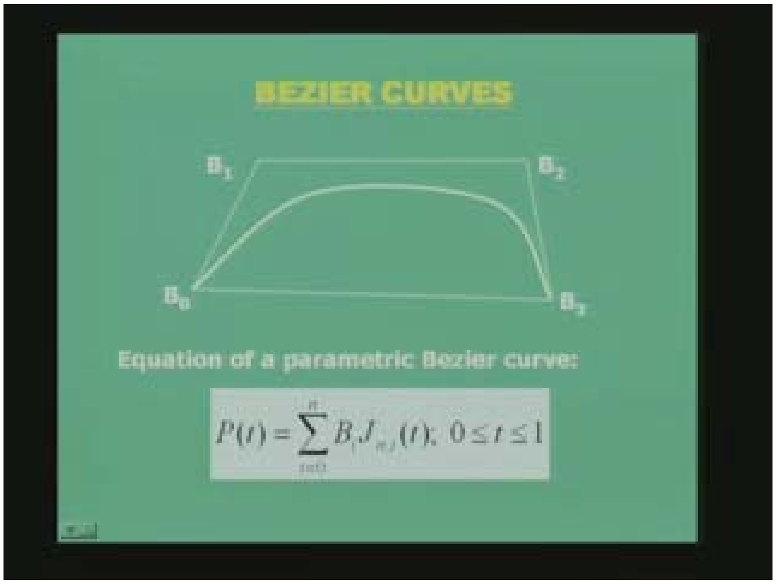
Computer Graphics
Prof. Sukhendu Das
Dept. of Computer Science and Engineering
Indian Institute of Technology, Madras
Lecture - 38
Curves and Surface Representation

In the last two lectures on curve representation we discussed about parametric cubic curves and cubic splines. We also introduced Bezier curves, the Bernstein basis and we will start from where we left off in the last class and continue with Bezier curves and then move over to surfaces. So we will have lots of equations and derivations so kindly be ready. And of course a few illustrations will be also accompanying the mathematical expressions for each of these curves representation. So Bezier curves we talked about this representation that it must lie within this polygon, the starting point, end point, tangents to the curve should be the first and the last edges of the polygon.

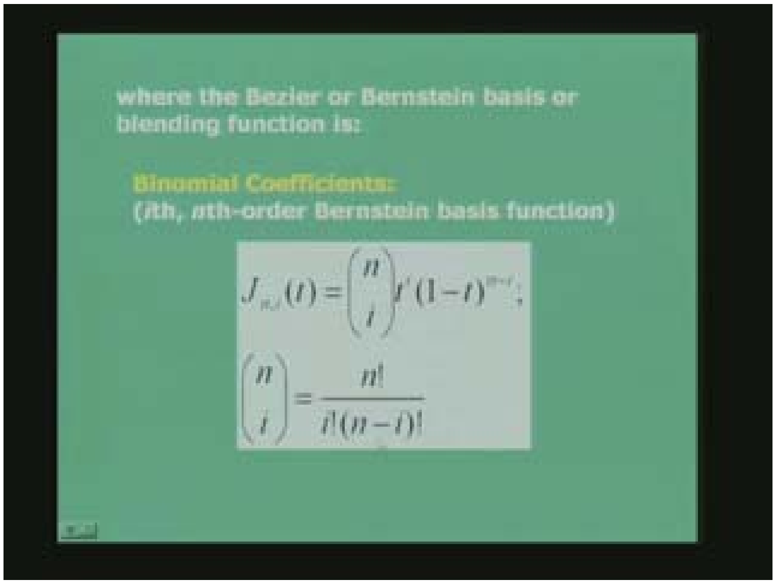
And the equation of the parametric Bezier curve t is running from 0 to 1 is given by this where this J and i is called the Bezier or Bernstein basis or blending functions or even binomial coefficients ith nth-order Bernstein basis functions are given by expressions of this form. And we will now see what do these shapes depending upon the value of n and i that is the order will be treated by two parameters and the value of t which varies from 0 to 1 we will see how the functions looks like and dictates the nature of the curve. So this B_{is} are also called the control points in the previous expression of the Bezier curve which we have seen and we will henceforth called the J_{ni} as Bernstein basis or Bernstein blending functions.

(Refer Slide Time: 00:02:52)



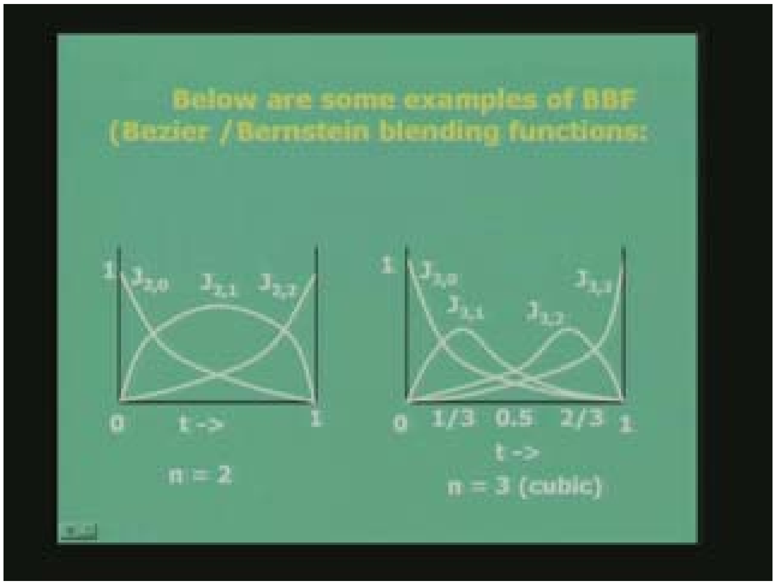
J and i is the ith nth order Bernstein basis function, n is the degree of the defining Bernstein basis function or the polynomial curve segment which is one less than the number of points used in defining the Bezier polygons.

(Refer Slide Time: 00:03:22)



So if you have n plus 1 as the number of points used to define the polygon that means the number of points if it is n plus 1 then n is the order of the Bernstein basis here. These are some examples of the BBF or the Bezier Bernstein blending function.

(Refer Slide Time: 00:04:03)



If you see here for n equal to two there are three possible values $J_{2,0}$ $J_{2,1}$ and $J_{2,2}$ because n is equal to 2i could vary from 0 to up to n. So you have $J_{2,0}$ $J_{2,1}$ $J_{2,2}$. So this is the first curve $J_{2,0}$ this is $J_{2,1}$ and $J_{2,2}$. Again hand simulated but if you generate it using a computer simulation using the expression given in the previous couple of slides back t varying from 0 to 1 you will get the same.

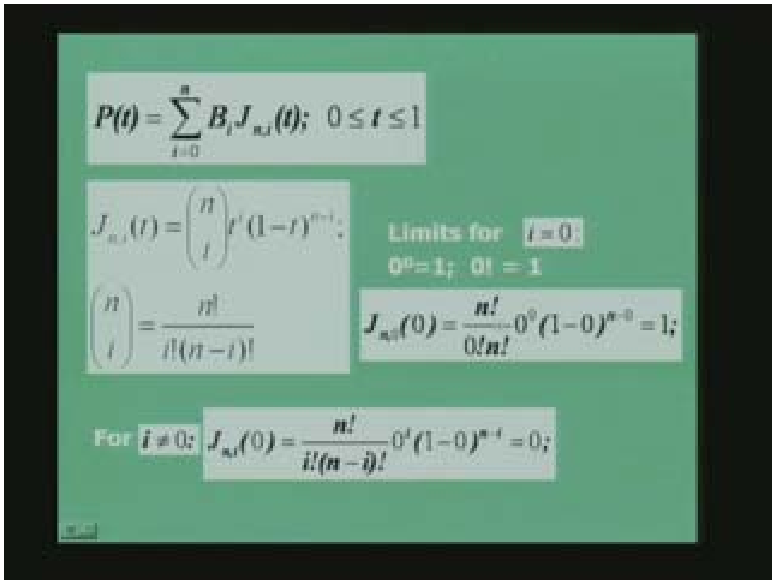
How many curves you will get for n equal to 3, for n equal to 2 you got three curves $J_{2,0}$ $J_{2,1}$ $J_{2,2.}$ If n is equal to three you will get $J_{3,0}$ $J_{3,1}$ $J_{3,2}$ and $J_{3,4}$ curves. So let us look at the nature of the four curves. Here is what you get for n equal to 3 cubic Bernstein basis t again varying from 0 to 1, $J_{3,0}$ is the first curve here $J_{3,1}$ $J_{3,2}$ and $J_{3,3}$.

As you can see it is a symmetrical nature of these curves and the smoothness at the ending points which lends itself to the curves fitting with respect to the endpoints and their tangents. But we will see the boundary conditions of this Bernstein basis first and see that they not only have good symmetric properties but also good starting and finishing properties in terms of their values and tangents as well. The n equal 2, the quadratic, n equal to 3 cubic Bernstein basis functions will look like. And interesting to note $J_{3,0}$ is this one is just replicated mirror reflection of $J_{3,3}$ similarly $J_{3,1}$ is a mirror reflection of $J_{3,2}$. Here also the same $J_{2,0}$ is a mirror reflection of $J_{2,2}$.

