Finite Element Method

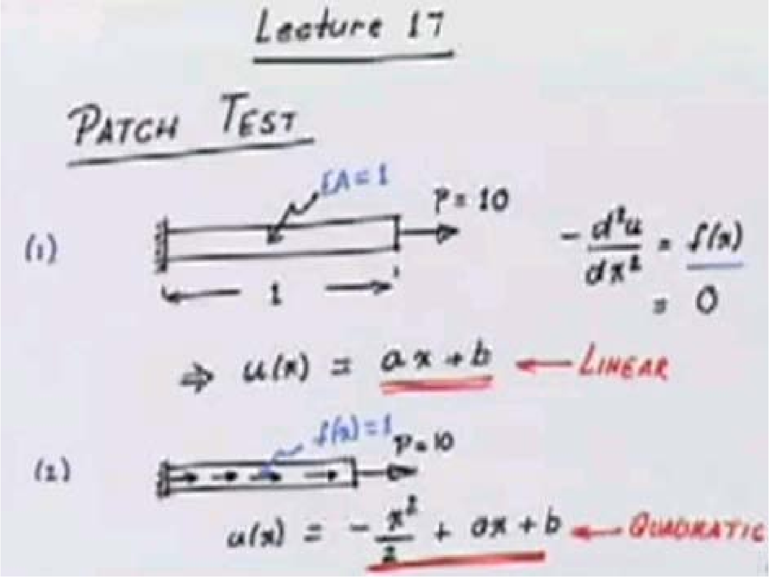
Prof C. S. Upadhyay
Department of Mechanical Engineering

Indian Institute of Technology, Kanpur

Module – 6 Lecture – 1

In this lecture, we are going to talk about some more things related to one dimensional finite element program. Till the last lecture, we had essentially talked about the finite element method in the context of the one dimensional problem and we had in a detailed way elaborated the various components that go into the one dimensional code.

(Refer Slide Time: 00:29)



Lecture 17

PATCH TEST

(1)

$$-\frac{d^2u}{dx^2} = f(x) = 0$$

$$\Rightarrow u(x) = ax + b \leftarrow \text{LINEAR}$$

(2)

$$u(x) = -\frac{x^2}{2} + ax + b \leftarrow \text{QUADRATIC}$$

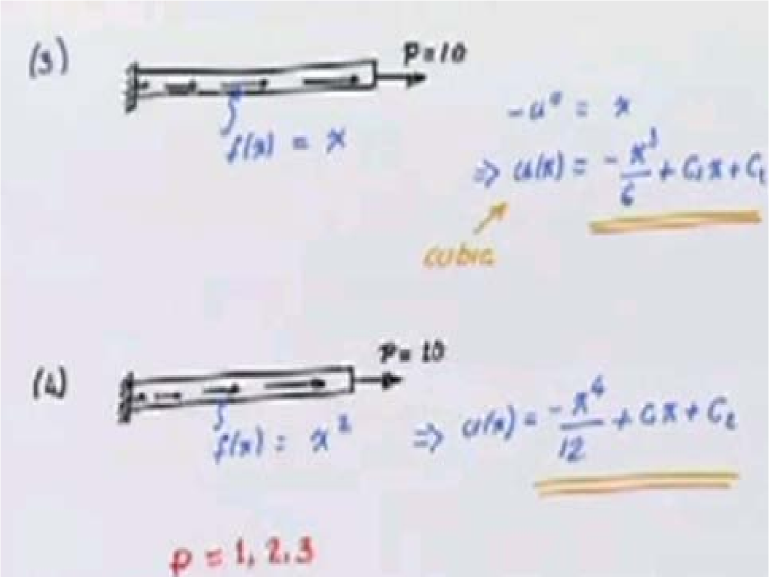
EA = 1
P = 10
1
f(x) = 1
P = 10

One has to put these components together and once they are put together, under the separate units called pre-processor, processor and post processor; then I have my complete finite element program. The question is that what is it that we would like to see with a finite element program? Certain things that we should check for right in the beginning before we start using this program to ensure that it is doing what it is supposed to do.

In a fleeting way it was mentioned in the last lecture, that we can do something called a patch test to check the one dimensional program. There are various definitions of this patch test given in various books. What we are going to do is something very simple. What is the basic idea? As I had mentioned let us take this problem (Refer Slide Time: 01:55). This is a bar again of length 1. Let EA be equal to 1 and currently I am applying no distributed forces. In this case I know my differential equation is going to be corresponding to the deformation of the bar. It is going to be –d square u dx square is equal to f (x) in the interval 0 to 1, f(x) for us is 0. This implies that for this case u(x) that is the exact solution will be equal to ax plus b. And from our initial condition that we have applied here the b would come out to be 0, but this is the representation. My exact solution is actually a linear. This problem I am going to call as problem 1 (Refer Slide Time: 03:25).

Similarly, what I would like to do is take problem 2. Here everything I keep the same as in the problem 1, but now I apply a uniformly distributed load f(x) is equal to1. When f(x) is equal to1 then if I look at this part on the right hand side (Refer Slide Time: 04:12) –d square u dx square is equal to1. u(x) here will come out to be minus x squared by 2 plus ax plus b. If you see the exact solution is a quadratic. I should write it we have the exact solution is a linear and here it is a quadratic.

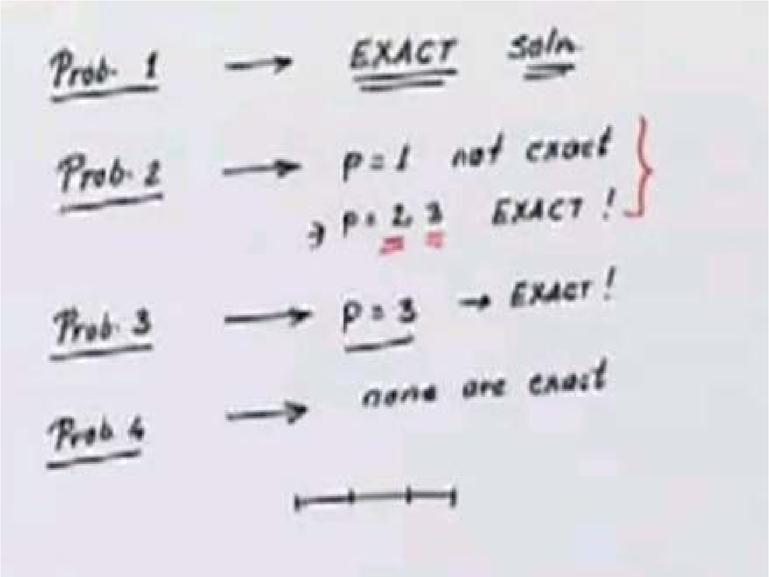
(Refer Slide Time: 05:08)


(3)
P=10
f(x) = x
$-u'' = x$
$\Rightarrow u(x) = -\frac{x^3}{6} + C_1x + C_2$
cubic
(4)
P= 10
f(x) : x^2
$\Rightarrow u(x) = -\frac{x^4}{12} + C_1x + C_2$
p = 1, 2, 3

Let us take the third problem. I keep everything fixed as problems 1 and 2, but here I am applying an f(x) which is a linear which is given by f(x) is equal to x. In this case I get minus u double prime is equal to x implies u of x is equal to minus x cube by six plus C_1 x plus C_2 where C_1 C_2 will come from our boundary conditions.

