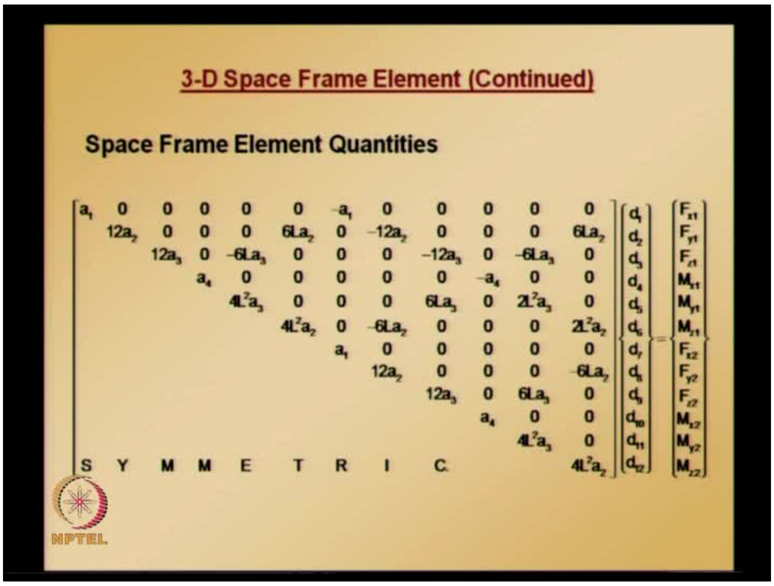
Finite Element Analysis
Prof. Dr. B. N. Rao
Department of Civil Engineering
Indian Institute of Technology, Madras

Module No. # 01
Lecture No. # 14

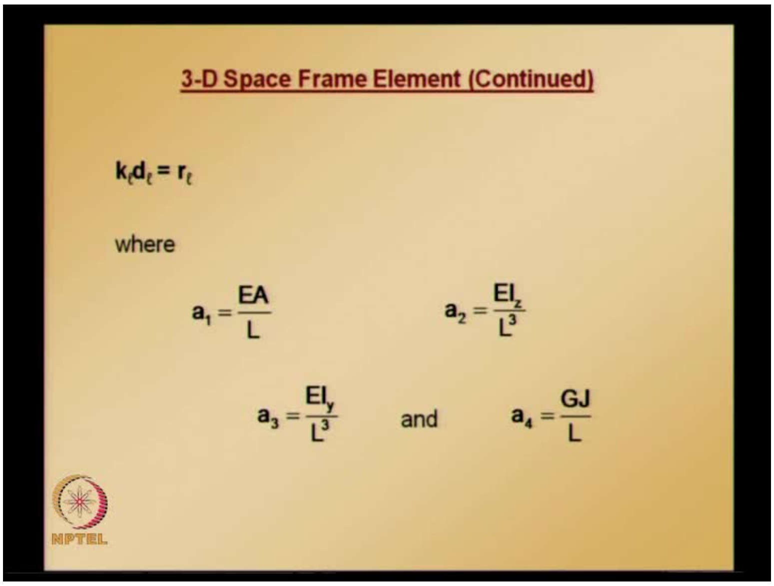
In the last class, we have seen three-dimensional space frame, details of how to derive the element equations, but what we have done in the last class is we have derived element equations in the local coordinate system; and to solve any problem we need to assemble or we need to put together all the element equations for all elements - frame elements - in the local coordinate system. Before we put them together, we need to know what the local to global coordinate system transformation is, and once we transform all the element equations and the global coordinate system, we will be assembling them and then we will be applying boundary conditions; then, we will be solving for the unknown nodal values.

(Refer Slide Time: 01:16)



Let us see, what we have done in the last class; this is the element equation in the local coordinate system for a 2 node three-dimensional space frame element. Please note that while deriving this element equation we assume axial force effects and bending effects in two planes - x y and x z planes, and also torsional effects; all these are uncoupled, and also we made an assumption that this is valid under small deformation condition.

(Refer Slide Time: 02:29)

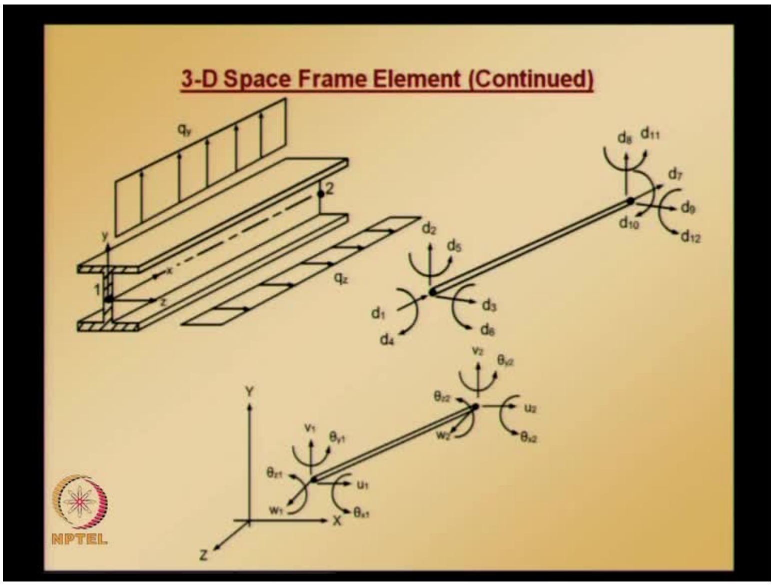


You need to keep this in mind whenever you are using this element equation. Now, here we have some coefficients - a 1 a 2 a 3 a 4, and we have seen in the last class what these are; the element equation in the local coordinate system can be written compactly in this manner: k l d l is equal to r l; the coefficients are given here in - a way - the first coefficient a 1 is some sort of measure of axial rigidity; coefficients a 2 a 3 are some sort of measure for flexural rigidity; and, a 4 is for torsional rigidity.

So, using these element equations we can assemble for all elements frame elements, and each of these equations we need to convert into the global coordinate system. So, in today's class, what we will be doing is, we will be looking at how to get this transformation matrix local to global transformation matrix.

The equations that you have seen just now, that is - element equations, are in the local coordinate system; before assembling these equations we must perform transformation to the global coordinate system in the global coordinate system capital X capital Y capital Z coordinates.

(Refer Slide Time: 04:20)



The nodal displacements and forces in the global coordinate system - capital X capital Y capital Z; we will see both in the local coordinate system. These are the displacement values and rotations of these two node space frame elements in the global coordinate system, that is - u 1 v 1 w 1 are the displacements in X Y Z directions at node 1, and theta x1 theta y1 theta z1 are the rotations at node 1 in the capital X capital Y capital Z coordinate system; similarly, u 2 v 2 w 2 and theta x2 theta y2 theta z2 are the corresponding displacements and rotations at node 2.

Now, we need to know what the relation between the displacements and rotations in the local coordinate system are and the displacements and rotations in the global coordinate system.