Let us look at the end points. So this was the Bernstein basis of function 0 to 1 and this was the Bernstein basis Jn of i is defined as this. We have noted this down in the previous class or note this down right now. Also, the t varying from 0 to 1, the end combination i is given as this particular expression.

Let us look at limits for i is equal to 0 for this particular function, with the constraint that 0 factorial which we will need and the 0 factorial and 0 to the power of 0 both are equal to 1. So we put this in mind and look at the limiting condition when i is equal to 0 here. If you look, that if we substitute J_n of 0 of the t is equal to 0 if you substitute here this condition where i is 0 as given here, t is also 0 you will get it as 1. This is what you will get so that is interesting to note. That means the n0 of the 0 the starting point when t is equal to 0 for any order will start at 1.

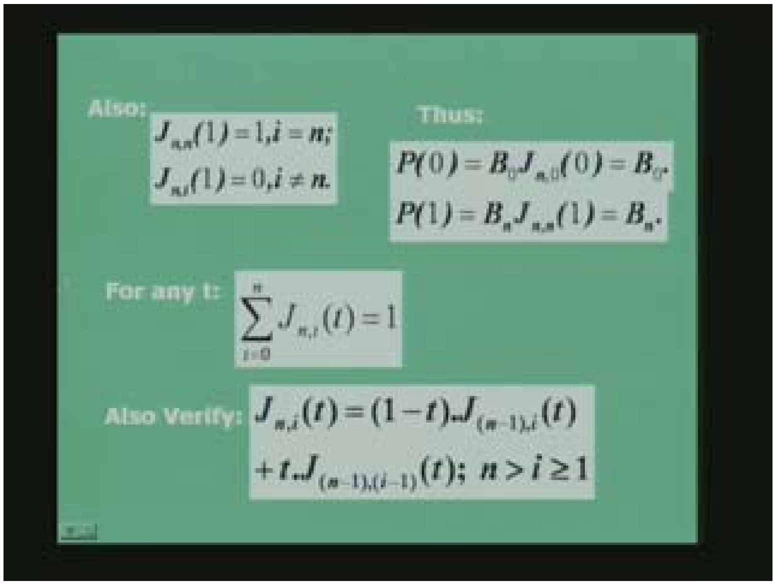
(Refer Slide Time: 00:06:53)



For any order when t is equal to 0 the first curve will start at 1 where if an i is not equal to 0 that means for any other non starting curve and i is not equal to 0 means 0_J and 0 that means $J_{n,1}$ and $J_{n,2}$ and so on that this will start at 0. This is interesting, the first curve starts at 1 and all other curves start at 0. Can we verify these using the previous diagram for the quadratic and cubic case? Let us go back, yes you can see here $J_{2,0}$ starts at 1 and all other curves start at 0, the same thing here. $J_{3,0}$ starts at 1 and all other curves start at 0.

Did you follow the logic? I again repeat, $J_{2,0}$ starts at 1 and all other curves $J_{2,1}$ and $j_{2,2}$ for the quadratic case starts at 0. Here the cubic case starts at 1 all other curves also start at 0. So this is what we have derived here the Jn 0 of 0 for any n, n could be 2, 3 quadratic cubic case starts at 1 all other curves for i is not equal to 0 they start at 0. What about the finishing case when t is equal to 1 J_n of n if you substitute back into the expression when i is equal to n, if you substitute here in this expression i is equal to n you should be able to obtain 1 and J_n for all other i not equal to one you should get 0.

(Refer Slide Time: 00: 11:37)



That is also an interesting phenomena, that means the last curve and i is equal to one will finish at 1 and all other curves will finish at 0. You compare this boundary condition when t is equal to 1 with the condition here at the initial condition when t is equal to 0 the starting condition. All curves start at 1 and they finish at the first curve the first curve starts at 1 and all other curve starts at 0. In the first the finishing condition is concerned and the ending condition bound when t is equal to 1, all other curves will end at 0 the last curve will finish at 1. Let us go back and check these conditions as well.

You see here the last curve $j_{2,2}$ finishes at 1 all other curves finishes at 0. $J_{3,3}$ finishes at 1 all other curves finish at 0. So this is the interesting symmetrical starting and finishing condition which you see. The first curve starts at 1 the last curve finishes at 1 all other curves except the first one start at 0 all other curves except the last one finishes at 0 and the same thing here. All other curves except the first one start at 0 all other curves except the last one they all finish at 0, this is interesting.

The Bernstein basis which is illustrated using the quadratic and cubic case in this figure and it is true for any n, 4, 5 whatever you visualize you must remember. If you remember these two figures or those boundary conditions mathematically it will be easy for you to visualize the Bernstein basis nature. Of course you can look into the derivatives of this expression and find out that some of these curves except the first and the last one will have a peak somewhere.

If you look back here that the $J_{2,1}$ has a peak at t is equal to 0.5. The $J_{3,1}$ and $J_{3,2}$ will have peaks at 1 by 3 and 2 by 3 respectively, that is interesting. And all of these curves intersect at 0.5. And there of course there are other interesting properties of this Bernstein basis which we will see. I hope you have copied at least these two curves because whenever we come across new properties in the series of discussions here with Bernstein basis you should be able to get back to this figure yourself. It could be difficult for me to roll back but you should be able to visualize this figure from your notes and see that the property is satisfied so far. Right now just remember the starting and the finishing conditions for the first and the last curve, 1 and 0 and this is also 1 and 0.

Starting at 1 and all other starting at 0, the last curve finishes at 1 all others finishes at 0, the same thing for the cubic case. So, we will roll forward, we have seen the starting case and we have seen the finishing case. Thus P of 0 since J_n of 0 starts at 1, the first point B of 0 the B_0 is the P of 0, the B_0 which is the coefficient to be evaluated for the Bernstein is the P_0. So how do we evaluate the other one using boundary conditions B_n also will be P of 1 because J_n of n is equal to 1 we have seen that if you substitute you will get 1. So what you get as the coefficients B_0 B_n are nothing but the starting and the finishing points. And so the curve must start at B_0 now and finish at B_n.

We talked of these as one of the conditions in the last class. We talked of this polygon with the end points; the curve must start at the first point and finish at the last point. Hence, that is what is the mathematical condition here P of 0 is B of 0 B_0 and P of 1 is B_n.
For any t this is also another interesting feature but for any t sum of all the values of the entire Bernstein basis for any value of t starting from 0 to 1 is equal to this.

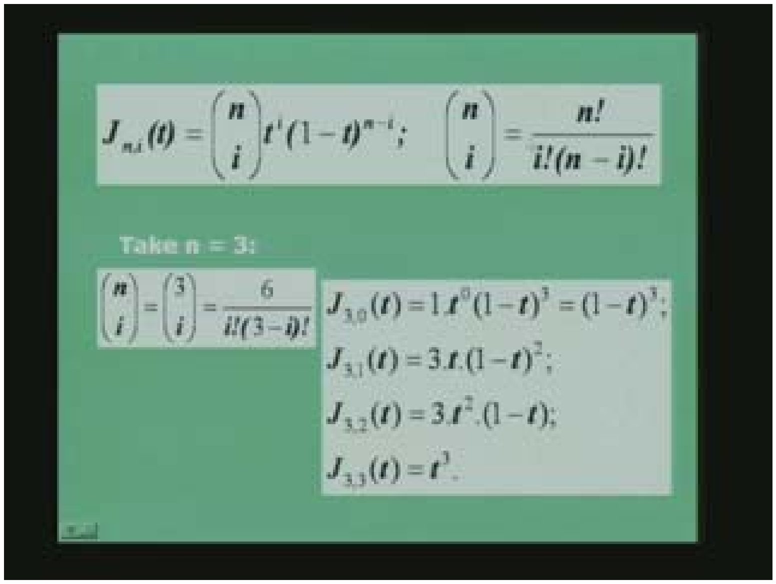
Look at the normalization nature of the curve. Take any t, take cubic or even higher order for larger values of n if you take, take any value of t sum up all of them and you will get a value 1. And I will leave this as an exercise for you to find out mathematically even for the quadratic or a cubic case for any value of t or for any general value of n that this value is 1. It will be a very interesting proof, try this out. I should leave some points as a home exercise so this is one of them for the class today.

So, coming back for any t this is what you should try to prove or find out. Also verify, this is also an interesting proof where the J and i of t can be represented as J_n minus 1 t and n minus 1 i minus 1. That means the next order curve of the next higher order basis can be represented in terms of previous basis of lower order basis of J_n minus 1 i and j minus 1 and i minus 1 that will be very interesting.

This is something like a decomposition in the sense that one higher order curve can be broken into two smaller parts or two lower order curves can be joined together to build a higher order curve. And of course I should be more than 1 and all that this is a boundary condition here and limit for the case of i which should be of course less than n and it should be more than 1. Although I placed in the reverse way but you can follow here. This is what I meant that if you are talking of n is equal to 3 any particular curve for any arbitrary i can be derived provided those limits of i are maintained from the quadratic curves let us say, two quadratic curves can yield a third degree curve and so on. By summation of some of the product forms 1 minus t and t factor will lead to 1. You can also verify this mathematically as a proof which is possible for you to derive.

