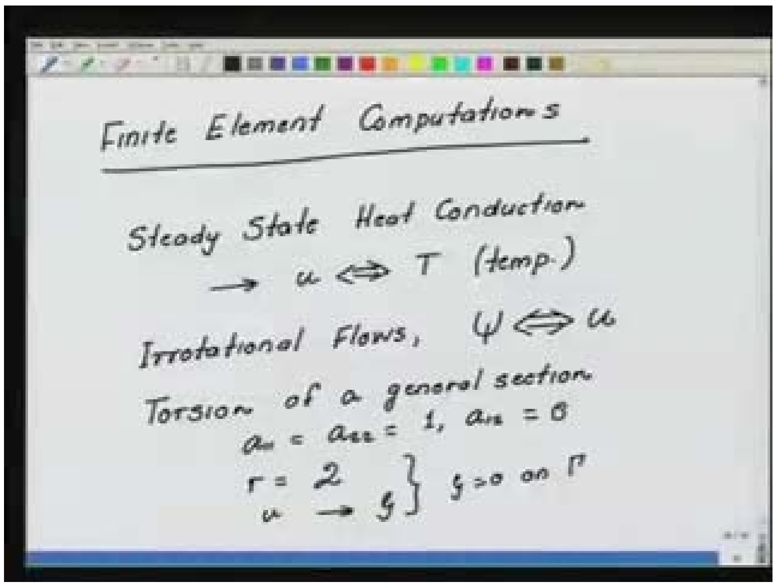
Here we do have the issue of not being able to exactly mesh the control of the domain availability. That gives rise to essentially what we can call as discretization error; that is an error in representing the domain with our collection of elements. The triangles or rectangles or whatever we take - that is one issue.

Second thing we see is that as far as the boundary conditions are concerned, we have to apply them; that is, in part of the domain I may have g given or in part of the domain and I may have u specified. How do I know where to apply what point condition? For that, you see that now the boundary is given on the boundary contours. We have to know for every element whether an element, element is the triangle, that is called the triangle by generic name tau. So we have tau can be of various size; one could be an interior triangle; interior element that is an element lying completely in the interior of the domain.

We could have at tau which is the boundary element. That is an element which shares at least a side with the boundary of the domain. It could share two sides for all we know. It is an element with shares a side with the boundary. So these are some essential differences that we have with respect to what we had done earlier, with respect to the one-d problem. We have to identify when we are making the mesh that these are the elements which are boundary elements and which are the elements which are interior elements.

Now for the boundary element, we also have to identify which are the edges lines on the boundary, because boundary conditions have to be applied along edges, not along the nodes. We will be applying the boundary conditions along the edges of the element that we have and this way we go element by element. These are some issues that we should have in mind when we go ahead and do the finite element computation.

(Refer Slide Time: 34:34)



Now the question is that I have taken a very simple second order differential equation. Now for this simple second order differential equation does it represent anything meaningful from an engineering application point of view? In terms of that, there is a whole class of problem for which this second order differential equation is what we have to solve in order to get a solution.

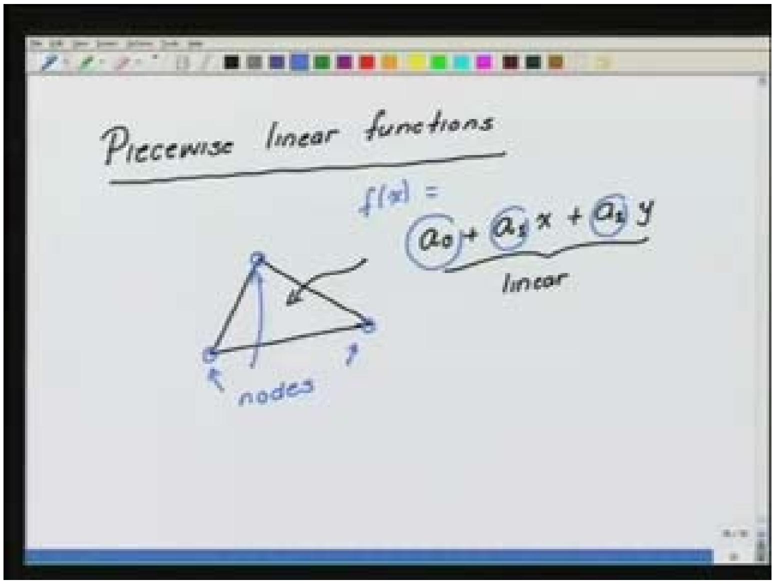
For example, if I look at the problem of steady state heat transfer, then in this case heat conduction. So in this case, you see that u is equivalent to… the u that you have taken here, the generic q to the T; T is the temperature. The differential equation look exactly the same rather a_{11}, a_{12}, a_{22} are nothing but the thermal conductivity. So this is one problem where I can use this differential equation to get a solution.

What we will do is, see when from the programming point of view the program for a differential equation then we interpret where we can interpret. The whole class of engineering problems can be covered by this program, whatever we are going to do. Here the flux q essentially we have to consider the heat flux in the design. Similarly, you have the irrotational flow, the steady state irrotational flow; so irrotational flows, in this case the u is equivalent to the stream function psi. In the steady heat conduction problem earlier, the r represents the heat flow term. Here I can have various r of either it could be 0 or I could have a source anything, whatever.

Similarly, I have the torsion of a solid section; torsion of a general section. So here, in this case a_{11} is equal toa_{22} is equal to 1 and a_{12} is equal to 0 and r becomes equal to 2. If I take this and the function u represents the handle stress function, I could call in theta. So you see that there are many such problem where I can apply the generic second order differential equation that we have written to get the solution. In the torsion problem, the boundary conditions are very specific; theta is equal to 0 on gamma. That is, in this case, the whole boundary is a Dirichlet boundary that we complete here; so we can solve for the Prandtl's state function, we can obtained the j, the term which goes into the modulus of torsional rigidity. So all these things we can do if we are able to solve this second order differential equation that we have taken.

