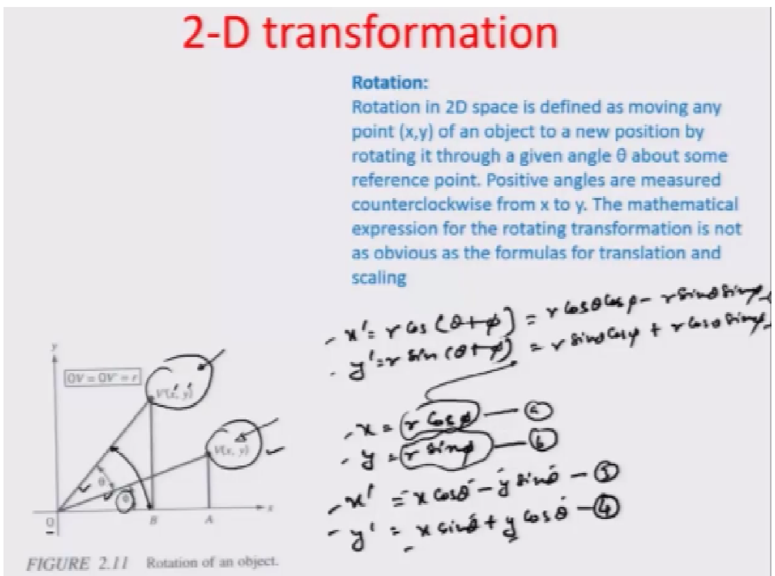
Design Practice - 2
Prof. Shantanu Bhattacharya
Department of Mechanical Engineering
Indian Institute of Technology-Kanpur

Lecture - 03
Geometrical Transformation

Hello and welcome to this Design Practice 2 module 3. In the last module we were discussing some geometrical transformations related to very simple operations like translation in 2D space as well as scaling. We also just about introduced the rotation and we would like to do that in detail today.

(Refer Slide Time: 00:40)



So when we talk about rotation a particular problem in hand could be something like a vector V which has coordinates x, y rotated by an angle theta as you can see right here in this figure and the new position of the vector V goes to V dash where the location coordinates could be x dash and y dash and the only difference here is that you know although the coordinates are changing but the radial vector that is from the origin O all the way up to the point V remains similar.

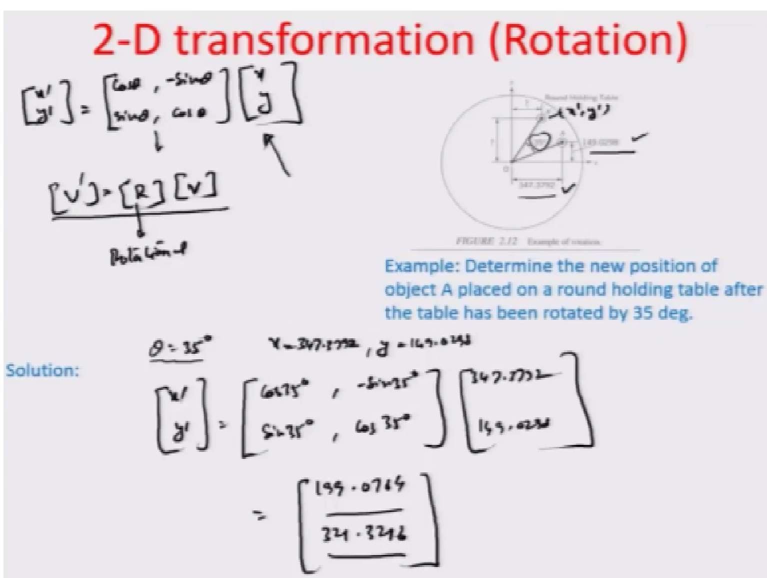
We can also assume that V was located in a manner that the radial vector make an angle phi with respect to the x axis and we have to sort of transform or make a transformation equation in a manner so that in terms of x and ys and this vector r and the angles that are given we could be able to predict what is the x dash and y dash. For doing that let us just look geometrically how we will express this different coordinates.

The coordinates x dash in this particular thing can be represented as very simply as r cos of theta plus phi where theta plus phi is the complex angle as the vector moves to the new location V dash with respect to the x axis. Similarly, y dash can be represented as r sin of theta plus phi and from this information we can probably get you know the representations of x dash and y dash in terms of x and y coordinates.