So these are the terms which you need to remember to derive these two conditions for any t. Sum of this J_{ni} is equal to 1 and the n of i t nth ith order Bernstein basis represented in terms of lower or n minus 1 and n minus 1 and i minus 1 to derive these two you will need these formulas which are the expressions of the Bernstein basis, you know that.

(Refer Slide Time: 00:13:30)

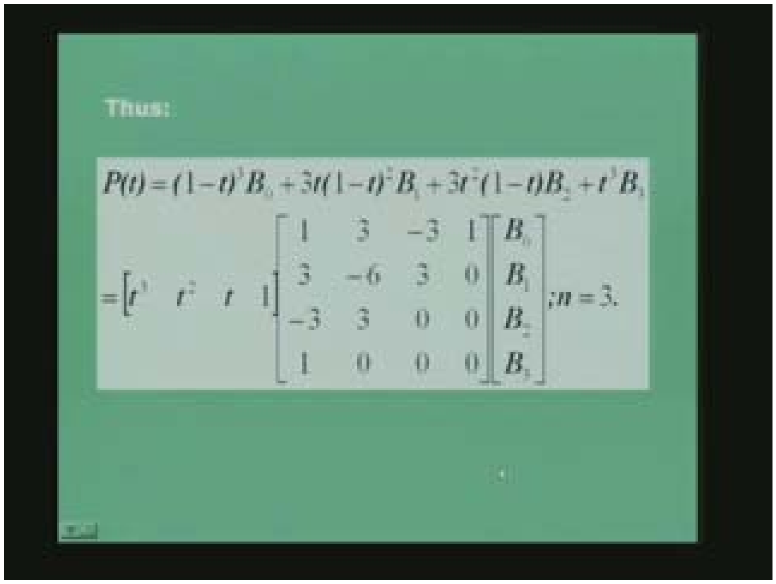


I will give you a few examples of the Bernstein basis, this factorial terms and the combinations where n and i combination can be written as this. You can use these to derive what you will say as the $J_{3,0}$, $J_{3,1}$, $J_{3,2}$ and the $J_{3,3}$ for the cubic Bernstein basis as you can say when n is equal to t we say that there are four curves.

For quadratic n is equal 2 there will be three curves $J_{2,0}$, $J_{2,1}$, $j_{2,2}$ and for a cubic case n equal to 3 there will be $J_{3,0}$, $J_{3,1}$, $J_{3,2}$ and $J_{3,3}$. These are the expressions. I leave it as an exercise for you. Use these expressions and very easily substitute them and this is also given very simply as it is here, simply substitute that and this is what you get. You see, all of these are cubic polynomials of t the $J_{3,0}$ and you have already seen the nature of these curves. We have seen those two figures a couple of slides earlier for the quadratic n equal to 2 and the cubic n equal to 3 case they are nothing but these expressions which you are seeing in the slide right now.

Have a look, with the first curve will start from 1 and finishes at 0 and these two also will start at 0 and finish at 0 and this curve will start at 0 and finish at 1. These were the properties we discussed for the Bernstein basis. I repeat; this first curve starts at 1 when t is equal to 0 and finishes at 0 when t is equal to 1. These two curves for t is equal to 0 and 1 the boundary conditions will start at 0 end at 0. And this curve for t equal to 0 will start at 0 and wind up at the value 1 when t is equal to 1.

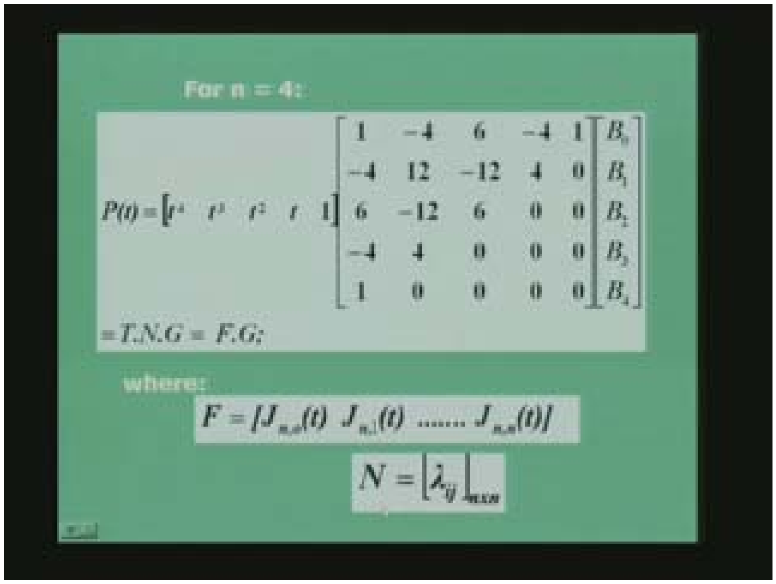
(Refer Slide Time: 00:14:57)



You can rearrange in that form which we discussed couple of classes earlier or in the previous classes as well where the P of t for the cubic Bernstein basis or Bezier curve of order three when you substitute these $J_{3,0}$ from the previous expression on to the expression of the Bezier you will have this at B_0, B_2, B_3, B_1, B_2 and B_3 are the geometric points. We define the polygon and if you substitute and rearrange I leave it as an exercise for you to write it in forms of T N and B, T N and G for n equal to 3 this is what you get as your matrix formulation of the Bezier curve.

Again this is a third home exercise which I am leaving now. There are lots of derivations here, if I spend time on these I thought you should practice it yourself to gain confidence. Simple manipulations are not much difficult, you should still practice and not simply copy them from the slide. I leave it as an exercise for you to try these after the class when you go back. This is what you have as your T N and B which you can write from this particular expression of your cubic basis.

(Refer Slide Time: 00:17:44)



For n equal to 4 in fact this is what you keep trying again. It will be t to the power 4 raised and there should be 5 because one more than the number of points, it should be one more than the order of the Bezier curves. So you have five control points for the polygon and B_0 to B_4 and I leave it to you again as an exercise to obtain these matrices as we have done for the cubic cases here.

Remember, we first derived this that means you first derived this J_3 is then we wrote the P(t) and then wrote this expression t n terms of T N and G. So you do that same thing for the case n equal to 4 that means you derive your $J_{4,0}$, $J_{4,1}$, $J_{4,2}$, $J_{4,3}$ and $J_{4,4}$ that is five of them.

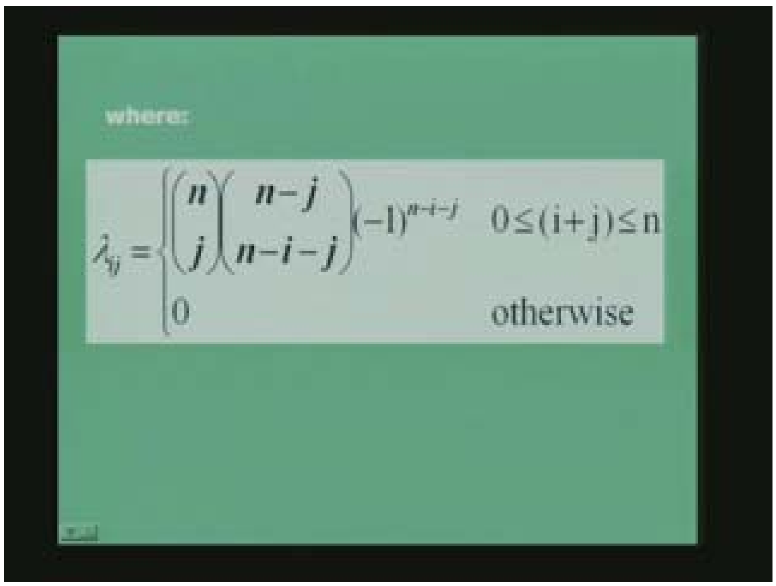
I repeat again; $J_{4,0}$, $J_{4,1,}$ $J_{4,2}$, $J_{4,3}$ and $J_{4,4}$ that is five of them. You derive them, obtain their nature then write the expression of P and P of t and then try to group these terms into different categories and write the expression as given in the matrix form T N and G. And sometimes it is written as F dot G where G is this geometric constant vector and this triple ending function F could be visualized as a weightage parametric curve in terms of raise to the power 4. So this is what we have for F where the J_{is} can be written as a matrix form in terms of these J_{nis} J_{n0} J_{n1} up to J_{nn}.

And what are these J_{nis}? These J_{nis} are nothing but what we have derived if it is n equal to 3 then you have $J_{3,0}$, $J_{3,1}$ up to $J_{3,3}$. If n is equal to 4 you would have had $J_{4,0}$, $J_{4,1}$ up to $J_{4,4}$. That is what you would have defined as your F. But in terms of T and G if you look at the t it is a very straight forward matrix and it depends upon the order n equal to 4 which starts from t to the power of 4. If n equal to 3 you will have only these four terms t cube to 1 and this matrix N which is the most interesting matrix I can define in terms of lambda ij.

I have formulation for this lambda ij to define the elements of this matrix for n equal to 4 or the previous matrix n equal to 3 here. You look at this matrix in fact the lower triangular one is almost as in all matrix, element 0s in fact they are the upper left triangular elements only. That is what we can define in terms of this lambda ijs and the formula in this.