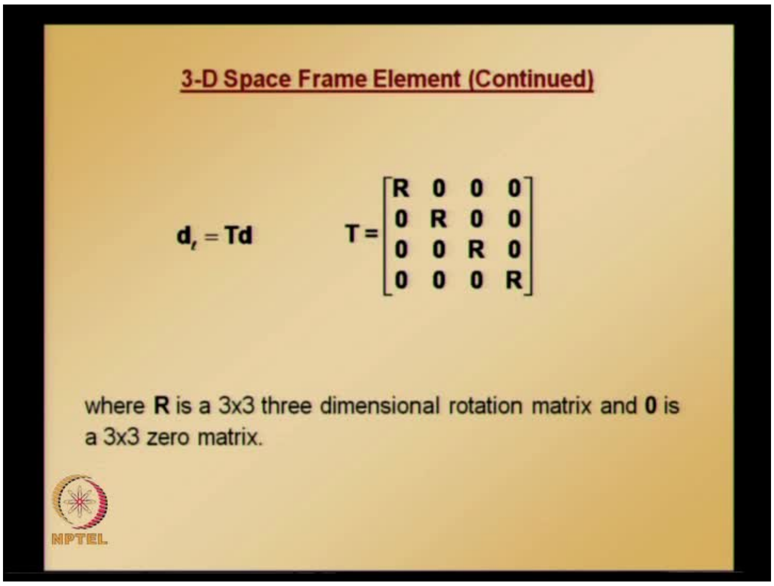
This is what we have in the last class - a space frame is shown in the figure and a local x-axis, local y-axis and local z-axis are indicated there; also the displacements and rotations in the local coordinate system are defined - d 1 d 2 d 3 are the displacements at node 1 in the local coordinate system; d 4 d 5 d 6 are the rotations at node 1 in the local coordinate system; similarly, d 6 to d 12 are the corresponding displacements and rotations at node 2.

Now, we need to know what is the relation between these d's, that is - d 1 to d 12; and, what is the relationship between that vector and v 1 u 1 v 1 w 1 theta x1 theta y1 theta z1 and u 2 v 2 w 2 theta x2 theta y2 theta z2 put together in a vector form - what is the relationship between these two vectors?

The local to global transformation matrix is developed by considering three components of displacements and rotations at each node, as vector quantities; thus, the complete transformation matrix - here you have 12 quantities in the local coordinate system and 12 quantities in the global coordinate system.

So, transformation matrix is obviously going to be 12 by 12 matrix consisting of 4 identical - and this 12 by 12 matrix consist of 4 identical 3 by 3 rotation matrices like here.

(Refer Slide Time: 07:23)

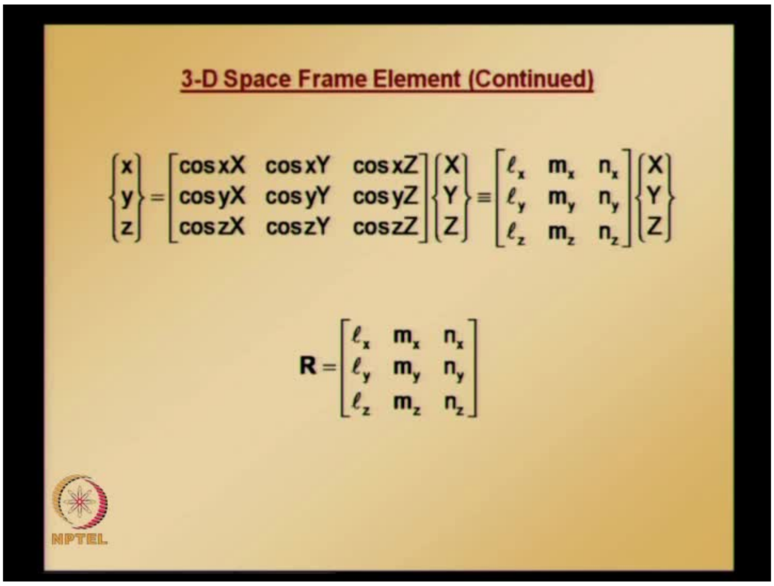


Local displacement vector is compactly written as d l and global displacement vector is written compactly as d, and the relationship between local displacement vector - which includes both displacements and rotations - the relationship between the local displacement vector and global displacement vector is given there; they are related through transformation, and the transformation consists of 4 identical 3 by 3 rotation matrices; so, total size of transformation matrix is going to be 12 by 12.

Now, we need to know what this R looks like; so, R is a 3 by 3 three-dimensional rotation matrix; 0 there is also a 3 by 3 null matrix - 0 matrix.

Now, the rotation matrix R transforms a vector quantity from local to global coordinate system; since, the displacements, rotations, forces and moments are all vector quantities they can all be transformed using this R matrix. The components of a vector along local x y z coordinates are simply projections of its components in the global coordinate system - capital X capital Y capital Z, along the local axis; in matrix form this transformation matrix R can be written in this manner (Refer Slide Time: 09:22) Which you already saw when we were solving space trusses - this transformation matrix. The components of this rotation matrix are simply direction cosines involving angle between the local coordinate system and global coordinate system.

(Refer Slide Time: 09:18)



Here, cos of, let us say, small x capital X is cosine of angle between small x and capital x-axis; similar meaning is applicable for all other components; the rotation matrix here - this whatever equation here - it gives a relationship between local coordinate system x small x small y small z and the global coordinate system capital X capital Y capital Z; with this we can define a rotation matrix R like this - which consists of all the direction cosines l x is nothing but cosine of angle between small x and capital X coordinates or axis; similar meaning is where applicable for all other components - m x n x l y m y n y l z m z n z.

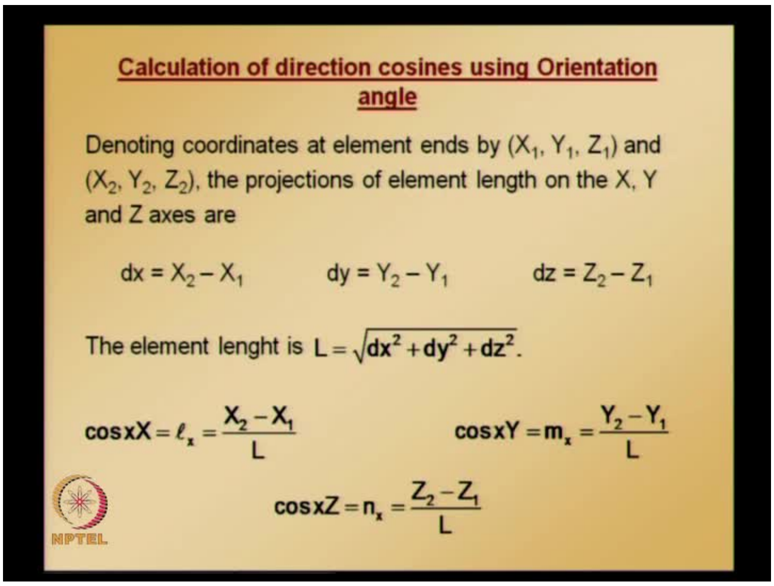
So, 9 direction cosines are needed to establish this rotation matrix R for each element in a space frame; so, just knowing the element and coordinates is not enough to establish all the 9 direction cosines; like, if you recall, in truss problems - when we are solving truss problems - we used only l x m x n x, but here we require - even though this transformation matrix is written at the time we have looked at this one, but we have not used l y m y n y l z m z n z. To calculate these - what I am saying is - just knowing the element and coordinates is not enough, we need additional information about one of the local principle axis; we will see some methods to see how to calculate all the components.

There are two methods to accomplish this one - the first method needs an orientation angle, and the second method needs coordinates of a third point that explicitly define either local x z or local x y plane of bending.

Formulas for calculating these direction cosines based on the two methods that I just mentioned, that is - one method is called orientation angle method and the other method is called third node method, we will be looking in detail at these two methods to calculate all the components of the rotation matrix, that is, l y m y n y l z m z n z in addition to l x m x n x.

Now, let us look at each of these methods for calculating the direction cosines for - finally we will be using it to solve this 3D space frame problem; now, for a while we will be concentrating on the two methods for calculating these direction cosines.

