Finite Element Analysis

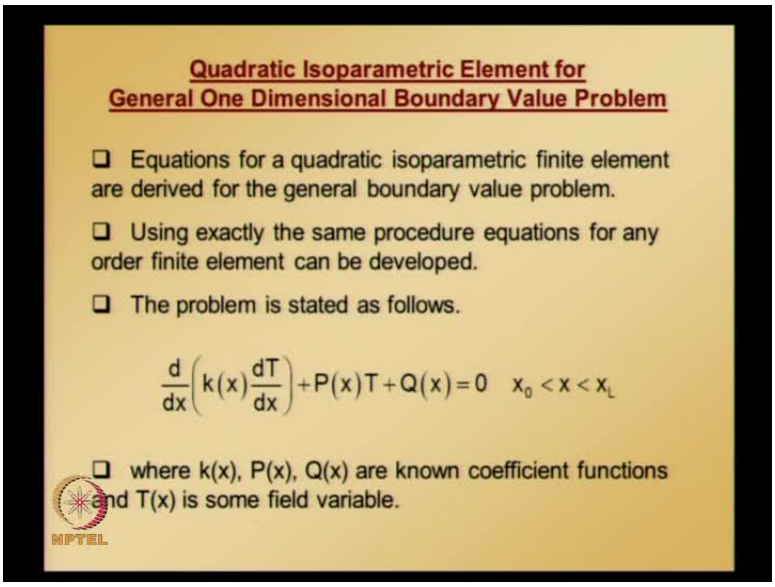
Prof. Dr. B. N. Rao
Department of Civil Engineering
Indian Institute of Technology, Madras

Module No. # 01
Lecture No. # 19

As a part of higher order elements for one-dimensional boundary value problems in the last class, we have seen how to derive shape functions for 3 node element, and also, we have seen this derivation of shape functions for higher order elements using two kinds of procedures. One is starting with a polynomial having number of coefficients is equal to number of nodes for the element and the other way is we derived using Lagrange interpolation formula.

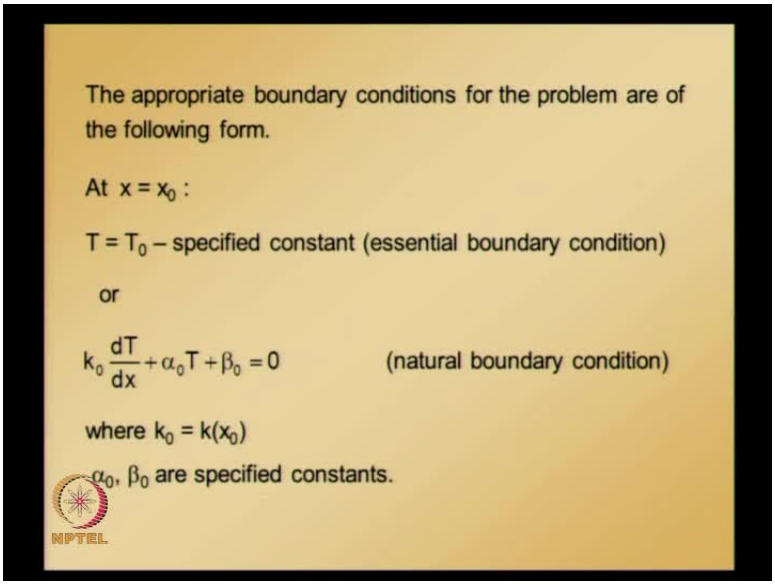
So, we can use any of these methods to derive shape functions for higher order elements and in the last class, we have seen how to derive shape functions for a 3 node quadratic element and also we have seen through examples that the placement of interior node influences the way the shape functions behave and also we looked at isoparameter mapping concept. Also, we have derived the condition for the placement of the interior node in case of a 3 node quadratic element. The distance of the interior node from either of the extreme nodes should not be less than L over 4, where L is length of the element, 3 node element.

(Refer Slide Time: 02:26)



In today's class we will continue; so, what we will be doing in today's class is we will be deriving element equations using quadratic element for general one-dimensional boundary value problem which we already did earlier, using 2 node linear element. So, equations for a quadratic isoparametric finite element are derived for general boundary value problem using exactly the same procedure. Equations for any higher order finite element can be developed.

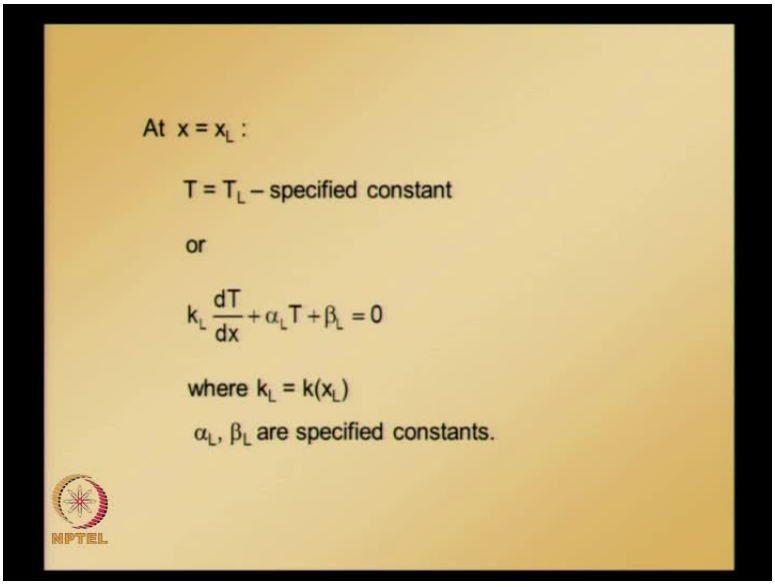
(Refer Slide Time: 04:06)



The problem statement is as follows, for which we are going to develop the element equations. This is the second order differential equation where K p Q are some known coefficient functions and T is field variables, some field variable. We need to solve the second order differential equation over the domain x going from or the lower bound of x is x naught and upper bound of x is x L. We have derived element equations using 2 node linear element for the same general one dimensional boundary problem earlier. The boundary conditions you need to solve.

This problem subjected to boundary conditions, the boundary conditions are as follows. Appropriate boundary conditions for the problem are of the following form: at x is equal to x naught either you can have essential boundary condition specified that is, T field variable; T is equal T naught a specified constant or a boundary condition which involves derivative first derivative of the field variable. That is here (Refer Slide Time: 04:45), k naught times derivative of T with respect x plus alpha naught times T plus beta naught is equal to 0. This you can easily using the thumb rule that we looked at in the earlier classes, you can easily check that the second boundary condition is natural boundary condition and here k naught is k evaluated; that is known coefficient k evaluated at x is equal to x naught and alpha naught beta naught are some specified constants.

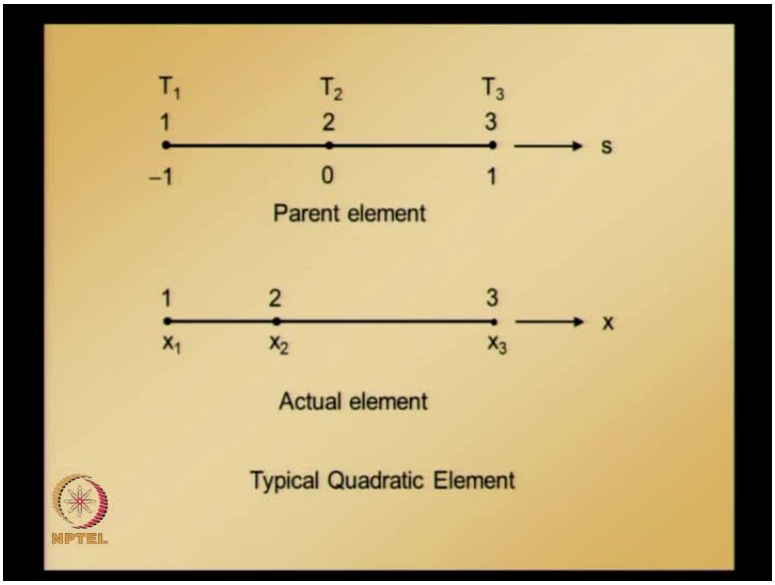
(Refer Slide Time: 05:30)



So, this is either of these boundary conditions can be specified at x is equal to x naught and let us see the boundary conditions at x is equal to x L. At x is equal to x L again, either essential boundary conditions can be prescribed or a natural boundary condition can be prescribed; where k L is k the known coefficient k evaluated at x is equal to x L and alpha L and beta L are some specified constants.

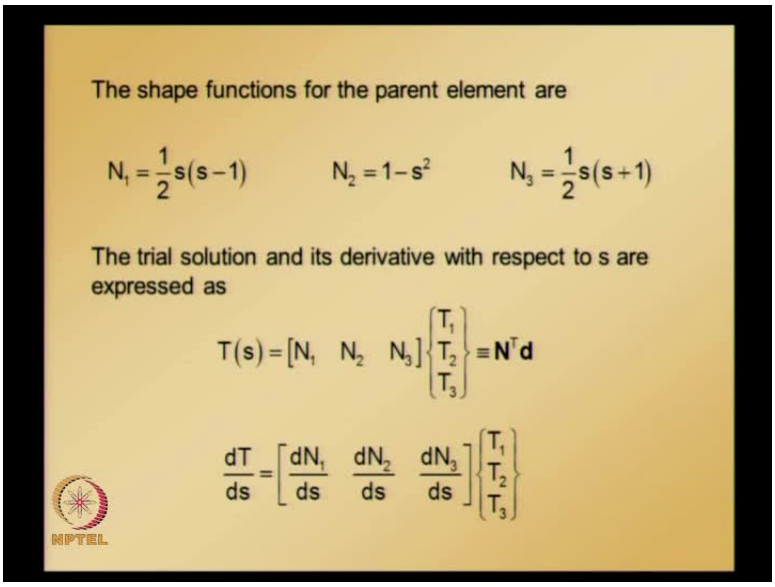
So, we need to develop quadratic isoparametric element equations for general one dimensional boundary value problem which is stated here subjected to this boundary conditions.

