Finite Element Method

Prof. C.S. Upadhyay

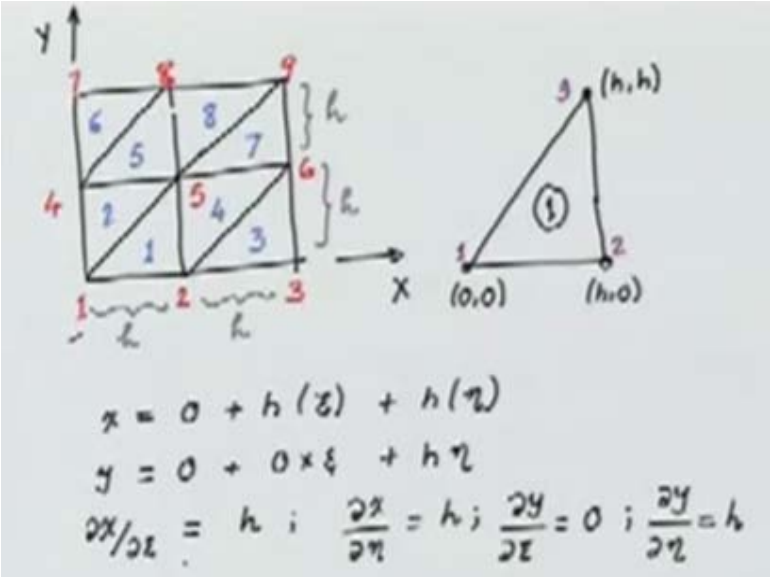
Department of Mechanical Engineering

Indian Institute of Technology, Kanpur

Module -7 Lecture – 3

Let us start this lecture with the example that we considered in the previous class. We have taken a square domain for simplicity and in this domain we have this mesh of triangles. Let us number the elements in the nodes of the mesh. These are nodes 1, 2, 3, 4, 5, 6, 7, 8 and 9. These are elements 1, 2, 3, 4, 5, 6, 7 and 8. This mesh has 8 elements and 9 nodes and as you see the way I have drawn, it's a square mesh and all the elements have the same shape. The straight edges of the elements are of size h. Both the straight edges of a given element are of size h. That is, the domain is of size 2h/2h.

(Refer Slide Time: 00:28)

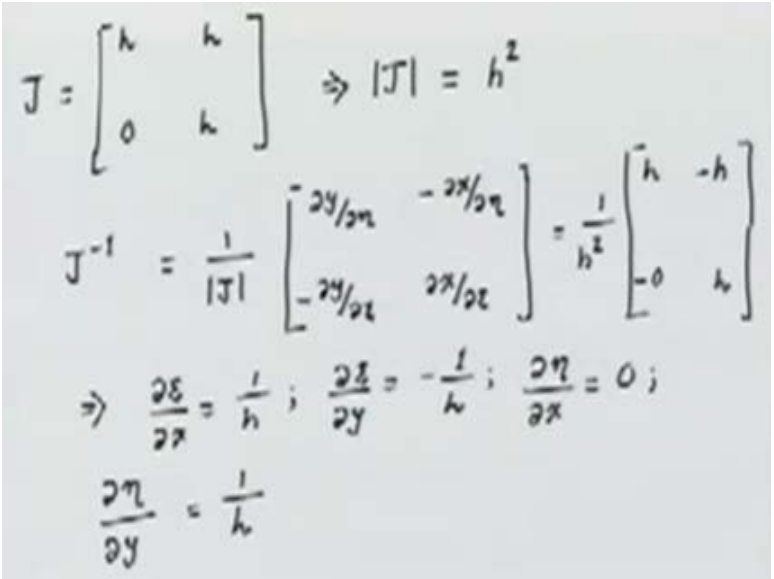


For this domain, I would like to do the element stiffness matrix calculations using the linear approximation over the domain. Let us go through the steps that we have done earlier. Let's take element 1 as an example. Element 1 is this. If I look at the coordinates of element 1, say, the coordinate system that I have employed is like this: this is the origin o, this is the x-axis and this is the y-axis. The coordinates of the N nodes of the

element 1 are (0, 0), (h, 0) and (h, h), and remember that the first one is the global node 1, the second one is the global node 2, the third one is the global node 3. Now I give local numbering to the nodes - this is local 1, this is local 2, this is local 3 and for the element, this is the first node, this is the second node, this is the third node and corresponding to the global numbers, 1 is equivalent to the global 1, 2 is equivalent to the global 2 and 3 is equivalent to the global 5. Given this, I can now write what is x: x is equal to x_1 N_1, in the master of the same representation. x_1 is 0, N_1 doesn't matter. So, 0 plus x_2 N_2, that is, h into x_2 N_2. What is N_2? N_2 is psi plus x_3 N_3. What is x_3? It is h; h into and what is N_3? It is eta.

Similarly, y is equal to y_1, which is 0, plus y_2 N_2. That is, (y_2 is 0) into psi, plus y_3, which is h into eta. These are my representations of x and y. Given this representation of x and y, now I can easily find what is del x del psi. Del x del psi equal to h; del x del eta is equal to h; del y del psi is equal to zero, because y is not a function of psi; del y del eta is equal to h. I am simply using my representation of the x and y.

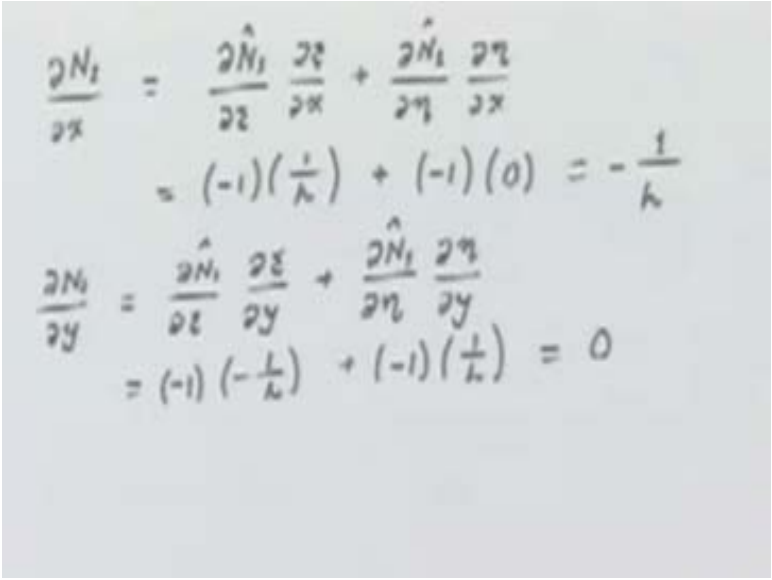
(Refer Slide Time: 05:14)



For the first element, these are the del x del psi etc that you get. If you remember, our Jacobian matrix becomes essentially del x del eta from the previous slide, which is h; this is h; this is zero; this is h. This is equal to the Jacobian matrix. This implies, the Jacobian

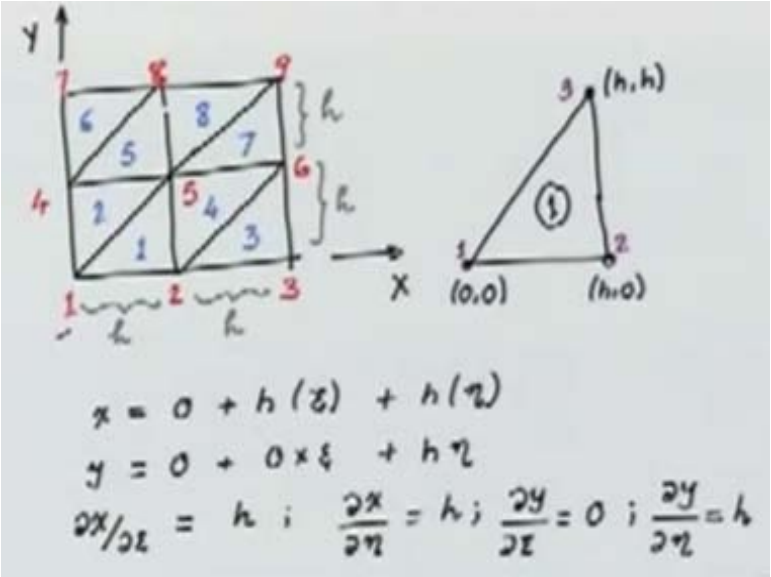
itself is equal to h squared. This is what I get as the Jacobian. If I get this Jacobian, then I would like to now get the metrics of the inverse transformation. If you remember that the inverse transformation will have J inverse is equal to one by Jacobian into del y del eta, del x del psi, minus del x del eta, minus del y del psi. This is equal to 1 by h^2 into h h; (and take the negative of diagonal terms), minus h and minus 0. This implies del psi del x is equal to 1 by h; del psi del y is equal to -1 by h; del eta del x is equal to 0; del eta del y is equal to 1 by h^2. Given this matrix of the inverse transformation, we are now in a position to find the x and y derivatives of the shape functions.