(Refer Slide Time: 13:37)

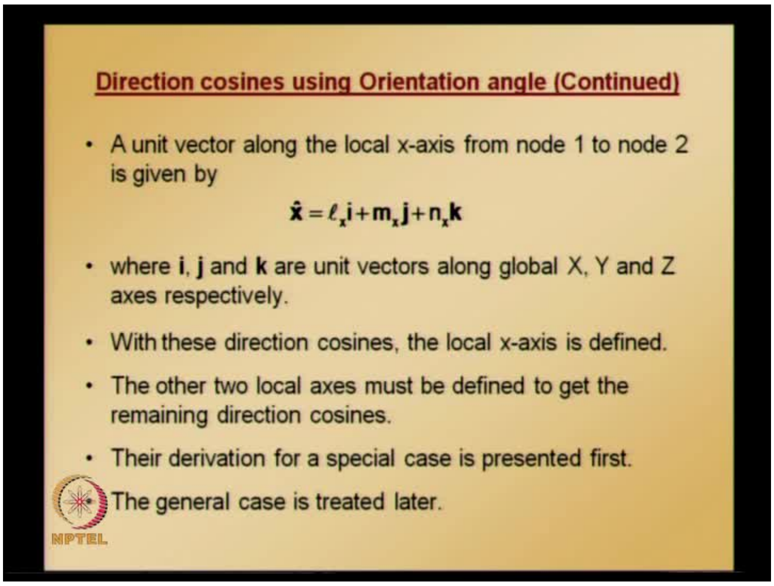


Denoting the coordinates at element ends, I am taking a two node space frame element; let the coordinates at the element ends be denoted by capital X capital Y capital Z - that means, we are here basically expressing the coordinates in the global coordinate system; one end the coordinates are capital X 1 capital Y 1 capital Z 1, the other end coordinates are capital X 2 capital Y 2 capital Z 2.

The projections of element length on capital X capital Y capital Z axes - you are already familiar with this: dx is equal to difference between the x coordinates of the two end; similarly, dy dz, so once we know dx dy dz we can calculate the length of this space frame element; also, if you recall, we have obtained when we were solving 3D space truss problems - also l x m x n x, that is, cosine of angle between small x-axis capital x-axis; similarly, cosine of angle between small x-axis and capital y-axis that means angle between the local x-axis and global x y z-axis they are l x m x n x; and, you have seen how to calculate this even while solving the 3D space truss problems.

This gives us - using for a space frame element on knowing the coordinates of the end points, this is how we can calculate l x m x n x; once we have this l x m x n x we can write a unit vector along local x-axis from node 1 to node 2, that is, a local x-axis is defined connecting the end having coordinates capital X 1 capital Y 1 capital Z 1; and, node 2 is defined as the end having coordinates capital X 2 capital Y 2 capital Z 2.

(Refer Slide Time: 16:43)



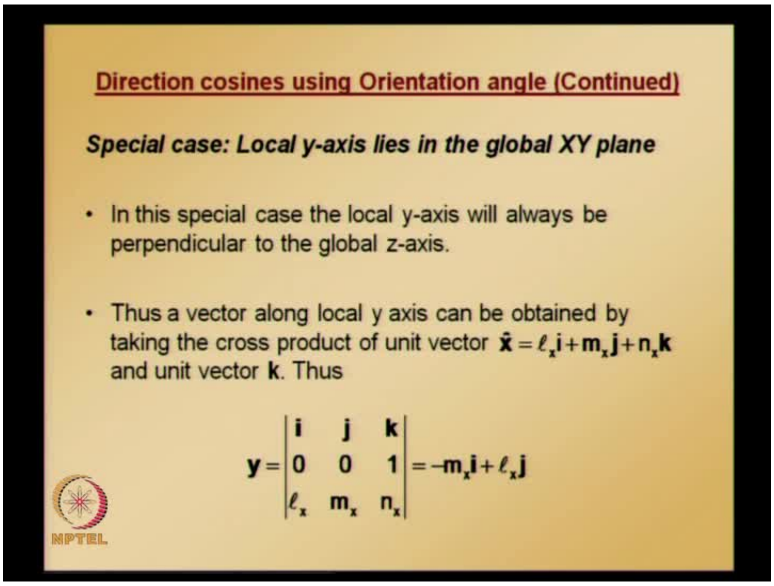
So, a unit vector along the local x-axis from node 1 to node 2 using this l x m x n x we can write it as this one; so, this is the unit vector from going from node 1 to node 2, where i j k are unit vectors along global X global Y global Z axes respectively. With these direction cosines local x-axis is defined and the other two local axis must be defined to get the remaining direction cosines.

If you recall rotation matrix components, we need to know the orientation of local x local y local z axes; so, we need to know other two local axis to get all the components of the rotation matrix.

To get the other two local axes - here first we will be looking at a procedure or derivation for a special case first, and then later that special case will be generalized for any case.

What is this special case? In the special case, let us assume, that the local y-axis lies in global XY plane.

(Refer Slide Time: 18:25)



Basically, what we have done till now is - we got the unit vector along local x-axis; so, in a way we know the direction of local x-axis. We need to know the other two axis, so we are looking at a procedure to calculate the other two local axes.

Here, we are looking at a special case, where, let the local y-axis be in the same plane as global XY; in this special case local y-axis will always be perpendicular to global-z axis.

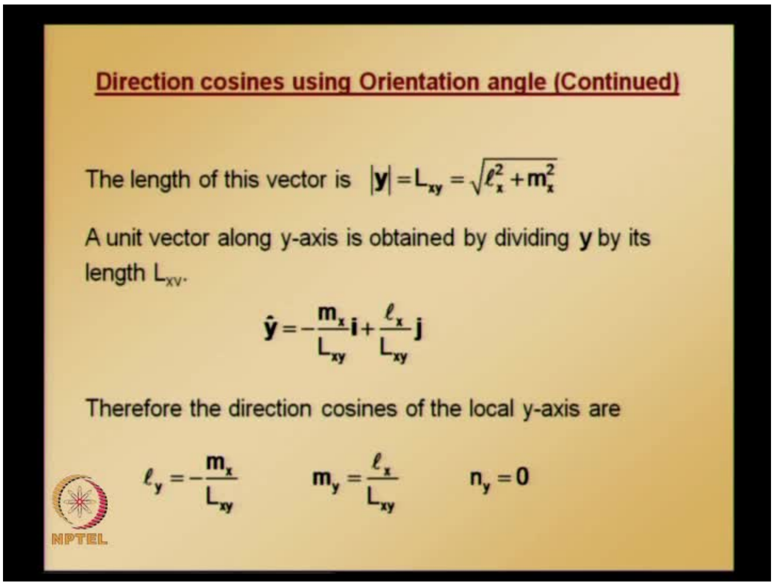
So, a vector along local y-axis can be obtained by taking cross product of unit vector along local x-axis, which we just calculated using l x m x n x; and, a unit vector along global-z axis. Please note, that a unit vector along x-axis is given by I; a unit vector along y-axis is given by j; and, a unit vector along z-axis is given by k. You know that the cross product of a vector along x-axis with a vector along y-axis gives us z-axis; a vector along z axis, similarly, cross product of a vector along y-axis and z-axis gives us a vector along x-axis; and, we also know that the cross product of vector along z-axis and x-axis gives us a vector along y axis.

With this, what we are basically doing is - we got unit vector along local x-axis and we are dealing with a special case in which local y-axis is in the same plane as global XY plane; and, also we know that local y-axis is going to be perpendicular to the global z-axis, and a unit vector along global z-axis is given by this k.

If we take a cross product of unit vector along global z-axis and unit vector along x-axis, which we just obtained, we get unit vector or vector along - it is not going to be unit vector - we are going to get a vector along local y axis.

Cross product of k and unit vector along local x-axis gives us a vector along local y-axis, and this is not going to be unit vector - we need to normalize it. So, the length of this vector can be obtained from m x and l x and using the length of vector if we normalize this, then we are going to get unit vector along local y-axis.