Earlier, if we were to look at how V was placed x could have been represented there as r cos of phi and y could have been represented as r sin of phi. So if I open these complex angles the relationship that would emerge is r cos theta cos phi - r sin theta sin phi and y dash could emerge out to be r sin theta cos phi + r cos theta sin phi.

And we already know that these two equations a and b we call them are having or are indicating the relationship between the radial vector and the angle phi which was initially there for the vector V in space with respect to x and y. So if I wanted to just substitute these values back into these two equations which we formulated let us call them 1 and 2.

(Refer Slide Time: 02:58)



The whole expression could then be written down as x dash equals x times of cos theta - y times of sin theta. And y dash equals x times of sin theta + y times of cos theta. So these are the two emerging equations which would be transforming the x dash y dash in terms of the angle that the vector V had rotated by and the old coordinates at V which were x and y respectively.

So I can actually very simple, in a very simple manner represent this equation in a you know in terms of a matrix equation and we can say that x dash y dash in this particular case equals cos of theta - sin of theta and sin of theta cos of theta times of x and y. So if I simply applied the mathematics multiplication rule I would have equations 3 and 4 emerge which would give a relationship between x dash y dash and x and y.

So in other words this being the rotational transformation vector can map the vector V into the vector V dash through a simple linear form of equation V dash equals R matrix times of V matrix. This is known as the rotational matrix. Now one thing one has to be very careful about is that really rotation is not just about the angle but also the plane in which that angle has been executed. For example in the earlier case where V rotated to V dash we can see that the rotation is described about the z axis.

In this case does not change really because the rotation is about that particular axis okay. So in any event if we can we construct a 3D space in mind there can be rotation about any of the axis. For example there could be rotation about the x axis or rotation about the y axis and we will revisit this letter when we do 3D transformation and try to represent rotation in the 3D space where we will see that about the x, y, and z all the 3 axis there will be different transformation equations which will actually transform the points very easily after rotation mathematically.

So let us do some problem example in this rotational transformation in 2D. Let us say we have a point A in space which is given by the location coordinates 347.3792 x coordinate and 149.0298 y coordinate and we are rotating this point A to a point A dash by an angle 35 degrees and we want to find out what are the new coordinates just using this transmission equation that we have formulated in the earlier step.

So we will substitute theta as 35 degrees and when the theta is in the anticlockwise direction the convention says that you should treat theta to be positive whereas if it is in the clockwise direction the theta is generally treated as negative. So the same trigonometric conventions will be used in the transformation at hand as well. So we will consider this to be positive 35 degrees because it is moving in the counterclockwise direction.

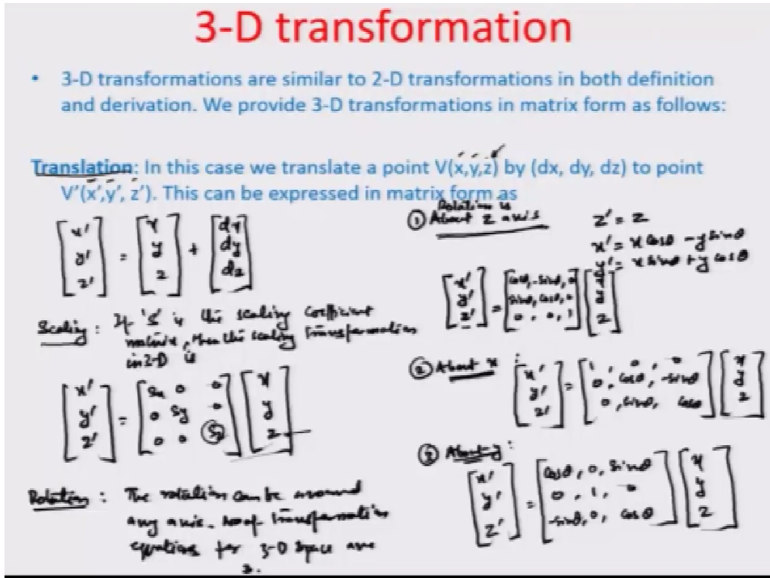
The coordinates x and y are represented again by two very rather long numbers 347.3792 and 149.0298. Intentionally it has been made so that you cannot just, you have to use the matrix transformation, otherwise, becomes little complex. So when we talk about finding out what are the location coordinates of x dash y dash which is the point A dash in space after the rotation has been executed, the x dash comes out to be your let us say the matrix x dash y dash comes out to be equal to cos of 35 degrees - sin of 35 degrees.