If you now look at this solution, this is now a cubic (Refer Slide Time: 06:04). I could also create a fourth problem; let me create it such that my f(x) is a quadratic, if my f(x) is quadratic implies my u(x) is going to be minus x to power 4 by 12 plus C_1 x plus $C_{2.}$ It is going to be a fourth order. Our program is going to be designed to have orders of approximation 1, 2 and 3. If I solve these problems now using orders of approximation 1, 2 and 3, here what we are doing? We are constructing problems for which we know the exact solutions and not only the exact solution; the exact solutions are polynomials of some order.

(Refer Slide Time: 07:49)



Prob. 1 → EXACT soln

Prob. 2 → p = 1 not exact

⇒ p = 2, 3 EXACT !

Prob. 3 → p = 3 → EXACT !

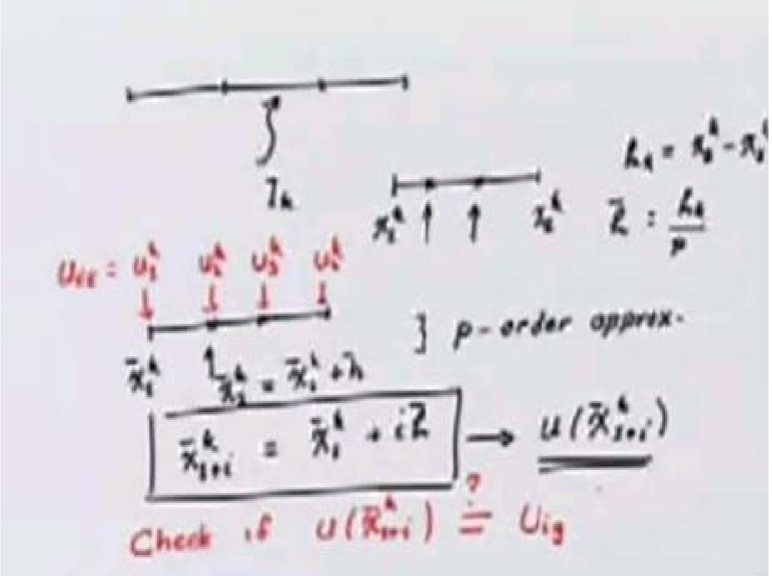
Prob. 4 → none are exact

If I now try to solve this problem using the finite element method using P is equal to1 2 3. For problem 1, all three should give me the exact solution, because my approximation is of order greater than or equal to 1, my exact solution is linear. I know from the completeness property of my approximating function I should be able to exactly represent the linear using this.

If I take problem 2; for P is equal to 1, it is not exact, but for P is equal to 2 and 3 it has to be exact. If I take for problem 3 which is why the previous problem is exact again because it is a quadratic polynomial and the quadratic polynomial should be exactly represented by our approximations of order 2 and 3 because of the completeness. Problem 3 the exact solution is a cubic. Here only for P is equal to 3 it is exact. For P is equal to 1 and 2 it is not going to be exact. If I ask you the following question what about problem 4? In problem 4, none of them are exact. Because the exact solution now is the fourth order polynomial and we are only able to approximate it using the linear quadratic and the cubic.

How are we going to approximate? You can take let us say three elements in the domain; three elements of size 1 by 3 1 by 3 and do the whole thing, but what we were going to see in the cases when it is not exact? If I refine the mesh that is, I add more and more points I should get closer and closer to the exact solution of the problem. For the polynomials, for the approximations, for this problem for this one we said that the solution is exact. How do you check whether that exactness is coming out in terms of the degrees of freedom that you obtain? If you plot you may get the plots very close and you can conquer or with kind of draw conclusion that you have obtained the exact solution. But we should be able to check it numerically upto the machine precision have been specified. How do we do?

(Refer Slide Time: 11:08)



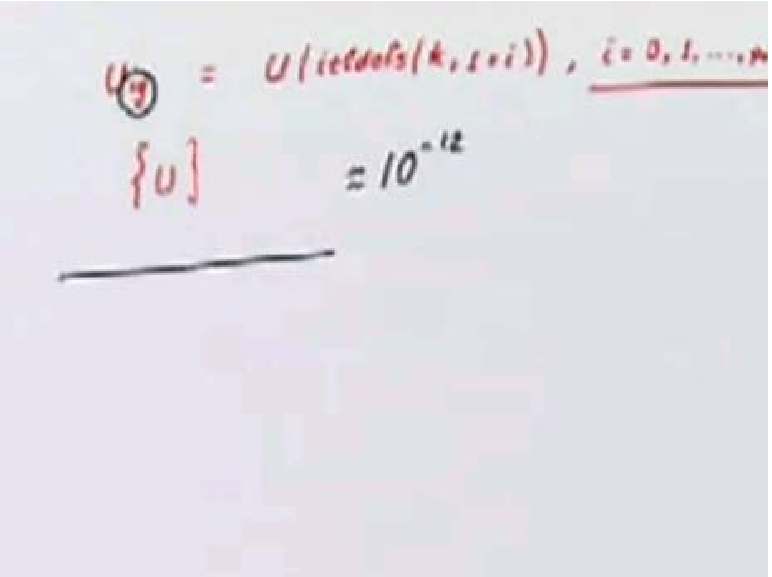
I_k
$h_k = x_2^k - x_1^k$
x_1^k
x_2^k
$\bar{h} = \frac{h_k}{p}$
$U_{ik} = u_1^k \quad u_2^k \quad u_3^k \quad u_4^k$
} p-order approx.
$\bar{x}_1^k$
$\bar{x}_2^k = \bar{x}_1^k + \bar{h}$
$\bar{x}_{1+i}^k = \bar{x}_1^k + i\bar{h}$
$\longrightarrow u(\bar{x}_{1+i}^k)$
Check if $u(\bar{x}_{1+i}^k) \overset{?}{=} U_{ig}$

Let us say that I have this 3 element mesh. What do I do? I take any element I_k for this element I know this is going to be my x_1 of k this is going to be my x_2 of k now given the order of approximation P, I am going to have h_k which is equal to x_2 of k - x_1 of k; this we have done many times over.

We will take this h bar equal to h_k divided by P. We will take this smaller p and now we are going to divide this into smaller pieces. We are going to add these additional points and you see how many points we are adding, essentially it will be h_k by P. We have taken these pieces and we will do this then what do we do further? We are going to do the following that is the element I have the first node I am going to call x_1 of k then give it another name x bar. Then the next node I am going to form not node a point x_2 bar k is equal to x_1 bar k plus h bar. Then I will find the next point and so on. That we have I am able to construct all the P plus 1 points in the element corresponding to a P order approximation. At these points now once I have the exact solution expression any generic point will have x_{1plusI} of k bar is equal to x_1 of k bar plus i h bar.