(Refer Slide Time: 06:18)



Now, let us look at a typical quadratic element; both parent element and actual element are shown in parent element. Node 1 is placed at s is equal to minus 1; node 2 is placed at s is equal to 0; node 3 is placed at s is equal to 1 and whereas, in the actual element x 1, x 2 and x 3 are the coordinates of nodes 1 and 2,… 1, 2, 3. Here, for mapping to be valid or for Jacobian to be positive - to remain positive everywhere over the entire element that is, s going from minus 1 to 1, the location of node 2 should satisfy the condition that we derived in last class; that it should not be closer than L over 4, where L is total length of the element. It should not be closer than L over 4 to either node 1 or node 3.

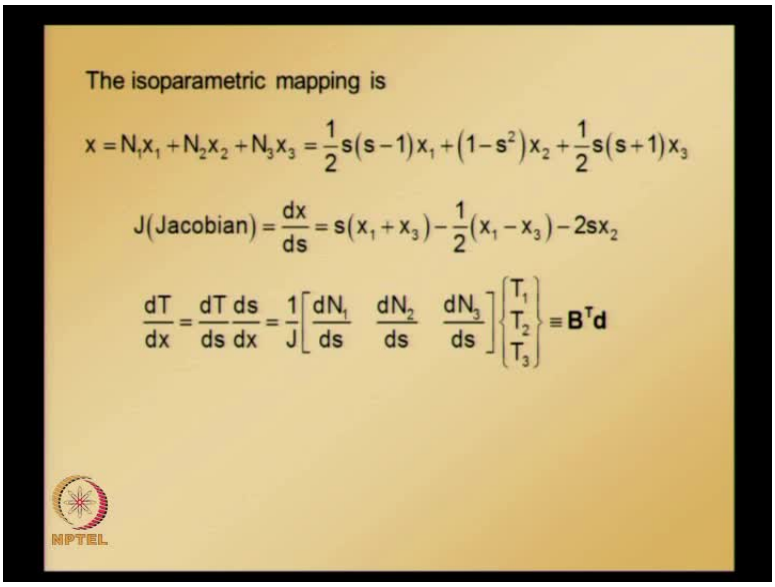
(Refer Slide Time: 07:45)



The shape functions for parent element also we have derived in the last class and they are reproduced here. Shape functions for parent element or these are the shape functions by substituting s is equal to 1, we can notice that N 1 is equal to 1 and N 2 N 3 are equal to 0. Similarly, similar kind of check can also be made by substituting s is equal to 0 in N 2 and also s is equal to minus 1 in N 3.

So, the trial solution and its derivative with respect to s can be written in this manner where N 1 N 2 N 3 are put in a matrix and T 1 T 2 T 3 are put in a vector and compactly, this can be written as N transpose d where d consists of all the nodal parameters.

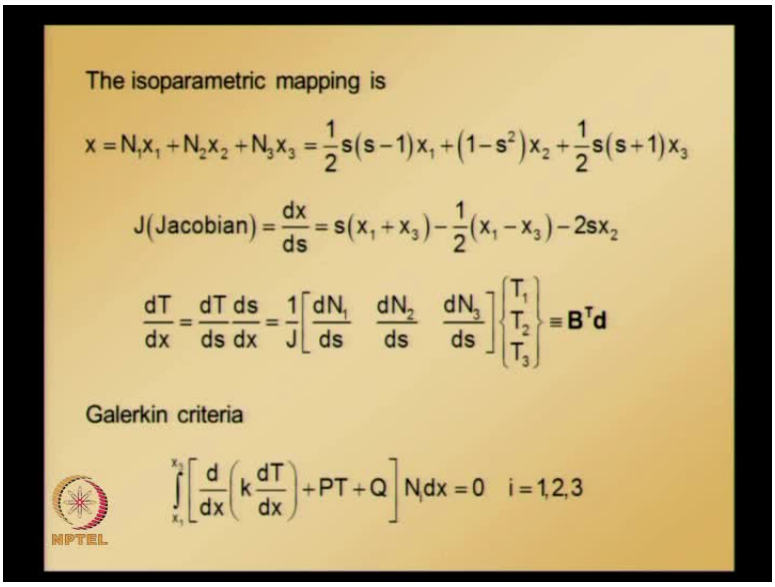
(Refer Slide Time: 09:04)



Now, derivative of T with respect to derivative of field variabile T with respect to s is given by this one and now isoparametric mapping x is equal to sigma N i x i. Or here, there are 3 nodes; so isoparametric mapping is x is equal to N 1 x 1 plus N 2 x 2 plus N 3 x 3. Substituting, N 1 N 2 N 3 which are shape functions of parent element, we get this one and once we have this relation which is relating x 2 s, we can easily calculate what is the derivative of d derivative of x with respect to s (Refer Slide Time: 09:35). That is d x over d s which is Jacobian and we require this Jacobian or inverse of Jacobian because, we require to calculate derivative of T with respect to first derivative of field variable with respect to x which can be written as d T d s times d s d T over d s times d s over d x. so, there d s over d x to get that we require this Jacobian which is d s over d x is nothing but 1 over J.

So, using this equation first derivative of field variable with respect to x, first derivative of T with respect to x can be calculated which can be compactly written as B transpose d where B is defined as 1 over J times derivative of first derivative of shape functions with respect to s. That is, d N 1 over d s, d N 2 over d s, d N 3 over d s. Now, we have the field variable expressed in terms of nodal values or nodal parameters and also its first derivative. So now, we can use the Galerkin criteria to derive the element equations similar to what we have done in the earlier cases.

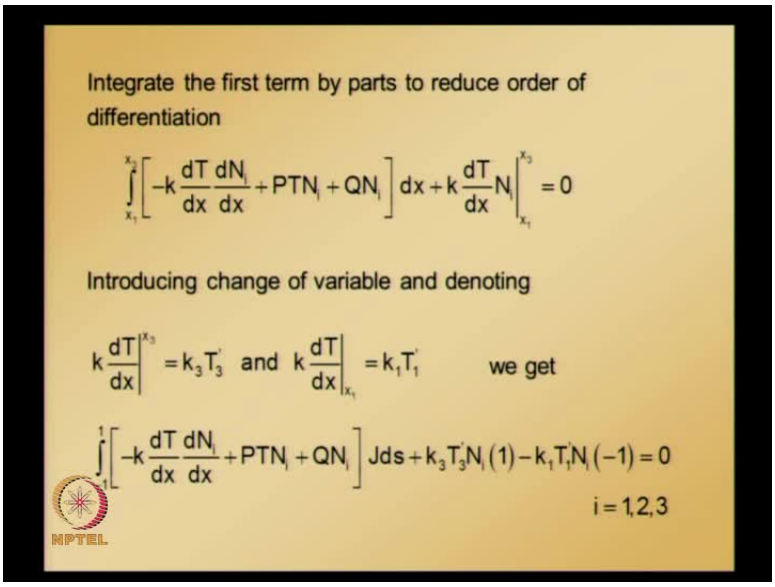
(Refer Slide Time: 11:33)



So, Galerkin criteria given differential equation, we need to multiply with weight function integrate over the problem domain and equated to 0 and here, when we are use adopting finite element weight function is same as shape function.

So, multiply the given differential equation with the shape function; integrate over the problem domain. Assuming the problem domain is going from x 1 to x 2 integrate the limits of integration are going to be x 1 x 2 and since, there are 3 nodes here I takes values 1 2 3 and expanding this or before doing that we can notice the first term inside the integral is second order derivative, is involved in the term. So, we can apply integration by parts; integrate the first term by parts to reduce the order of differentiation we get this.

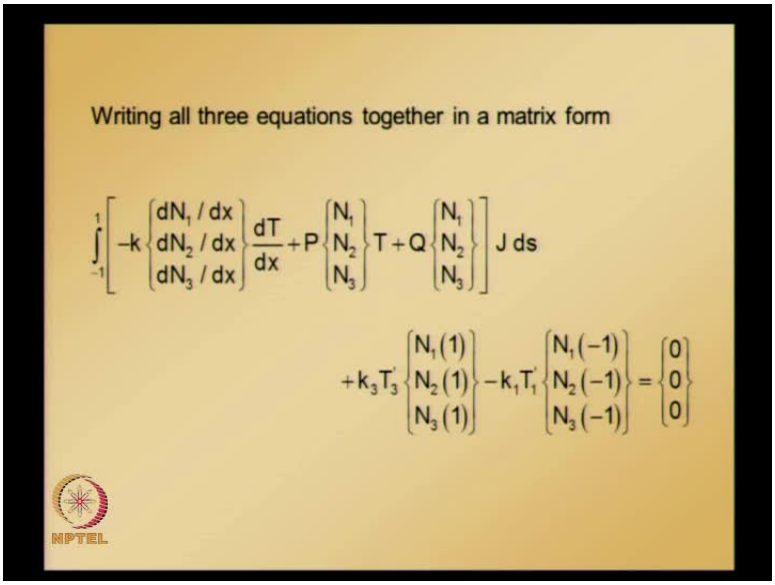
(Refer Slide Time: 12:35)



We have seen this kind of procedure earlier and we are solving problems using Galerkin criteria. So, same procedure you can adopt to do this integration by parts. Now, introducing change of variables because we need to change the limits of integration from x 1 to x 2 to minus 1 to 1.

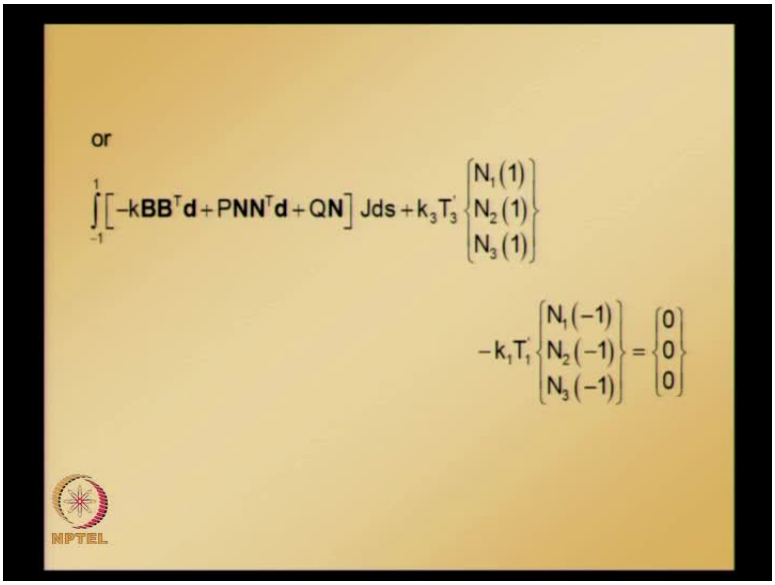
So, introducing change of variables and denoting k times first derivative of T with respect to x evaluated x at x 3 is k 3 T 3 prime and k times first derivative of T with respect to x evaluated at x is equal to x 1 is equal to k 1 times T 1 prime and then we are going to get the equation. After introducing the change of variables and using this notation, the previous equation can be written in this manner. Where d x is replaced with J d s and then rest of the terms follows. Once we substitute the first derivative of T with respect to x times k evaluated at x 1 and x 3 substituting the corresponding values we are going to get this.

