

The following content is provided under a Creative Commons license. Your support will help MIT OpenCourseWare continue to offer high quality educational resources for free. To make a donation, or view additional materials from hundreds of MIT courses, visit MIT OpenCourseWare at ocw.mit.edu.

PROFESSOR: Ladies and gentlemen, welcome to lecture number 8. In this lecture, I would like to discuss with you the numerical integration and some modeling considerations. You might recall that in the isoparametric formulation that we discussed in the early lectures, we used numerical integration. And this is an important fact, and I want to discuss with you in this lecture the Newton-Cotes formulas that we're using, and in particular, the Gauss integration-- the Gauss numerical integration that we are using in finite element analysis.

There are various considerations. First of all, I'd like to present to you a little bit of the theory that is being used and in particular, however, I also like to present to you some practical considerations. And finally, together with these aspects, I would like to also present to you some considerations regarding the choice of elements, and, of course, regarding the choice of the order of numerical integration that has to be used in the finite element formulation.

Well, just to refresh your memory, on this view graph, I put together once the various matrices that we have to evaluate. Here, we have to stiffness matrix, which of course, is obtained by the integral of $B^T C B$. B is the strain displacement transformation matrix. C is the stress strain law. Of course, this B -matrix is a function of x and y in the physical space, but if you use isoparametric interpolation in two-dimensional analysis, this B -matrix would be a function of r and s , as we discussed earlier.

The m matrix is given here, the R_B load vector is given here, the R_S load vector is given here, and the initial stress load vector is given here. We derived all of these quantities in an earlier lecture and all I like to now discuss with you is how do we evaluate these quantities using numerical integration.

In isoparametric finite element analysis, I mentioned already the displacement transformation matrix is a function of r , s , and t . The strain-displacement interpolation matrix is also a functional of r , s , and t in three-dimensional analysis. Of course, in one- or two-dimensional analysis, we only talk about r and s , respectively. Remember please, that r , s , and t vary from minus 1 to plus 1, and that we, therefore, need the volume transformation here, dV being the determinant of J , where J is a Jacobian transformation between the x - and y -, and r - and s -axis, or x -, y -, and z - and r -, s -, and t -axis. And of course, dr , ds , dt are the respective differentials that have to be used here, too.

We discussed that earlier, and just to refresh your memory again, what we came up with is that the stiffness matrix K two-dimensional analysis is given in terms of B transpose C , B , times determinant of J , dr , ds , and we are integrating from minus 1 to plus 1. The mass matrix is given, as shown here, where H is, of course, once again, the displacement transformation matrix. [ρ Row ρ] is the mass density.

Just a preliminary thought, please recognize that B involves derivatives of displacements, We have here derivatives of displacements, here we have the actual displacements. So the product of B transpose B will, in general, be of lower order than the product of H transpose H . Therefore, when we talk about numerical integration, we will find that to evaluate the M matrix exactly, we will have to use a higher order integration than in the exact evaluation of the stiffness matrix. I will refer to that again later.

We also mentioned already that in the numerical integration, the basic process is that we are evaluating the function F_{ij} , as shown here, at particular points, i , j , denoted by i , j . And the $\alpha_{i,j}$ are weight coefficients. Therefore, what we do is, once again, to evaluate this product at a particular point within the element, multiply that product by a weight coefficient, and then do the same for a certain selected number of points, and sum all these evaluations up, as shown in this equation. Now, what this means, of course, is that we really have to use specific points-- and I have to discuss with you what points we are using-- and how many points is another question that we have to also discuss.

Well, in 2x2 integration, in isoparametric finite element analysis, we would use four points altogether. 2x2 means two points this direction, and two points that direction. In three-dimensional analysis, we would, of course, talk about 2x2x2. You would have another two points in this direction, in the t direction. These points in Gauss integration, are given by these values here. In other words, point 4, for example, would have the values, r equals plus 0.577 and s equal to plus 0.577. Point 1 would have the same values, but both negative.

These values here are the Gauss values, and I will discuss briefly with you a little later how these have been obtained. Of course, Gauss derived them for us some 100 years ago, and we are now simply using them. In 3x3 integration, we talk about nine integration points altogether. Once again, three integration points lay, as so to say, in this direction, and into that direction. If we use three-point integration in three-dimensional analysis, we would have altogether 27 integration points because each of these layers, of course, applies then three times, and because we also have three layers into the t direction, and therefore altogether, 9x3, being 27 integration points.