(Refer Slide Time: 07:41)



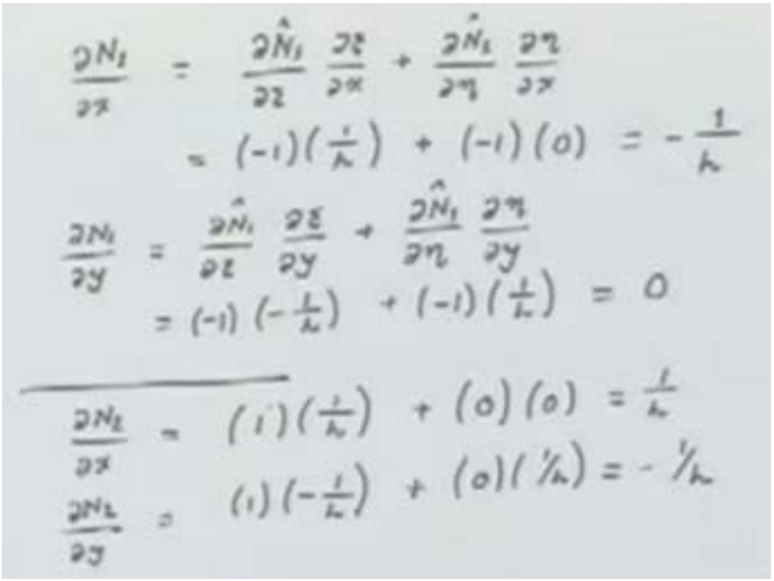
If we want to find the x and y derivatives of the shape functions, we have, del N_1 del x is equal to del N_1 hat del psi, del psi del x plus del N_1 hat del eta, del eta del x. Now N_1 was one minus psi minus eta. If I write it, del N_1 del psi would be -1 into del psi del x (del psi del x from the previous page is 1/h), so, 1 by h plus del N_1 del eta, which is -1, into del eta del x, del eta del x is 0. This becomes – 1 by h. Similarly, del N_1del y, would be equal to del N_1 hat del psi, del psi del y plus del N_1 hat del eta, del eta, del y. Again, this will be equal to -1 into (if I refer to the previous slide, del psi del y is equal to 1 by h) minus 1 by h plus minus one into (del eta del y for us is 1 by h) 1 by h. This becomes plus 1 by h, minus 1 by h, which is 0.

(Refer Slide Time: 09:46)



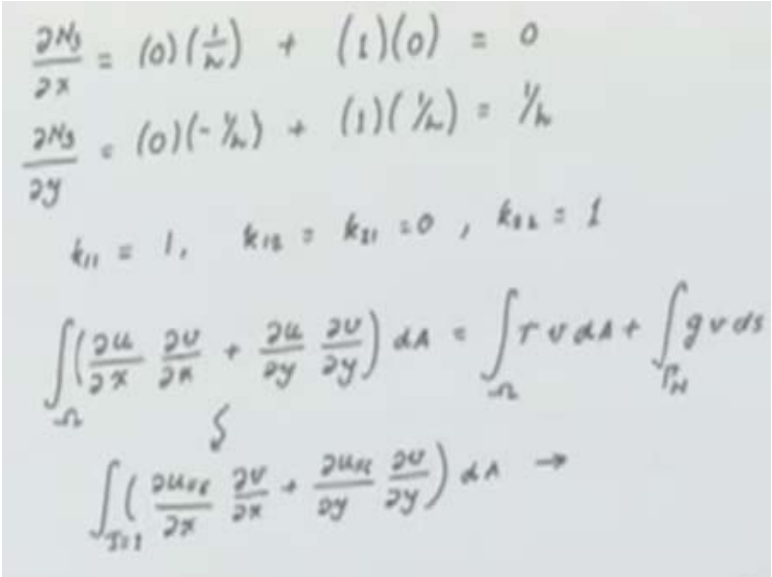
Let's go back and look at the element and see if it makes sense. If you look at the first shape function and I want to get it in terms of x and y, it turns out that this shape function will be nothing but 1 minus x by h. Because it has to have a value one at x equal to 0, it is 1; it has a value 0 at x equal to h. So, if it is in the physical element, it is 1 minus x by h. So, del N_1 del x is equal to minus 1 by h and del N_1 del y has to be 0 and that's exactly what we got out of our transformations.

(Refer Slide Time: 10:27)



Similarly, I can have del N_2 del x is equal to del N_2 del psi, which is 1 into del psi del x, which is 1 by h plus del into del eta which is 0 (because N_2 is del into hat del eta because N_2 hat is psi) into del eta del x, that is, 0. This is going to be 1 by h. del N_2 del y is equal to del N_2 del psi, which is 1, into del psi del y, which is minus 1 by h plus del into del eta, which is 0 into 1 by h, which doesn't matter. This is minus 1 by h. del N_3 del x is equal to del N_3, del psi. del N_3 del psi is actually 0 because N_3 is eta.

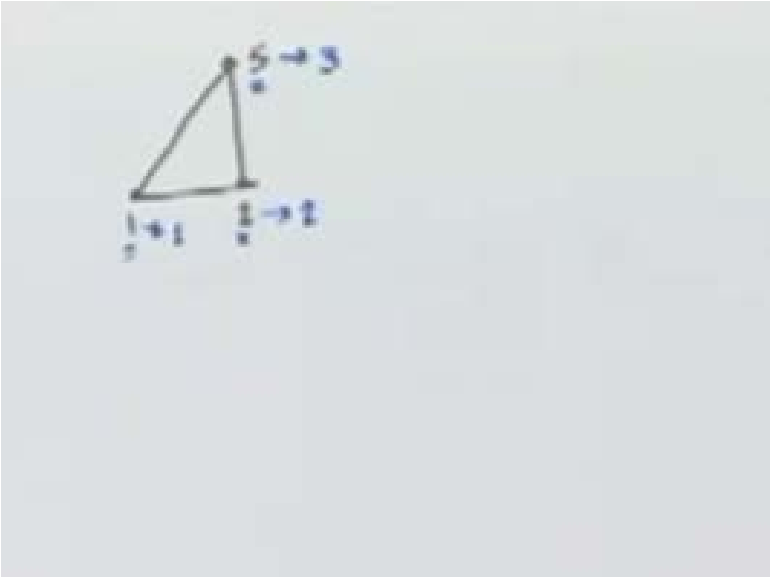
(Refer Slide Time: 11:57)



We have del $N_{3,}$ del psi which is 0, del psi del x which is 1 by h plus del $N_{3,}$ del eta which is 1 into del eta del x, which is 0. This is going to be 0. del N_3 del y is equal to 0 into del eta del y, which is minus 1 by h, plus del $N_{3,}$ del eta which is 1 into del eta del y which is 1 by h. This becomes 1 by h. I have these three quantities and as we had said last time, let us take a very simple differential equation. We had taken the generalized partial differential equation in terms of one variable, which is u, with the coefficients k_{11} k_{12}. What we are going to do is, we are going to take the differential equation k_{11} is equal to 1, k_{12} is equal to k_{21} is equal to 0 and k_{22} is equal to 1. Using that differential equation, we have the variation formulation over the whole domain omega will be given as del u del x into del v del x plus del u del y into del v del y dA is equal to integral over the omega, if there is a source term, r v dA plus on the Neumann part of the boundary we have g v d s, which was the boundary line for the contour.

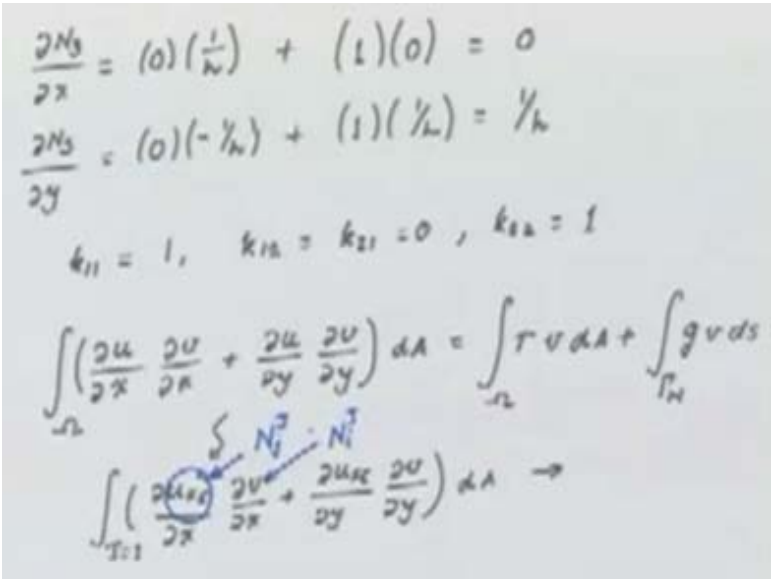
This is our differential equation. From this, as we have discussed many times, we are interested in the part of the integral on the right hand side and the left hand side coming from our particular element and that integral is going to give us our stiffness matrix and the force vector pumps. Let us take the elements: tau equal to one, for the first element I will have del u_{FE} del x del v del x plus del u_{FE} del y del v del y, this integrated over dA will be somehow related to our stiffness matrix.