Sin of 35 degrees cos of 35 degrees times of these two x and y coordinates okay which represents the initial state of the vector at point A. So if we apply the matrix multiplication rule here, the x dash and y dash would come out to be equal to 199.0764 and 321.3216. So that is about how x and y can emerge into x dash y dash executed through a rotation.

Now the advantage behind this process is that supposing you have multiple rotations in space that normally is the case when we are talking about solid modeling where we are having a model and we are turning it around, rotating it at different locations and then re-representing it or replotting it a very easy computational way is emerged for finding out what are those new coordinates which will need to be joined to formulate the object in the rotated manner as was you know from the initial object.

So it gives kind of quick way of computation for the CAD backend software to figure out what would be the new location provided the old locations and the rotation angle were mentioned duly and the axis about which the rotation took place mentioned duly. So let us now go from 2D to 3D space and try to see how these transformations would change.

(Refer Slide Time: 08:47)



So when we talk about 3D transformation obviously there is an addition of a new coordinate z in addition to the x and y which would happen and in case of translation as we had seen we will just incorporate another new coordinate here and say that the translated vector x dash y dash z dash can be represented as the original vector which was x, y, z plus a distance vector which is actually the distances along the x, y, and z axes respectively as one move in in the 3-dimensional space.

So a point V which was at initially x, y, z location if translated by dx, dy, dz location, definitely moves to another point V dash which was which is represented as x dash y dash z dash location. So that is how we can represent these translation case. Similarly, we can also represent the scaling case in this particular case let us say the scaling which would come out is that you know if s is let us say the scaling coefficient matrix.

Then the scaling transformation in 3D is provided by or is given by x dash y dash z dash equals to sx 0 0, 0 sy 0, 0 0 sz times of x y z and if I wanted to, so I am just adding an additional coordinate z here and the scale along the z direction in order to translate this into the 3D scaling transformation equation and then when we talk about rotation so obviously we have already recorded that rotation can be around any of these axes x y and z.

So the rotation can be around any axes. Therefore the number of transformation equations for 3D space are 3 one along the x axes, one along the y axes and along the z axes. So let us look at those equations. I already talked about one along the z axis. So let us say when the rotation is about z axes the total equations which will emerge is that the z coordinate does not change. So we write z prime as z; z prime being the later on coordinate after the rotation is executed.

Similarly, the x prime which is the later on x coordinate can be represented as x cos theta - y sin theta. We just did it in the 2D case before this okay and then y dash or y coordinate changes to y dash and so y dash can be represented as x sin of theta + y cos of theta. So if I were to again represent this whole thing in terms of linear algebra and wanted to find out if there is a relationship between the new position vector V dash and the old position vector V.

We can represent that in terms of x dash y dash z dash equal to this rotation vector here times of the coordinate vector which was earlier that is x, y, z and this rotation vector can now be represented as cos of theta - sin of theta 0, sin of theta cos of theta 0 and 0 0 1. So therefore if I were to just look at this new addition, obviously you have already learnt about the x dash y dash earlier so z dash would be equal to 0 into x + 0 into y + 1 into z.

So z dash and z are similar to each other. So that is what rotation about z axis can be coined as. Similarly, you can have about the x axis and in this particular case the transformation equation can be x dash y dash z dash equal to 1, 0, 0. Obviously the x dash becomes equal to x because it is about x axis; 0 cos of theta - sin of theta and 0, sin of theta, cos of theta times of x y z.

And similarly, I would have rotation about the y axes represented as again x dash y dash z dash equal to cos of theta, 0, sin of theta, 0, 1, 0, sin of theta, 0, cos of theta this is minus sin theta I am sorry; minus sin theta, 0, cos of theta. Just I will rewrite this part again so that there is no confusion, minus sin theta times of x y z. So these 3 equations represent rotation about the 3 different axes that is z axis x axis and y axis.

And so in a nutshell what we did in a 2D space is now being translated into a 3D space as translation, scaling, and rotation and set of equations have emerged because of this. So now I would like to indicate a very elementary problem which generally comes you know when we talk about computations using algorithms and that is about you know how much capable a processor would be in terms of you know processing different operations, mathematical operations.