(Refer Slide Time: 22:54)

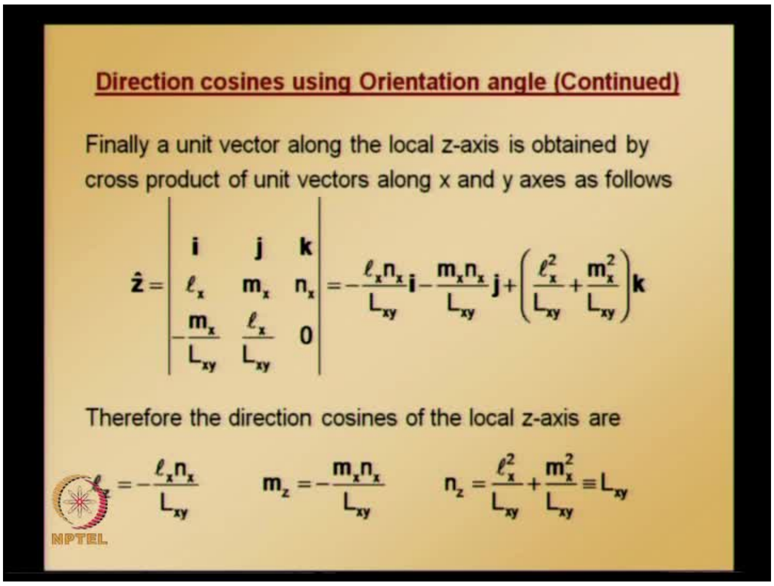


This is the length of this vector. Unit vector along local y-axis is obtained by dividing vector y with its length.

So, the direction cosines in the local y-axis are these. Now, what we have done is, we have calculated direction cosines in the local x-axis and using vector product between unit vector along global z-axis and local unit vector along local x-axis, we obtain a vector along local y-axis and by normalizing it we got the direction cosines of local y-axis.

Now, we have got local x-axis and we have got local y-axis. If we take cross product of it - cross product of these two - that is, a vector along local x-axis and a vector along local y-axis, if we take a cross product of these two we are going to get a vector along local z-axis; again, normalizing that with respect to its length we are going to get unit vector along global z-axis; so, this is how this procedure goes.

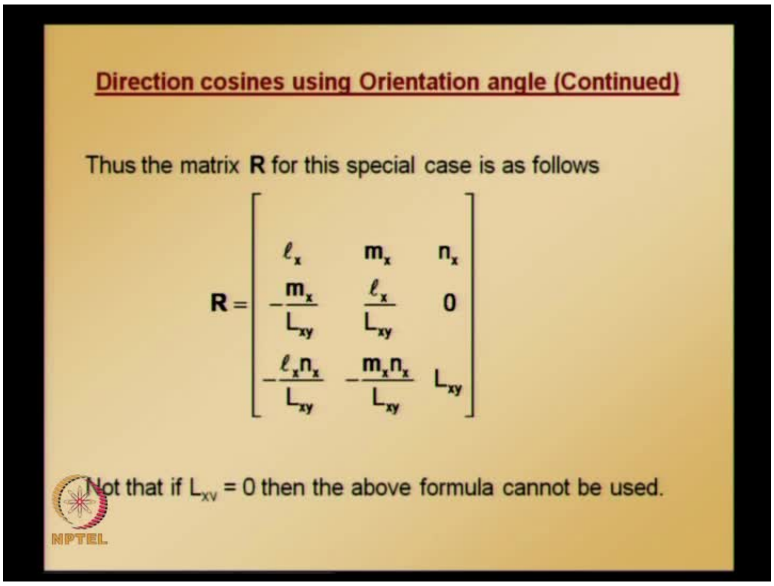
(Refer Slide Time: 24:40)



Now, finally a unit vector along local z-axis is obtained by cross product of unit vectors along local x and local y axes. This is not going to be a unit vector unless we normalize it or we can take product of unit vectors along local x-axis and local y-axis and then we automatically get unit vector along local z-axis.

So, the direction cosines once we obtain this vector we can easily find what are the direction cosines; direction cosines of local z-axis are l z m z n z; so, we obtain all the direction cosines - we got l x m x n x, we also obtained l y m y n y, and just now we got l z m z n z; then, we can put all these together and we can get the rotation matrix.

(Refer Slide Time: 26:08)

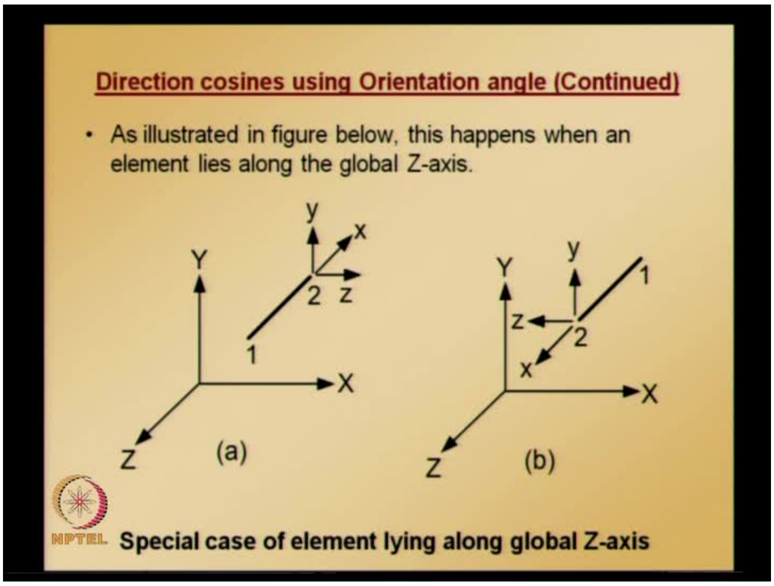


The rotation matrix for this special case - what is this special case? Special case shear is local y-axis is in the same plane as global XY plane; so, under that condition rotation matrix can be obtained using this equation.

But, this equation has a small problem - let us say, here in the denominator you have L xy; if L xy is 0 then will have problem with this formula. Note that, if L xy is 0 then the above formula cannot be used; because, L xy is in the denominator - when you have something in a denominator it goes to infinity, so we will have problem with this equation and calculating the rotation matrix.

We will see under what circumstances L xy is going to be 0; as illustrated in the figure below, this happens when element lies along global Z-axis.

(Refer Slide Time: 27:29)

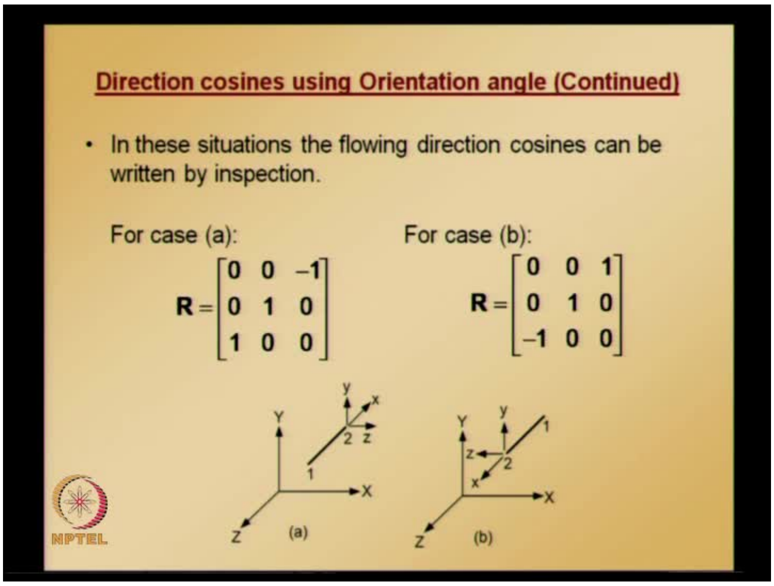


Here, two cases are shown: figure a and b; in both cases you can see L xy is going to be 0 and L xy is defined as l x square plus m s m x square root of that.