Given this points (Refer Slide Time: 13:40) I am going to feed them to a routine which returns me the exact solution. At these points, but we also know that these are the exact solution how do I compare these point values with my finite element solution, by the construction of shape functions we know that these points also correspond to U_1 of k U_2 of k U_3 of k U_4 of k and so on. That is these have the corresponding shape functions defined with respect to these points if I transform from the master to the physical and vice versa. We know that the degrees of freedom or the elements of the solution vector U that we have formed, they correspond to the value of the finite element solution at these points in the element. What are we going to check? We are going to check if U at x_1 bar 1 plus i, this thing is equal to question mark $U_{ig.}$

(Refer Slide Time: 15:16)



$U_{ig} = U(\text{ieldofs}(k, 1+i))$, $i = 0, 1, \ldots, p$

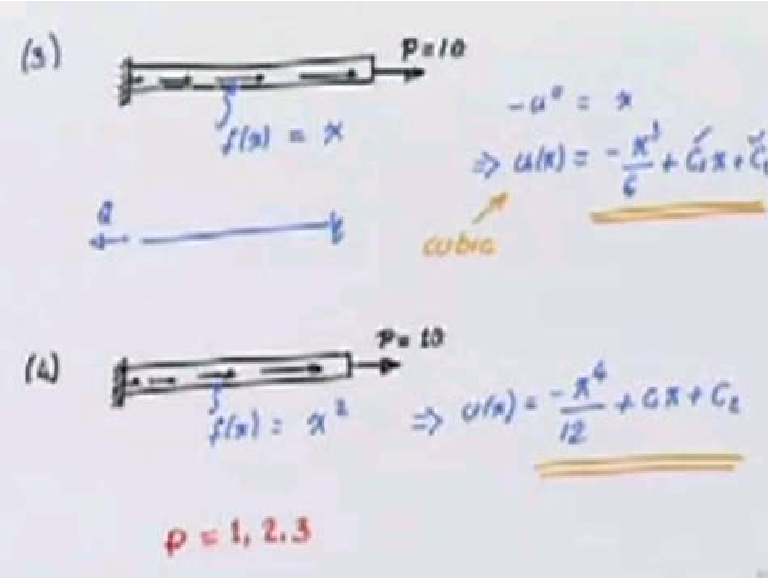
$\{U\}$ $\approx 10^{-12}$

What is U_{ig}? U_{ig} is equal to the U ieldofs for the element k, 1 plus i, i going from 0 P plus 1. If I take all these i is essentially it simply means that I go to the solution vector U from there I extract. This global entry of the solution vector which corresponds to the i eldofs k, 1 plus i. It corresponds to U at this point.

If all these values are 0, are equal up to the last decimal digit that is essentially error between the two is 10 to the power of minus 12 or less. Then we say our nodal value are sufficiently accurate are upto numerical precision it is exact. If these nodal values are exact or the point values that is, the degree of freedom correspond to these point values and they are equal to the exact solution, then we can say that atleast for the cases where the exact solution has to be equal to the finite element solution, the finite element solution is the correct one.

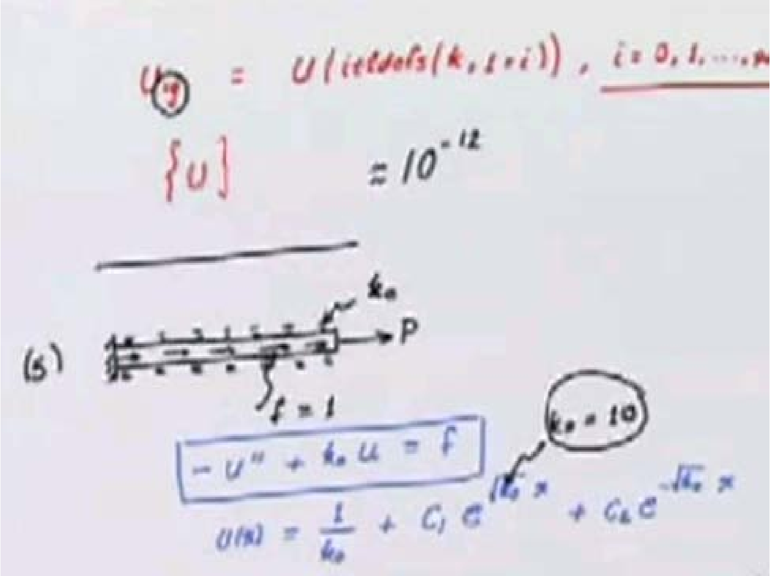
This check has to be done element by element and we should output the values and check. Once this set of tests is done then we know that atleast a part of our element calculation matrix, element matrix coming out of the stiffness only not the spring part, is correctly done the load vector calculations are properly done and the boundary conditions are properly applied.

(Refer Slide Time: 17:21)


(3)
P=10
f(x) = x
$-u'' = x$
$\Rightarrow u(x) = -\frac{x^3}{6} + C_1x + C_2$
cubic
(4)
P= 10
f(x) = x^2
$\Rightarrow u(x) = \frac{-x^4}{12} + C_1x + C_2$
p = 1, 2, 3

If I go up, I can certainly devise variants to these problems. You see one can play around with the boundary condition that one applies. Imagine I can also do this (Refer Slide Time: 17:31), I can do this, can apply the spring boundary conditions, I can do various things to essentially get the effect of the various boundary conditions and for these various boundary conditions only these coefficients C_1 and C_2 are going to change. The nature of the solution is not going to change. I can play with all the boundary conditions that we have there and check whether the boundary conditions are being properly applied or not.

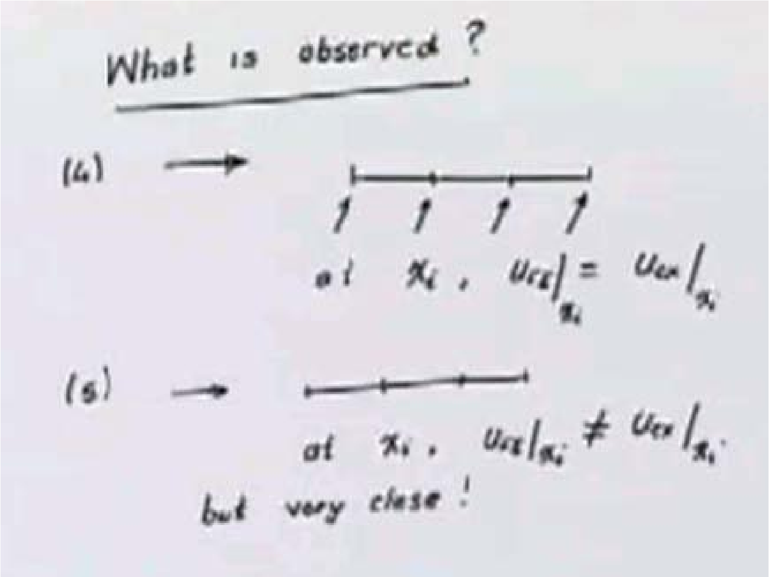
(Refer Slide Time: 18:03)


$u_{(9)}$ = U(ieldofs(k, 1+i)), i = 0, 1, ..., n
{u}
$\approx 10^{-12}$
(5)
k_0
P
f = 1
$-U'' + k_0 U = f$
$k_0 = 10$
$U(x) = \frac{1}{k_0} + C_1 e^{\sqrt{k_0} x} + C_2 e^{-\sqrt{k_0} x}$

All the parts of the program are checked; solver is also checked, boundary condition application is checked and also the part where we output the values of the solutions at nodes. The post processor to a certain extent is also checked.

Let us go to another set of problems. I will call this problem 5 (Refer Slide Time: 18:32). Here I can have a P, here I can have f and I am putting this distributed spring support - spring constant k_0. Let f be 1 let's assume. In this case what happens to the exact solution? In this case differential equation looks like this (Refer Slide Time: 19:12). Minus U double prime plus k_0 U is equal to f. In this case U of x will turn out to be plus C_1 e to the power of root k_0 x plus C_2 e to the power of minus root k_0 x. Let us say we choose k_0 equal to 10. We see that in this case the solution is not a polynomial the exact solution is not a polynomial but it is rather in terms of the exponentials or I can write it in terms of cos hyperbolic and the sine hyperbolic functions. In this case I expect that using linear quadratic or cubic approximation I should not be able to exactly capture these functions but we should be able to do a good enough job, be close to it.