(Refer Slide Time: 14:55)



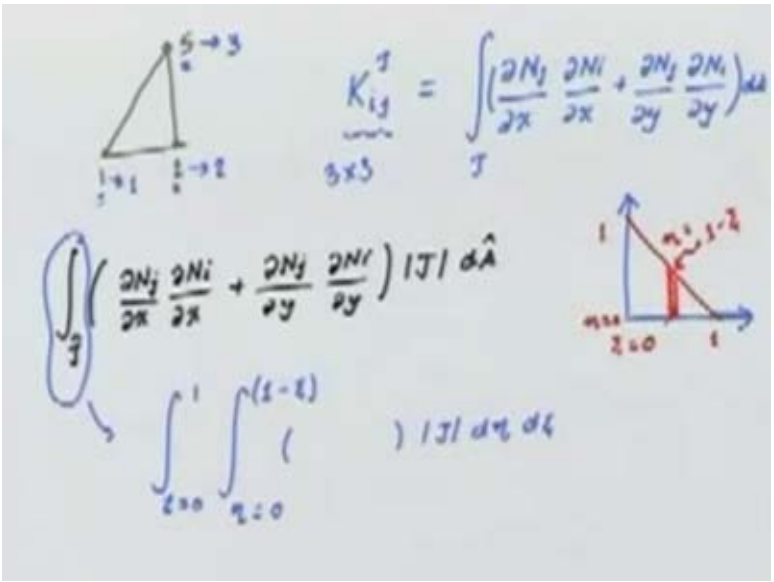
We know how to do it. In the first element if you remember, the active basis functions correspond to the degrees of freedom 1, 2 and 5, which correspond to the local 1, local 2, local 3. If I write my u_{FE} in the element in terms of the global phi_1 phi_2 and phi_5 because these are the only non zero ones, v also in terms of the global phi_1 phi_2 and phi_5, which means my element only contributes to the global equations 1, global equation 2, global equation 5 and to the columns 1, 2, and 5 of the global equations. This is exactly the way we have discussed it for the one-D problem.

(Refer Slide Time: 15:42)



Which means, from our previous representation, I will be talking only in terms of N_j of the element tau instead of u_{FE} and instead of v I will put N_i of the element tau, where, the i and the j go from 1 to 3, because I have only three functions defining the linear in the element.

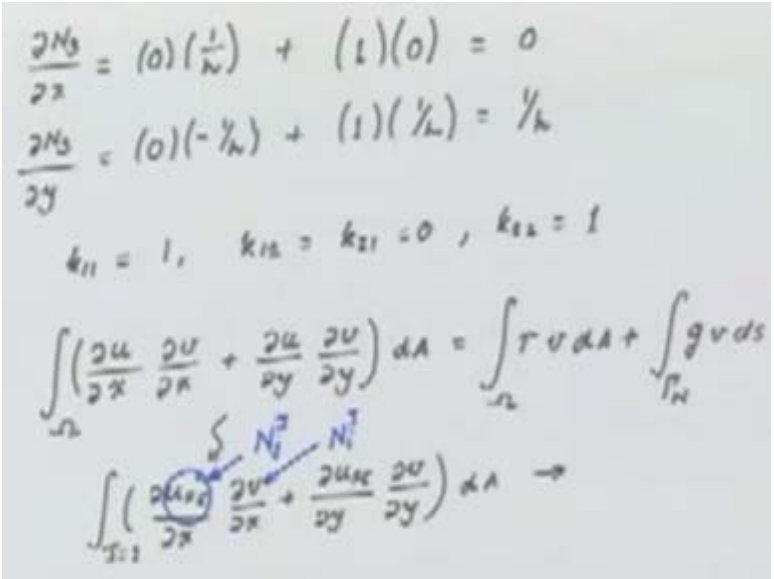
(Refer Slide Time: 16:06)



So, the element stiffness matrix will then be K_{ij} for the element tau, which is here equal to one, this is equal to integral over the element tau of del N_j del x del N_i del x plus del N_j del y del N_i del y dA. I have to evaluate each of these entries and this has to be a three by three matrix, because I have only three unknowns in the element. How do I evaluate these entries? Well, we have done most of the job. I take this integral, map it to the master element, that is over tau hat and here I will have retain these things and that these derivatives are obtained with respect to the master coordinate system, into Jacobian into dA hat. This integral is actually equivalent to writing an integral over if you see in the master element, the way we have defined the master element, psi goes from 0 to 1 and eta also goes from 0 to 1. If I look at the area of the strip, equation of this line is equal to eta is equal to one minus psi and this line is eta equal to zero.

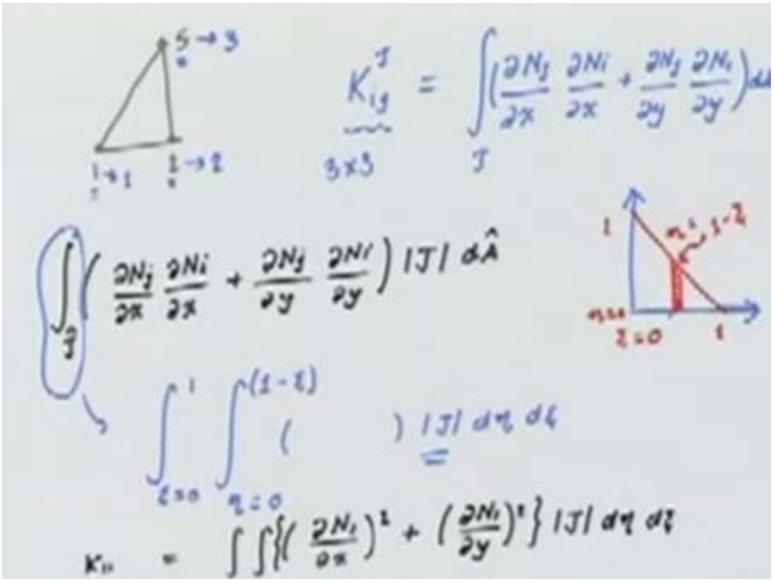
Then I can write the integral as psi going from 0 to 1, integral over eta goes from (I am looking at the area of the strip) 0 to 1 minus psi into whatever is my integral into the Jacobian into d eta d psi.

(Refer Slide Time: 18:56)



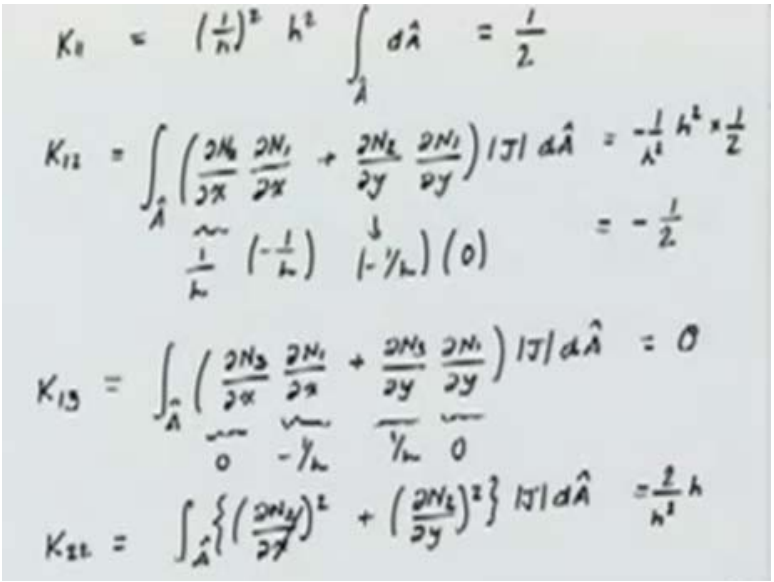
This is the integral I am supposed to do. For us, the job is very easy, because if you see, our derivatives del N_1 del x del N_1 del y del N_2 del x del N_2 del y and so on are all constants. So here we are simply integrating constants.

(Refer Slide Time: 19:06)



The Jacobian is also constant so our integral gives us, let us say I want to find k_{11}, will be equal to integral over the psi integral over the eta of del N_1 del x whole squared plus del N_1 del y whole squared into Jacobian into d eta d psi. For us this del N_1 del x whole squared will be equal to (del N_1 del x is equal to minus 1 by h, del N_1 del y = 0), let us put that and Jacobian is equal to h squared.