The basic process of numerical integration is that if we want to integrate a function. And I've shown here an actual function-- this heavy line is the actual function f-- if we want to integrate that actual function, with respect to x here. What we do is we are putting an interpolating polynomial down as an approximation to that function. Now, if we want to integrate this function from a to b, the simplest interpolating polynomial would be to put a straight line from this point to that point. And then the area under the straight line is assumed to be approximately, at least, really sufficiently close, to the area under the actual function f.

So what we are doing then is we are replacing the integral of the actual function f, over this integral, by the integral of the interpolating polynomial over that integral. If we do that with just two points, and the two points are these two, we will derive the trapezoidal rule. If we use those three points, and I have here a view graph that shows how we would use three points. One point here, second point there, and now

we have also a third point there. Then we would obtain the Simpson's rule.

Once again, instead of integrating the actual function, f , which is shown by this line here, we are, in fact, integrating the area under this red line. This is a little the weak. Well, now there it comes. We are integrating the area under the red line. In other words, the assumption is then that this area is close enough to the area under blue line, and what we need to do is to evaluate, of course, the actual function, f , at these three points. And we will have to somehow develop certain constants, the α_i, β_j 's that I talked about earlier that multiply this function, f , in order to evaluate the red area here accurately.

Well, if we do that-- in other words, if we integrate the red area using these three function points then we obtain the Simpson's rule, and this is just one rule of the more general Newton-Cotes integration. In the Newton-Cotes integration, we use equally-spaced sampling points. In other words, if we want to integrate from a to b , the simplest way of proceeding is to use a sampling point here, and a sampling point there. And then, of course, as I pointed out earlier, we are putting, really, a straight line down and we're assuming that the area under this straight line is equal to the area to under the actual function, f , that might be looking like that.

Well, that is the trapezoidal rule and Newton-Cotes integration with n equal to 1. In other words, with two constants here, is equal to this trapezoidal rule. What we are saying is that the area from a to b -- now I used here r , this is the r -axis. say, from a to b -- this is the actual integration-- is replaced by the summation of the value at this point, the F_0 value, the F_1 value, and each of these values is multiplied by a certain constant.

So in this particular case, for the trapezoidal rules that I'm talking about, the right hand side looks as follows-- b minus a , that is this length here, times two values, the C_0 , n is equal to 1 in this particular case, times F_0 . Plus C_1 times F_1 . And of course then, plus an error, R in this particular case, n , once again, is equal to 1, plus R_1 . This error of course, is assumed to be very small in the analysis, and we would actually neglect it. So once again, what we are saying is that we can replace

the integration of F of R dr , from a to b , via this summation here. Notice that b minus a times that constant, that is our α_1 , that I referred to earlier. And b minus a times that constant would be another α value. And, of course, these α values here multiply the actual function values at this station, and at that station.

Well in the Simpson rule, n is equal to 2, b minus a is still there, but now we're talking about C_0 2 0 times F_0 plus C_1 2 times F_1 plus C_2 2 times F_2 , and of course, an error of 2's there. Well, the constants, the C_0 1, C_1 1 and so on. The constant C_n have been evaluated for us, and they are tabulated in this table here. Notice, that for the trapezoidal rule, we have the simple constants, $1/2$, $1/2$. The error here is shown here, 10 to the minus 1 times the second derivative of F . So if the second derivative of the actual function is 0, well, then our integration is exact. Of course, what that means is that our function is indeed a straight line. Because only then the second derivative is 0. So our integration is exact when we integrate a straight line. Well, if you look at our earlier graph here, surely, if our actual red line is a straight line between this point and that point, then our integration, using the trapezoidal rule, must be exact. And that is verified by looking at the error bound, given in this table.

The Simpson's rule has constants, $1/6$, $4/6$, $1/6$. n is equal to 2 in this particular case, as I mentioned earlier. The error here is now given to the force derivative. And so on. Here, we have integrals 1 to 6. In practice, however, we would only use this the trapezoidal rule, the Simpson rule, and we might then still use this rule here. Notice that the error of this rule here is of the same order as the error of this rule, involving five points. So the four-point integration rule, of course, is preferable when you compare to the five-point integration rule.