(Refer Slide Time: 14:37)



Now, I take values 1 2 3; so, expanding this, that is writing all the equations all 3 equations together in a matrix form, we are going to get this. Please note that here shape function N 1 is going to be 1 when it is evaluated at s is equal to minus 1 and shape function N 3 is going to be 1 when it is evaluated at s is equal to 1 and N 2 is going to be 0 for s is equal to minus both for s is equal to minus 1 and s is equal to 1.

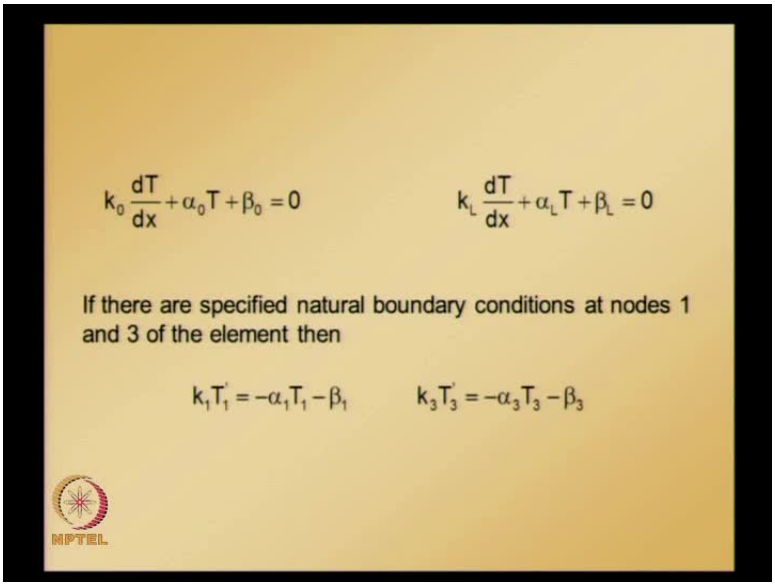
(Refer Slide Time: 15:26)



So noting all these things, we can further simplify this and we get this one where definition of B B or B transpose is used to write this. The previous equation in a compact

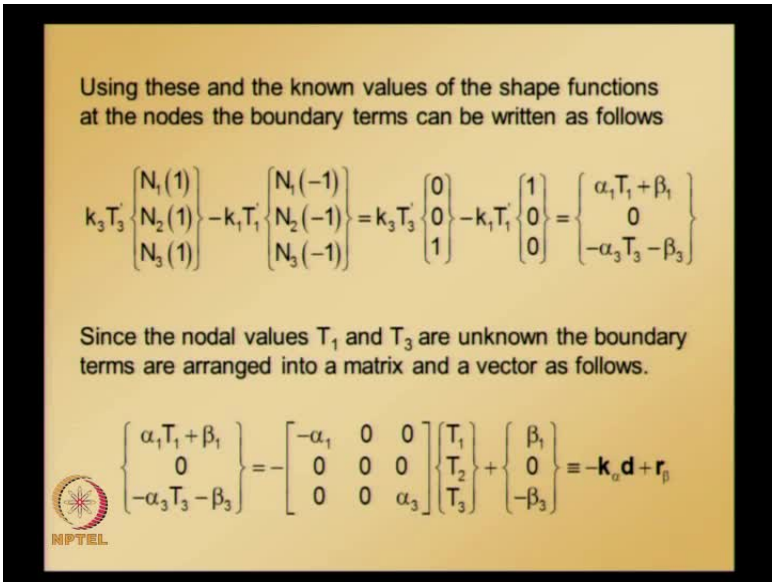
manner like this and from the natural boundary conditions that are given, we can further write k 3 times T 3 prime and k 1 times T 1 prime in terms of alphas and betas.

(Refer Slide Time: 16:09)



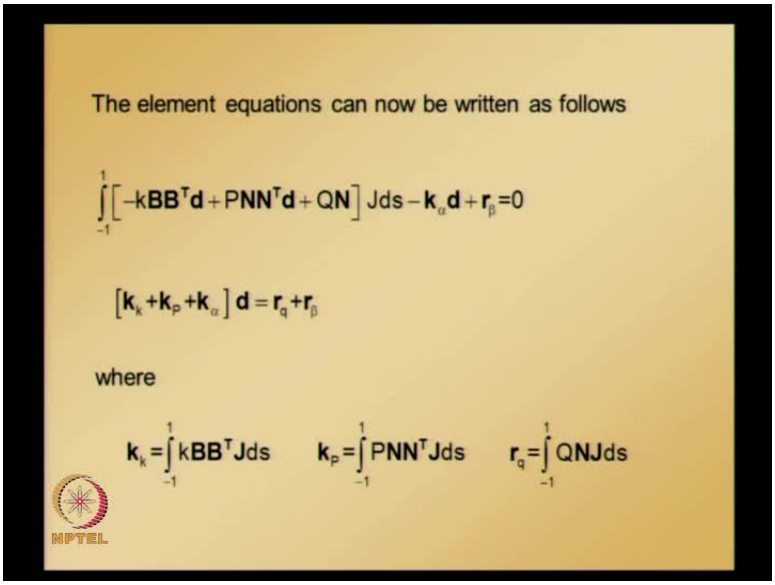
These are the boundary conditions; natural boundary conditions that are given and the element is chosen such a way that the lower limit is x 1 and upper limit is x 3. So, if the specified these boundary conditions are specified at nodes 1 and 3 of element then, these equations can be written in this manner; k 1 times T 1 prime is equal to alpha 1 times T 1 minus beta 1. Similarly, k 3 times T 3 prime is equal to alpha 3 T 3 minus beta 3.

(Refer Slide Time: 17:17)



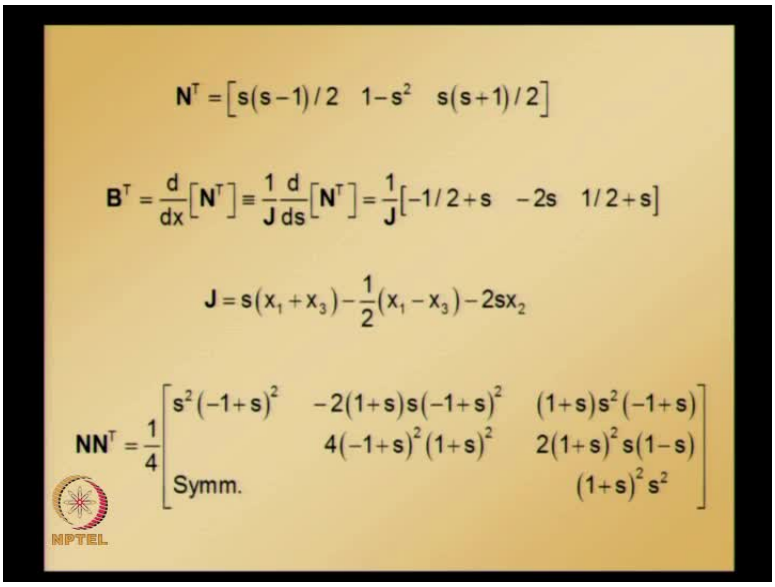
Substituting these into the previous equation using this and the known values of shape functions at nodes, the boundary terms can be written in the following manner and which can be again rearranged. Since, nodal values T 1 and T 3 are known are unknown the boundary terms are arranged into a matrix and vector form as given here. So, this alpha 1 times T 1 plus beta 1 0 minus alpha 3 times T 3 minus beta 3; this vector can be written as minus k alpha times d plus r beta. So substituting this the element equations can be written in this manner and defining further defining K k K p and k alpha.

(Refer Slide Time: 18:28)



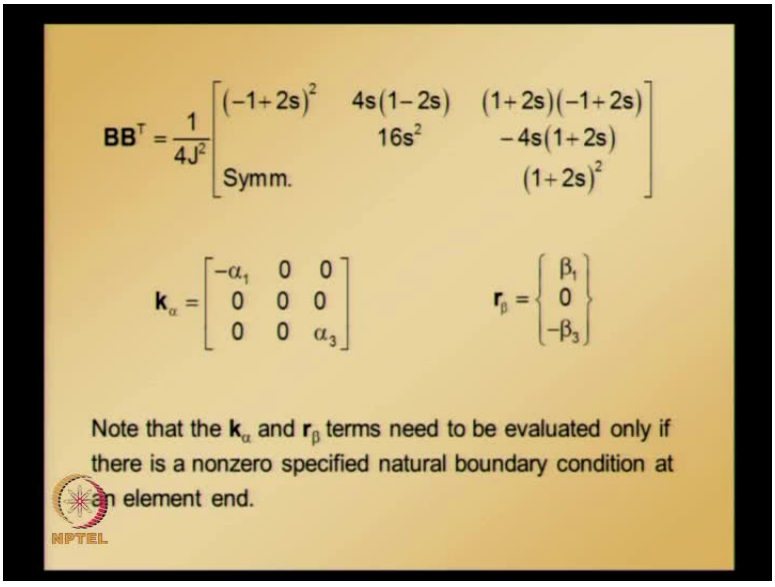
The equation can be written as: K k plus K p plus k alpha times d is equal to r q plus r beta, where each of the terms are defined like this. You can see here in K k you have B trans B B transpose and K p N N transpose is present.

(Refer Slide Time: 19:40)



So we will further simplify and try to figure out what B B transpose is and N N transpose is because, we require those to perform this integration from minus 1 to 1. And, N transpose is defined like this and B transpose is nothing but derivative of shape function with respect to x is defined like this, where J Jacobian is obtained once we know the nodal coordinates of all the 3 nodes and now N N transpose is given by this one. Similarly, B B transpose can be computed and also we have this k alpha and r beta.