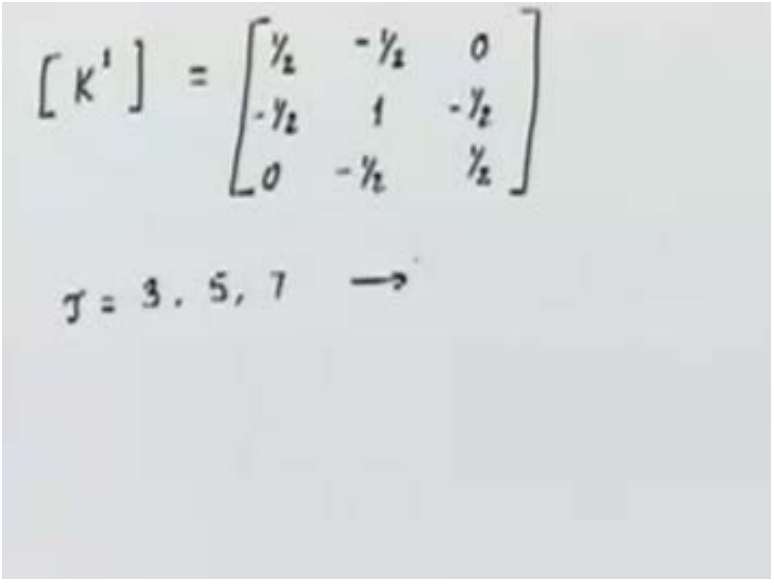
(Refer Slide Time: 20:05)



Let us put all those things in our expression so K_{11} is equal to (del N_1 del x is equal to minus 1 by h) 1 by h whole squared into the Jacobian. Jacobian is also a constant, so this is h^2 into integral over the area of the master element dA hat. Essentially, this whole integral is nothing but area of the master element, but area of the master element is a right angle triangle with sides of length one, so area of the master element is half. So, this becomes one by two. Similarly, I can find K_{12}. K_{12} will be integral over the area del N_2 del x del N_1 del x plus del N_2 del y del N_1 del y into Jacobian into dA hat. For us, del N_2 del x if we compute, del N_2 del x was equal to 1 by h this is equal to 1 by h del N_1 del h is equal to minus 1 by h. If I look at del N_2 del y, it will be minus 1 by h and del N_1 del y will be zero. This quantity essentially becomes equal to minus 1 by h squared into the Jacobian h squared into the area of the master triangle, that is, half. This is minus half.

Similarly, K_{13} will be equal to integral over A hat, del N_3 del x del N_1 del x plus del N_3 del y into del N_1 del y into Jacobian into dA hat. If I do this, I know this is minus 1 by h and this quantity is zero. So, del N_3 del x is zero and this is 1 by h. The whole integral is zero actually because integrant is zero. Similarly, I can find what is K_{22}. K_{22} will be integral over A hat of del N_2 del x whole squared plus del N_2 del y whole squared into Jacobian dA hat. This whole thing will become two by h squared into h squared into half. This becomes one.

(Refer Slide Time: 23:52)

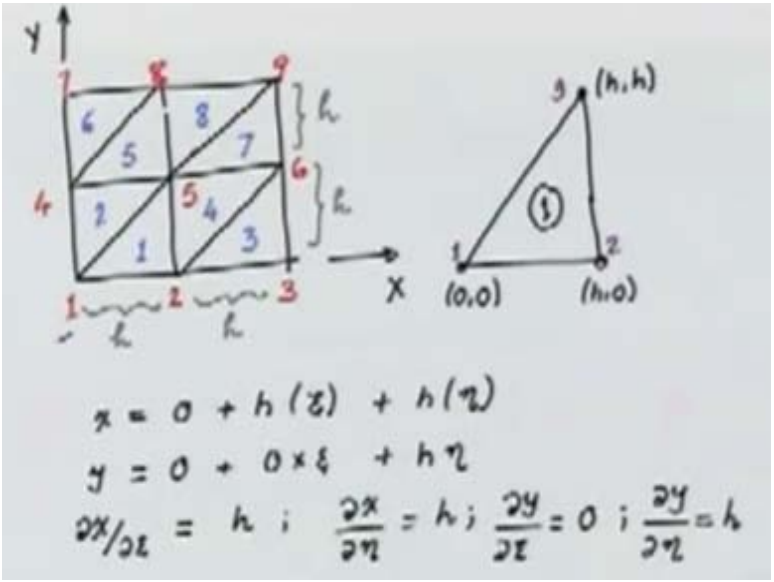


If we do this exercise all the way through, then I will see that my K for the element one is equal to half minus half, zero, and because this stiffness matrix is symmetrical, I don't have to compute all the terms. I can get them from symmetry. This is half, K_{22} was 1 and if we do the calculations, we will find out that K_{23} is minus half. This one will be zero. This is minus half and this is half.

One thing you should see is that if I add the rows or the columns, all the terms in a row, or all the terms in a column of the stiffness matrix, then I get zero, which tells me that this stiffness matrix is actually singular. It should be because if I look at an element as a whole domain, then obviously, the element should give me a singular stiffness matrix, because the global rigid modes are not constrained.

The global rigid modes correspond to the constant u. We have to fix one value of u. In fact, the rank deficiency of this matrix is 1. This one can check out. If I can find the elements stiffness matrix for element 1, then similarly, I can find the element stiffness matrices of elements 3, 5, and 7. Why?

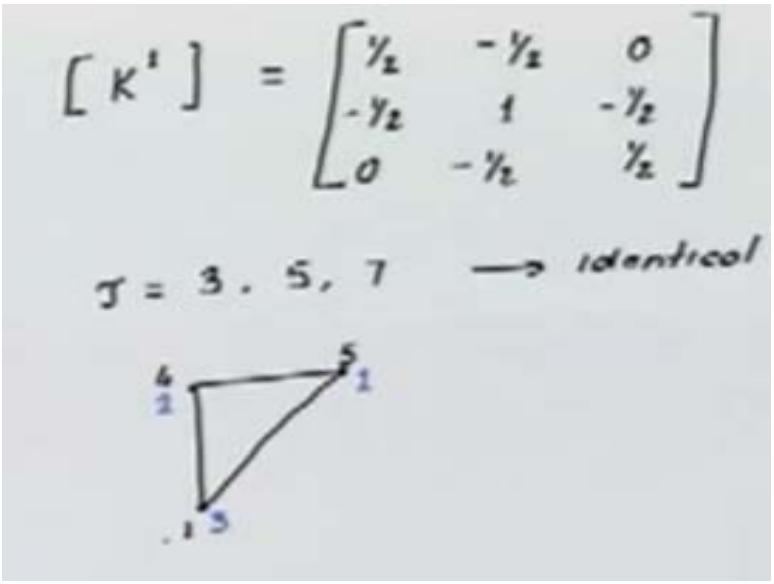
(Refer Slide Time: 25:30)



Because, if I go to my first slide, you see that the first element, the third element, the fifth and the seventh have exactly the same geometry. They are essentially the same triangles,

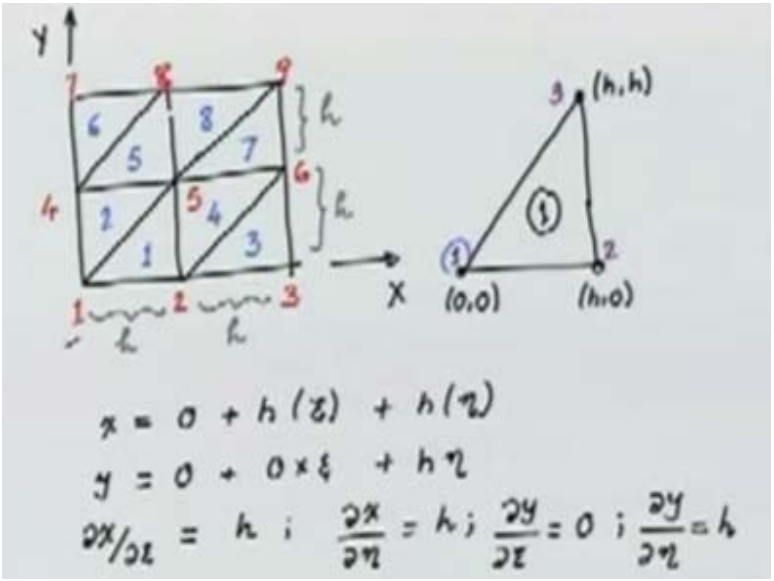
just translated by some distance in the x direction or y direction. The derivatives of the N_1 N_2 N_3 in each of these elements will remain exactly the same as that I got in element 1 and because of those derivatives being the same, the metrics of forward and inverse transformation will also be the same. The stiffness matrices corresponding to the elements 1, 3, 5 and 7 will be identical.