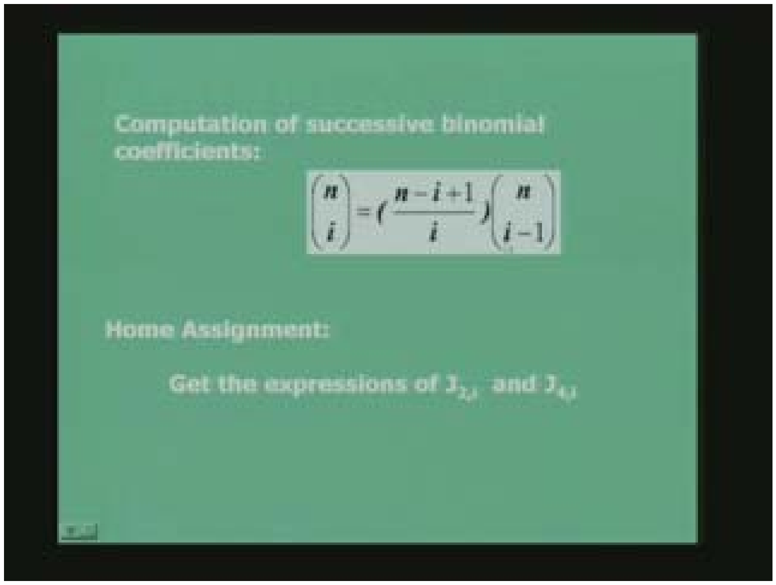
Please note down the lambda ijs. If i plus j is lying between these and 0 to n then you can use this formula. If i plus j exceeds the value of n you use the value 0 this will help you to see that these elements are non 0s in the left upper triangular matrix in the lower right triangular matrix either for n equal to 4 or for n equal to 3 as you can see here the upper left triangular has non 0 values lower left triangular values has 0s. This is for n equal to 3 and for n equal to 4 the same thing happens.

(Refer Slide Time: 00:18:40)



The upper left triangle is non 0, lower left triangular and this is formulated with the help of this lambda ijs because i plus j if it lies between 0 to n. You are talking of elements which are lying in the left upper triangular matrix, very interesting, lower right triangular matrix elements if you sum up i plus j the elements index to that element of the matrix will roll over and in that case use the value 0.

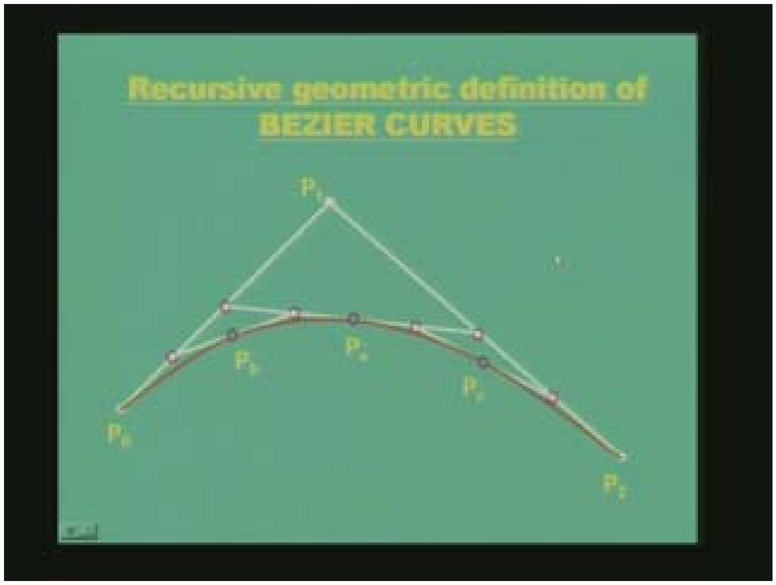
(Refer Slide Time: 00:19:44)



So this is the expression which you have for the lambda ijs and the coefficients of the successive binomial coefficients also can be derived. I consider this as another home exercise for you and I also request you to obtain the expressions for $J_{2,i}$ and $J_{4,i}$. Computation of successive binomial coefficients also can be obtained using this expression. I leave this as an exercise for you to verify this.

This is very simple because this formula was given to you earlier. So that is very simple and trivial, it should be easy for you and also I leave it as an exercise to obtain the expression for $J_{2,i}$ and $J_{4,i}$. $J_{3,i}$ is what we have derived. I leave it as an exercise, the quadratic case and the quadruple case with the case of $J_{4,i}$.

(Refer Slide Time: 00:26:43)



So we will move on to a very interesting property outside mathematics which we will see as the recursive geometric definitions of Bezier curves. Let us say we take three points P_0, P_1 and P_2 and if you have three points that means n equal to 3 what should be the order of the Bezier Bernstein basis or polynomials or even the Bezier curve? The order should be n minus 1. So if n is equal 3 it should be a second degree. So, second degree curve lying within that polygon starting from P_0 to P_2, tangents should start from P_0 to P_1. Follow that edge first then go somewhere through the middle within the polygon and end tangentially in the arc P_0 to P_2.

And again if you look back it should start at P0 tangentially here, the point up to this. Of course you can obtain all these using the mathematics given in the coordinates P_0, P_1, P_2 in 2D or 3D you can obtain the curve. But let us look at the geometric recursive definition which will be very interesting. That means can I find the set of points which will lie on the Bezier curve starting from P_0 to P_2 and controlled by P_1.

Of course you can have more number of points. Nobody stops you from having more number of points that is P_0, P_1, P_2, P_3 and so on. But it will be interesting to first have a look at these and then of course extend this philosophy if you have more number of points. Just look at a triangle, this definition is very interesting.

We take the bisector point P_0 to P_1, this labor point i says the bisector, the midpoint between P_0 to P_1 is marked here. Similarly, I mark a midpoint between P_0 to P_2 it will be somewhere here. That is the midpoint from P_0 to P_1 and the next midpoint from P_0 to P_2 is here. I join these two midpoints; the first midpoint of P_0 to P_1 is joined with the next midpoint between P_1 to P_2. And then I obtain another midpoint of this new line, midpoint will be somewhere here which I will label as P_a. This is the first point I obtain where the curve must pass.

The Bezier curve passes through the point P_a it will start at P_0 wind up at P_2 for value t is turning from 0 to 1, t_1 to t_2 in this case 0 to 1 and it must pass through that point P_a within that triangle. This is the definition of the first curve. Now we can use these definition recursively to look for points to the left of P_a and to the right of P_a. That means within the triangle points between P_0 to P_a, have a look into these figure. Now we have to look for points between P_0 to P_a and the points between P_a to P_2 because we are now guaranteed that if the curve starts from P_0 passes through P_a and winds up at P_2 we have to look for a point to the left of P_a which lies between P_0 to P_a and then again also obtain points to the right of P_a which lies between P_a and P_2.

Now as you got the point P_a from the triangle P_0, P_1 and P_2. I hope that idea is clear or should I roll back once again, this was the starting point. That means you had a triangle P_0 P_1 P_2 and I obtain a point P_a how? I roll it once again for you for a clarification, how do I get this first point P_a?

Given a triangle P_0 P_1 P_2 my problem is to get P_a. I get the first midpoint the left between P_0 to P_1 then get the next midpoint P_1 to P_2 join these two lines obtain the midpoint of this line which is joining the midpoints. That is the method by which I get it. If I could do that to get P_a from P_0 P_1 and P_2, I repeat, if I can obtain this P, remember I leveled it to a different shade than the other midpoint because this is the point on the curve. So this point P_a is a point on the curve obtained from the points P_0 P_1 and P_2 a triangle.

I can similarly obtain from these three points, what are they? P_0 then this first bisector and P_a, these three will form another triangle, another polygon P_0 this point the first midpoint P_a will form another polygon and using that I can get another point here through which this curve must pass. Let us do that, I will roll it straight without any steps. So from that point I roll forward, two midpoints, one bisector and then this midpoint again. This is the same of what I did to get Pa. let me roll back one step.

I am concentrating on P_0 this first midpoint and P_a and visualize only this part of the triangle and see how I can get this midpoint here which is a new point. And this is the recursive part that I am using the same logic of P_0 P_1 P_2 to get P_a. As I use this logic here I am calling that function again and passing these points P_0 and a midpoint here and P_a to get the second point to the left of P_a and this is the point which will come. And this new point P_b is another point which will be lying between P_0 and P_a and the curve must also pass through that, recursively I have used that. And I can do it to the right of P_a between P_a to P_2, I can use a similar definition and if I use this I can get another point let us try that.

Look at the line bisector joining midpoints and the midpoint again and that is the P_c. So now you see I have got three points, first I have a function called from P_0 P_1 P_2 to get P_a. Then I recursively call those same function and pass P_0 this point and P_a and then P and to the right of it, P_a second bisector and P_2. And I can keep doing this again and again. I can generate points between P_0 to P_b, P_b to P_a, P_a to P_c, and P_c to P_2. All of these will start from triangles and I can recursively keep calling them and finally I will get a set of points and then that is what the Bezier curve will look like.