(Refer Slide Time: 20:30)



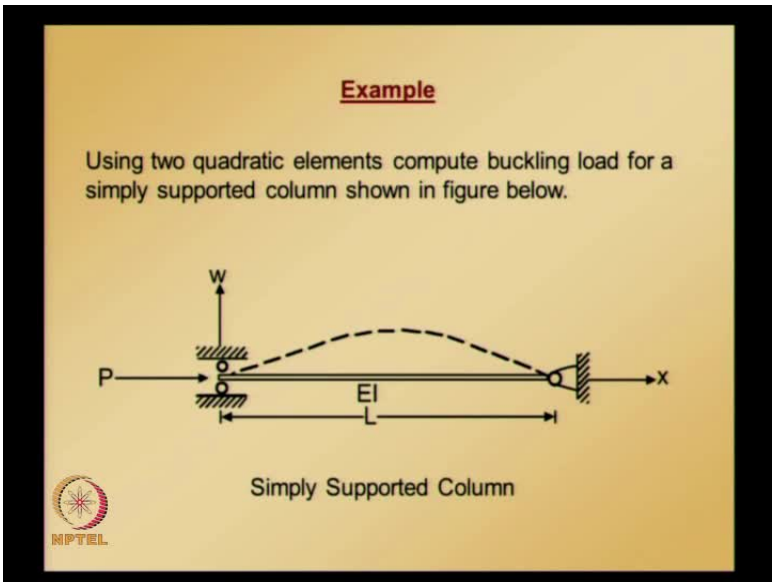
So, once we know all these values we can easily write the element equations for a quadratic element for this particular general one dimensional boundary value problem. And, please note that here, this k alpha and r beta shows up only if non-zero specified natural boundary conditions are present. Note that k alpha and r beta terms needs to be evaluated only if there is a non-zero specified natural boundary condition at an element end.

Suppose, if at node 1 natural boundary condition is specified and at node 3 essential boundary condition is specified then, alpha 1 is going to be a non-zero value whereas alpha 3 is going to be 0. Similarly, beta 1 is going to be non-zero value and beta 3 is going to be 0 and vice versa. If natural boundary condition is specified at node 3 and essential boundary condition is specified at node 1 then, alpha 1 is going to be 0 and beta 1 is going to be 0; alpha 3 and beta 3 are going to be non-zero values which we need to figure out from the given natural boundary condition.

So, now we have the element equations for a quadratic element for a general one dimensional boundary value problem. So, now we can look in applications of this and what we will be doing is, we will solve the same column buckling problem that we have see solved using the element equations corresponding to linear element.

We will solve the same problem now using quadratic element. If you recall, when you used linear elements, when we discretize the domain using 4 linear elements, we notice that the solution that we obtain has are the critical buckling load. Critical buckling load that we obtained as a solution using 4 linear elements for column buckling problem; we notice that there is about 5 percent error.

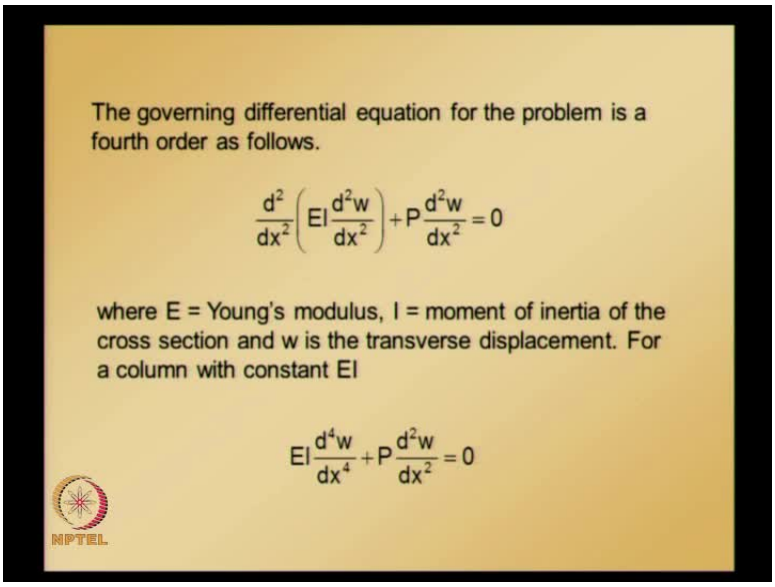
(Refer Slide Time: 23:25)



Now, let us see how can we, how much improvement we get using this quadratic element for the same number of nodes for the entire domain? So, here we will be solving the same problem using two quadratic elements compute buckling for simply supported column shown in figure below. We will be using same discretization or we will be using same number of nodes for discretization. When we are solving this problem using linear elements, we used 5 nodes. So, we will be using 5 nodes here also, except that we will be using two quadratic elements and see how solution accuracy improves.

So, before we proceed, just we will briefly review what are the boundary conditions and how this, the governing differential equation corresponding to this problem can be brought into the form of general one dimensional boundary value problem for which we develop the element equations.

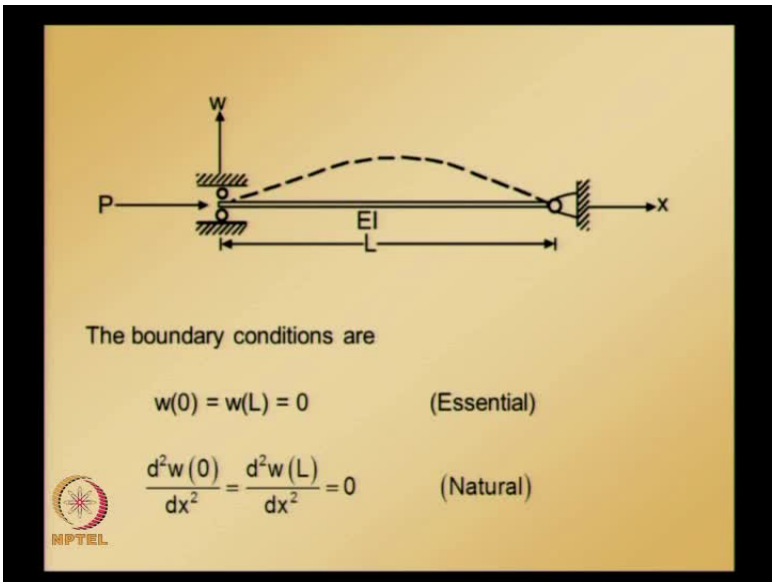
(Refer Slide Time: 24:35)



So, the governing differential equation for this simply supported column buckling problem, is a fourth order differential equation subjected to the boundary conditions. Here E is Young's modulus, I is moment of inertia of cross section, w is transverse displacement and for a column with constant EI.

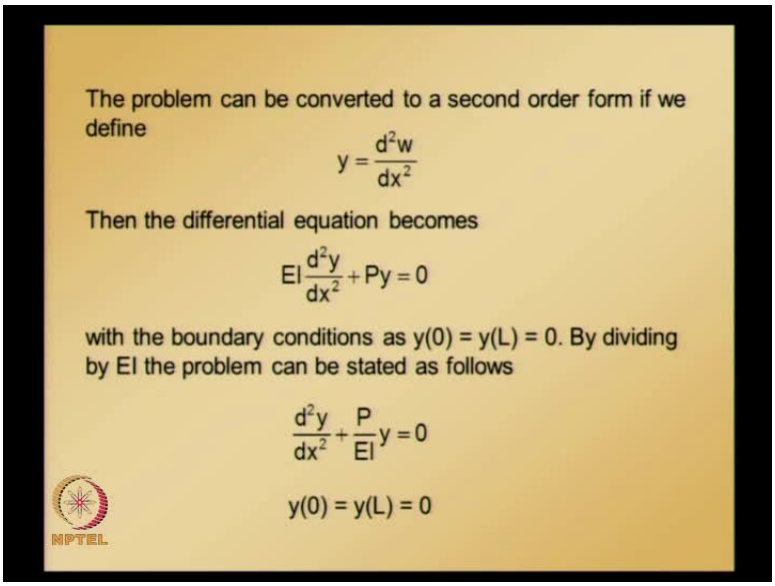
The previous equation can be written in this manner because, EI is constant. We can take it out of the integral and we get this equation and this is governing differential equation which needs to be satisfied over the entire length of column that is x going from 0 to L. Now, let us look at what are the boundary conditions. Since, we are dealing with simply supported column transverse displacement at both ends is going to be 0 and also curvature at both ends is equal to 0.

(Refer Slide Time: 25:57)



So, that can be written mathematically like this. and, it can be easily verified that the first set of boundary conditions are essential. Second set of boundary conditions are natural. This first set of boundary conditions are zeroth order equations. Second set of boundary conditions are second order equations. But, the this problem simply supported column buckling problem is not directly going to fall under the category of general one dimensional boundary value problem. We need to make some substitutions similar to what we did earlier.

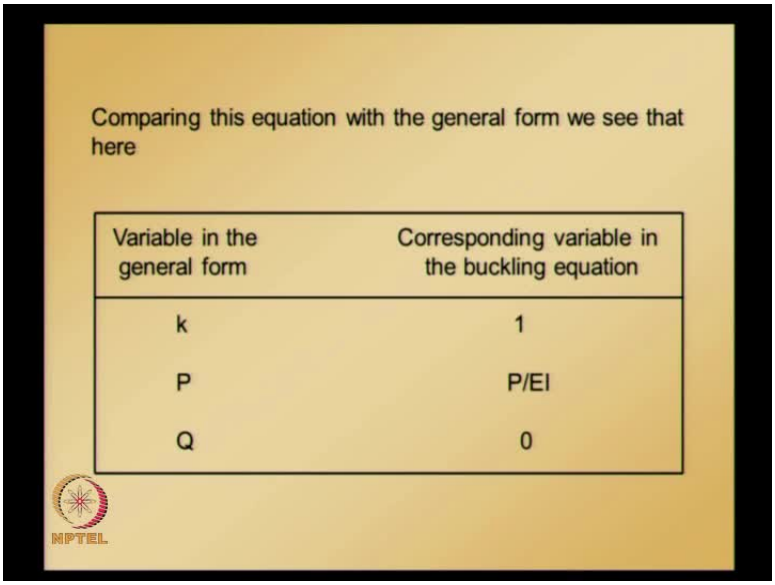
(Refer Slide Time: 26:35)