(Refer Slide Time: 26:12)



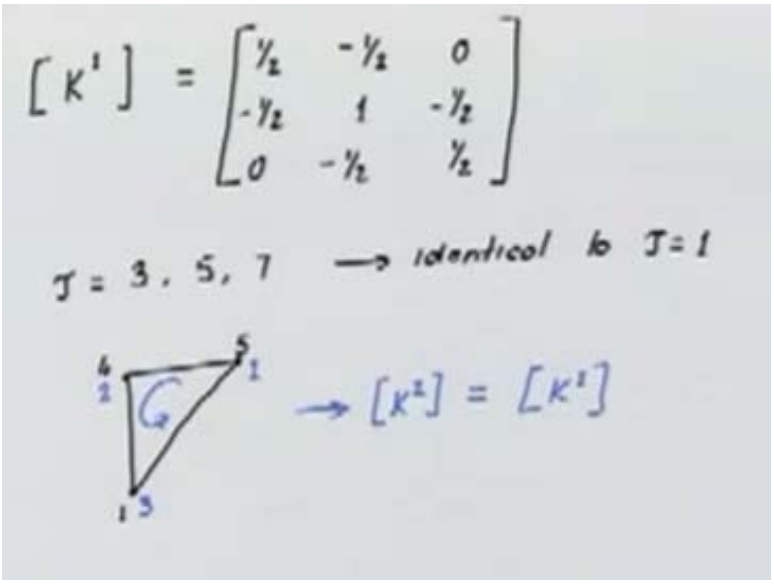
This will be identical to tau equal to one, as far as the stiffness entry matrices is concerned. If I take element 2, element 2 is like this, with the global nodes 1, 4, and 5. I can do the local numbering for element 2 in such a way as this. This is my local 1, this is my local 2, and this is my local 3.

(Refer Slide Time: 26:53)



If I do that numbering, then I am actually following the numbering which I had done for element 1. If I turn the element 1 around, that is, rotate it around, I will get exactly the same local numbering as I had done for element 2.

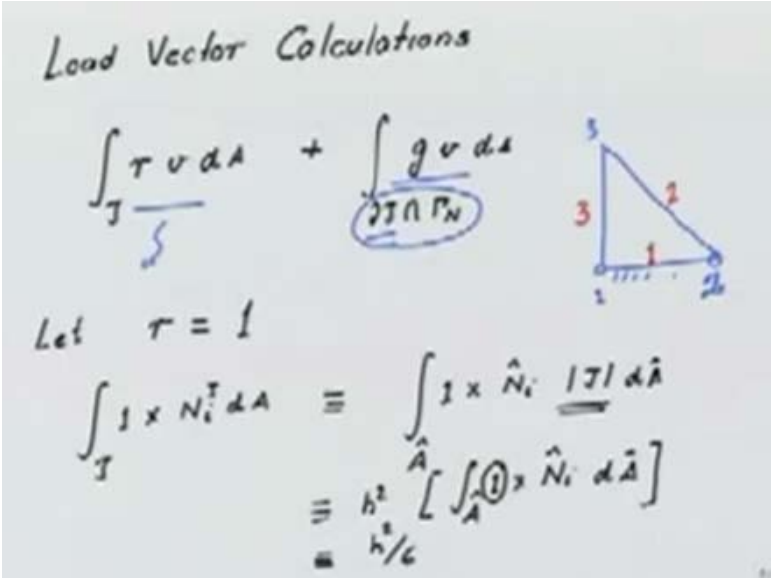
(Refer Slide Time: 27:10)



If I do this numbering for element 2, then the local enumeration is completely up to me. The only thing I have to ensure is that the enumeration is done in a counter clockwise

manner. I always number the local numbers in a counter clockwise manner. Then for the element 2 also you will see that K of two essentially equals K of one. In a way, I have now been very easily able to obtain for this particular problem, the element stiffness matrices for all the elements concerned.

(Refer Slide Time: 27:54)



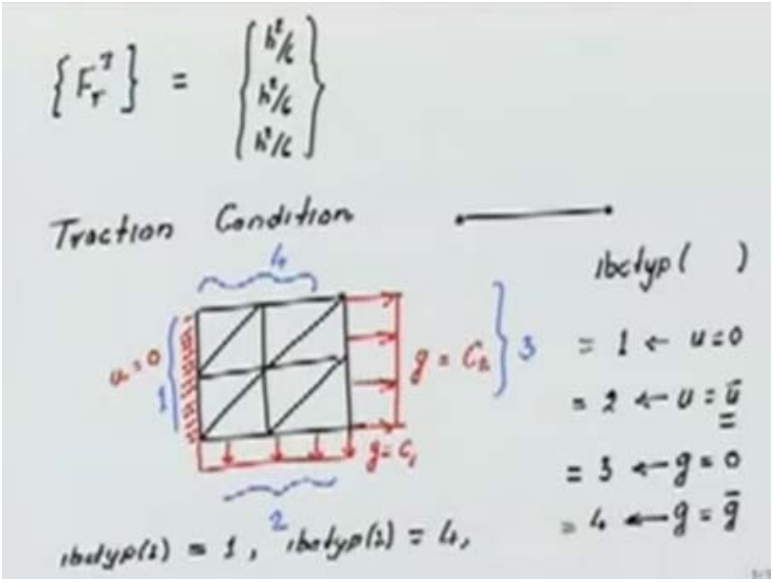
Next, we have to do the load vector calculations at the element level. When I am talking of the load vector calculations, then I am essentially talking of the integral of the element over the element of r into v dA plus the part of the Neumann boundary, which intersects the edges of the element g v d s. We also have to take care that this is essentially due to the internal source term or the body force term and this part is due to the boundary traction terms or the flux terms and we have to see whether the element shares an edge, because the element has the three edges. I can define the three edges of the element that is in the master element. This is 1, this is 2 in the master element, this is 3, then in the master element I will call this as my edge one, this as my edge two and this as my edge three.

If I have an edge, the edge actually has the 2 extremities given by the 2 nodes sitting at the two edges. Edge is like a one-D element and so for each edge I have the two nodes corresponding to the edge and I can have either edge 1 or edge 2 or none of the edges

lying on the boundary of the domain. And if it is lying on the boundary, I have to figure out whether it is common and if it is actually on the Neumann boundary. If it is, then I have to apply the Neumann conditions there. We are first going to talk about this part. Let r be equal to one. Let us take a constant value of r and we have taken r is equal to one. At the element level again, which of the vs are non-zero - it is essentially the N_1 N_2 N_3 of the element. I will have to do the integral over the element of 1 into N_i tau dA.

But, this is equivalent to doing integral over the master element A hat of one into N_i hat into Jacobian into dA hat. This Jacobian is a constant. So this is essentially equal to h squared into integral over A hat of one into N_i hat dA hat. If I do this integral with respect to psi or eta, you will see that these integrals will come out to be the same for each of these shape functions. Remember, this is true because we have taken r as a constant. The integral is going to come out to be h squared by 6.

(Refer Slide Time: 31:20)



I am now in a position to write the F for the element tau and this F tau is going to be same for the mesh that we have taken for all the elements. Remember, here we are talking only of the part coming out of r. So F tau r is equal to h squared by 6, h squared by 6, h squared by 6.

Let us go back. If instead of this r I had r as some function of x and y, then here I have to map this as a function of r hat of psi and eta. This integral now has to be done piece-wise. That is, for each of the shape functions I have to find the integral. I cannot say now that the load vector term coming out of this integral is the same for each of the shape functions. This is one thing you should keep in mind. I have to do the explicit integration. Next let us see how to take care of the traction condition. Earlier, in the one dimensional problem that we had done, as far as the traction condition is concerned, we had actually taken the two end points of the element of the domain and the boundary conditions were applied only at these end points.

Now we are talking of a two-dimensional domain, where the boundary conditions are applied on edges, on the boundary edges. So we have to somehow apply those conditions into our integral representation. In the one-D case it was very easy because it was at end points and I could go to the corresponding load vector entry and add numbers. Here we have to do integrals over the boundary edges. That is why we will do them not at the end of our element calculations, but while we are doing the element calculations. This is one major difference between the one-D case, that we do not wait till the end of the whole assembly process and then apply the boundary conditions but rather, we do it while we are doing our element calculations,

So how are we going to do this? Let us again take the domain and also make the mesh along with it. Let us say, on this edge I fix the u, that is, u = 0. On this edge I have given a constant flux, that is, g is equal to some constant C_1. On this edge I could have the flux g is equal to C_2 and this edge is actually a flux condition with zero as the value of the flux. We see that there are distinct edges of the domain on which I am specifying the boundary conditions. I am going to now identify these edges where the boundary conditions are changing. I am going to identify two things as far as the domain is concerned: the distinct edges where the domain profile is changing and the edges where boundary conditions are changing and are of either one type or the other type.