This is the Bezier curve. I did mention that this curve will pass through the points. It has to pass through the point not only it will start from P_0 and wind up at P_2 but it also must pass through P_a P_b and P_c and the rest of the points also can be generated using similar function. I hope you have understood this logic but I will probably roll back once more. This was the starting point. So the first method was to get the P_a. This is how you get the bisecting and that is where you get Pa.

The recursive call from P_0 bisector, first bisector and P_a will give me another point P_b and another recursive call for the triangle P_a second bisector and P_2 and the bisector of P_1 and P_2 is this point and P_2 so this is the triangle. I can get the third point P_c and the curve will pass through these points. And I keep taking the smaller triangles and keep recursively calling them and keep generating points and this is what the set of points which I join to get this Bezier curve.

This is a very interesting recursive geometric definition of Bezier curves. I hope you like the methodology with very little mathematics in terms of matrix multiplications, cubic degree manipulations, permutations, combinations, starting point, end point with Bernstein basis. It is a very interesting phenomena here, how recursively I can keep doing this function of obtaining two bisectors of the triangle edges joining them get the bisector. This is the main function, I keep recursively calling them as I keep developing new points and generate newer and newer triangles to the left and right of each set of points I generate. So this is what it is.

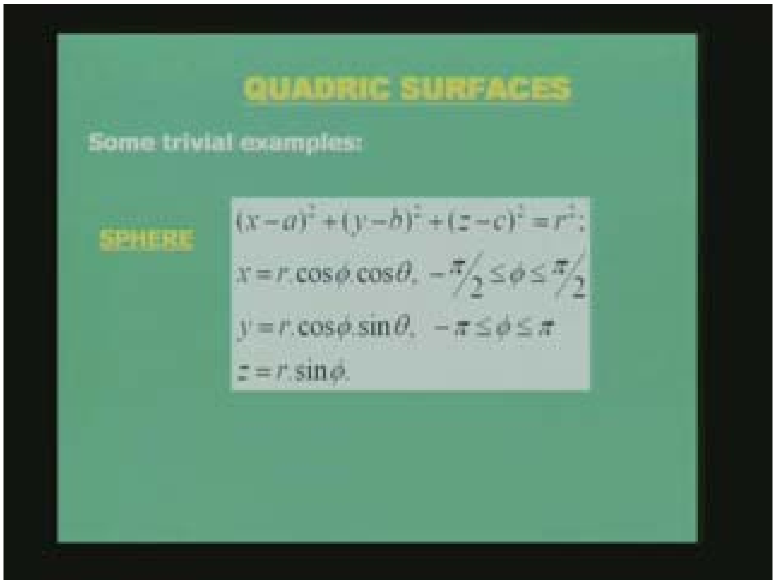
Now look into the points, so now if you feel here that the curve starts from P_0 and passes through these points P_a P_b P_c and has 2 of course end at P_2 you can see that the overall nature of this curve can be controlled by oscillating. If you keep changing this P_1 position keeping P_0 and P_2 unaltered all these points P_a P_b P_c and other points of the curves will change and so these are my control points.

I keep changing P_1 and my curve gets varying and I can add a few more points. And I can add a few more points and all these have to follow that particular logic. So I leave it as an exercise for you to read about what are called the B-splines represented as blending functions. Of course remember we have talked about cubic splines and Bezier curves. I think with the time available to us we have to move to surfaces. So I leave it as an exercise for you to read about cubic splines using Bernstein basis, splines using Bernstein basis which are called as B-splines.

Remember, we talked of cubic splines and Bezier curves with Bernstein basis. Merger to splines with Bernstein basis are called B-spline. Very interesting functional forms read about B-splines which are also widely and commonly used represented as blending functions. Also, read about conversion from one format to another. And knots control and knot points how they play an important role and also the condition when B-spline could become a simple Bezier curve.

I leave these as an exercise for you to read due to time limitations available to us in the series of lectures. We will move on to surfaces. And this is a trivial example of a sphere which you know, the Cartesian coordinate form, a b c the center of the sphere in 3D, R as the radius and this is the parametric form in fact it is representation of the sphere or equations of this sphere in polar coordinate form r theta and fi in 3D.

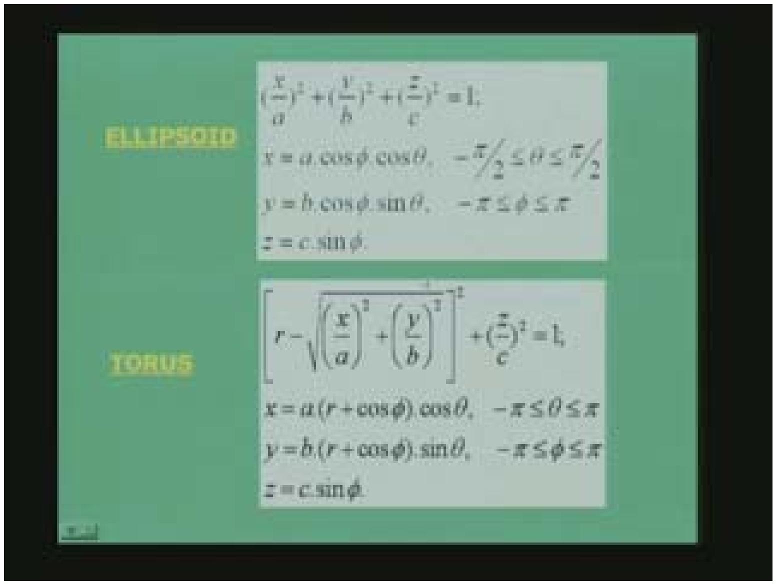
(Refer Slide Time: 00:30:07)



I started with the simple example because the sphere is the most common example in 3D surface which we used. But before we move on to generic expressions of quadric surfaces and in fact complex surfaces based on even Bezier we will go through a very few other examples very fast before moving on to generalized expressions and see illustrations as the last part of the coverage of the curves and surfaces representation. So look back, I hope you are able to copy these expressions. The first one is of course very simple and straight forward, look into the ranges of theta and fi. This is the equation of the ellipsoid. It has three parameters a b c the Cartesian coordinate form on the top and the polar coordinate at the bottom, again the ranges of theta and fi are given.

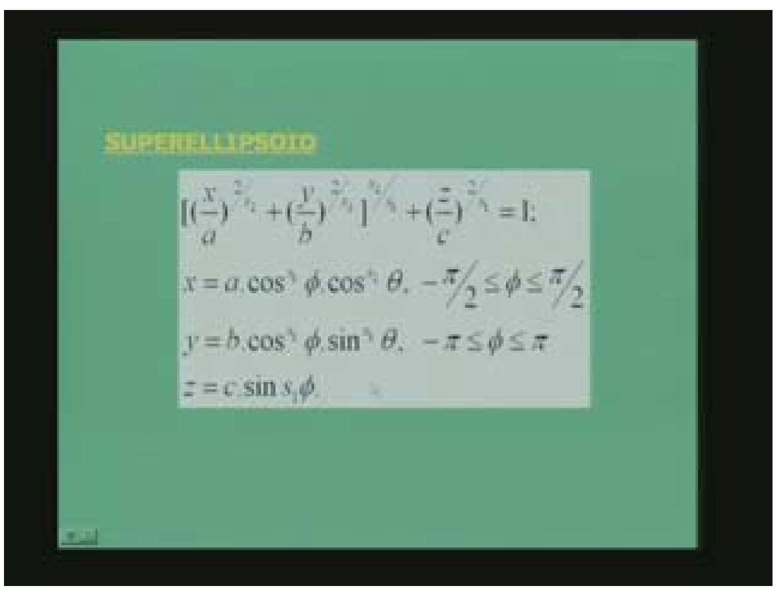
That is an example of the torus. And the expression is given in both Cartesian coordinates form with respect to r theta and fi. It is a little bit different than what we had seen in the previous case of a sphere and ellipsoid. This is the example of torus. This is an example of a super ellipsoid where it is an ellipsoid but you have these coefficients s1 s2 which you can vary and give various types of structures to the ellipsoid as you feel necessary. So these were the three different forms of expressions of special types of quadratic surfaces.

(Refer Slide Time: 00:31:31)



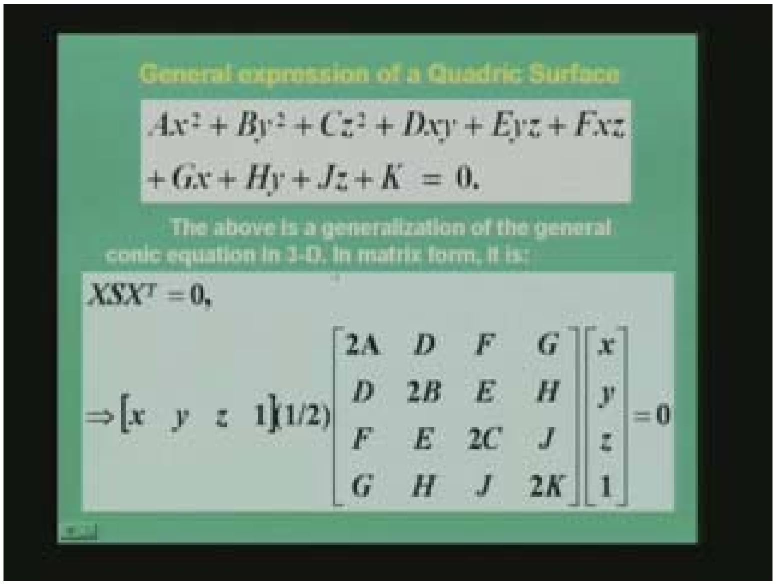
We will look into the general expression of a quadratic surface which is an extension of the general conic which we have seen earlier. Remember, there were about six parameters for a generalized conic for which we used boundary conditions to evaluate about a couple of classes back.