(Refer Slide Time: 20:39)



What is observed ?

(4) →

at x_i, $u_{FE}\Big|_{x_i} = u_{ex}\Big|_{x_i}$

(5) →

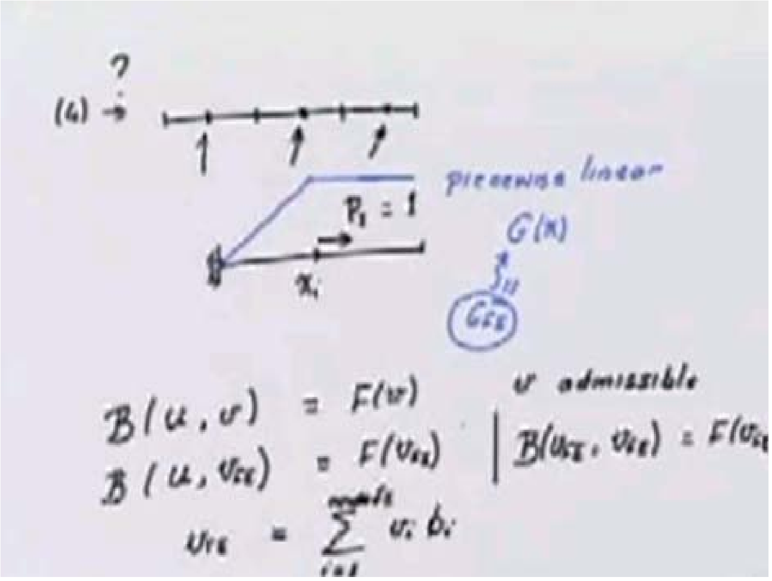
at x_i, $u_{FE}\big|_{x_i} \neq u_{ex}\big|_{x_i}$.

but very close !

In this case, what should be observed? What is it that we observed? In this problem, problem 5 as well as in, I would also talk of, problem 4 and 5. In problem 4 you note that if I am using this three element mesh (Refer Slide Time: 21:10). Then at these points x_i which are my global nodes. At these points by the finite element solution is equal to the exact solution at x_i is equal to $x_{i.}$

At other points you may be close but you will never be exact, for problem 5 this is true for all orders of approximation 1, 2, 3 so on. For problem 5 we observe that if you take these nodes at x_i U_{FE} at x_i is not equal to U exact at $x_{i,}$ but it is sufficiently close. And this is how close improves or becomes better and better as the P is raised.

(Refer Slide Time: 22:36)



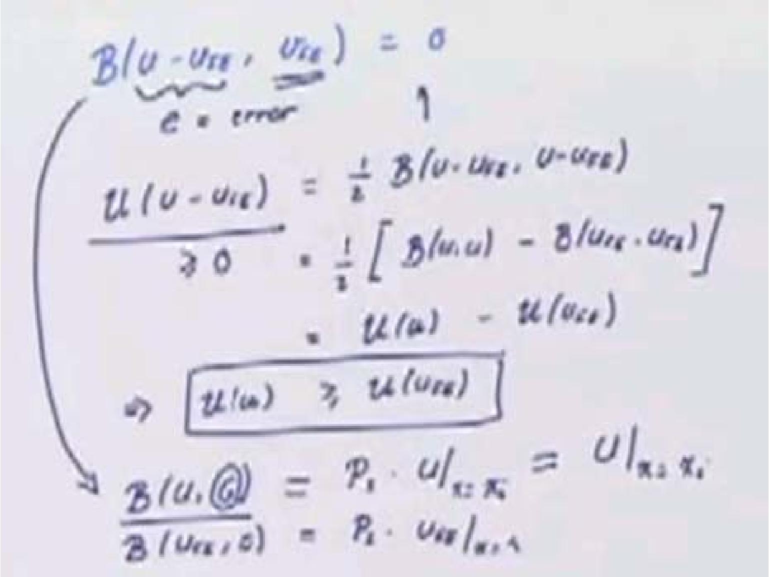
$$B(u, v) = F(v)$$
$$B(u, v_{FE}) = F(v_{FE}) \quad \Big| \quad B(u_{FE}, v_{FE}) = F(v_{FE}$$
$$v_{FE} = \sum_{i=1}^{nodes} v_i \phi_i$$

Can you say the reason why this is going to be like this? Why not something else? Can we say that if I am using a quadratic approximation what can be said about these points? Why should the solution not be the same as the exact solution at these points, in the general case for problem 4, for example? This can be given in a very simple mandate way. Let us take the problem corresponding to the node x_i (Refer Slide Time: 23:05). We have the mesh, we have the node x_i. At this node we apply a point load equal to size one and remember here we have fixed. Let's take the problem as it is. Here there is no end force. I am simply applying a force in the middle.

In this case we know how to solve this problem, in this case the solution would be a piece linear. Let us call this solution G - G of x. This solution is a piecewise linear. My question is as we have discussed earlier by putting a node here by having a mesh, should I be able to represent this exactly using my finite element solution or not answer is yes. This will be equal to the finite element solution. If I took this problem, solved it, using the finite element method, this would give me the finite element solution would give me exactly the solution to this problem. Let us call it by G_{FE}. What we have done earlier, we know in terms of the bilinear forms that B (u, v) is equal to F(v) I am simply recapping for admissible weights. So v is admissible.

We know what we mean by admissible that v at the point with geometric condition is given, is 0. Similarly we also know if it is true, this is for the exact solution for any problem. If this is true for any v then it should also be true for so called v finite element. And by v finite element we understand that I can write v in terms of my basis functions that are used in the approximation. This going to be true further, I will also have the finite element solution how are we obtaining it u finite element, v finite element is equal to v finite element.

(Refer Slide Time: 26:08)



$$B(\underbrace{u-u_{FE}}_{e\,=\,\text{error}}, \underline{v_{FE}}) = 0$$

$$\underbrace{\mathcal{U}(u-u_{FE})}_{\geq 0} = \frac{1}{2}B(u-u_{FE}, u-u_{FE})$$

$$= \frac{1}{2}\left[B(u,u) - B(u_{FE}, u_{FE})\right]$$

$$= \mathcal{U}(u) - \mathcal{U}(u_{FE})$$

$$\Rightarrow \boxed{\mathcal{U}(u) \geq \mathcal{U}(u_{FE})}$$

$$\underline{B(u, v)} = P_1 \cdot u|_{x=x_1} = U|_{x=x_1}$$

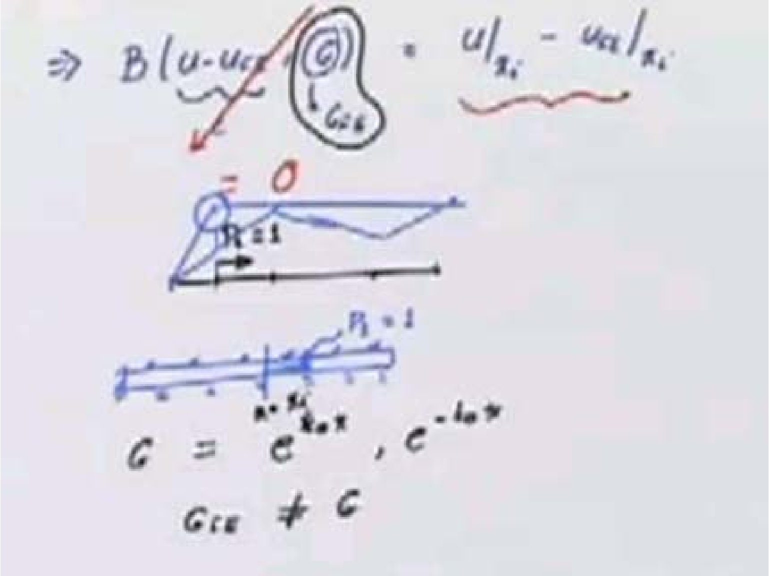
$$B(u_{FE}, v) = P_1 \cdot u_{FE}|_{x=x_1}$$