The other important point is that if we use the Newton-Cotes formally, we are really always involving the endpoints of the integration. In other words, if we want to integrate from a to b then we will involve these two points in the actual integration. So if we talk about the integration through the thickness off the shell, or through a beam element thickness, then we would have here the top and bottom surfaces

involved.

Bad point-- well, I shouldn't say bad, but certainly it's not a very advantageous point, of the Newton-Cotes integration is that we need to involve a fairly large number of integration points, or integration point stations to obtain an acceptable accuracy when we talk about finite element analysis. And therefore, the Gauss numerical integration is a more attractive scheme. And, in fact, we will find that in finite element analysis, we almost always use Gauss numerical integration. They're only certain special cases in analysis of beams, plates, and shells where we, in fact, use the Newton-Cotes integration. And the reason then, of course, is that we want to obtain also surface stresses and strains. That would be, for example, one reason.

Now, in Gauss numerical integration, the basic concept is quite similar to the concept that we just discussed. That we again use an interpolating polynomial to lay through the function over the interval, a to b . The important point, however, now is that we use-- if we look at our function here. Here is the interval, a to b . Our x is here. If you look at that function, it's that we, in Gauss numerical integration, do not use equal-spaced intervals. In Newton-Cotes integration, we use equal-spaced intervals, and involves the end points. In Gauss numerical integration, we are using points, stations, 1 to 2 , and our stations that can be anywhere in the interval.

And Gauss has developed the optimum stations for us. in other words, what he has said, at that time is let us optimize the particular locations of the integration points stations, and in addition, let us optimize the weights, α_1 to α_n . In the Newton-Cotes integration, we only optimize, so to say, the α values. The stations were fixed because we involved, for example, in the trapezoidal rule, we involved this end point, and that point. In the Simpson's rule, we used the midpoint. And so on. In Gauss integration, we say, well, let us does not necessarily fix those stations but rather let us just optimize the location of these stations.

Well, since we then can optimize the values of the α_1 , to α_n , this of course are n , components, and the stations R_1 to R_n , which are also n components. We really can optimize two n components. Now, if we look at the fact that we're dealing

with two n components that we optimize now, and the fact that the interpolating polynomial of course, also involves one constant, then we find that the interpolating polynomial can now be of order, $2n - 1$. Whereas in the Simpson's rule, the interpolating polynomial with n stations could only be of order, $n - 1$. You see, in the Simpson's rule, for example, with three stations, the n was, in the Simpson's rule equal to 3, but the order of the interpolating polynomial was 2. In other words, a parabola could be integrated exactly.

Now in the Gauss integration, we are talking about the interpolating polynomial that can be of order, $2n - 1$, a much higher order for n stations. Here, with the Newton-Cotes rule for n stations, we only get an interpreting polynomial of $n - 1$.

Well, Gauss has developed the specific stations, R_1 to R_n , to be used [UNINTELLIGIBLE] in order to fit the interpolating polynomial through the actual function that we want to integrate. And of course, we also know now the error that is involved, R_n . The error of course, being much smaller than in the Newton-Cotes formula because we are dealing with a higher-order interpolating polynomial. The Gauss values are given here. When n is equal to 1, well, we only use the midpoint and the alpha value is 2. When n is equal to 2, in other words, we are talking about two-point integration, then r_i is equal to plus minus 0.577, and 1. In this particular case, we are using the fact that we want to integrate $4 - 1$, to plus 1. In other words, our r here, is this. We coordinate here, and we're integrating from minus 1 to plus 1. So in two-point integration, we using a point here, which is minus 0.577, and another point here, which is plus 0.577.

Notice that we are feeding in these points to a higher-order accuracy in a computer program, typically, in a computer program that runs on a CDC machine with 14-digit precision. We would actually put these points in to 14-digit precision. Of course, they are actually put into this [? outputing. ?] They are there, and the computer program just picks up this value and evaluates then, of course, the function that we want to integrate at that station. The factor that the function has to be multiplied by is given here.

In three-point integration, we use three points. This one is the easiest to look at-- it's right to center point, and then there are two more points, one at minus 0.77 and one at plus 0.77. In finite element analysis, we know, of course, in isoparametric finite element analysis, I should really say, we know, of course, that our integration runs from minus 1 to plus 1. So these are the actual values to be used. And these are the actual values that you would find, for example, in the computer program such as ADINA.