(Refer Slide Time: 00:32:04)



So we will look back into the slide and see the general expression of a quadric surface which has ten A B C D E F G H I J. So there are ten parameters to the generalized expression or a general expression of a quadric surface.

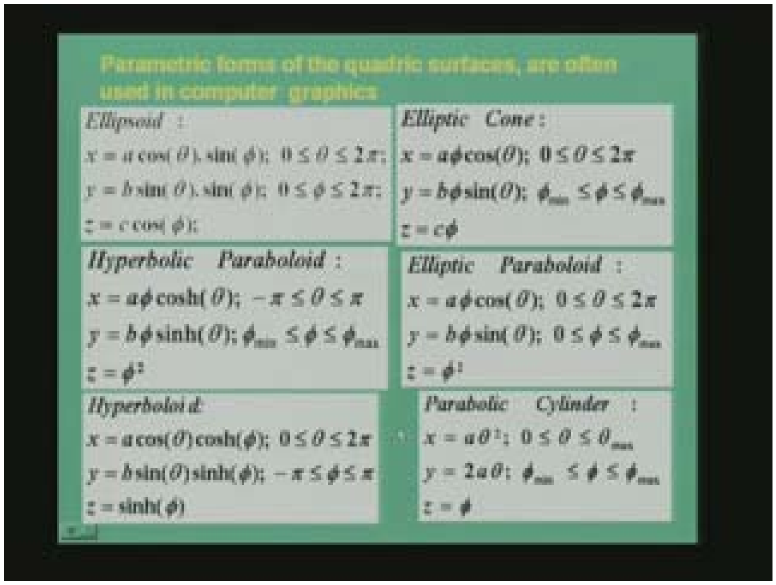
(Refer Slide Time: 00:32:33)



And the above is a generalization of the general conic equation in 3D in matrix form takes the shape like this. This we have done for the conic as well as in 2D and in 3D we have a surface a quadratic. So we have this X and the X transpose and this is the symmetric matrix S. Now you can insert the factors two here, if you remember we did that with the conic in 2D and take care of this 1 by 2 and two terms. These factors in the diagonal and the off-diagonal terms are differing due to the nature of the symmetric matrix you need a factor of two extra there.

So it is possible that you can avoid A B C and K, you can avoid the factor 2 there and you do not need a 1 by 2 factor if you provide the terms of D F E G and H. These are the terms which are contributing to the off-diagonal terms symmetric nature of the matrix if you put a 2 there this matrix S has a way and shape without having this factor of 1 by 2 and 2 inside. So you know these concepts from the generalized conic. Now parametric forms of the quadratic surface are often used in computer graphics.

(Refer Slide Time: 00:34:37)

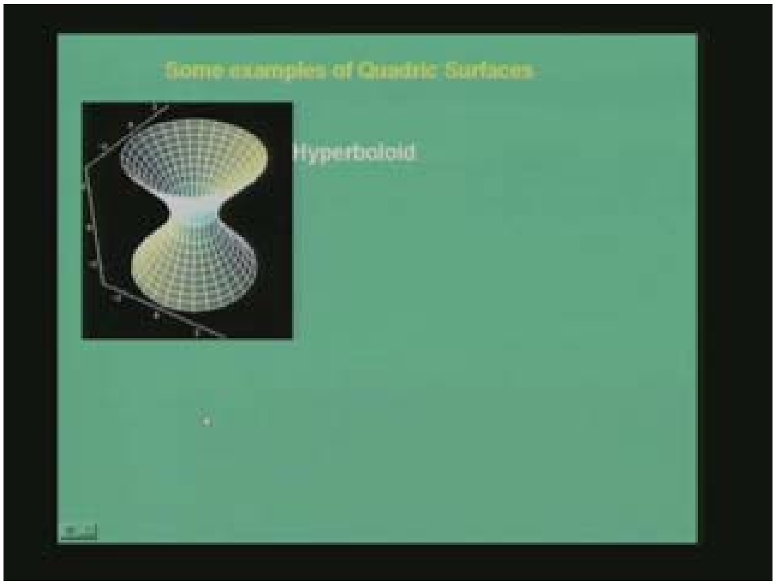


Mainly all the time you do not use generalized expressions because they are difficult to handle in mathematics unless you are very keen to handle matrix manipulations. Parametric forms give a good visualization. So again we will go through the different and special types of quadric surfaces and their parametric forms as much as possible. So let us look back, ellipsoid is given here, we have already seen this earlier, we will see into this elliptic cone. Now, if a is equal to b it is a perfect cone, elliptic cone will have a is not equal to b, that is an expression of cone in parametric form.

Hyperbolic paraboloid well we will see the examples of these surfaces in the next slide but just note down these expressions. And now we have a cosine hyperbolic and sine hyperbolic terms and z is fi square hyperbolic paraboloid are very interesting properties.

Elliptical paraboloid also as given here and then we have the hyperboloid in general given by a product of a cosine and a cosine hyperbolic multiplication and a sine hyperbolic term in z and of course the parabolic cylinder. So we have about six of these, at least four of these expressions or for four of these cases will see example in the next slide. Let us start with the hyperboloid, the expression which I have given here.

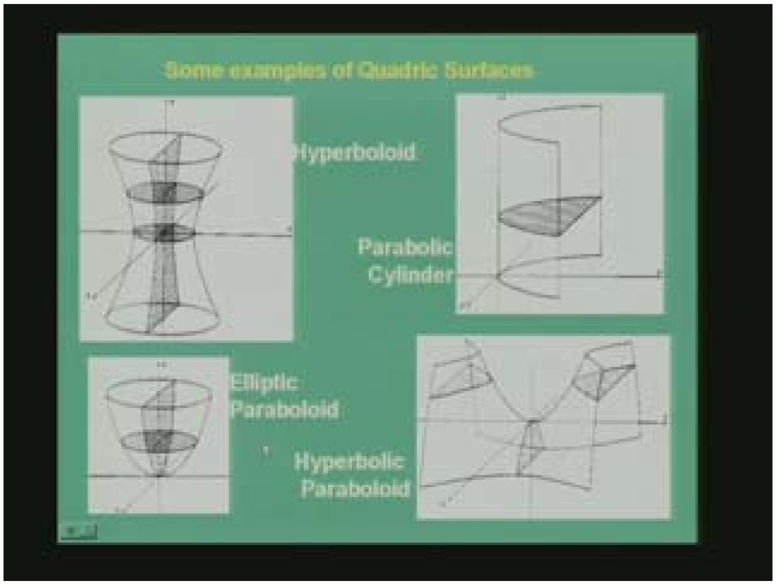
(Refer Slide Time: 00:34:53)



Let us look at how the hyperboloid looks like. That is an example for a hyperboloid. It is a wire frame diagram shaded with a color to give you not the effect of intensity illumination like Phong and Gouraud but this color indicates the direction. So you have three axis x, y and z and that is how the hyperboloid will look like in terms of the wire frame diagram. We will provide a sketch of this hyperboloid which will actually look like something like this. I think it is easy for you to visualize these sketches where there are certain planes provided at the origin on the top and along the x y plane.

We have along the z plane as well and this is the sketch from the wire frame diagram given here. And then the expression of the hyperboloid was given in the previous slide as in the left bottom corner. That is the hyperboloid for you, the wire frame and the sketch diagram we will go ahead and see other shapes such as the parabolic cylinder which is very interesting.

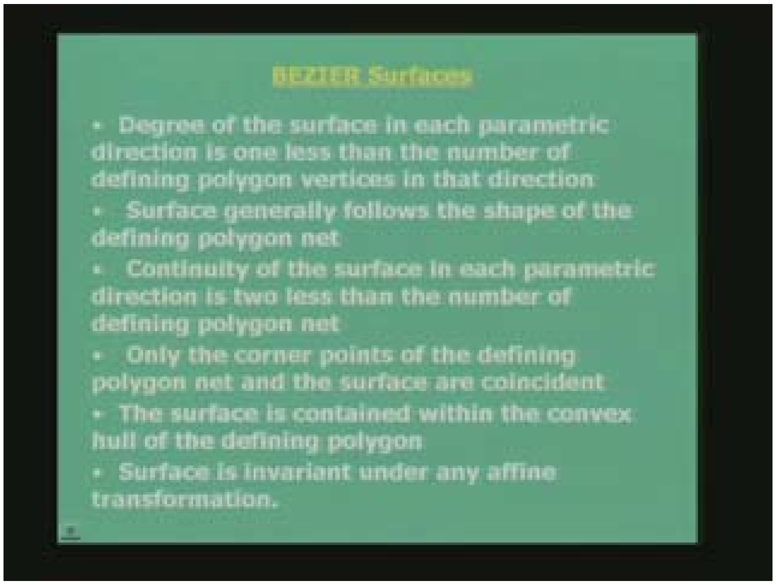
(Refer Slide Time: 37:18)



The parabolic cylinder is easy for you to visualize. It is a cylinder with a circular arc would have been a normal cylinder since the arc is parabolic now and it is called a parabolic cylinder. We had expressions of these earlier as well. Elliptical paraboloid is very interesting. The elliptical paraboloid looks like a bowl. Expressions were given in the previous slide. And then we have the most interesting the hyperbolic paraboloid. Hyperbolic paraboloid's sketch is given here. And I request you to use a computer simulation tool box in computer graphics to generate these surfaces or use your own c plus plus OpenGL concepts and draw these figures and create a wire frame. These are the examples of quadratic surfaces, hyperboloid, parabolic cylinder, elliptical paraboloid and hyperbolic paraboloid as well.