So, if you see these two vectors which are oriented in the way they are in figures a and b you can easily check that l x and m x are 0 for this; so, L xy is going to be 0.

In this special case, what we can do is we can write the rotation matrix. Instead of using that formula in that equation we can actually go back and carefully inspect these and see - basically, what is rotation matrix? It consists of cosine of angle between local axis and global axis.

(Refer Slide Time: 29:50)



We can easily figure out by inspection - instead of going and using that formula - we can easily inspect and find what is cosine of angle between small x capital X, cosine of angle between small x capital Y, similarly other components of the rotation matrix and doing that in these situations, the following direction cosines can be written by inspection: for case (a) you can easily do that kind of inspection and write what is rotation matrix. And it turns out that rotation matrix for a vector which is oriented as shown in figure a is given by whatever rotation matrix is there on the slide under - for case (a).

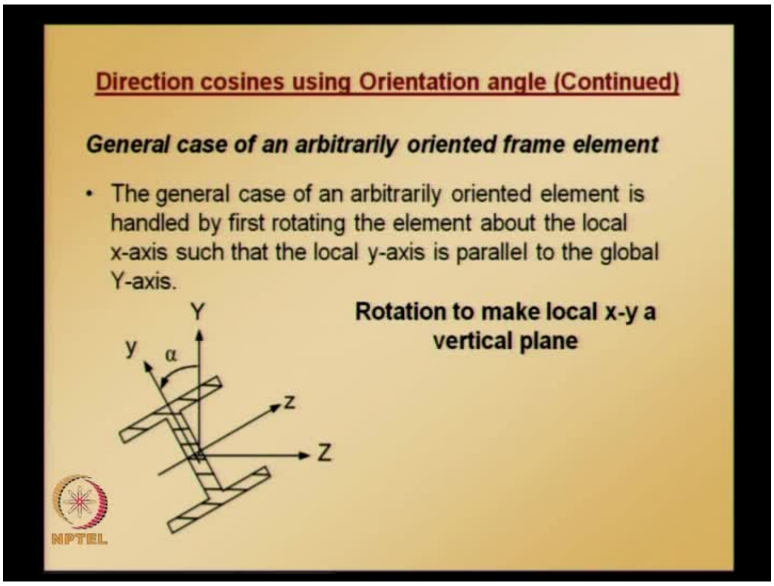
Similarly, if space frame element is oriented in the way it is shown in figure b, rotation matrix can be obtained by inspection and rotation matrix looks like whatever is shown in the slide under - for case (b).

You need to keep these things in mind, not just blindly applying the formula; because, this formula is going to fail under some circumstances in that case we need to get this rotation matrix by inspection.

What we have seen so far is - we have seen a case in which local y-axis is in the same plane as global XY or local at y-axis is in global XY plane, so under that special case this is how we can calculate rotation matrix.

If L xy can be calculated we can use the formula; if L xy cannot be calculated and if it 0 because l x m x are 0, in that case, we can calculate by inspection. Now, let us look at a more general case - general case of an arbitrarily oriented frame element.

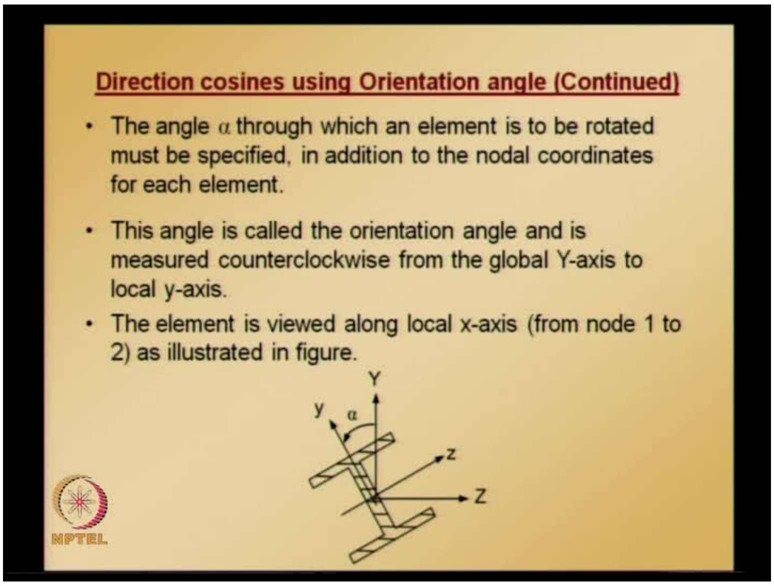
(Refer Slide Time: 32:02)



The general case of an arbitrarily oriented element is handled by first rotating the element about local x-axis, such that, local y-axis is parallel to global Y-axis. So, what we will do is - in the previous case you have local y-axis is in the same plane as global XY, and this is a more general case in which this local y-axis is not in the same plane as global XY; and, in this case what you can do is, you can rotate this frame element about local x-axis such that local y-axis is parallel to global Y-axis - that is shown here. Since, rotation is about x-axis - here in this figure - x-axis is coming out of the board so it is not shown, so the axes which is coming out of the board that is x axis.

Rotation is taking place about x-axis, and somehow do some rotation about local x-axis to make this local y-axis is parallel to the global Y-axis; this alpha is angle between local y-axis and global y-axis measured in the counter clockwise direction.

(Refer Slide Time: 34:21)



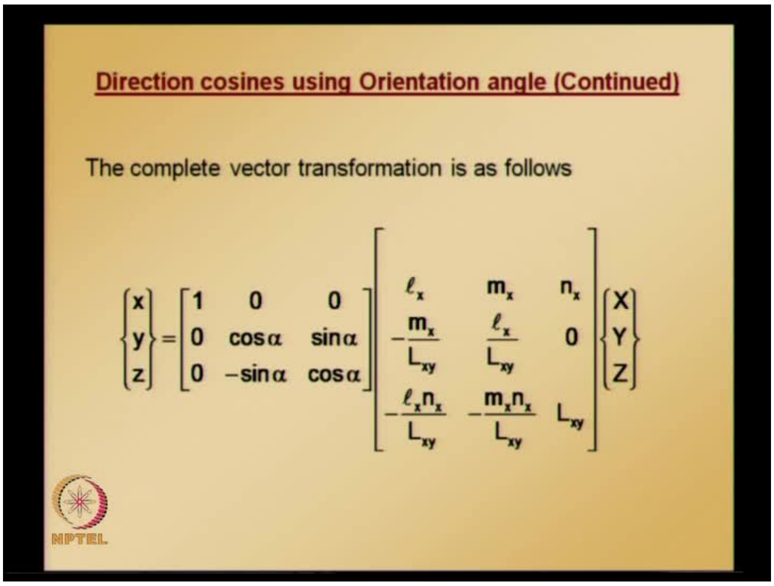
When we want to use this method both the angle alpha through which frame element is to be rotated must be specified, in addition to the nodal coordinates or end coordinates of this particular space frame element.

We also need to know this angle by which we need to rotate to make this local y-axis to become parallel to the global Y-axis; this angle is called rotation angle and is measured counterclockwise from global Y-axis to local y-axis.

That is why this method is called orientation angle method, and here as I mentioned element is viewed along local x-axis that is why you are unable to see x letter there; because, the axis which is coming out of the plane of the board is local x-axis.