From these two equations we are getting we can get B (U minus $U_{FE,}$ V_{FE}) is equal to 0 this quantity is called the error e. This say that the error in the finite element solutions is supposed to be orthogonal to all functions which can be represented using the basis functions, were used to form the finite element solution. If it is true then we can talk of the strain energy of (U minus U_{FE}) is equal to 1/2 B (U minus U_{FE}, U minus U_{FE}) and using what we have obtained here this will be equal to 1/2 [B (U, U) minus B (U_{FE} minus U_{FE})].

This is equal to strain energy of u minus strain energy of U_{FE} and this (Refer Slide Time: 27:20) we had said that implies strain energy of u is greater than or equal to strain energy of U_{FE} because the strain energy is positive; this itself is greater than or equal to 0, strain energy of error is greater than equal to 0 this has to be greater than this (Refer Slide Time: 27:50). This as yet has no connection to what we have done. From here if I have this relationship lets come back here, go back to our previous problem here (Refer Slide Time: 28:08), then if I take this problem this load, then for this if I take B of (U, G) will be equal to, if you work it out, it will be equal to P_1 into U at the point x is equal to $x_{i.}$

This is nothing but the unit load method that you have done. What this say is that the strain energy that is the work done by our existing state of stress against this strain due to G, is equal to the work done by the applied force against the existing displacement. This is the unit load method that we have done in class - in the mechanics class. This is going to be equal to U at the point x is equal to $x_{i,}$ but I know that G is G_{FE}. Further, if I do it like this (U_{FE}, G) (Refer Slide Time: 29:00).

(Refer Slide Time: 29:45)



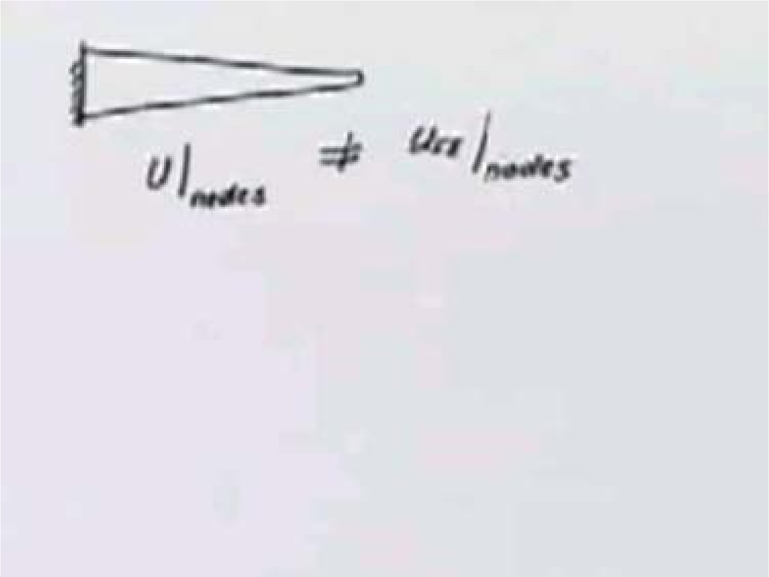
Then if I take the difference of the two things I will get implies B (U minus U $_{FE,}$ G) U at the point x_i at the node x_i minus U_{FE} at the node $x_{i.}$ But this G can be replaced by the G_{FE} and using the orthogonality of the error because this is the error, this side is equal to 0 (Refer Slide Time:

30:15) which means that this difference is equal to 0; the finite element solution that the node is equal to the exact solution of the node.

What happens for any other point? If I have any other point which is not a node let us take this point. If at this point if I apply the unit load, then the exact solution to this problem will be like this. This exact solution I cannot capture exactly using the basis that we have defined, because my finite element solution can only do this and so on. It will not be able to capture this thing. If it is not able to capture this thing, I cannot say that G_{FE} is equal to G. If I cannot say that G_{FE} is equal to G, then I cannot use the orthogonality property and I cannot say that U is equal to U_{FE} at any other points.

Similarly, if I take problem 5 the class of problems corresponding to problem 5, if I have this term (Refer Slide Time: 31:38) here with these springs applied input now with the node and at that node I apply a point load P_1 is equal to 1. In this case, the solution to this problem G is going to be essentially a cos hyperbolic and sine hyperbolic. The solution will be in terms of e to the power of k_0 x and e to the power of minus k_0 x piecewise. Certainly, I cannot represent G with the finite element representation in any in any case using P equal to 1, 2, or 3. I cannot exactly represent the G with $G_{FE.}$ If I cannot do it, then even for a node in this case, if this point where I am applying it is a node that is, an end of an element, even in that case this condition cannot be satisfied (Refer Slide Time: 33:00). This means that, I cannot use the orthogonality of the error and U at the node is not going to be equal to the U_{FE} at the node. This is contrary to what is generally assumed or given as a juristic proof in the normal finite element literature.

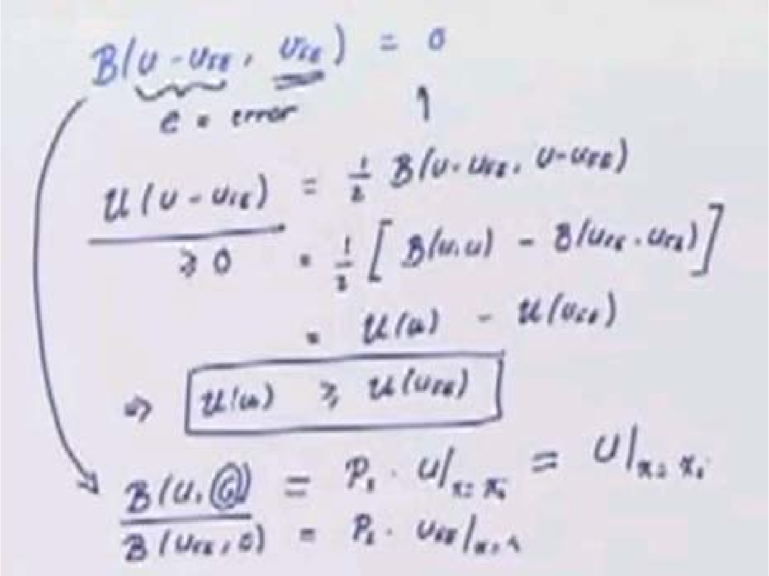
(Refer Slide Time: 33:42)



$$U\big|_{nodes} \neq U_{FE}\big|_{nodes}$$

These values are not going to be exact. Further, even if I do not have the spring and I have tapered member. In this case also you will see that is not equal to U$_{FE}$ at the nodes, but they are sufficiently close.