We will move on to Bezier surfaces which are extensions of Bezier curves. Degree of the surface in each parametric direction is 1 less than the number of defining polygon vertices in that direction. Now, these Bezier surfaces are extension of Bezier curves.

(Refer Slide Time: 39:14)



We talked of cubic splines, Bezier curves. We talked of quadric surfaces, now we move to Bezier surfaces as for a generalized conic we moved on to a generic surface, a parametric form or non parametric form implicit representation. So, for Bezier surfaces also the same could hold good where instead of a single curve I will have a set of Bezier curves embedded like in a wire frame, a mesh form which will try to represent a surface and that is what we call as a Bezier surface. And conditions which we talked about for a Bezier curve most of them hold good for a Bezier surface which so we are not going to talk about that that right now. So we talk of a polygonal net instead of a simple polygon in 2D for a Bezier curve.

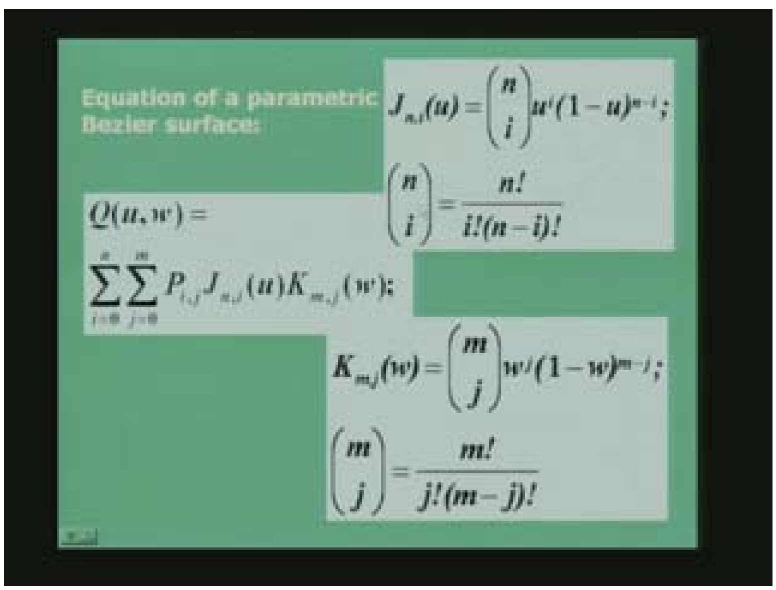
Now we have a polygonal net distributed in 3D. Assume it to be a wire frame net distributing and you are putting a surface on that so that is the net. So the degree of the surface in each parametric direction is 1 less than the number of defining polygon vertices in that direction and this is the same as for the curve where we had the number of vertices only for a polygon. In fact there is two or three directions now for a surface and the surface generally follow the shape of the defining polygon net so we have a polygon net, we will see how that is defined.

Continuity of the surface in each parametric direction is 2 less than the number of the defining polygon net. And only the corner points of the defining polygon net and the surface are coincident. And the surface is contained of course within the convex hull of the defining polygon. And the surface is of course invariant under any affine transformation. So if you look back into these points about conditions of Bezier surfaces they are almost similar to the conditions which we discussed in the last class about Bezier curves.

We discussed about a 2D polygon, the starting point, finishing point that the curve must be within the polygon hull that the tangent at the starting and the ending point must follow the nature of the polygon at the edges, edges of the polygon and starting and the finishing point. So in all those conditions the order must be 1 and less then the number of vertices. These surfaces are the extensions of all those curves. So if you look back I read it out again, the degree of the surface in each parametric direction is 1 less than the number of defining polygon vertices in that direction. Surface generally follows the shape of the defining polygon net.

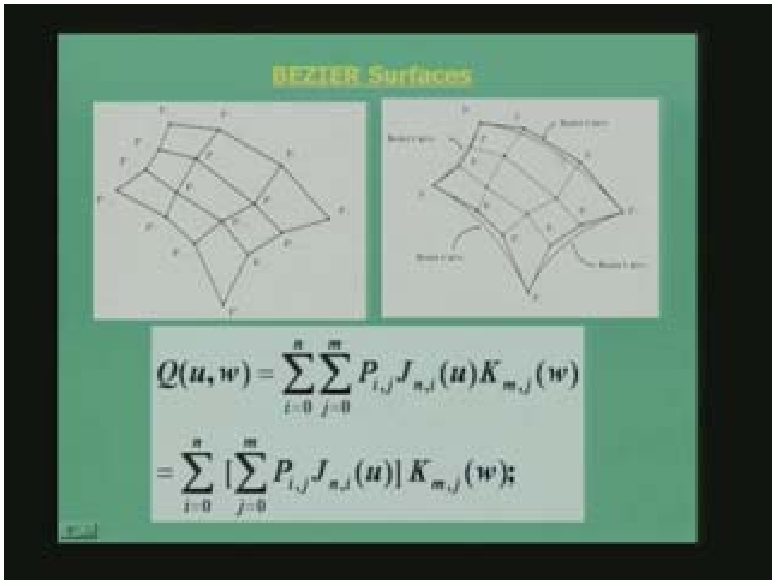
Continuity of the surface in each parametric direction is 2 less than the number of defining polygon net. Only the corner parts of the defining polygon net and the surface are coincident. The surface is contained within the convex hull of the defining polygon and it is invariant under any affine transformation so that is the equation of the parametric Bezier surface. So now you see there are two parameters u and w instead of t that is what you have and these are the control points P_{ijs} and J_{ni} of this is known to you earlier. Let us look at the expression of J_{ni} of u it is exactly the same of what we did for our parametric Bezier curve the K_{nj} also follows the same rule. K_{nj} of w follows the same expression, u replaced by w, n replaced by m, i replaced by j.

(Refer Slide Time: 40:50)



So the nature of these two expressions is same so we have two parameters u and w instead of t in the case of a Bezier curve. So what about the Bezier surfaces and how does it look like?

(Refer Slide Time: 00:41:08)



This is an example of a polygonal net. Like a matrix you have $P_{0,0}$ to P_m into n, m is equal to n in this square mesh and we want to fit instead of a single polygon in 2D or 3D you could have fit a curve through this now we have the whole mesh and we want to fit a surface on the 3D domain. So this is what it could result in.

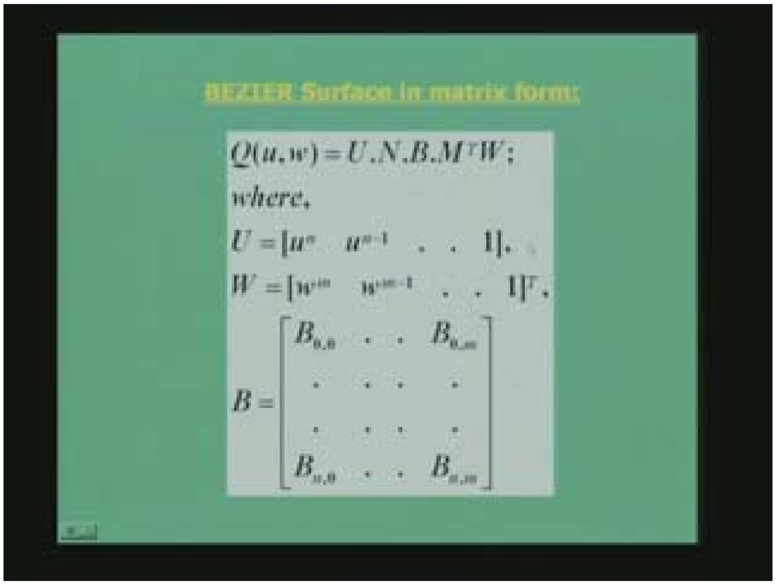
We can have a Bezier curve along all of these at the end and the surface should start from this Bezier curve and end at the other Bezier curve and it may not pass through all these points but it should have the other properties. So if you see here the Q(u, w) which is given by this expression j indices running from 0 to n and m can be represented by this expression where, if you look at this expression within this bracket, that is a Bezier curve for any value of i. So if any value of i running from 0 1 2 or 3 from 0 to n that is a Bezier curve for any vertical value of i.