If we want to integrate an interval from a to b-- I'd like to simply mention that, of course, the same scheme can directly be used. If we have the r_i and α_i for the interval from minus 1 to plus 1, then these would be the value to be used for the integration point stations. If our interval is not from minus 1 to plus 1. but from a to b, and this would be the weight that would have to be used for this particular case. The r_i in α_i , once again, are the same r_i and α_i that I listed here in this table.

Well, so far we really talked about one-dimensional integration. However, the same concept can directly be used for multiple dimension integration. In other words, if we talk about the integration, minus 1 to plus 1. For example, as we would integrate, of course, in the formation of a truss element.

If we now want to go through the integration of a two-dimensional element, then we would have to integrate a two-dimensional function in r and s , and that means integrating twice, once with respect to r , once with respect to s , and we can use the same concept that we used in [? analytical ?] integrations, and we first integrate with respect to r . That gives us one α_i here, and then with respect to s . And the final result is shown here, that we have to bring in a weight factor, α_i , and a weight factor, α_j . This one for the i integration, that one for the s integration. And of course, stations r_i and s_j are the stations at which we would evaluate the actual function.

Notice one important point that we could use a different order of integration into the two directions. For example, if we have an element such as this one, two-dimension

element, two nodes here, in the s direction. But say three nodes in the r direction. So our r -axis is this one, s -axis is that one. Then in this particular case, since we're using a higher order interpolation into the r direction, in this particular case, we might say, well, let us use three-point integration this way, but two-point integration into the s -axis. So here, two-point integration into the s -axis, but three-point integration into the r -axis.

And of course, the same concept applies also in three-dimensional analysis. In three-dimensional analysis as an example here, we would have the F now being a function of r , s , and t . Three times integration from minus 1 to plus 1. And the final result just derived the same way as in two-dimensional analysis as given here. Notice once again, we can use different orders of integration into the three directions.

I should also mention at this point that when we talk about an actual stiffness matrix, of course, we talk about an array, n by n , for three-dimensional element, say a 20-node brick. This would be a 60 by 60 stiffness matrix, and every one of these elements here, in that stiffness matrix would be an F , such an F . Every one of these elements would be integrated as shown here.

Let us now look at the practical use of numerical integration. Well, the first important point is that the order of integration that is required to evaluate a specific element matrix can, of course, be evaluated by studying the function, f , to be integrated. In other words, if I want to evaluate a certain stiffness matrix, K . Here is the stiffness matrix given. If I want to evaluate that stiffness matrix exactly, I could look at each all of these elements, and there are now many of these F functions in there. Let us look at that one. I would look at that F value and identify its dependency on i , s , and t . Linear, cubic, parabolic and so on.

And then I could, by entering into-- by the use of the Gauss numeric integration, or the Newton-Cotes integration, and by looking up the errors involved, I could identify what integration order I need in order to evaluate this integral exactly. That can be done and for simple elements, relatively simple elements-- in other words, 4-noded

elements, that are square, or 8-noded elements that are square. We can fairly simply evaluate the order of integration that is required.

However, when the element is curved. In other words, when we talk about a general elements like that-- the required integration order is not that easily assessed, in order to evaluate the stiffness matrix exactly. And the reason for it is that we have the Jacobian transformation in there. Therefore, in practice, the integration is frequently not performed exactly, but we have to remember that the integration order must be high enough.

Well, considering then the evaluation of the element matrices, we should notice the following requirements. For the stiffness matrix evaluation, we note that the element matrix should not contain any spurious zero energy modes. What do I mean by that? Well, I mean that if I have an element such as that element here. We know that if you're talking about a plane stress element, there only three rigid body modes. One rigid translation this way, one rigid translation that way, and one rigid rotation of the elements that way. These are the only three rigid body modes that the element stiffness matrix should contain. It should, of course, contain these in order to satisfy the convergence requirements that we discussed earlier. What I'm saying now is that my integration order should be high enough so that there is no additional 0 eigen value.

If there is an additional 0 eigen value, I might run into serious numerical difficulties in the solution of the overall governing equilibrium equations. And I will give you just now a very simple example.

The second requirement is that the element stiffness matrix should contain the required constant strain states. So basically, what I'm saying here is that the numerically integrated stiffness matrix should still contain the requirements that we discussed, which the element must satisfy for convergence. For the mass matrix evaluation, here we only need to satisfy that the total mass must be included. In other words, the numerical integration should pick up the actual mass of the element. If that were not the case, then considering the total analysis, we would

have lost some mass. The force vector evaluation valuation here, also is a total loads. Must be included, must be picked up, so to say, by the numerical integration.