Now the question is how do we construct the approximation? That is the discretization. What we did earlier in the one-d case was we said that the simplest possible function for the second order differential equation which satisfy the C zero continuity were piecewise linear.

(Refer Slide Time: 38:54)

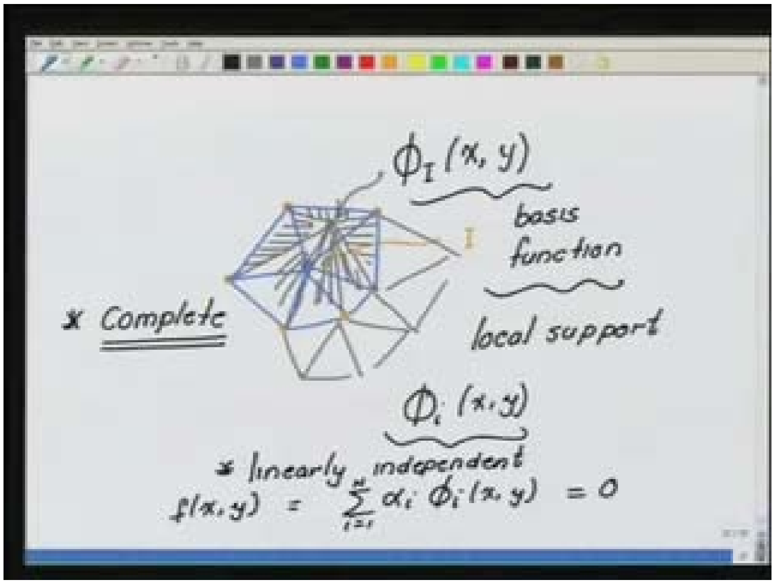


So why not try the piecewise linear function? We will have the piecewise linear functions. What do you mean by piecewise linear function? That is I have a triangle which is my generic element, over this element I defined a function, an approximating function which is given in terms of the linears in x and y. That is a_0 plus a_1x plus a_2y. So the element that is linear.

You see that this linear now in the two-d case requires C constant a_0, a_1, and a_2 where in the one-d case this thing was not there; it needed only the two constant a_0, a_1. This required C constant, you know that this has to be represented in terms of values of the function, if I take any generic function which is a linear, it can be represented in terms of the values of the function at three points and these three points we chose as the nodes of the element. We will essentially construct the basis function or the shape function here in terms of the C node value function.

Now using this how can we construct the basis functions? We are going to do the elaborate construction of the basis function in the next class. Here what we are going to do is at least let us see graphically what these basis function will look like. So let us say that we have a node.

(Refer Slide Time: 41:07)



Like in the one-d the piecewise linear were defined with respect to a node. We have a node here, now the difference between the one-d and two-d is that the node could be connected to many elements; it need not be connected only to two elements. So I could have this kind of a connectivity for this node, where these are all the nodes at the other ends of the element that are connected to this nodes. So let call this middle one as node I. With respect to this node, I can make the basis functions exactly like you have done in the one-dimensional case as having a value 1 at the point of the node, the location of the node - and it [ ] of linearly to 0 at the other extremities of the element; that is, at the other node of the element sharing this node.

It is essentially if you look at it forms the tent like structure; so the things forms a tent. Beyond this point that is in all the elements which are outside this region this function is going to be 0. This function that you have constructed will be called the basis function phi_i as a function of x and y. Just like in the one-d case this will be our basis functions which is piecewise linear in the element that are connected to this node I. In all other elements, this function has a value 0. What do we have? The same feature that you have built in the one-dimensional case we also incorporate with the function in the two-dimensional case; that is, they have local support. These functions have local support. If I keep on defining these functions with respect to all the nodes I in the mesh then I have the set of basics function phi_i which are now generalized hat function, that is these are more like tent like functions in two dimension. So these are functions that we would like to create in order to construct our approximation. What more property this functions have to satisfy? Remember that these properties are generic and any construction in any dimension will have to satisfy this problem. So they have to be linearly independent; are these functions linearly independent? Now again, extrapolating the argument we have done in the one-d case.

If I look at a function made by a linear combination of this phi_i then I know that at a particular node only the corresponding phi_i is going to be of size 1 at all other node this phi_i is going to be 0 and at this particular node all other phi_i s are going to be 0 . If I take f x y is equal to sum over all the i's 1 to n $alpha_i$ phi_i if I take the x y to be the co-ordinate of the ith node then a [back] node if phi_i becomes 1 all other phi_i become 0 , so I get alpha i is equal to 0. So if I get alpha I is equal to 0 it tells me that when this thing is equal to 0 we can only v equal to 0 when all of the alpha is 0.

So these functions phi_i are in the linearly independent; they have to be complete. Complete means that the linear combination of this phi_i should be able to exactly represent any linear polynomial in the domain. If see you that is quite a trivial case because these will indeed in each element represent the linear; so you take the collection of this element and you will get the completeness over the whole domain.

So these are certain things that we have to keep in mind when we are making these basics functions in the two-dimensional case. Local support is one property, completeness is another property, and linear independence is another property.

In the next class, what we are going to do is we are going to go into the specific construction of these basis function. We will do it for the standard general Cartesian coordinate x y, then we will move over to a concept similar to what we have done in the one-dimensional, that is the master element. In the master element, you see that definition become much easier, management of the functions becomes much easier and it is also going to be directly incrementable in a computer program.