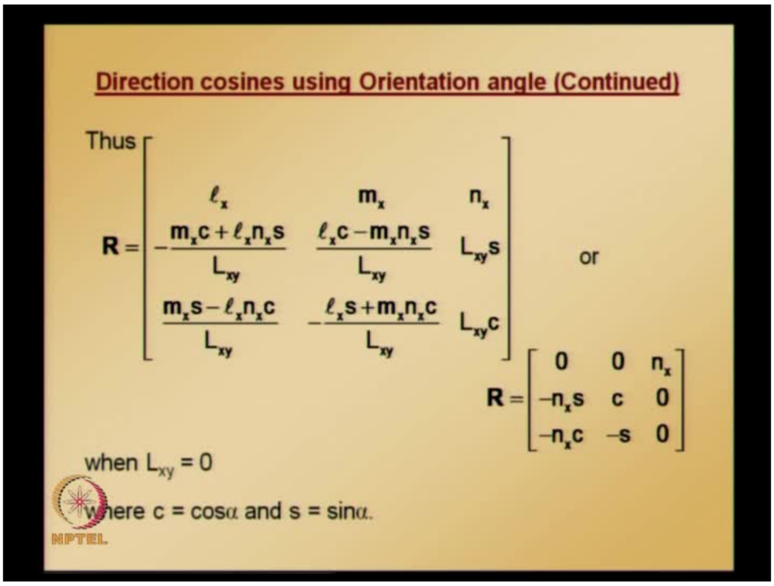
Once we rotate in this manner, and once we make local y-axis to be parallel to the global Y-axis then rest of the procedure is same; and, as what we discussed for the special case.

(Refer Slide Time: 35:58)



So, the complete vector transformation is given by this; if you see the first part of this transformation it is nothing but - coming from rotation that we are doing about local x-axis to make local y-axis to be parallel with global Y-axis; second part of this transformation is the same as what we got for the special case.

(Refer Slide Time: 36:50)



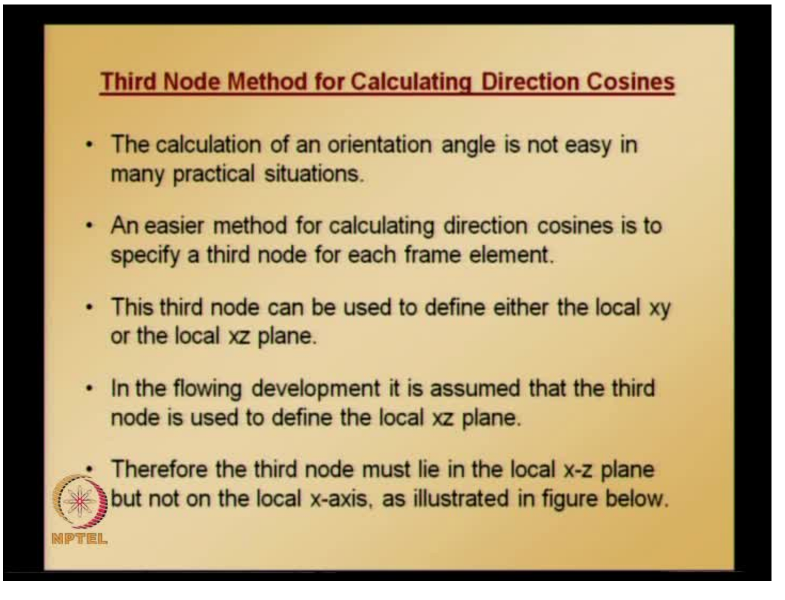
You need to just multiply the first part to the second part, which we already obtained for the special case; once this multiplication is performed then rotation matrix for this case - for arbitrarily oriented frame element case - is given by this; this is more general, this is applicable even for the case where local y-axis is in the same plane as global XY, and here also L xy is in the denominator and in the special case where you have seen earlier if L xy is 0 then there is a problem in using this equation.

So, in the limit L xy tends to 0 this rotation matrix becomes this; so, in the case L xy is equal to 0 you can use the second equation that is shown on the slide, which is applicable when L xy is equal to 0 and c is nothing but cosine of angle alpha and s is nothing but sine of angle alpha.

Alpha is - we already looked at - alpha is angle through which the frame element is to be rotated about x-axis - local x-axis - such that local y-axis is parallel to the global Y-axis.

This is how we can obtain rotation matrix or matrix consisting of direction cosines, which we will be finally using for transformation of global displacement vector or local displacement vector - the global displacement vector or vice versa.

(Refer Slide Time: 39:00)

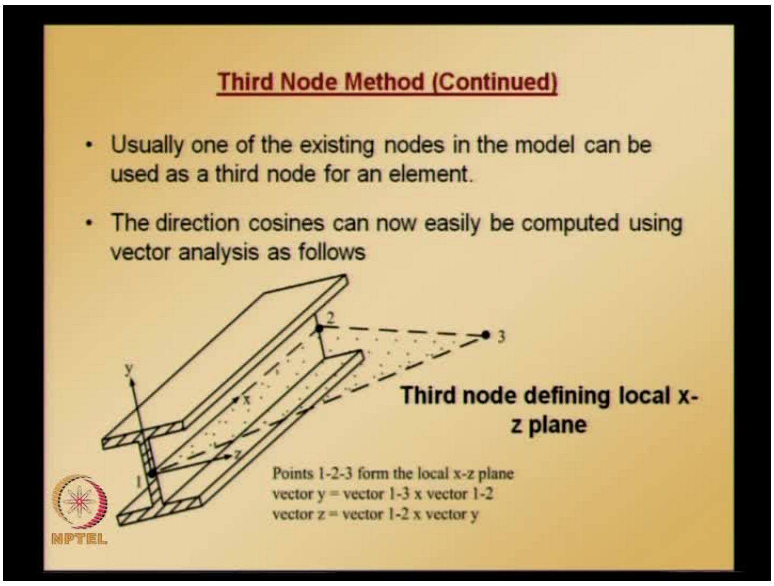


Now, look at the other method, which is called third node method for calculating direction cosines. So, choice is going to be yours - you can use any of these methods depending on the information that is available.

In third node method for calculating direction cosine - calculation of an orientation angle is not easy in many practical situations; under this case you can use this third node method. An easier method for calculating direction cosine is to specify a third node for each frame element and how to specify these and all these details it will be clearer to you when we actually look into more details in a while.

This third node can be used to define either local xy or local xz planes; in the following development it is assumed - means, here we are going to derives equations - it is assumed that third node is used to define local xz plane; therefore, third node must lie in the local xz plane but not along local x-axis; this is going to be clearer to you when you look in the figure.

(Refer Slide Time: 40:35)



Usually, one of the existing nodes in the model can be used as third node for an element. Direction cosines can now be easily computed using vector analysis.

This is how you can define the third node; if you see here a arbitrarily oriented space frame element is shown with local x-axis, local y-axis and local z-axis shown there; and, node 1 and node 2 are the two end points of this element - space frame element - and node 3 is third node is that is the node that we require for defining local x-y plane or x-plane in this case it may be x-y plane in some other case.