(Refer Slide Time: 33:27)



$$B(\underbrace{U - U_{FE}}_{e\,=\,error}, \underline{U_{FE}}) = 0$$

$$\underset{\geq 0}{\underline{\mathcal{U}(U - U_{FE})}} = \frac{1}{2} B(U - U_{FE}, U - U_{FE})$$

$$= \frac{1}{2}\left[B(u,u) - B(U_{FE}, U_{FE})\right]$$

$$= \mathcal{U}(u) - \mathcal{U}(U_{FE})$$

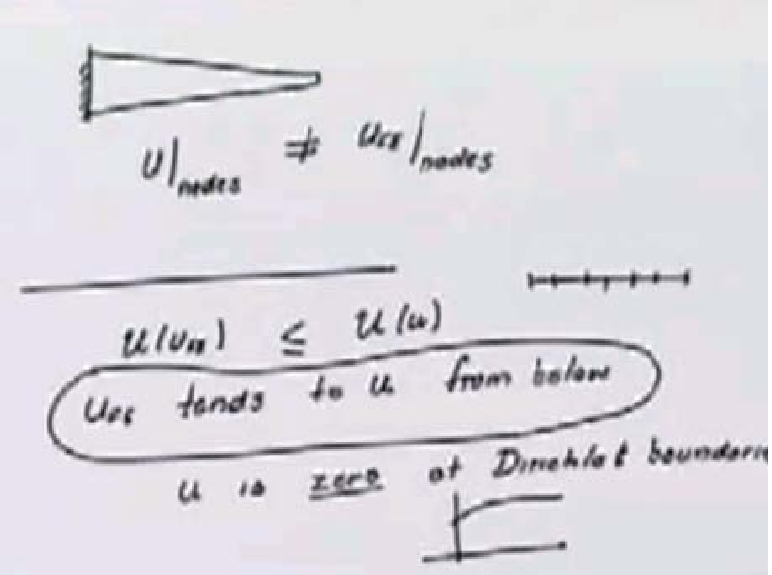
$$\Rightarrow \boxed{\mathcal{U}(u) \geq \mathcal{U}(U_{FE})}$$

$$\underline{B(u, \textcircled{a})} = P_1 \cdot u\big|_{x=x_1} = U\big|_{x=x_1}$$

$$B(U_{FE}, a) = P_1 \cdot U_{FE}\big|_{x=x_1}$$

If I go back to what I had done here (Refer Slide Time: 33:55), if I look at this, this also gives us a way of checking what is happening. I know that my finite element solution has been obtained how do I see that if I refine the mesh further or I keep on increasing the P I am doing the right thing that is, I am converging to the exact solution that is through this?

(Refer Slide Time: 34:25)

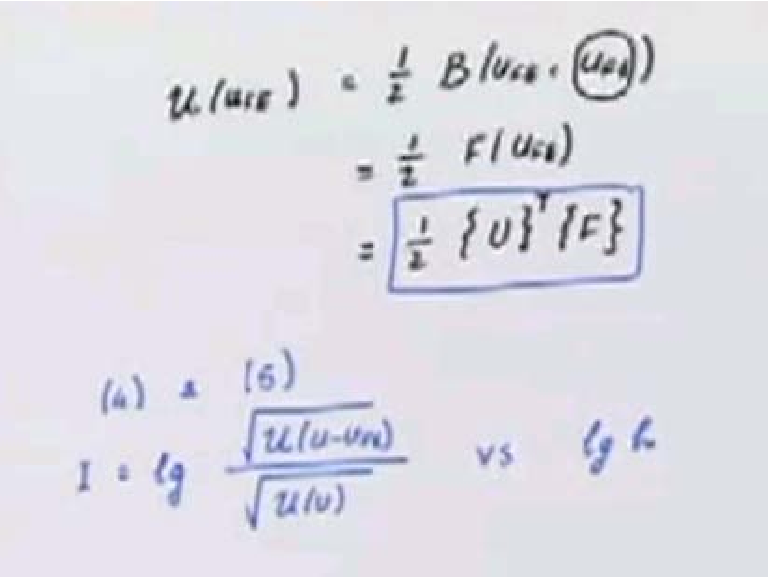

$U|_{nodes} \neq U_{FE}|_{nodes}$
$\mathcal{U}(U_{FE}) \leq \mathcal{U}(u)$
U_{FE} tends to u from below
u is zero at Direhlet boundar

In the case you know that the u of u_{FE} is less than or equal to U of u itself. What we say that the finite element solution reaches or tends to u exact from below, in terms of the strain energy. If I am able to compute the strain energy of the finite element solution, then If I take the sequence of the meshes that is, I take the mesh, which is a crude mesh may be this one (Refer Slide Time: 35:10), then I refine the mesh, then I take the sequence of the finite element solution I should get the strain energy for the crudest mesh is lesser than the strain energy for the next refined mesh, is lesser than the strain energy for the next refined mesh.

This is what we should be able to observe. This is always going to be true provided my exact solution is set to 0 at Direchlet boundaries. For the case where I have a Direchlet boundary condition, there I have to have that case where u is 0. If u is not equal to 0 then this is not going to be true.

This is a check of what is happening with my approximation. I should able to get this monotonically converging to a higher number (Refer Slide Time: 36:15). If I plot for the various meshes I should get like this - convergence to an exact from below.

(Refer Slide Time: 36:34)



$$\mathcal{U}(u_{FE}) = \frac{1}{2} B(u_{FE}, u_{FE}) = \frac{1}{2} F(u_{FE}) = \boxed{\frac{1}{2} \{U\}^T \{F\}}$$

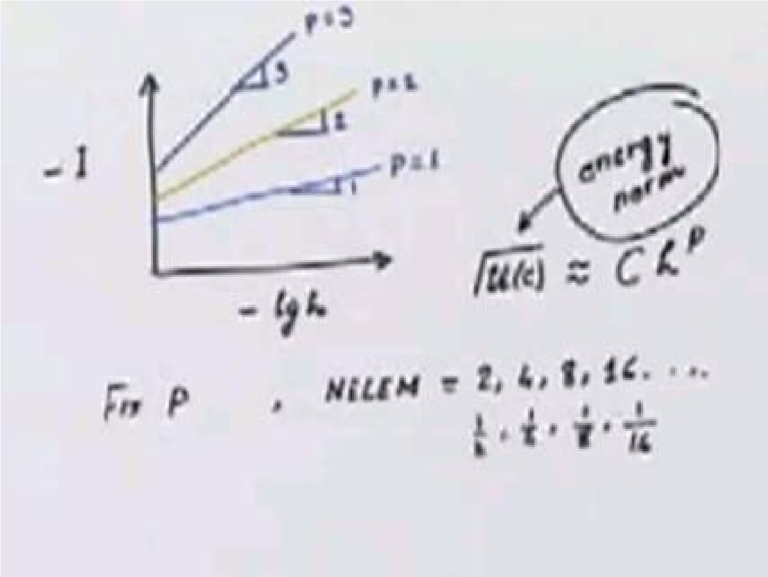
(4) & (5)

$$I = \lg \frac{\sqrt{\mathcal{U}(u-u_{FE})}}{\sqrt{\mathcal{U}(u)}} \quad \text{vs} \quad \lg h$$

How do I compute u_{FE}? If I see the strain energy of the finite element solution is 1/2 of B (U_{FE}, U_{FE}) but I know that u_{FE} is a choice for a V. This is certainly going to be equal to half of F of U_{FE}. This is going to be equal to half of my displacement vector multiplied by the load vector. I solved for the U transpose somehow I store the load vector, then this is my load vector after applying boundary conditions, then U transpose F should give me what is the strain energy.