Let us look at this requirement here, and then I have a very simple example. Here we have the 8-node element, plane stress element, which is supported here on a hinge, and here on a stiff string. I apply a load, p , there. If I use 2×2 Gauss integration, I get absurd results. If I use 3×3 Gauss integration, I get the correct results. The correct results, of course, being that the elements simply rotates about this point, a . It simply rotates about the point, a , as a rigid body. And, of course, extending in that rotation's a spring at point b .

The absurd results here come about-- using 2×2 integration-- because the stiffness matrix k does not contain just three rigid body modes, but four rigid body modes, with 2×2 Gauss integration. Now, three rigid body modes are suppressed by the hinge here and the spring there, they are suppressed. But if I look at the fact that we are now having four in the stiffness matrix, if I suppress three, I still am left with one rigid body mode, and that means one 0 eigen value.

So this is still an unstable structural stiffness matrix, and when I subject that stiffness matrix to a load p , I get absurd results. Therefore, this is a very simple example that demonstrates that the Gauss integration has to be high enough in order to prevent instabilities occurring in the solution of the equations. In practice, of course, what happens frequently is that we are using these elements in a mesh, so we have an 8-node element here, and another 8-node element here. And if we use 2×2 integration for this element, 2×2 integration for that element, we might very well be able to solve the equations very nicely, and we might, in fact get good results in a analysis of a more complicated problem.

Because the stiffness of this element and that element, they, each other, compensate for the loss for the rigid body mode that is still available in this one element here. In other words, although this one element is unstable, when you look at it numerically, when we solve the equations for a single element, as shown here. When the element is surrounded by other elements, they provide enough stiffness

into this element so that the overall solution of the mesh can still be obtained, and we obtain, in fact, good results.

However, if you do use reduced integration, we call this reduced integration because we are not evaluating the stiffness matrix exactly. If you do use reduced integration, you should be aware of this fact that for a single element, or for a certain element mesh layouts, we can run into numerical difficulties.

Let's look briefly at the stress calculations. Surely, stresses can be calculated at any point of the element. And they would be evaluated as shown here. The stress strain law is here. Here we have to the strain-displacement interpolation matrix. Here, we have seen the nodal point displacement vector. Of course, these nodal points displacements are now known, they are now given. This is an initial stress vector which is also given. You should notice that the stresses are, in general, discontinuous across element boundaries. If we look at a [? practical ?] analysis, we might actually not evaluate the stresses at the boundary of an element, but we might evaluate the stresses at the integration points. So if you look at a mesh of elements, as shown here. We might evaluate the stresses at the 2x2 integration points, shown here for that element. If we were to evaluate the stresses of this element at this boundary, and of this element at that boundary, we would see a stress jump.

I should mention here, just as a side note, that in fact, one can show that the stresses at the Gauss integration points are somewhat better predicted in a finite element analysis than along the boundary of the element. So that is another reason why we use, rather, why we predict stresses rather, at the Gauss integration points, and not on the boundaries of the element, using directly this approach.

As an example here, let us look at a very simple problem. Here we have a cantilever modeled with two 8-node elements. Now remember, that these two 8-node elements each contain the parabolic displacement variation. So since this is a bending moment that we apply to that cantilever, and since this bending moment, of course, gives parabolic displacement variation. The finite element solution would be

correct, will be exact in this particular case. And therefore, there would be no stress jump here.

Since the elements contain the displacements that shall be predicted, that are the analytically-correct results. The finite element mesh will predict these displacements, and, of course, we have stress continuity. This is shown here by-- we have dissected these elements here. This is element 2, this is here, element 1. And we have plotted the stress distribution along the line a, b, c. Here's the line, a, b, c. And as you can see, there's 900 here, and there's 900 there. And of course, a linear variation stresses, so no stress jumps there.