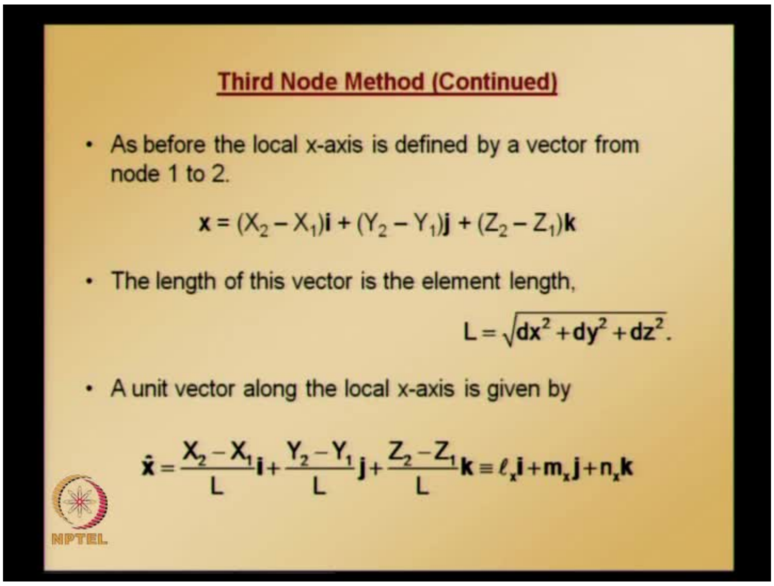
This third node - where can we obtain this third node from? You can choose because here only one space frame element is shown for illustration purpose, but in general if you take any structure you will have many frame elements; so, this third node you can take from some other node of some other member. This will be clearer to you when we look at a problem.

Points 1-2-3, you can see from the figure form local x-z planes; vector y we can obtain by taking cross product of vector 1-3. If third node is specified we can find what is the vector going from node 1 to node 3, once we also know the local vector in the local x-axis that is the vector going from node 1 to node 2.

If you know these two vectors and if we take cross product of these two we are going to get vector along local y-axis; once we get vector along local y-axis and if we take cross product of vector along 1-2 - vector along nodes 1-2 - is nothing but local x-axis a vector along local x-axis; if we take cross product of that with a vector along local y-axis we get a vector along local z axis. So, this is the basic logic behind this method. So, once the third node is defined we can do all the calculations to calculate local x-axis, local y-axis and local z-axis vectors along these; once we know the vectors along local x-axis, local y and local z-axis, we can find what are the direction cosines; and, once we know the direction cosines we can get the rotation matrix, through which we can do the transformation.

Now, let us see, how to get vector along local x-axis. You can see local x-axis is lying along a vector, which is going from node 1 to node 2; so, we can easily calculate - as before local x-axis which is defined by vector going from node 1 to node 2.

(Refer Slide Time: 44:18)

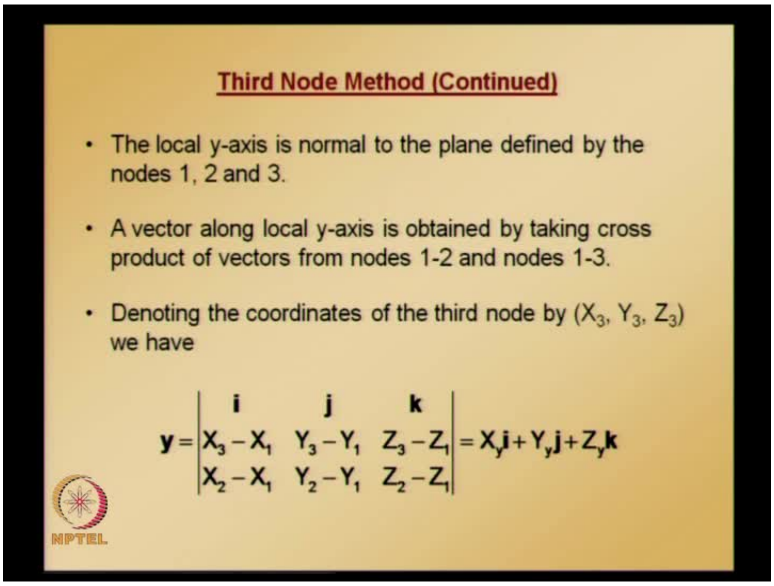


So, vector along local x-axis we can obtain using this. And, if you normalize this you are going to get unit vector; length of this vector is given by dx square plus dy square plus dz square; dx is defined as capital X 2 minus capital X 1; dy defined as capital Y 2 minus capital Y 1; dz defined as capital Z 2 minus capital Z 1.

When you normalize this vector along local x-axis with its length we are going to get unit vector along local x-axis. Now, local y-axis is normal to the plane formed by nodes 1 2 3.

First we need to calculate vector connecting nodes 1 and 3; and, once we take cross product of that with respect to a vector connecting nodes 1 and 2 we are going to get the vector along local y-axis. Local y-axis is normal to the plane defined by nodes 1 2 3, a vector along local y-axis is obtained by taking cross product of vectors from nodes 1-2 and nodes 1-3.

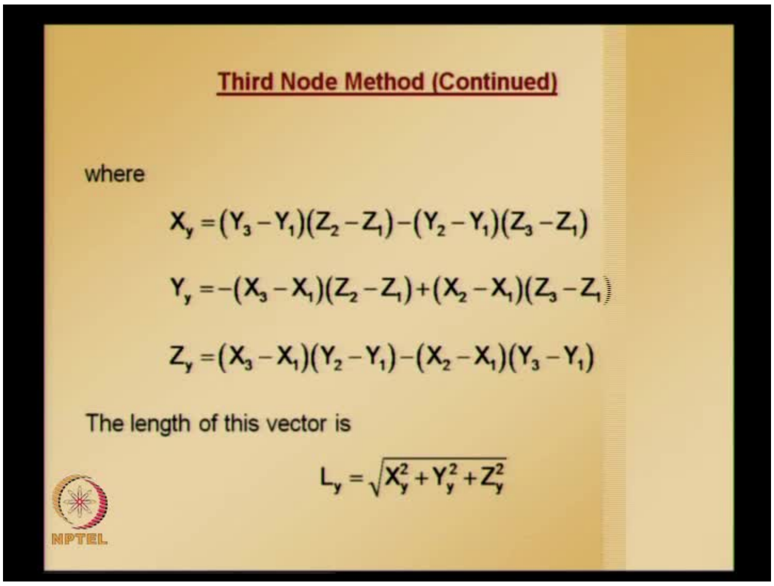
(Refer Slide Time: 45:58)



If we denote the coordinates of third node by capital X 3 capital Y 3 capital Z 3, we can easily find the direction cosines of vector connecting nodes 1 and 3. Once we know these we can take cross product of the vectors connecting nodes 1 3 and 1 2 then we are going to get vector along local y-axis.

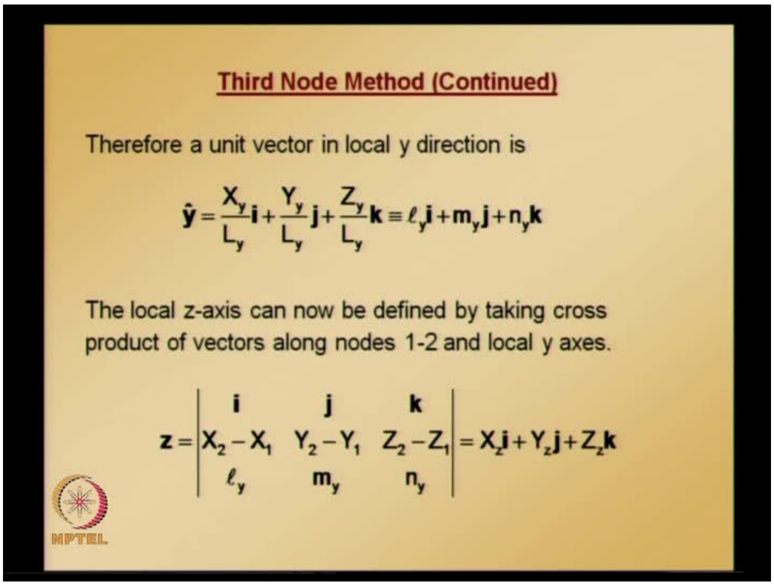
This vector along local y-axis can be obtained by taking or finding this cross product, and here this is compactly written in terms of X y Y y Z y; and, these are defined like this - after carrying over the cross product we get this.