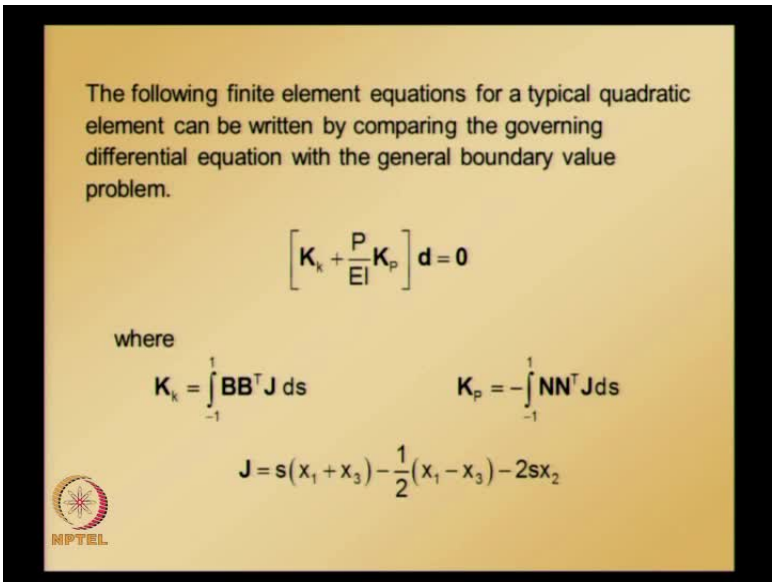
So, problem can be converted into second order form. If we define y is equal to second derivative of w with respect x square because, the general one dimensional boundary value problem that we for which we derived element equations, is a second order differential equation; somehow, we need to bring this a fourth order differential equation into the second order differential equation so that, we can write the element equations for this column buckling problem. So, during the substitution the given differential equation can be rewritten in this manner and also boundary conditions gets modified or this differential equations can also be rearranged and written in this manner where boundary conditions are also indicated. y is equal to 0 both at x is equal to 0 and x is equal to x L.

So now, we have the column buckling problem in the same form as general one dimensional boundary value problem.

(Refer Slide Time: 28:05)



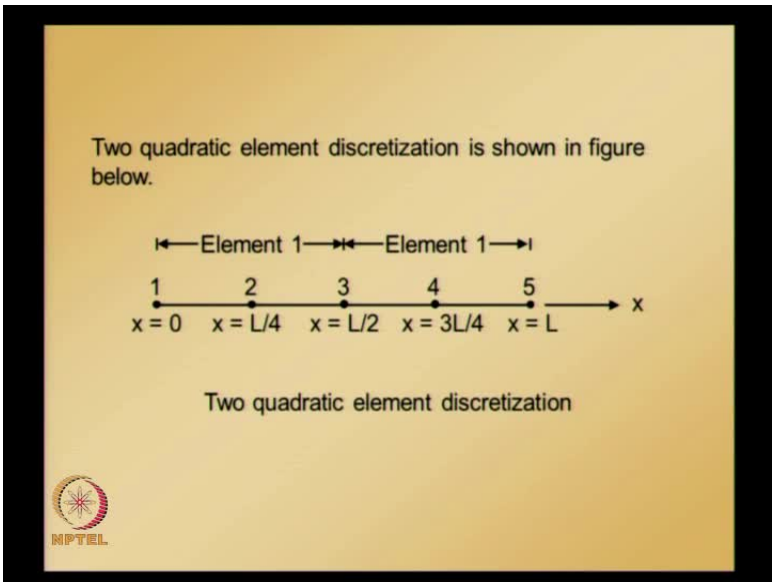
(Refer Slide Time: 28:28)



Before we proceed further, we can make a comparison of both equations. Comparing this equation with general form, we see that the corresponding variables we can identify and once we identify the corresponding variables, we can write the element equations. The following finite element equations for a typical quadratic element, can be written by comparing the governing differential equation with general boundary value problem.

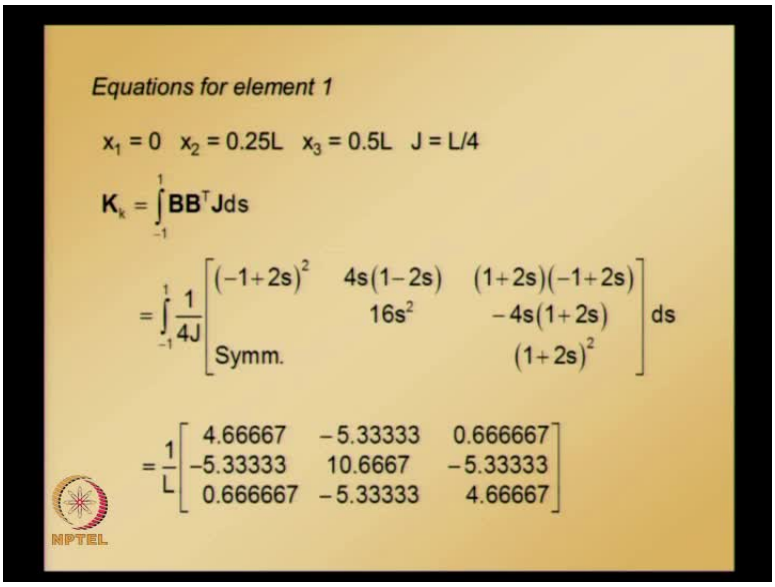
So, that is what we did and these are going to be the element equations and where K k and K p and J are defined. These are the element equations; so once, we decide the discretization for a particular problem and once we have the nodal coordinates, coordinate values available, we can go to each element and get these quantities K k K p from where we can easily write the element equations.

(Refer Slide Time: 29:43)



So, now let us look at the column buckling problem. Two quadratic element discretization for this column buckling problem which is for which the domain is going from 0 to L. The domain 0 to L is discretized using two quadratic elements or total 5 nodes and all nodes are equally are equispaced. Element 1 comprises of nodes 1 2 3; element 2 comprises of nodes 3 4 5 and there is a mistake element 2 2 instead of 2 1 is printed. So, element two comprises of nodes 3 4 5; element 1 comprises of nodes 1 2 3 and you can notice that for both elements 1 and 2, lengths are same and also the position of nodes locally is as the same distance from each other. So, element equations for element 1 and element 2 are going to be same, since EI is constant.

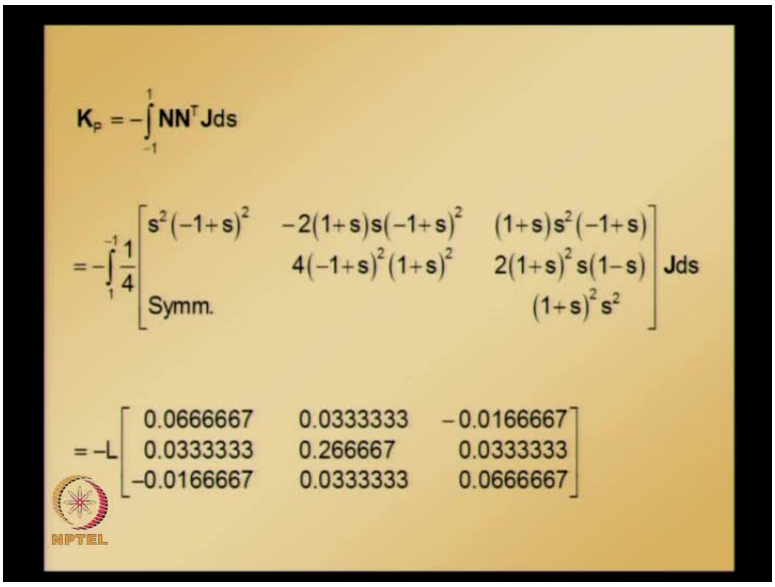
(Refer Slide Time: 31:13)



Now, let us see element equations for element 1. Noting down the coordinates of 3 nodes and length of this element is L over 2 and once we have these coordinates x 1 x 2 x 3 we can easily calculate J Jacobian value. You can see here Jacobian for this nodal spacing and for this 3 node element is L over 4 which is a constant and which helps us. Or, in a way, it makes our job easier when we are doing integration because, J is constant which we will realize or which you will notice when we are actually trying to integrate to evaluate K k. K k by definition is integral minus 1 to 1 B B transpose J d s, since, J is constant, happens to be constant for this particular element a in which the nodes are located at the locations 0.25 L and 0.5 L.

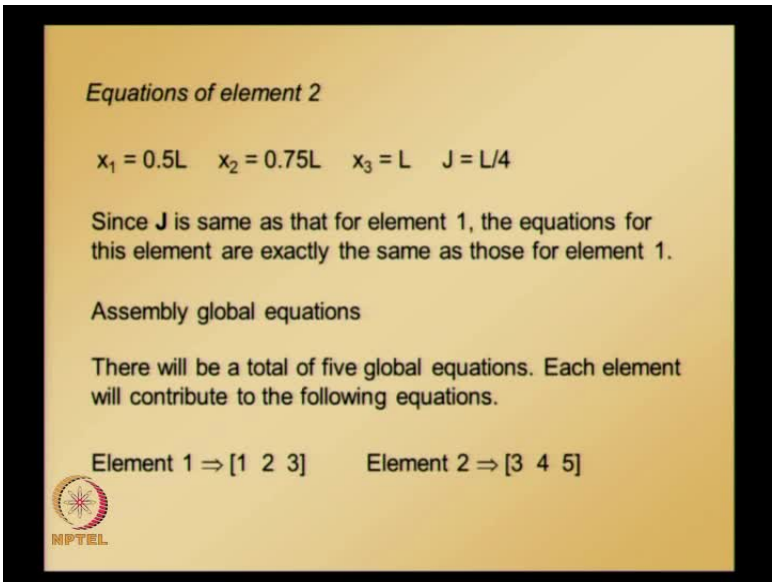
So, since J is constant integration become simpler substituting B B transpose and also J value, this can be further simplified by performing integration.

(Refer Slide Time: 33:27)



We get K k as this one now. We need to assemble or we need to compute what is K p. K p is integral minus 1 to 1 N N transpose J d s. Substituting the values of shape functions vector of shape functions, we get this J is again L over 4 and substituting J value and further simplifying by integrating we get this. So here these integral or evaluation of this integral becomes simpler in one way because, J is constant. If J is not constant then the integrand is going to be more complicated; then sometimes to evaluate this integrals is going to be complicated where we will be adopting numerical integration techniques which we will be discussing later.