However, if you look at the same problem, the same cantilever, I should say, with now an edge share load, as shown here. Then these elements cannot represent the exact displacement distribution in the cantilever because they only contain a parabolic variation in displacement. And in this case, we have the stress discontinuity. $\tau \times x$, in other words, the stress into this direction here, so normal stress. For this element along, for element 2, along a,b,c, you can see 1744 up there. And then [UNINTELLIGIBLE] a neutral axis because of symmetry conditions, of course. And for element 1, along a, b, c, we have 1502 right there. So there is, in fact, this stressed discontinuity that I was talking about earlier. The share stresses here have also a slight discontinuity, as you can see, 291.38, 296.48.

The reason that we have this stress discontinuity here is, of course, that the elements cannot represent the exact solution. And therefore, they each try to, in the integral sense of the finite element formulation, try to predict the solution accurately within its domain, and the result is that we have a stress discontinuity between elements.

Finally, let us look then at some modeling considerations. In order to solve a problem, we should really have a qualitative knowledge of the response to be predicted. We should also have a thorough knowledge of the principles of mechanics and the finite element procedures available. And this, of course, is the subject of my lectures here to you.

If we have these two knowledges, then we should still know a little bit more about what kind of elements we should use, how one element performs versus another. And here, the first remark that I like to make is that parabolic undistorted elements are usually most effective. Here, please see that I'm saying usually most effective because in some earlier lecture, I talked about the analysis of fracture mechanics problems, where we, in fact, distort elements to pick up stress singularities. But if we don't talk about stress singularities or special situations, usually the parabolic undistorted elements are most effective.

Here, I have put together in a table the kinds of elements that I would recommend for usage. These are typically also the elements that I used in general purpose code such ADINA. Tress or cable element, the 2-node element is quite effective. In some cases, we want to use a three-node element or four-node element, but the 2-node element is cheap and very effective in modeling. For two-dimensional plane stress, plane strain, axisymmetric analysis, the 8-node or 8-node element. Those are the elements most effective, and as I just mentioned, we should use them undistorted, in other words, rectangular.

The three-dimensional analysis then. In three-dimensional analysis, we would effectively use the 20-node brick element, which is really the counterpart of the 8- or 9-node element. For the 3D that is curved, we would use three-noded elements or four-noded elements-- better even four-noded elements. Of course, if we have a straight beam, then we would use simply the engineering [? animation, ?] two-noded beam.

But if you talk about a curved beam, then the isoparametric beam that I presented to you in lecture 7 is very effective. The plate and shell elements that I would recommend are this element here, for plate analysis-- I mentioned that already earlier. And for the shell analysis, the nine-noded element and the 16-node element. Even for plates, it might pay sometimes to use the 16-node element, which is very effective, and particularly works for very, very thin plates and shells. It does not have the deteriorating behavior of the locking phenomenon that I referred to in lecture 7.

Let us look also at some mesh considerations. If we perform an analysis, we might find that in some area of the element idealization. We want to use 4-node elements lower order elements, and in another area, we want to use higher order elements. Well, here, I've show a transition region going from a 4- to a 5- to take an 8- node element. Notice that we have full compatibility along this side. So this is a comparable element transition from 4- to 8-node elements. And similarly, of course, in three-dimensional analysis, you could go from 8-node bricks to 20-node bricks by using such transition elements.

An alternative approach would be to use a 4-node element here, and say, two 4- node elements there, but notice that if you do so, the displacements at node a, have to be constraint, if you want to preserve compatibility here. Notice that along BC, for this 4-node element, we have a linear variation and displacements because we have only UB, VB, UC, VC there. And of course, here, for these two 4-noded elements, we have a linear variation from here to there, and another linear variation from here to there in displacements. To preserve full compatibility along this line, these two constraint equations have to be used. And that means, really, that VA and UA are set to the mean of UB, UC, and VB, BC, as shown here.

Another transition approach is shown here on the last view graphs that I wanted to present to you. And here we're going from 8-node elements, or 7-node elements, if you wanted to use 8-noded elements here, then of course, we would have to put another node in there each. So here we're going from two layers of 8-node elements, over into one layer of 8-node not elements. Notice that in this particular transition region here, we are using a distorted element. Of course, that means the order of accuracy that we can expect here in the stress predictions and so on, is not quite as good as we would like to see it. So this transition region should be away from the area of interest. the area of interest being somewhere over here, far way from the transition reason.

However, we do have a compatible element layout, and that, of course, is an important point, which I mentioned to you earlier, that we would like to preserve compatibility as much as possible. And for that reason, of course, we have

developed these variable number node elements. This concludes what I wanted to say in this lecture. Thank you for your attention.