So as you see here both the scaling as well as the rotation matrix are represented by mostly a multiplier, mostly represented in a multiplier form where there is the basic vector multiplied by the transformation matrix brings out the final vector or the final state of the geometry. However, in case of translation the equation that is being used as an addition equation where the basic migration along the x, y, and z axes of the point x, y, z determines the final coordinates.

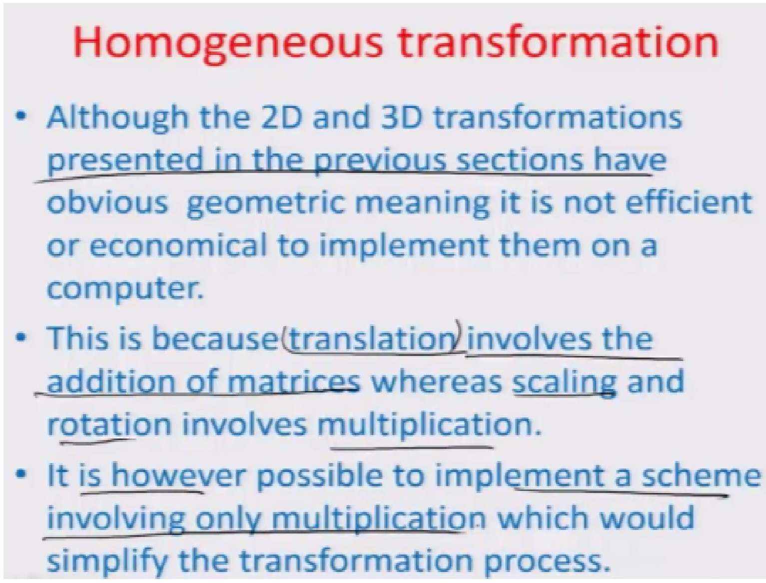
And therefore it may be worthwhile to convert into a system which is all using multiplier matrixes rather than you know having separate addition you know operation as well as multiplication operation and that is probably computationally less extensive okay and so therefore we can try for converting this translation you know mathematics into something which would actually rhyme and rhythm with what you have for the rotation.

As well as the scaling matrixes of one transformation matrix and a vector matrix multiplied together to formulate the final matrix. In any case when we talk about series of operations for example something is translated in space, rotated in space and then again you know let us say scaled up. So it probably always better to have a series of multipliers and can catenate all the multipliers in turn.

So that you have one transformation equation which represents the final coordinate given the initial coordinates, computationally much more quicker okay and then you do not have to really get into the business of a separate processing of additions and a separate processing of matrix multiplication. So in order to do this we really have to change the state of the matrix from so we will see how you know you can convert this addition into multiplication through a change of state of the matrix from a n dimension to N + 1 dimensional space.

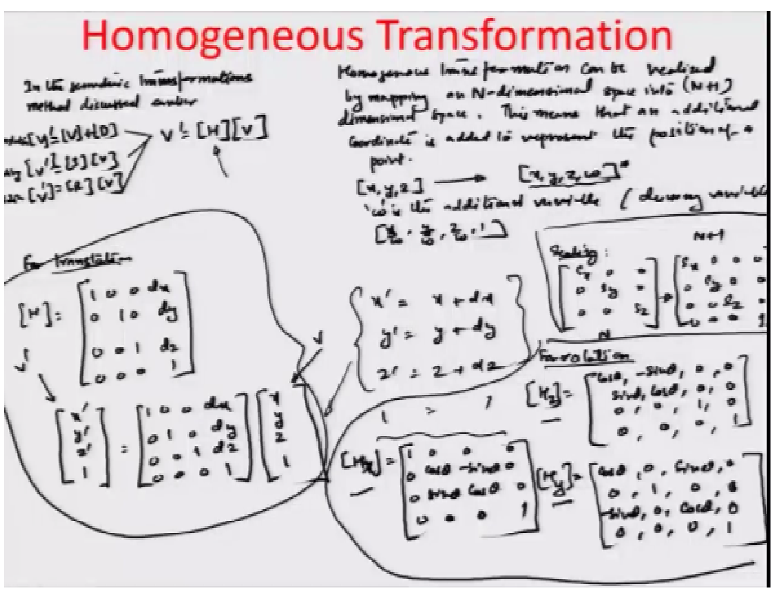
And so all you need to do there is to just add a dummy variable in order to execute the translation through a multiplier you know set of matrices rather than an addition of matrices.