In this example, we will say that this is boundary edge one. This whole edge is boundary edge one, this whole edge is boundary edge two, this whole edge is boundary edge three

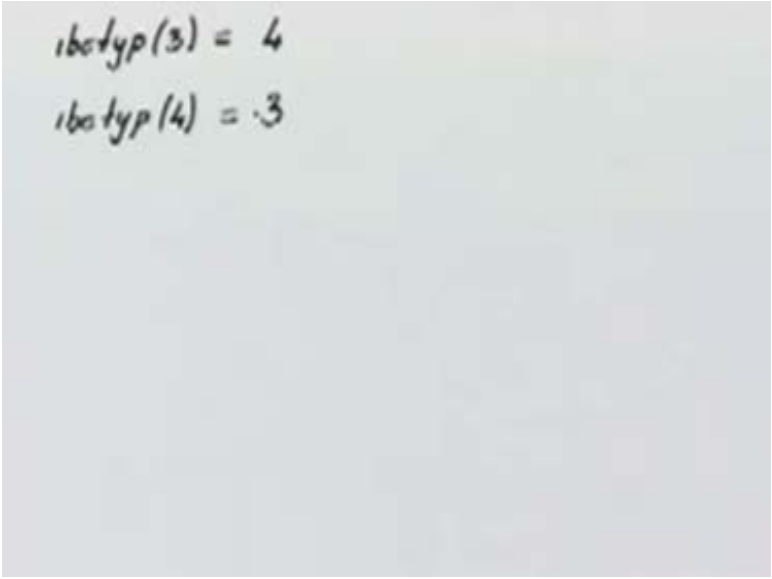
and this whole edge is boundary edge four. We have the four boundary edges in this particular problem. I will discuss this issue further, but let's take this particular case. On these four boundary edges now you see there are different types of boundary conditions. What I am going to now do is, I am going to apply the ibc type (remember that we had talked of ibc type associated with the 2N nodes telling us what type of boundary conditions is applied on this N nodes).

The ibc type is now going to be defined for the boundary edges. I will have four and I am going to say, this ibc type for a particular edge is equal to one. This is my choice. When I have homogenous dirichle boundary conditions, that is, when u is specified on the edge and u is equal to zero, this is equal to two. When I have non-homogenous dirichle boundary conditions, u is equal to u bar. In practical applications, we can take displacements applied on the boundary, which can be a displacement-controlled problem or a temperature controlled problem, where the displacement on the edge can be utmost a linear. We are not going to have very fancy profiles. From our implementation point of view, let us say it could be a linear.

So by this u bar we mean some non-zero value of u given on the boundary, that is, the given boundary condition. I will say it is three, when it is a Neumann boundary condition with homogenous type of boundary conditions, that is, g is equal to zero. It is a homogenous Neumann boundary condition. And, it is equal to four when it is a Neumann boundary condition, but g is equal to a non-zero value g bar. This is my way of representing each type of boundary condition that can exist on the given boundary edge and I am going to assign this number for each of these boundary edges.

For example, for ibc type one in this case, ibc type for the edge one is actually type one. Ibc type for edge two is 4 (g is non-zero so it is equal to 4).

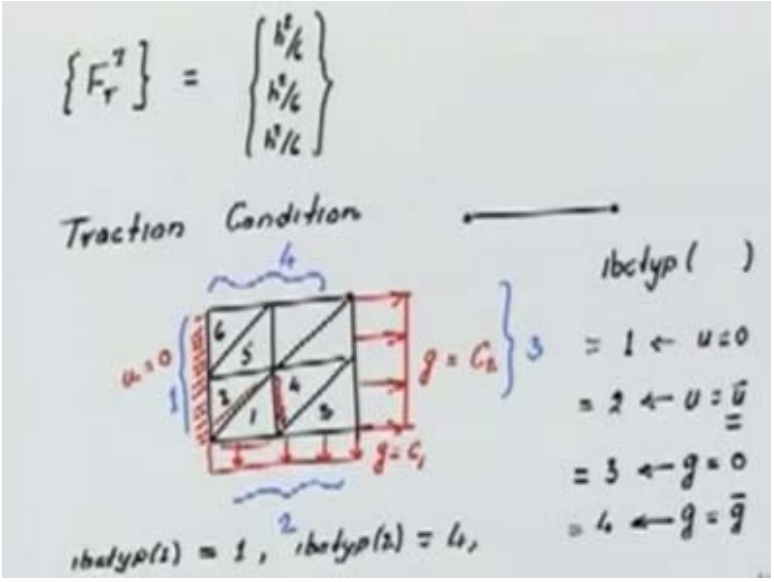
(Refer Slide Time: 38:39)



Similarly, ibc type for edge three is also non-zero Neumann, so it is also four and ibc type for edge four is the zero Neumann so it is three. My first job is to identify the edges where a specific type of boundary condition is applied. Now, for those edges I am going to specify what is the boundary condition type. Once I have the boundary condition type, let us say I choose my flux on the boundary, I am trying to solve a set of problems for which the flux on the boundary is at most linear and applied displacement on the boundary is at most linear. I can give the flux and the displacement as linear functions on the boundary, I can give them in terms of the x and y or I can give in terms of the normal and tangential components.

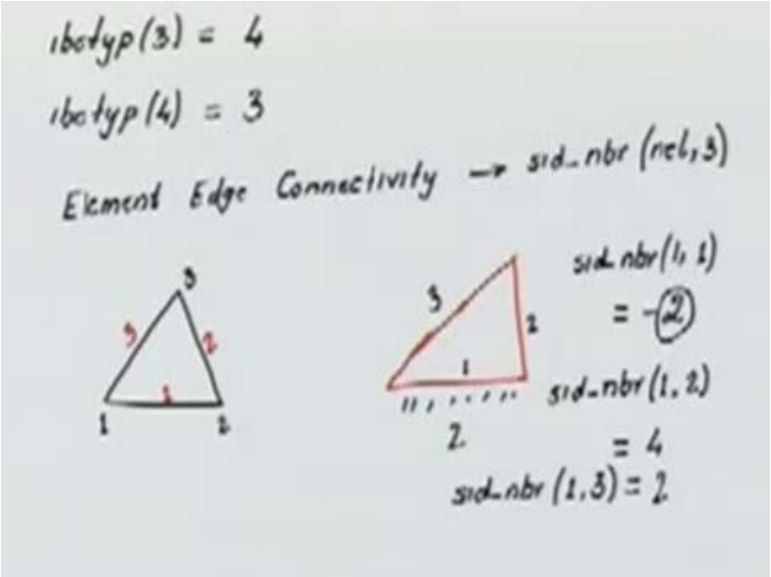
Let us say that we are giving it in terms of x and y. When we are giving the u and u bar and the g bar as a linear in terms of x and y, it will have three coefficients corresponding to the A_0, A_1 x, A_2 y. Those three coefficients have to be given as input data for the particular boundary edge, stored and then used. The question is, how do I now use this information to do my boundary integral calculation? How does one know apriori or while doing the computations that this element has an edge lying on the boundary? That information has somehow to be generated.

(Refer Slide Time: 40:41)



I am trying to think more like a computer here. If I go back to my previous picture, here element one has an edge on the second boundary; element two has an edge on the first boundary; element three has an edge on the second boundary and an edge on the third boundary; element four has no edges on the boundary. Similarly, element five has no edges on the boundary; element six has one edge on boundary one and the other edge on the boundary four. This way, I have for each element, either one or more edges on the boundary or none of the edges are on the boundary, that is, an element is a completely internal element, when it has no edges on the boundary or an element is a boundary element, when it has at least one edge on the boundary. We are interested in these boundary elements. How do I obtain that information? What I do is, while we are making the mesh, we essentially make a neighborhood information for the edges of the elements.

(Refer Slide Time: 41:54)



What we do is, we make this neighborhood information, what we call element edge connectivity and will give it a name, side underscore neighbor. This will have all the elements, that is, for any l elements in the mesh and for the three sides of the element. How have I numbered the sides? If I have, let us say, a generic element like this: this is the local one node, this is the local two node, this is the local three node, then the side with the ends as local nodes one and two is side one of the element, the side with ends as the local node two and three is the element side two and the side with ends as the local nodes one and three is the side three of three node.

This is how I am going to identify the three edges of an element. If I take element one and make a picture, it will have this as edge one, this as edge two this as edge three. On the edge one, I look at the side neighbor of element one for edge one. Side neighbor for the element one for edge one is boundary side two. Then I will say that the side neighbor of element one for edge one is boundary edge two.