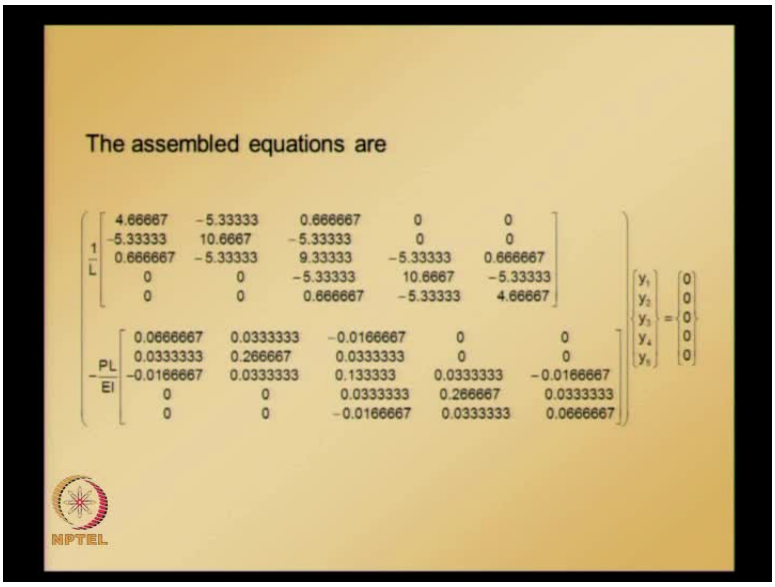
(Refer Slide Time: 34:44)



So, these are the element equations for element 1. We obtained K k and K p and now, element equations for element 2 are going to be similar. Since the nodal spacing is same and also material properties are constant for both elements, since J is same as that for element 1, equations for this element are exactly the same as those for element 1.

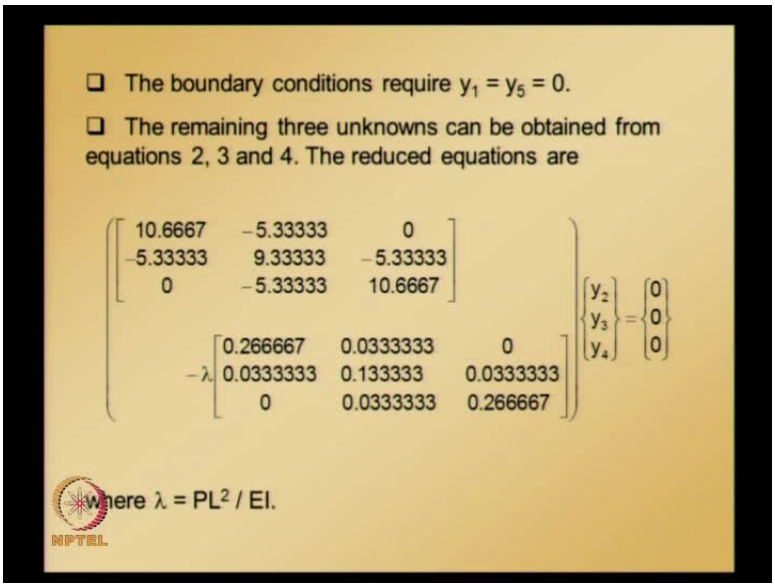
So, we got element equations for both elements: element 1 and element 2. Now, we are ready to assemble the global equations, assembly of global equations. There, will be a total of 5 global equations because, there are 5 nodes. Each element will contribute to the following equations. Element 1 contribution goes into 1 2 3 locations of rows and column locations of global equation system and element 2 contribution goes into 3 4 5 rows and column locations of the global equation system.

(Refer Slide Time: 35:59)



So, with that understanding since we already have element equations for element 1 and 2, we can write assembled global equations in this manner. We see that y 1 and y 5 correspond to x is equal to 0 and x is equal to L, respectively. So, y 1 and y 5 both are going to be 0.

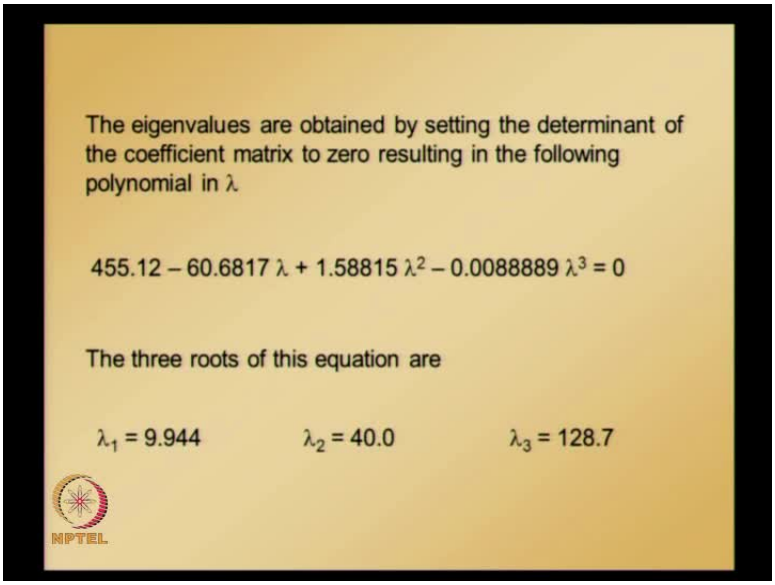
(Refer Slide Time: 36:35)



So, these are the boundary conditions y 1 y 5 is equal to 0 because, essential boundary conditions for this problem or y evaluated at x is equal to 0 0, y evaluated at x is equal to L is 0.

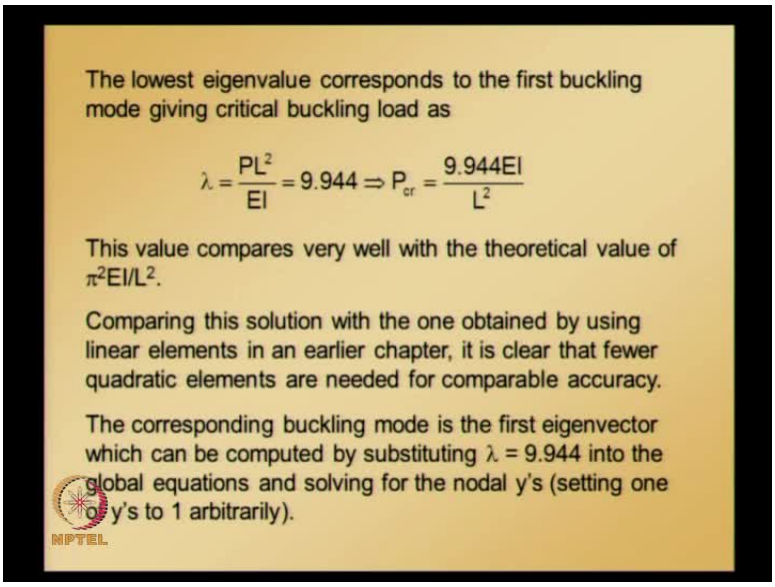
So y 1 y 5 are 0; so, in the previous global equation system since y 1 y 5 are 0, we can eliminate rows and columns corresponding to 1 and 5 locations and we get reduced equations. Remaining 3 unknowns can be obtained from equations 2 3 4 in the reduced equations obtained by eliminating 1 and 5 rows and columns from the global equation system is this one. Here, for further writing, briefly we introduced a notation lambda is equal to PL square over EI and you can notice that this equation system is for an eigenvalue problem. For this equation system to have a non-trivial solution, the determinant of the matrix should be equal to 0 and by equating determinant of matrix equal to 0, we get the eigenvalues.

(Refer Slide Time: 38:34)



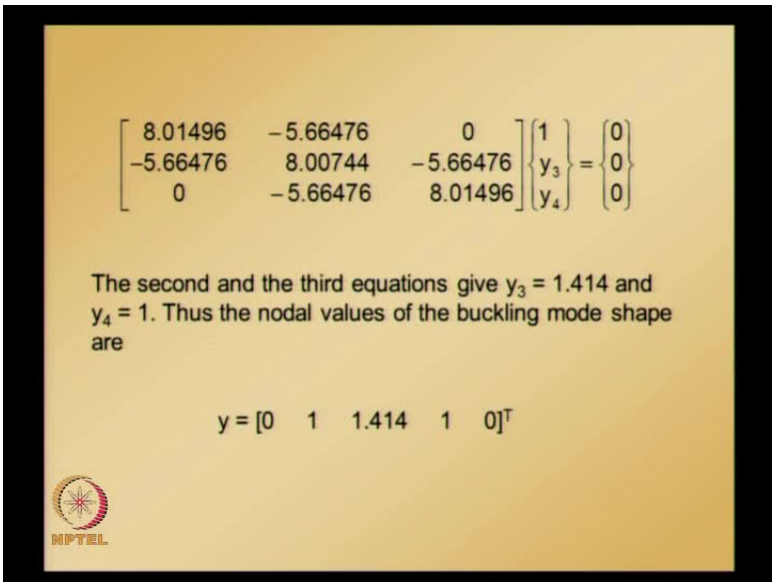
Eigenvalues are obtained by setting determinant of coefficient matrix to 0; resulting in following polynomial because the equation system is 3 by 3. We are going to get a polynomial of order 3 which has 3 routes. Solving for the 3 routes of this equation, we get lambda 1, lambda 2, lambda 3 and the lowest eigenvalue corresponds to the first buckling mode, as we have already discussed earlier when we are solving same problem using 2 node linear elements. So, by equating the lowest eigenvalue to the first buckling mode, we can obtain the critical buckling load.