And similarly if this is assumed to be a constant the outside expression also is a Bezier curve. So basically it is a combination of Bezier curves in two dimensions. You have a sequence of Bezier curves say along x and a sequence of Bezier curves along y they are all interpolated and combined together to relieve a Bezier surface. So you have a Bezier curve at the starting and the finishing point of these lines of the surface, the ends of the surface is not end points of the curve.

Ends of surfaces will be lines or curves. So in one direction which you have which is x let us say in x y plane which is U controlled by the parameter U and in other direction you have w which is controlling other direction. So that is what you have for your Bezier surface. If you look back you can have little Bezier curves at starting and finishing points and along each dimension u and w being discussed about and these two directions, the parameters.

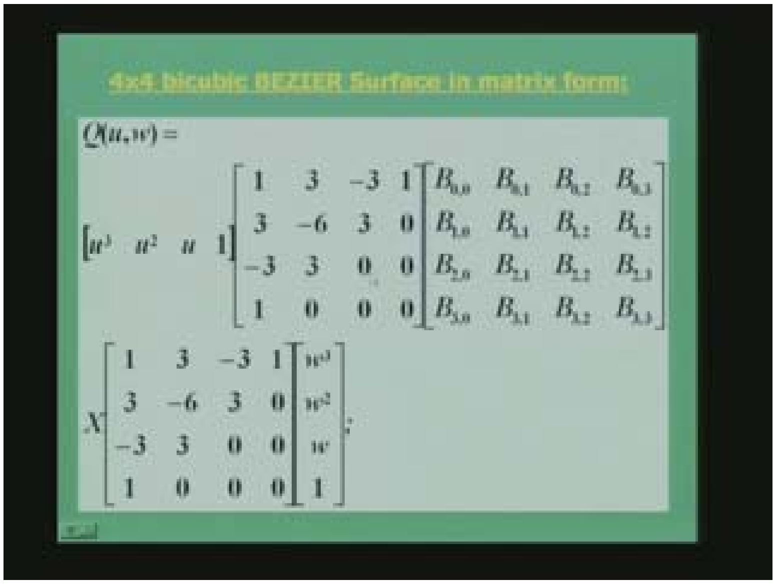
The Bezier surface in matrix form can be written as this where you talk about the ts in the case of curves so you have Us you have Ws and this is the matrix B. This is the matrix B which is the set of coordinate points which defines where the surface should lie, defines the geometry and well U and W are defined here already, U and Q as defined this is the parametric row vector and the column matrix B is the 1. So if we look at n and m what are n and m? That depends upon the degree of the curve you are fitting.

(Refer Slide Time: 00:43:03)



So if you have a 4 into 4 bicubic Bezier surface in matrix form this is what you will get.

(Refer Slide Time: 43:13)



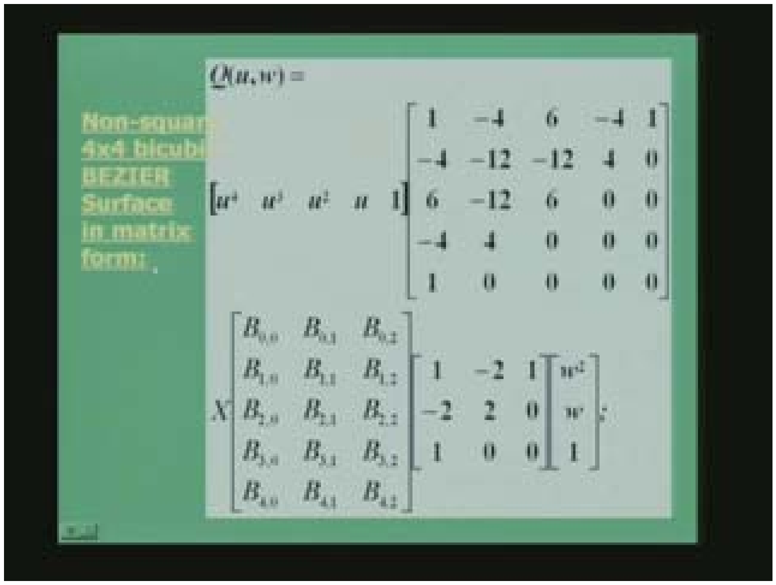
We will look back in to the expression of Q(u, v) you have U N B M transpose and W. u and w are given, this is u and this is your w. We discussed about this earlier depending upon the order. Then you have N B and M. When we talk of N B and M this is your N, this is your B matrix and this is your M. If you see, these two the N and M are same. N and M are same and they look like what can you guess? You have seen this matrix earlier. This N and mat M matrix one is the transpose, of course the other one since it is symmetric it does not matter much in this particular case. You have seen this matrix when we discussed about the matrix representation of the cubic Bezier curve.

Cubic Bezier curve had this matrix which we defined with those coefficients lambda ij at the beginning of the class today lambda ij for the N into M matrix. That is the matrix here which you have, the bicubic because there are two of those and these are your B_{ijs} the 4 into 4 points in space which define the polygon net. You have the four control points, 4 into 4 that is 16 of them which define the polygonal net and those are put into the matrix.
So, if you want to fit a 4 into 4 bicubic Bezier surface using a matrix equation you need this cubic matrix form from the Bezier curve you need these elements up to the order in which you want since it is bicubic it is cubic nature and you just put this 16 elements in this particular form.

It may not always be the case that you need to fit a bicubic curve only over 4 into 4 points in space. You may be given 5 into 3 a matrix of 5 into 3 points instead of 4 into 4 or it could be 4 into 5. So what do you do, you may need to fit a bicubic Bezier surface because that dictates the order of the surface and you may need to fit over control points given by the geometry condition which could be 4 into 4 that is 16 as we have seen earlier or it could be even 3 into 5 or 5 into 3.

So what happens if it is 5 into 3? Let us take a case. When we talk about non square 4 into 4 bicubic Bezier surface in matrix form and if you look at this 5 into 3 a non square matrix is what you are using and the matrix m into n is no longer the same, they are not symmetric, they are not the symmetric of each other.

(Refer Slide Time: 45:20)



This is the matrix which you obtained earlier for the case of the fourth degree Bezier curve. $J_{4,0}$ $J_{4,1}$ up to $J_{4,4}$ fourth degree Bezier curve when we discussed about this matrix this is the matrix form and the other one should be the quadratic. Therefore, based on the number of points given or the control points over which you have to fit the surface you have to decide on the order of the elements in the direction u and w. The parameters u and w will point at two different directions of the surface let us say x and y in this case. So u and w you will define that if there are five points along a particular direction you can choose up to fourth order.

If there are three points minimum we leave three points then you can use the second order and hence those matrices m and n which we discussed earlier will be dictated by the geometrical conditions of this polygon net. The geometrical conditions of this polygon net is going to dictate two matrices m and n in the formulation. If you look back into this expression here you already have what is U and W B matrix is defined by the geometry of the control points of the polygonal net. These N and M are the matrices dictated by the order of the Bezier surface which you are going to fit.

The order of the Bezier surface which you are going to fit is going to dictate the size of the matrix M and N in that expression. So if you see here these M and N which we discussed about the bicubic 4 into 4 or a 5 into 3 depending upon the direction U and V will all be dependent on B. So if you will roll forward you will see that for a 4 into 4 bicubic Bezier surface in a matrix form 16 points are given and this is the matrix element. Whereas if you have a non square polygonal mesh to be fitted and that is non square 5 into 3 but you want it will fit a 4 into 4 bicubic Bezier surface then on one side you may have up to the order four and on the other side you may have the order two. It all depends on the polygonal mesh available. The number of roll points available in each direction is going to dictate what is going to be your degree of the order M and N.

I leave it as an exercise for you earlier in this class to check up the equations or obtain the equations for $J_{2,i}$ and $J_{4,i}$ and that will help you to obtain this matrices M and N which is given in the slide here.

Please follow all the home assignments given and work out all the expressions which we have done in the last two classes specifically at the last class today and that will help you to master these equations in practice. And before you go to the program in environment using any simulation tool box or write a program using a high level programming language like c plus plus with OpenGL or PHIGS as your standards in any environment you should master this analytical expressions yourself, try to draw these curves then simulate and see how these curves come out when you try to plot them.

Either to simulate a curve or to fit a surface or to obtain a curve for a trajectory in animation or to fit an experimental data in 2D or 3D you are using these cubic splines, Bezier curve, Bezier surfaces or even B splines. We have not discussed the other types of curves and surfaces due to the time limitations available to us. But if you are able to master this you will be able to learn about any other types of surfaces in terms of B spline surfaces or coons.

There are various types of other surfaces which are talked about in literature and used by experimental people, computer graphics scientist and engineers to fit various plots in 2D and 3D or to come up with representations to represent the solid or to obtain a trajectory in a curve in 3D. So I would request you to kindly go ahead with these derivations and obtain all the different forms of matrices we have studied so far, also the different parametric expressions. Look into the blending functions where two or three different curves are meshed together to obtain from one to another and of course interesting geometric recursive definitions of the Bezier curves. That brings us to the end of the lectures on curves and surface representation in today's lecture. And in the last three lectures we have seen different types of representations of curves and surfaces.

And I would request you to kindly see them again, write them again, try to derive them again, run to the mathematics without any help if possible to obtain a control over this representation.