This is very easy to compute and for the sequence of meshes I should be able to check. Another thing that we should be able to check for problems 4 and 5, if I plot, I take this quantity I is equal to log of does not matter (Refer Slide Time: 38:12). This I can essentially plot verses log of h. If I plot this quantity, what does it say? It is the strain energy of the error, square root of strain energy divided by the square root of the strain energy of the displacement log of that I take. So I call as i, and then I should be able to see something very interesting when I plot.

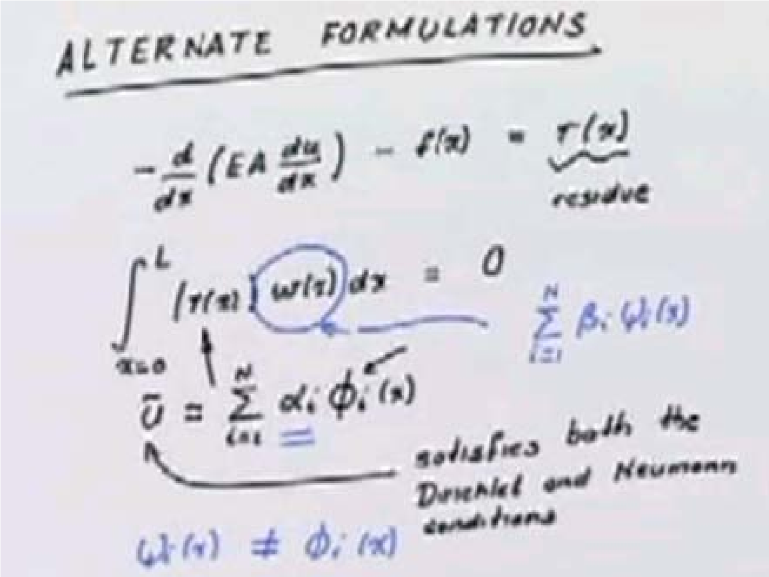
(Refer Slide Time: 38:43)



I plot here (Refer Slide Time: 38:45) as minus I minus log of h so I will get for P is equal to 1 one curve for P is equal to 2, another curve and for P is equal to 3 and this will be P is equal to 1, P is equal to 2, P is equal to 3 and the slope here, slope will be 1, slope here will be 2, slope here will be 3 (Refer Slide Time: 39:15). If I fix P to some value 1, 2 or 3 and then I choose my NELEM that is total number of elements is equal to 2, 4, 8, 16 and so on, all multiples of 2. Then essentially, I get h is equal to ½, 1/4, 1/8, 1/16, and so on. If I take the sequence of meshes these are more and more refined meshes, find for this P the solution finite element solution, from there I find the strain energy of the error, remember how we have given the strain energy of the error from our previous slide. It is given in terms of the differences of the strain energies of exact solution and the finite element solution.

Strain energy of the exact solution we can compute explicitly using the integrals. I can do this find the strain energy of the error and then find the relative error and plot. Plot these two, three things. For these values I should get curves like this (Refer Slide Time; 40:30). If you can get this again it is a good proof that a program is doing what it should be doing. That is we are seeing a Ch to the power of P rate of convergence in the square root of the strain energy of the error, which we have talked about earlier. That is, with the h as the mesh is refined I get the strain energy of the error square root of that converges as Ch to the power P. This quantity is called the Energy Norm. With this I think we have essentially understood most of what we needed to understand as far as this one dimensional problem is concerned.

(Refer Slide Time: 41:41)



ALTERNATE FORMULATIONS

$$-\frac{d}{dx}\left(EA\frac{du}{dx}\right) - f(x) = \underbrace{r(x)}_{\text{residue}}$$

$$\int_{x=0}^{L} \left(r(x)\right) \boxed{w(x)}\, dx = 0 \qquad \sum_{i=1}^{N} \beta_i \psi_i(x)$$

$$\bar{U} = \sum_{i=1}^{N} \alpha_i \phi_i(x)$$

satisfies both the Dirichlet and Neumann conditions

$$\psi_i(x) \neq \phi_i(x)$$

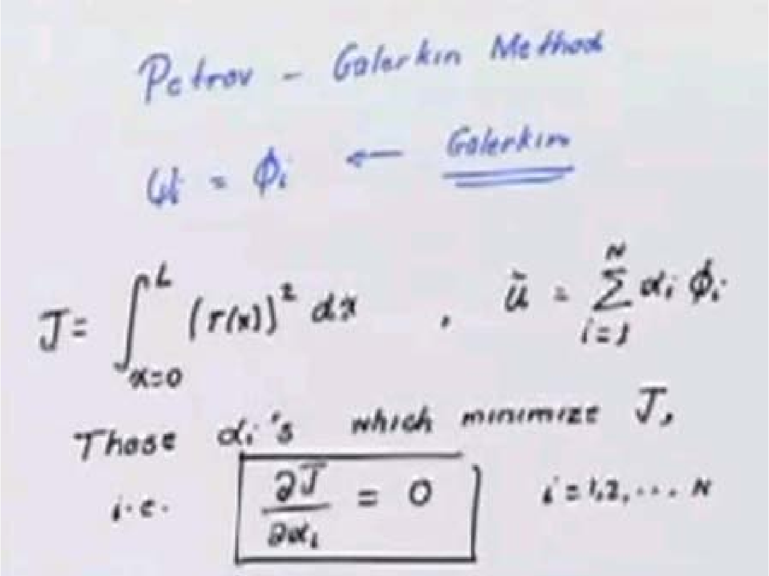
Let us in closing look at some alternate formulations that is, that we have done. If we remember either the minimization of the total potential energy of functional from where we got the variational formulation or we started with the residue that is my differential equation. If I take this quantity I will call it R of x. In the actual if you use a two solution of the problem R of x has to be equal to 0 (Refer Slide Time: 42:21). This I had called as the residue. What we had done? Multiplied this residue with w and integrated by part once transform derivative from u to the w and we got our weak formulation. What if we simply take integral x is equal to 0 to L of r of x into the weight function w dx; if I leave it like this. This has to be equal to 0 for our actual problem.

Here r of x now comes due a choice of some series representation of U. We could use some functions. We will say U tilde there it is an approximation i is equal to one to some N alpha$_i$ phi$_i$ x; phi$_i$ as we had done earlier are our basis functions (Refer Slide Time: 43:16). Put it here, I need this to find alpha$_i$s; I am not doing any integration by parts. First of all my phi$_i$s should be such that U tilde has to now satisfy both the Dirichlet and Neumann conditions at both ends. I am asking a little bit more from U tilde that it has to satisfy the given boundary condition. Not only that now the second derivative of U tilde has to be properly defined. I should take the phi$_i$s to be sufficiently smooth.

Let us say we have taken them to be polynomial in the whole domain. I can choose the w of x, because U tilde satisfies all the boundary conditions, w of x need not be constraint by the U. I can choose to be some sigma i is equal to 1 to N of beta i psi$_i$ of x, where the function psi of x are not the same as the function phi$_i$ of x.