(Refer Slide Time: 47:17)



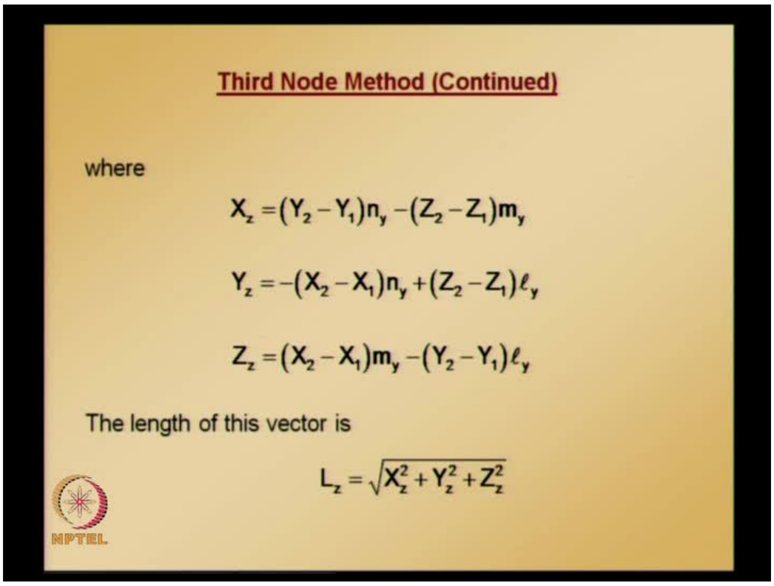
Length of this vector is given by square root of X y square plus Y y square plus Z y square; square root of that gives us length, and once we have the length multiply or divide the vector with its length we are going to get unit vector along local y direction.

(Refer Slide Time: 47:50)



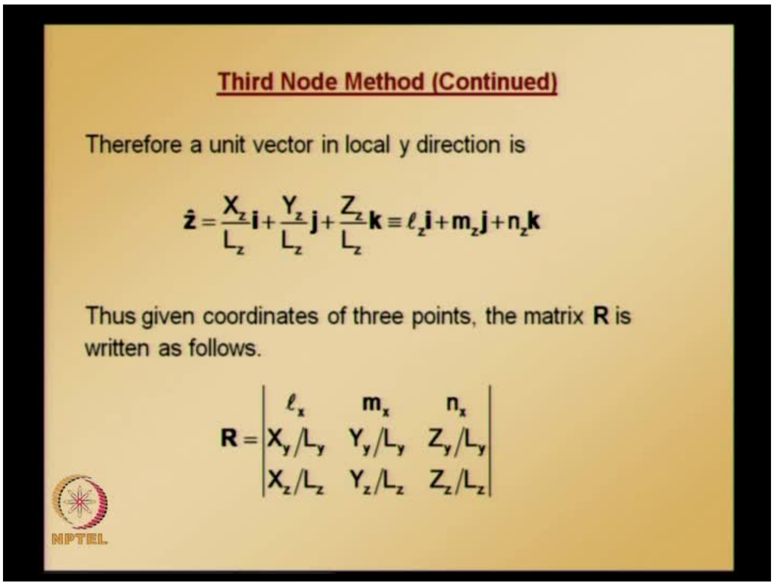
So, we have got unit vector along local x-axis unit vector along local y-axis; then, it is straight forward - vector along local z-axis can now be defined by taking cross product of vectors along nodes 1 2 - that is nothing but local x-axis and local y-axis. That cross product, once it is performed, we are going to get a vector and it is compactly written as X z Y z and Z z the meaning of these is given here.

(Refer Slide Time: 48:38)



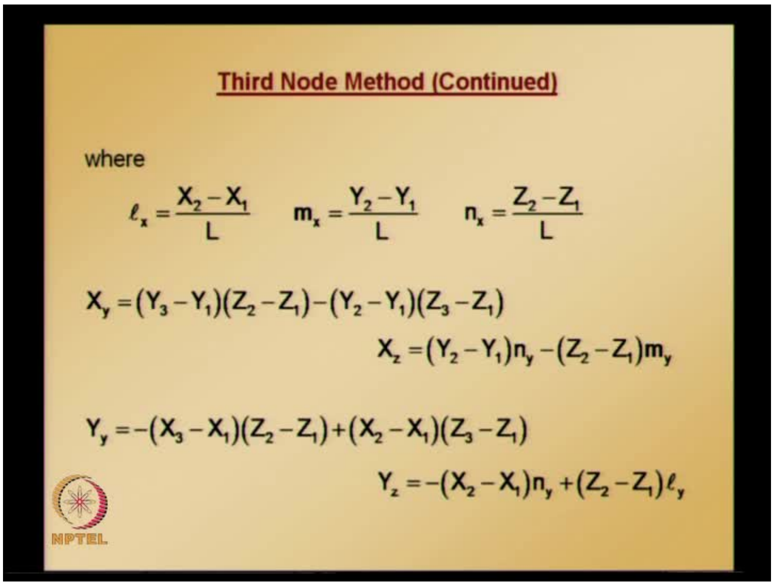
The magnitude of this vector which is along local z-axis is given by square root of X z square plus Y z square plus Z z square; so, that is going to be the length of this vector. When we normalize this vector, which is along local z-axis with its length we are going to get unit vector in local - there is a typing mistake there, it should be in the local z direction - unit vector in the local z direction is given by this.

(Refer Slide Time: 49:15)

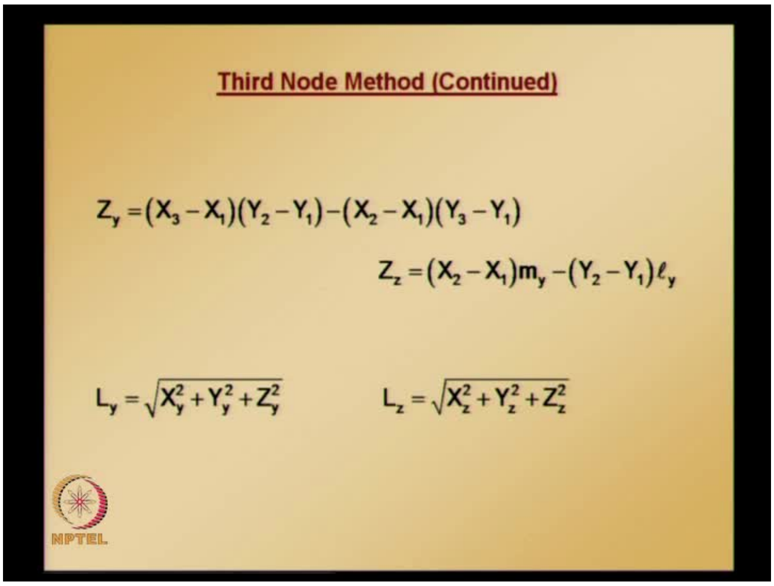


So, we obtained unit vectors along local x-axis, local y-axis and local z-axis; now, we are having all the information to get the rotation matrix. Thus, given the coordinates of three points - that is, coordinates of nodes connecting, or the coordinates of end points of the frame element along with that if a third node is specified, we can do this vector analysis to get this rotation matrix; and, rotation matrix can be obtained in this manner where all the components of this rotation matrix the details are repeated once again here.

(Refer Slide Time: 50:35)



(Refer Slide Time: 50:50)



So, this is how we can obtain rotation matrix - either by using orientation angle method or by using third node method. So, once we get this rotation matrix we can write transformation matrix from local to global coordinate system for this three-dimensional space frame element.

In the next class, we will be looking at a numerical example where we will be using all these concepts.