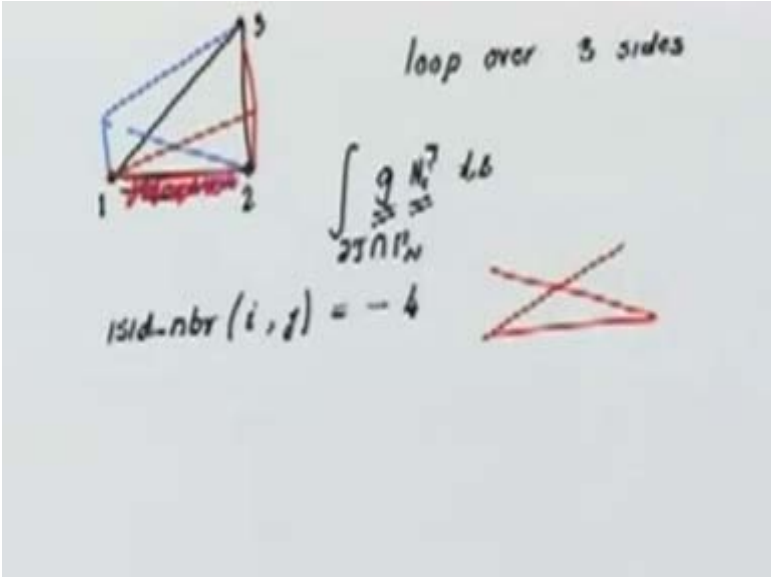
When it is a boundary edge I am going to assign a negative number with the number in the corresponding boundary edge. I will give minus two as the neighbor of this element on this side. As soon as I see minus two, I know that this side lie's on a boundary, that is, the second boundary edge. Similarly, if I look at side neighbor for the element one on

edge two, this is the edge two of the element one. This edge two has element four as a neighbor. Element four shares the same edge with this element. This will be equal to four. Side neighbor, similarly, for element one on the edge three is element two. This is the way I can create this edge connectivity information. Why do I need information? Firstly, to identify which edge of the element of a given element is on the boundary, that is, first of all, whether the element has any edge on the boundary or not.

Secondly, if it has, then which edges and which boundary segments do these edges lie on, because for the boundary segments we have no specified input data on the boundary conditions. Those are available to us. We have to transfer that to the integral on the boundary. Also, if you remember, we did some post-processing of the stress information in the one-D case, where we obtained this recovered stress field using information from the two elements on either side of a given element.

We can do the same thing in the two-D case by taking information from all the elements that are in a neighborhood of the given element of interest. For that again, we need to obtain this connectivity on the side information. Once we have this information, we do it for all the edges while we are making the mesh. And you see that by making this information we can also immediately figure out which element edges are lying on the dirichle boundary. And from there we can find out on which nodes or on which degrees of freedom of the domain values have to be assigned, because of the essential boundary conditions specified, that is, which of these ui’s in the representation of my finite element solution have to be given fixed values.

(Refer Slide Time: 47:21)



We will talk about that information a little later. I now have this side neighbor. Let us say that I go to element one and I start looping over the loop over the edges, loop over three sides of the element and check if that side has a negative neighbor. If it has a negative neighbor I check whether that negative neighbor is minus four. Why do we say that the negative neighbor has to be minus four? It's because we are talking of only computing the integral on the Neumann boundary of g and N_i tau ds.

We are only interested in the boundary segment were g is non-zero out of the Neumann boundaries, because, wherever g is zero, this integral becomes zero. I do not even have to bother about computing this integral. That's why I distinguish between the homogenous, Neumann boundary condition and the non-homogenous Neumann boundary condition.

We are only interested if Iside neighbor for the element jth side is equal to minus four. When this is there, then for that particular side I have to do this integral over the side. How do I do this integral over the side? We have to do this integral against the N_i tau. We have to ask the question, "Which of the element shape functions are non-zero on this edge"?

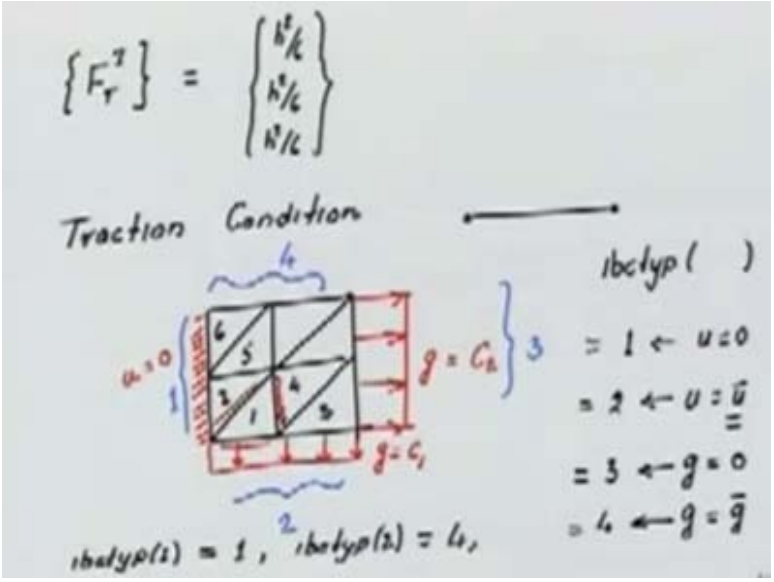
You see N_3 becomes zero all along this edge. The only non-zero shape functions on this edge are N_1 tau and N_2 tau. N_1 tau and N_2 tau are the only shape functions that are non-zero on this edge, which is my first edge of interest. So, given the edge, I can easily find out which of the shape functions of the elements are non-zero on this particular edge.

If we look at the shape functions on this edge they are now exactly the one-dimensional shape functions that we have studied earlier. Why? Because, we see along this edge is a line, along this edge N_1 tau is one at the first node, zero at the second node. N_2 tau is one at the second node, zero at the first node.

That's exactly the one-D shape function. We can use all that we have learned with respect to the one-D integration, numerical integration, to do the integration of the non-zero Neumann conditions against the shape functions on this edge. What we are saying here is: let this g also be a constant. As a simple case we are taking the g to be constant on this edge.

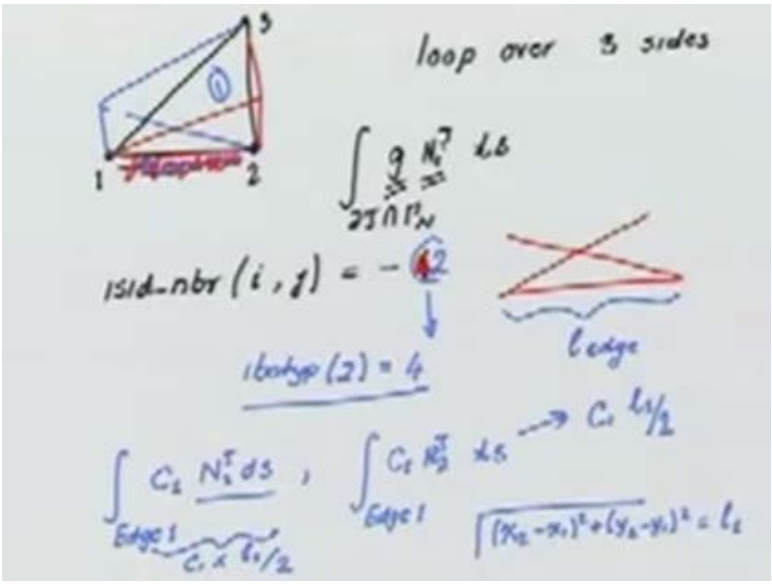
Let us say, if I am talking of the element one, then I find that the first edge of element one lies on a boundary. I made a mistake in my previous figure: the side neighbor is negative. The side neighbor is minus 2, minus 1, etc., but only when the ibc type for this particular boundary is, let us say, equal to four, only then am I interested in this boundary edge, otherwise I am not. I made a mistake in transferring the data. Only when the ibc type is four am I going to do the integrals on this edge.

(Refer Slide Time: 52:05)



I find that the first edge of the element lies on the boundary edge two. The boundary edge two has an ibc type which is four and now I can go ahead and do the integral of the given g on this edge against the shape functions which are non-zero on this edge.