(Refer Slide Time: 39:57)



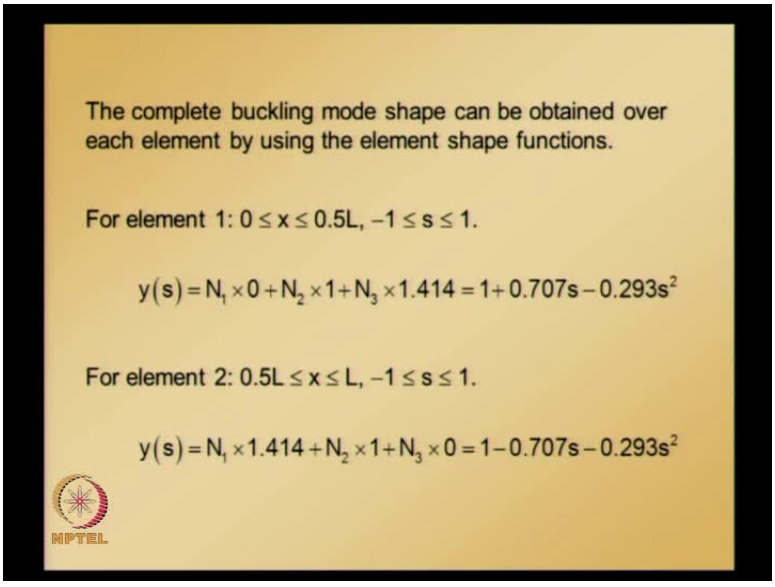
So, lambda 1 is equal to 9.944; that will be equated to PL square over EI. Back calculate what is P? We obtain P critical as 9.944 EI over L square and if you compare this with the theoretical value which is pi square EI over L square, it matches very well. This value compares very well with theoretical value of pi square EI over L square. Comparing this solution with the one obtained by using linear elements in the earlier lecture, it is clear that fewer quadratic elements are needed for comparable accuracy. The corresponding buckling mode we obtain buckling, sorry, eigenvalue. The corresponding buckling mode in the first, is the first eigenvector which can be computed by substituting lambda is equal to 9.944 into the global equations and solving for nodal y values. When we are solving for eigenvectors, we need to arbitrarily set one of the y values to 1 and solve for the other values. Similar, to the way we did for so when we solve the same problem using linear elements.

(Refer Slide Time: 41:29)



So, following the same procedure setting y 1 is equal 1 and solving for y 3 y 2 sorry, setting y 2 is equal to 0; y 2 is equal to 1. Solving for y 3 and y 4, we get the buckling mode shape. So, second and third equations give y 3 and y 4; thus, the nodal values of buckling mode shape are y 1 is equal to 0 which is essential boundary condition. y 2 is arbitrarily equated to 1 and we solved for y 3 and y 4 which happen to be 1.414 and 1 and y 5 is again 0.

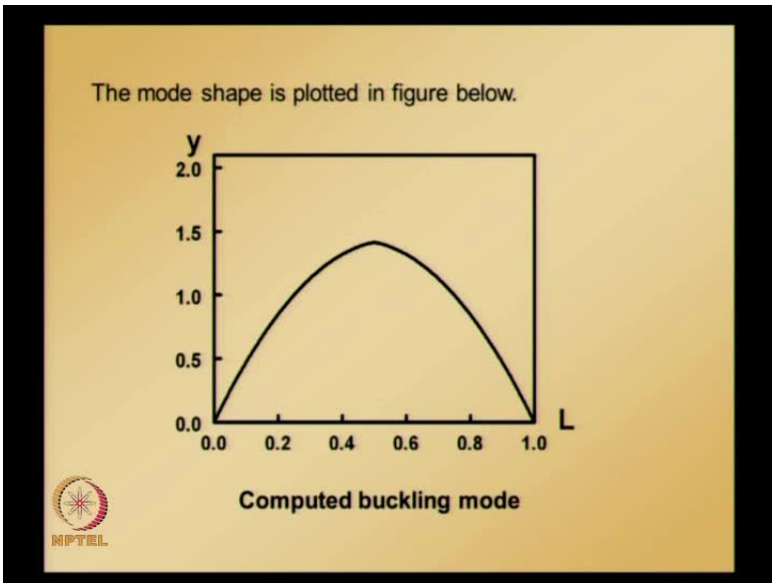
(Refer Slide Time: 42:33)



So, buckling mode shape is given by this nodal values and the complete buckling shape can be obtained over each element by using element shape functions. Once we got, once we get the nodal values that is y 1 y 2 y 3 y 4 y 5, we can do interpolation using finite element shape functions and get the complete buckling mode shape.

So, for element 1 which is going from x is equal to 0 to x is equal to 0.5 L and s goes from minus 1 to 1. Substituting, N 1 y 1 N 2 y 2 N 3 y 3 we get this equation which helps us to plot the mode shape of, for element 1 and similarly, for element 2 x goes from x x goes from 0.5 L to L s goes from minus 1 to 1. Substituting, the shape functions N 1 y 3 N 2 y 4 N 3 y 5 we get this equation.

(Refer Slide Time: 43:56)

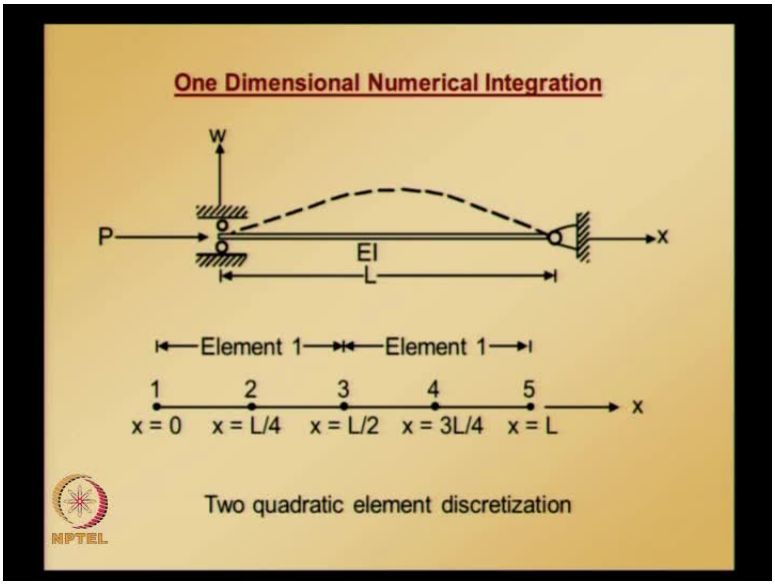


Once we have these two equations which represent how the mode shape is changing over element 1 and element 2, we can plot this and get the mode shape for the entire domain going from 0 to L.

So, now we have seen one application of this general one dimensional boundary value problem which is column buckling problem. You have noticed that the evaluation of these integrals when we are solving this problem becomes simpler because, J happen to be a constant J is not a constant then the integrand become complicated and we need to evaluate these integrals using some numerical techniques.

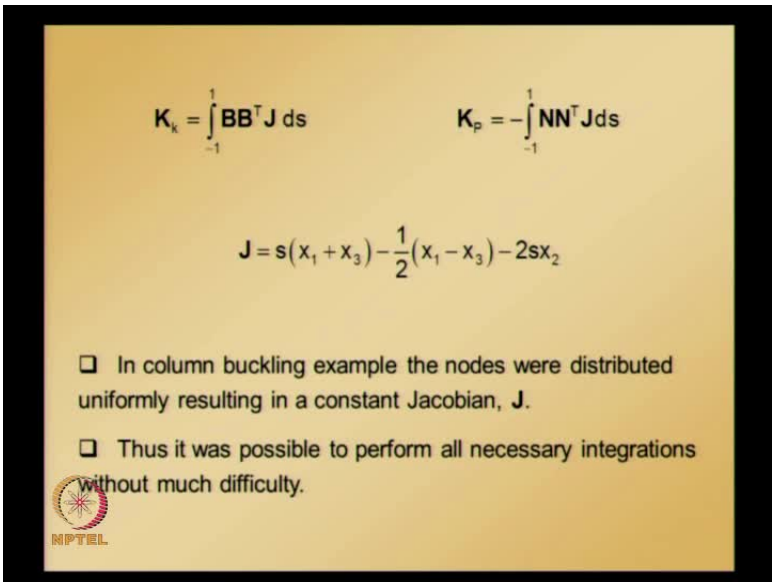
So, we will be discussing some numerical or one of the numerical techniques which is widely used in finite elements that is Gaussian quadrature for evaluating the integrals, so that we will be discussing now.

(Refer Slide Time: 45:20)

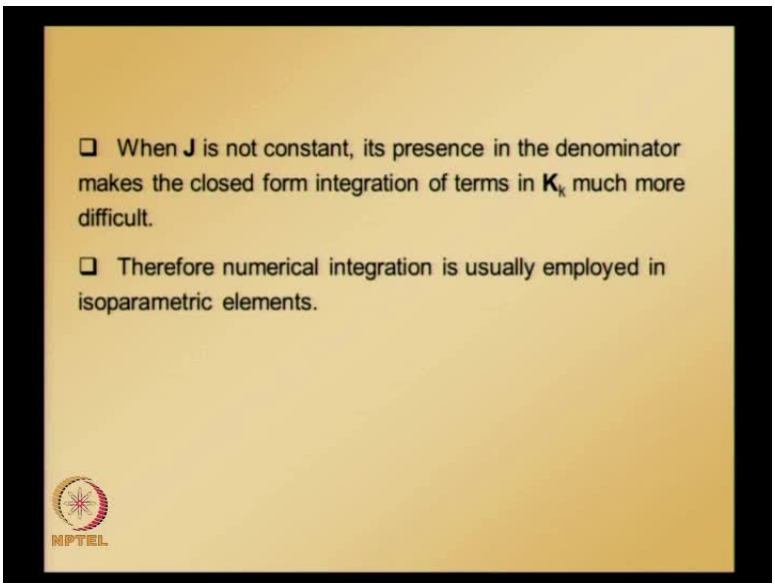


So, one dimensional numerical integration and the just I want to highlight the importance of this. This is the column buckling problem that we looked at and you can notice that the discretization involved 2 elements and nodes are locations of nodes are also shown for both elements 1 and 2. Before we solved for critical buckling load or node shapes, we require to evaluate for each of the elements of these integrals. For this particular (Refer Slide Time: 46:00), for the problem that we solved it happened that J is constant.