When psi$_i$ of x is not equal to phi$_i$ x this is a method you can do it like this, provided that you have chosen psi$_i$ in such a way that we get n linearly independent equations in terms of the unknown alpha x. If I get these equation then I can invert the matrix that comes you see that this will also lead to a matrix problem in terms of the unknown alpha I solve this and we get a solution.

(Refer Slide Time: 45:41)



Petrov - Galerkin Method

$W_i = \phi_i$ ← Galerkin

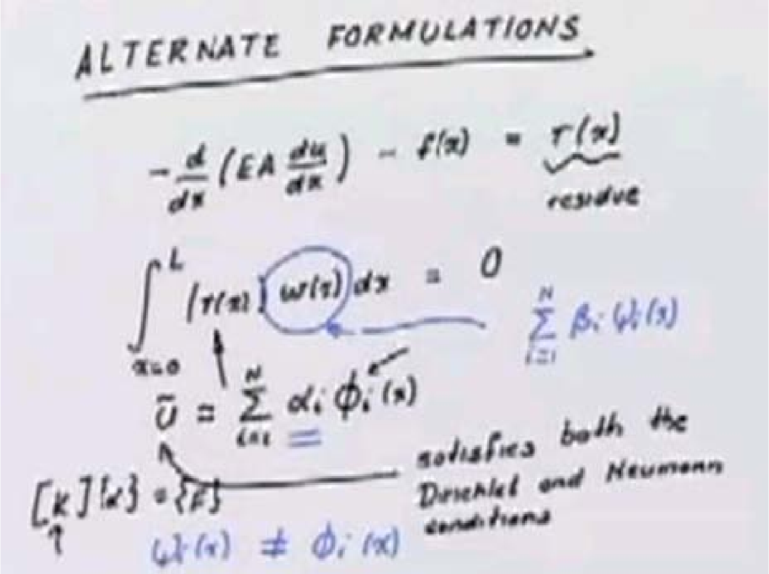
$J = \int_{x=0}^{L} (r(x))^2 \, dx$, $\tilde{u} = \sum_{i=1}^{N} \alpha_i \phi_i$

Those α_i's which minimize J,

i.e. $\boxed{\frac{\partial J}{\partial \alpha_i} = 0}$ $i = 1, 2, \dots, N$

This method is called the Petro Galerkin method in case I use psi$_i$ equal to phi$_i$. It becomes the Galerkin method and in fact we can do the integration by parts. To get the weak form, we saw that whatever we have done is essentially using this Galerkin method. Another way doing this which is quiet popular in fluid mechanics computation is I say that I will define a functional J dx. Again choose u tilde is equal to sigma i is equal to 1 to N alpha$_i$ phi$_i$ such that the phi$_i$ is satisfying all the given boundary conditions. Then I am looking for those alpha$_i$'s which minimize Ji we want the partial derivative of J with respect to the alpha$_i$ to be equal to 0, i going from 1 to N. I have in terms of the alpha$_i$'s N such equations.

(Refer Slide Time: 47:35)



ALTERNATE FORMULATIONS

$$-\frac{d}{dx}\left(EA\frac{du}{dx}\right) - f(x) = \underbrace{r(x)}_{\text{residue}}$$

$$\int_{x=0}^{L} r(x)\, w(x)\, dx = 0$$

$$\sum_{i=1}^{N} \beta_i \psi_i(x)$$

$$\tilde{u} = \sum_{i=1}^{N} \alpha_i \phi_i(x)$$

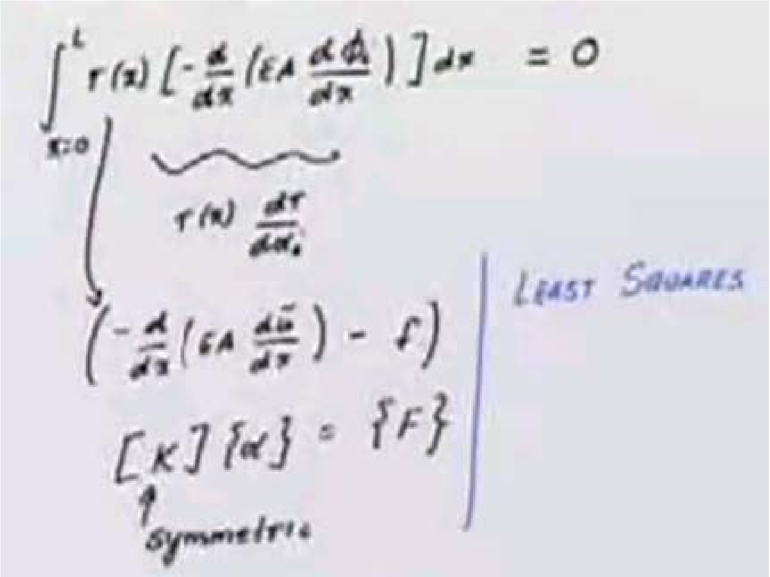
satisfies both the Dirichlet and Neumann conditions

$$[K]\{\alpha\} = \{F\}$$

$$\psi_i(x) \neq \phi_i(x)$$

If I go back and look at the Petro Galerkin here for i and J, if i replace i with J i need not get the same entries so here also I will have some K into alpha is equal to F, but the K K$_{iJ}$ will not be equal to K$_{iJ}$ need not necessarily equal to K$_{iJ.}$ This matrix is going to be unsymmetric and that comes with its own set of problems.

(Refer Slide Time: 48:08)



$$\int_{x=0}^{L} \underbrace{r(x)\left[-\frac{d}{dx}\left(EA\frac{d\phi_i}{dx}\right)\right]}_{r(x)\frac{dr}{d\alpha_i}} dx = 0$$

$$\left(-\frac{d}{dx}\left(EA\frac{d\tilde{u}}{dx}\right) - f\right)$$

$$[K]\{\alpha\} = \{F\}$$

If I do things like this here then del J del alpha I will give me integral x is equal to 0 L r of x into if I go to the r that we have taken minus d dx of EA d phi$_i$ dx this is equal to zero which has the u tilde is sitting there into minus d dx of EA d phi$_i$ dx this is nothing but the derivative of r of x with respect to alpha$_i$.

This is equal to the whole thing r of x into dr d apha$_i$. This has to be equal to zero. You see that in r of x I will be having r of x as minus d dx of EA du tilde dx minus f. If I substitute for u tilt in terms of the series then again I get and I set up the problem this. I will get K symmetric, which is something we often desire. This this whole approach is called the method of least squares.

There are various other alternatives that we can for example the collocation method and so on where collocation means that w is chosen as the Galerkin delta**.** Here I am going to talk about that here. We have various approaches to getting our system of equations provided we have made our series of representation for U that is the crux of the whole thing.

How we get the system of equation is our choice and depending of the problem at hand we will use one way or the other way. In the next class, in the next lecture we are going to talk about essentially an extension of the 1D problem to look at a different class of 1D problem. Essentially, what we know as the beam problem, why we are going to look at the beams? Because the beam problems corresponding to the Euler-Bernoulli beam theory leads to a forefather differential equation.

If you have the differential equation we continue with what we have done or we have to make certain modification in the structure of things that we have done in order to handle this problem, that is what we are going to focus on. We are going to look at the beam problem essentially, talk of addition to the programs that we have written one dimensional program in order to handle the beam problem and we will go from there.