(Refer Slide Time: 18:39)



So that is what is known as homogenous transformation and I think I have just mentioned this that although the 2D and 3D transformation presented in the previous section have obvious geometric meaning, it is not efficient or economical to implement them because the translation involves the addition of matrices as far as translation is concerned or you know multiplication as well as scaling and rotation is concerned.

(Refer Slide Time: 19:11)



So it is possible however, to implement a scheme involving only multiplication which would be simplifying the transformation process and in the geometric transformation methods discussed in the previous sections the scaling or the translation or the rotation is represented as V dash is V plus D translation, V dash is scale matrix into V matrix, scaling and V dash is rotational matrix into V matrix for rotation.

So we want to actually convert all this into a form which is V dash is some homogenous you know set of matrices which can transform the points into translation, rotation, scaling so on so forth into initial vector. So we bring in homogenous transformation which can be realized by mapping an N-dimensional space into N + 1 dimensional space. This means that an additional coordinate is added to represent the position of a point.

So geometrically it has no meaning. It is only a dummy kind of a variable but it will enable you to get into that you know from a 3 into 3 matrix into a 4 into 4 matrix where this transformation you know homogeneity can happen for all the 3 operations mentioned here. Let me just write down the operation just for the sake of so this is translation, scaling, rotation. So in fact what we are asking here is to convert x, y, z into x, y, z, and w; w is the additional variable you can call it the dummy variable also and you can use it for normalization of the state of space here.

So the coordinates here can be scaled into x by w, y by w, z by w and 1 just because it is a dummy variable can be used for scaling and this addition serves to convert the addition into multiplication matrices. Let us see how. So for translation now we will be using multiplier matrix. Let us say the h matrix which can be represented as 1 0 0 dx, 0 1 0 dy, 0 0 1 dx, 0 0 0 1 and let us see if this transmission happens well.

So let us actually just try to represent the addition matrix, the translation let us say a set of coordinates x dash y dash z dash and 1 the dummy coordinate generated through multiplying the h matrix that we have just formulated with the additional set of matrices x, y, z 1. This is the coordinate, initial coordinate of the vector V with the dummy variable 1 introduced. This is the initial coordinate of the vector V dash after the translation.

Final coordinate V dash of the vector after translation with the dummy variable 1 again introduced and if I just multiply these we would have again this set of equation x dash equal to x + dx, y dash equal to y + dy, z dash equal to z + dz and obviously an equality of 1s on both sides.

So the dummy coordinate does not change because of the translational process. So that is how you can multiply by changing the you know the dimension of the space to a higher order and trying to convert the submission into a set of multipliers.

So obviously all the other vector also should have a higher order. For example for scaling if we wanted to use homogenous transformation the vector would convert from the earlier vector sx 0 0 , 0 sy 0, 0 0 sz to now you know N + 1 that is 4 by 4 matrix from N-dimensional matrix. So this will now be sx 0 0 0, 0 sy 0 0, 0 0 sz 0, and 0 0 0 1. So this is how the scaling vector would transform.

Obviously the dummy coordinate here also will not change after the scaling operation has been executed and for rotation again we have again three different vectors in place. So we have rotation along the z represented through again cos theta, - sin theta, 0 0; sin of theta, cos of theta, 0, 0; 0, 0, 1, 0; 0, 0, 0, 1. So that is how the rotation around z is represented. Rotation around x is again represented through a N + 1 dimensional matrix that is a 4 by 4 matrix.

In this particular case it would be because it is x it will be 1, 0, 0, 0; 0, cos of theta, - sin of theta, 0; 0, sin of theta, cos of theta, 0; and 0, 0, 0, 1 and similarly for the rotation along the y the matrix would again be represented as cos of theta, 0, sin of theta, 0; 0, 1, 0, 0; - sin of theta, 0, cos of theta and 0, 0, 0, 1. So that is how you would represent the rotation around the y. So these are the homogenously transformed matrices for rotation.

This is for scaling and this right here is that for translation and so we are left with now only a set of multiplier matrices which we can concatenate together to do complex business of you know translation of an object in space or rotation and a combination of different operations is based on object. So we will actually try to see a problem example in the next module. In the interest of time I will close this particular module. Thank you very much.