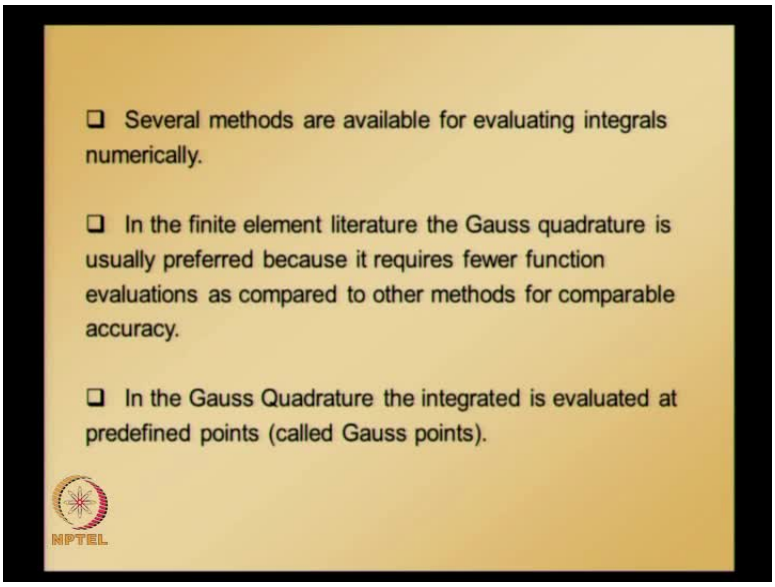
(Refer Slide Time: 45:45)



(Refer Slide Time: 46:30)



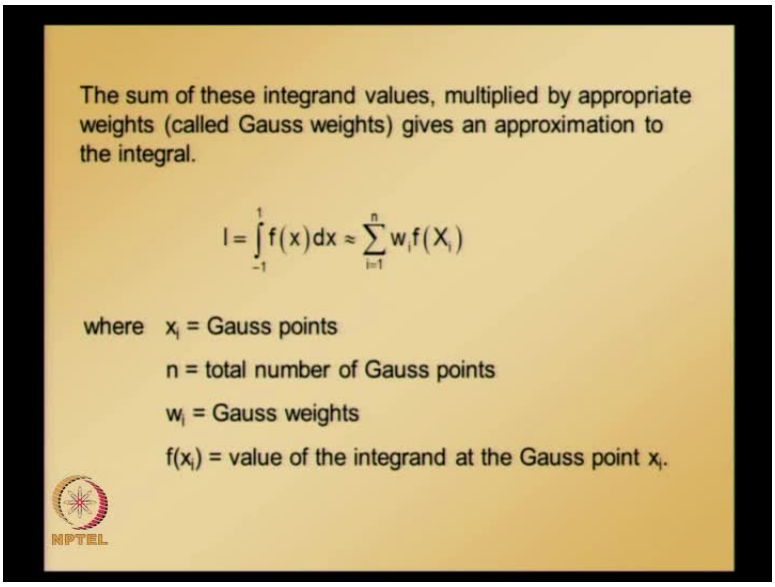
(Refer Slide Time: 46:56)



In the column buckling example, nodes are distributed uniformly resulting in constant Jacobian and thus, it was possible to perform all the necessary integrations without much difficulty. And, the problem arises when J is not constant or when the integrand becomes complicated when J is not constant. Its presence in the denominator makes closed form integration in terms of K k much more difficult. Therefore, numerical integration is usually employed in isoparametric elements there are several methods available for evaluating integrals numerically.

In finite element literature, the Gauss quadrature is usually performed because it requires few, it requires fewer function evaluations as compared to other methods for comparable accuracy.

(Refer Slide Time: 47:45)

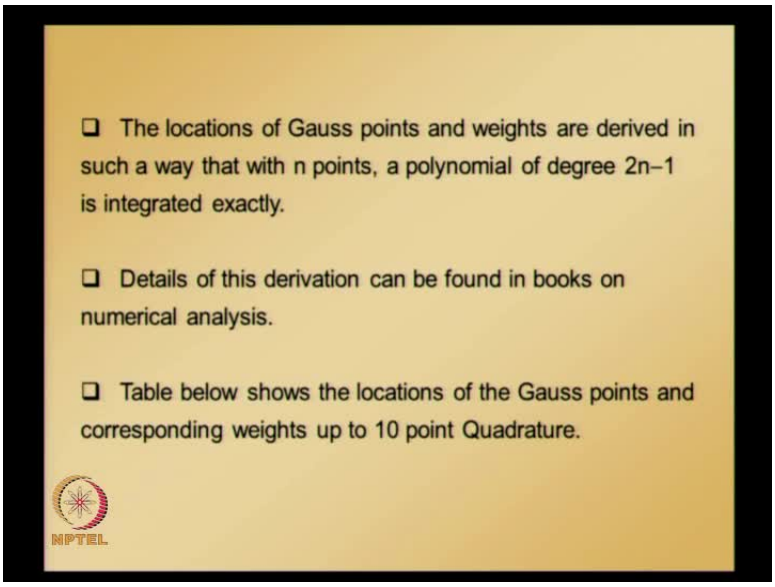


That is the reason why Gaussian quadrature is so popular in finite element method or in fact, in any other numerical technique like boundary element method. In Gaussian quadrature, the integrand is evaluated at predefined points called Gauss points and sum of these integrand values multiplied by appropriate weight called Gauss weight, gives an approximation to the integral.

So, for example, if a function f x needs to be integrated from minus 1 to 1, instead of doing this evaluating this integral in a closed form manner, we can evaluate this function f at some specific locations of x. Some of these values of function evaluated at some locations multiplied by some weights. If we sum it up, we are going to get, or the integral is going to be approximated by summing the function value evaluated at some specific points, multiplied by weight; and, only thing we require is to know where to evaluate this function.

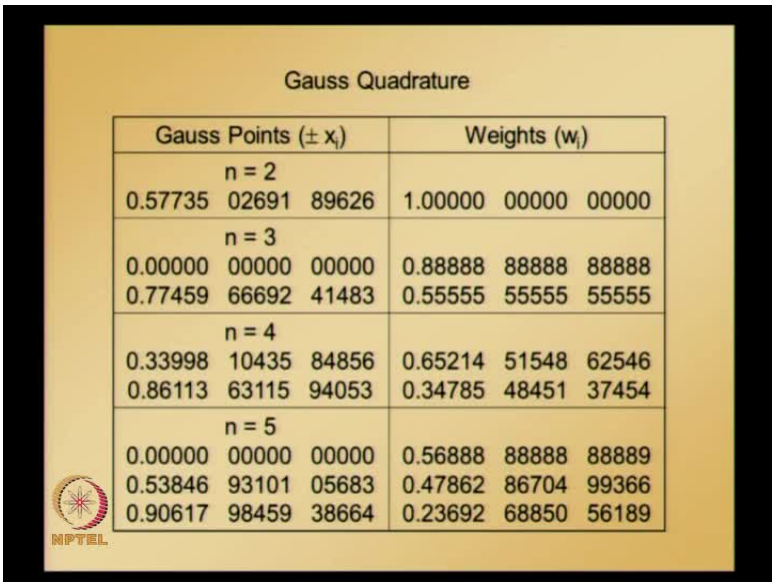
What is the corresponding weight we need to multiply with? Here, x i is the location where we need to evaluate function; is called Gauss point. Total number of N is total number of Gauss points; w is weight of Gauss point or Gauss weight; f x i is value of integrand whatever is there inside integral evaluated at Gauss point.

(Refer Slide Time: 49:37)



The locations of Gauss points and weights are derived in such a way that N points with N points a polynomial of degree 2 N minus 1 is integrated exactly. So, the locations values of Gauss points that is x i and w i a derived based on this condition that using N points we can or when we are deriving this. It automatically follows that if we adopt N points, we can integrate polynomial of degree 2 N minus 1 accurately.

(Refer Slide Time: 50:50)

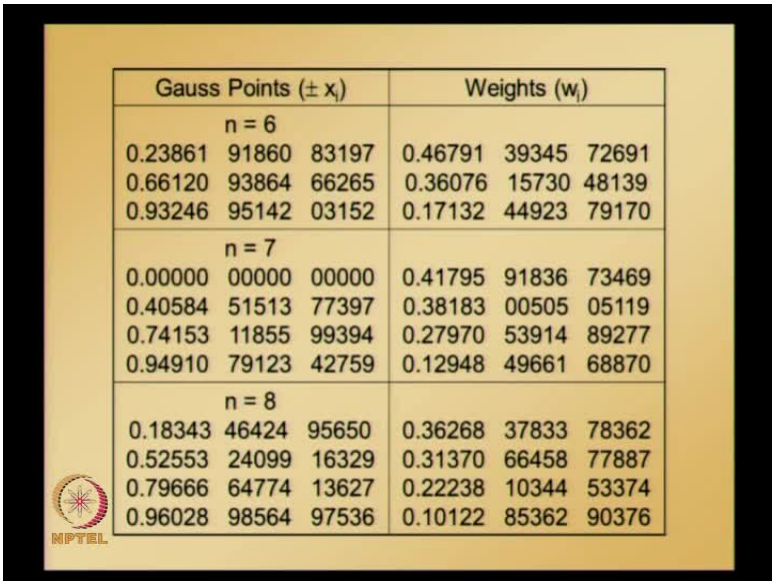


Gauss Points ($\pm x_i$)	Weights (w_i)
n = 2	
0.57735 02691 89626	1.00000 00000 00000
n = 3	
0.00000 00000 00000	0.88888 88888 88888
0.77459 66692 41483	0.55555 55555 55555
n = 4	
0.33998 10435 84856	0.65214 51548 62546
0.86113 63115 94053	0.34785 48451 37454
n = 5	
0.00000 00000 00000	0.56888 88888 88889
0.53846 93101 05683	0.47862 86704 99366
0.90617 98459 38664	0.23692 68850 56189

The details of this procedure - how to obtain this Gauss points and weights, can be found in text books on numerical analysis. Here, directly the locations of the Gauss points and

corresponding weights up to 10 points, 10 point quadrature are given in the table below. Here, Gauss points and weights are shown for N is equal to 1, N is equal to 2, N is equal to 3, N is equal to 4, N is equal to 5. For N is equal to 1, Gauss point is 0 and weight is equal to 2. So, that is not shown in the table and you can easily read the location or the position at which, when it evaluates the integrand from this table. You can notice that for both locations and weights, a space is there between the numbers which just gives us freedom. Or, it is up to the user to decide how many significant digits accuracy is required; depending on that we can select the rest of the digits from the table.

(Refer Slide Time: 52:06)



Here, N 2 to N 5 weights points and weights are shown and rest of the points and weights are shown in the next table N 6 N is equal to 6, N N is equal to 7 and N is equal to 8. Similarly, N is equal to 9 and N is equal to 10; the values of Gauss points and also weights are given and these values of Gauss points and weights can also be obtained using any of the commercial software like mat lab.

So, in the next class we will be looking at some examples through which we will demonstrate, how to evaluate integral using Gauss quadrature, using the values of Gauss points and weights given in this table.