(Refer Slide Time: 52:30)



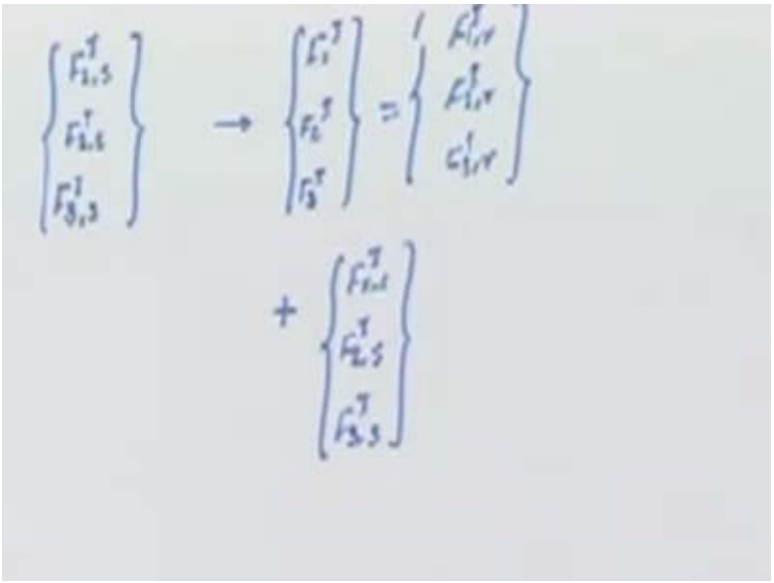
How do I do it? If I take the first element, let us say this is element one, I will have integral over edge one, edge one of the element of g which is C_1 into N_1 tau ds and over

edge one of C_1 into N_2 tau ds. I have to obtain these two integrals. C_1 is a constant so I know what is an integral of the linear one-D shape function along an edge, when I know the length of the edge. I will have to find the length of the edge. How do I find the length of the edge? Length of the edge is very simple. It is essentially square root of, assuming I take the first edge, $(x_2 - x_1)$ squared plus $(y_2 - y_1)$ squared, which is equal to l_1.

Similarly, I can find the length of all the edges of the element. I know the length of this edge is l_1 for the first edge, l_2 for the second edge, l_3 for the third. I can find them from the physical coordinates. This integral, we have done enough of this, is simply equal to C_1 into the length of the edge divided by two. Why? It's because, the integral of N_1 and N_2 is the same and N_1 plus N_2 is one. Integral of one along the edge is the length of the edge and because they are the same this becomes l_1 by two.

Similarly, this term is also C_1 l_1 by two for the edge of the particular element. I have these two integrals for this element. What do I do? I have obtained F_1 tau or F_2 tau or F_3 tau or I will now obtain the same thing for the sides. Then I am going to have my F_1 tau. So I loop over the sides, compute these integrals that we have done and add it to the F_1 tau side, F_2 tau side, F_3 tau side depending on which shape function is non-zero on this edge.

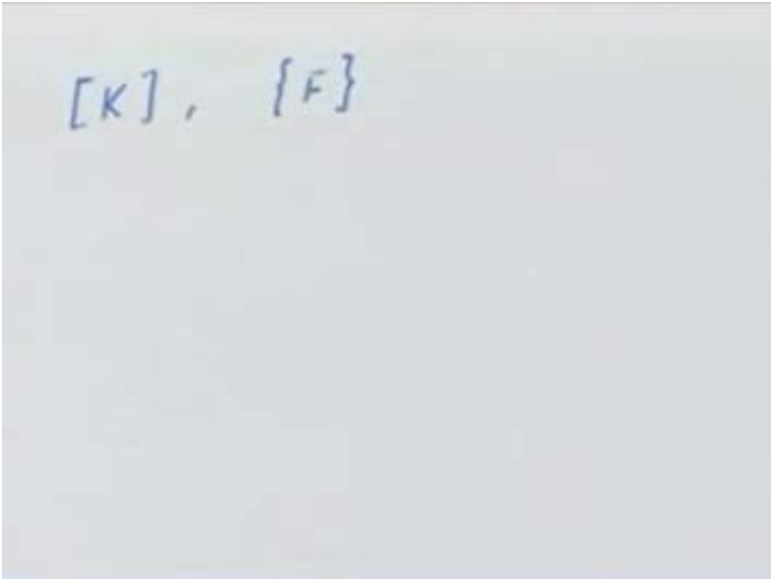
(Refer Slide Time: 55:44)



For example, on the first edge I will only get to the F_1 tau F_2 tau contributions. F_3 tau will be zero and other places something else will happen. So I obtain this vector after looping over all the three elements. Then my total F_1 tau is equal to F_1 tau or F_2 tau or F_3 tau or plus F_1 tau s F_2 tau s F_3 tau s. I have now essentially found, while I am doing the element calculations, I will do an integral over the area of the element and an integral over the edge of the element. Area integral will give me the stiffness matrix entries and the load vector entries come out of the distributed source term, while the integral over the edges or over the sides gives me the load vector entries coming out of the non-zero Neumann boundary condition.

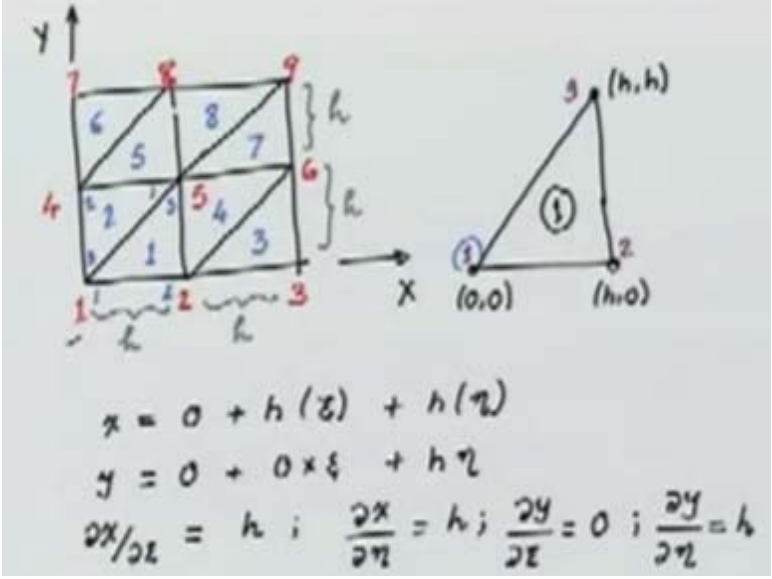
In general I am going to always initialize (obviously before doing the calculations in the element) this vector to zero, I will initialize this vector to zero and initialize this vector to 0. After doing my integrations and so on, this term, this vector will have some entries, this vector will have some entries simply add them up and I am done. I can actually obtained all the information about the Neumann boundary conditions at the element level itself and do essentially the adding of these contributions to the element load vector in one shot. Once I have that, I will have all the information that I need as far as the Neumann boundary conditions are concerned.

(Refer Slide Time: 57:52)



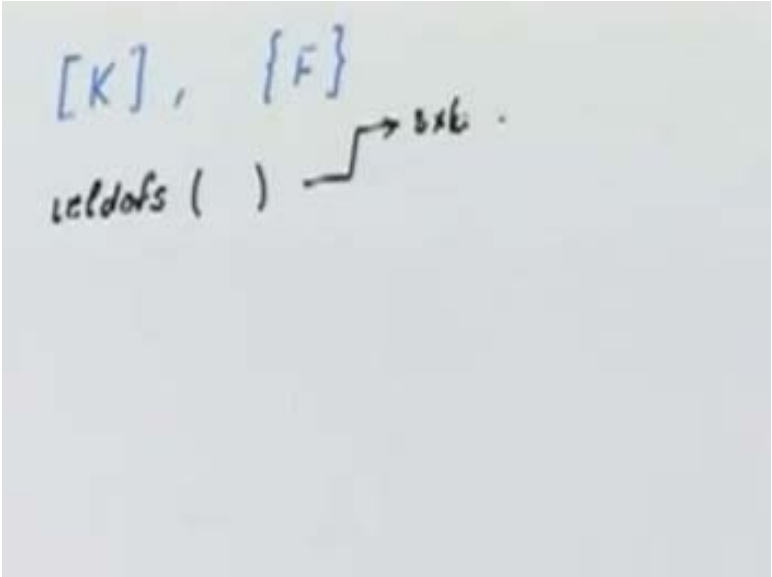
The next step is that we have obtained the element stiffness matrix, element load vector. What do I do? I will have to assemble them to obtain the global stiffness matrix and load vectors. Global stiffness matrix will be K global and global load vector will be F global. What do I do? For that again I need to construct the information, which tells me where the element degrees of freedom go to globally, that is, which global degrees of freedom do they correspond.

(Refer Slide Time: 58:23)



Remember that for the first element, for example, local one corresponded to global one, local two corresponded to global two, local three corresponded to the global five. For element two, the local one corresponds to the global five, the local two corresponds to global four and local three corresponds to global one and so on. I have to somehow construct this information. Again I can do as and when I am making the mesh, because when I am making the mesh, then I can go and see which global node my local one corresponds to, because, I am anyway storing the name of the global nodes assigned to the element.

(Refer Slide Time: 59:21)



There I can go and construct, what we had done earlier - the degrees of freedom array or the vector ieldofs. This is a vector which for an element I, we will have a starting point of in this case 3 into i. I will stop here.