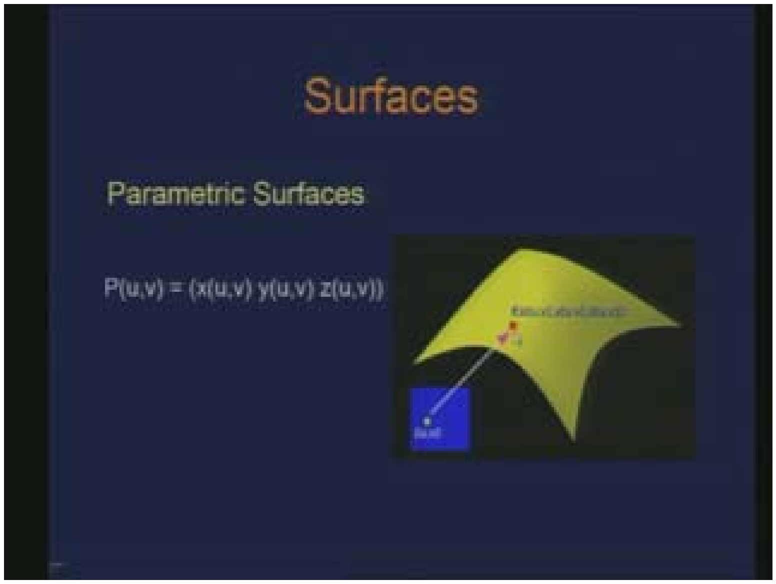
Introduction to Computer Graphics
Dr. Prem Kalra
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi
Lecture # 17
Surfaces (Contd…)

We have been talking about surface generation. So we started with the simplest example where a surface is obtained as an interpolant between four points. The four corner points are given to you and you want to have a simplest definition of the surface. Then we actually extended the problem definition. Instead of having just the four corner points given to you we specified two boundary curves and we obtained a surface interpolating these two boundary curves. And then the problem was further extended where instead of having two boundary curves we had four boundary curves.

In the case of two boundary curves the type of surface we obtain is the ruled surface. The interpolant for four corner point was bi linear interpolant which was an extension of linear interpolation. So again ruled surface was a combination of linear interpolation and the definition of the boundary curves which could be of any order. Then when we had four boundary curves given passing through the four corner points then what we constructed was the Coon's patch which is obtained as summation of ruled surfaces but we had to subtract the bi linear interpolation.

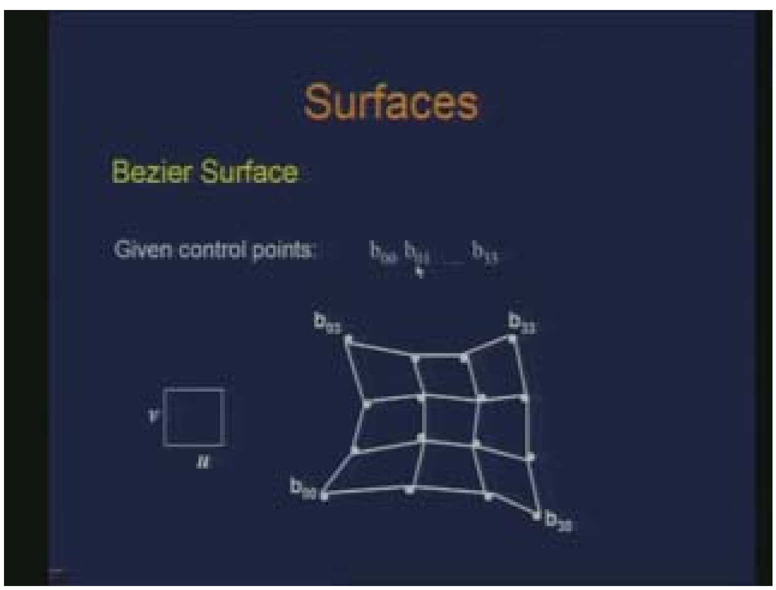
Today we will look at some of the surfaces which are more like extensions of the curves. For instance we have studied Bezier curve, B Splines and so on and so forth. Then how we generate surfaces extending those ideas which we used for curves? If we look at these parametric surfaces again as nothing but the parametric functions defined for the x, y and z using the parameters u and v which is defined in a domain which could be a rectilinear domain.

(Refer Slide Time: 03:49)



So, given the parameter value u and v where is the corresponding point on the surface? That is the idea of defining parametric surfaces. Now, for Bezier surfaces the problem which gets imposed is the following. Just the way we had in Bezier curves control points are given to you and you are required to construct the curve corresponding to or associated to those control points.

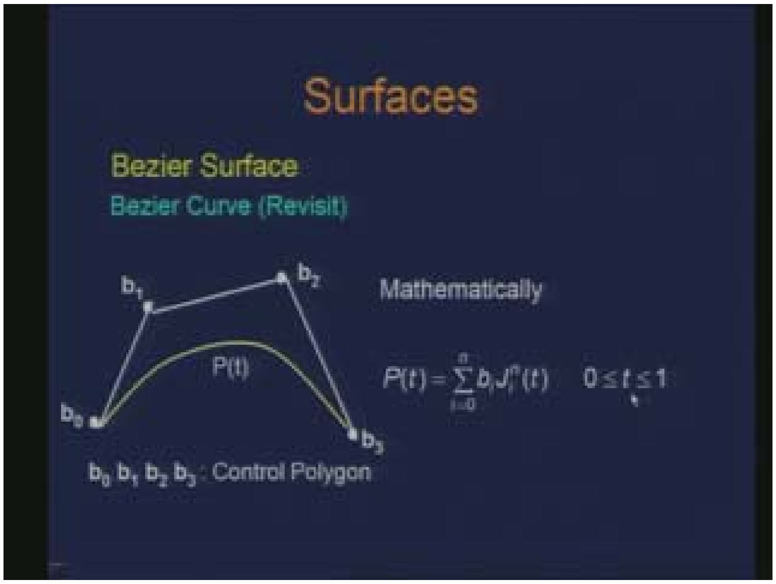
(Refer Slide Time: 04:21)



So here also we have control points given to us in the manner b_{00} b_{01} to b_{33} so the control points may look like something like this, it is basically a lattice or a grid of control points.

And this is the parametric domain we have for u and v. So we basically have the control points, index changing for u which is 0 then to 1 then to 2 and then to 3 and similarly the corresponding index for v. So we are basically extending the problem in an additional dimension. Just the way we had curves now we have an additional parameter v. Just to recapitulate what Bezier curve look like given the definition of the control points such as b_0 b_1 b_2 b_3.

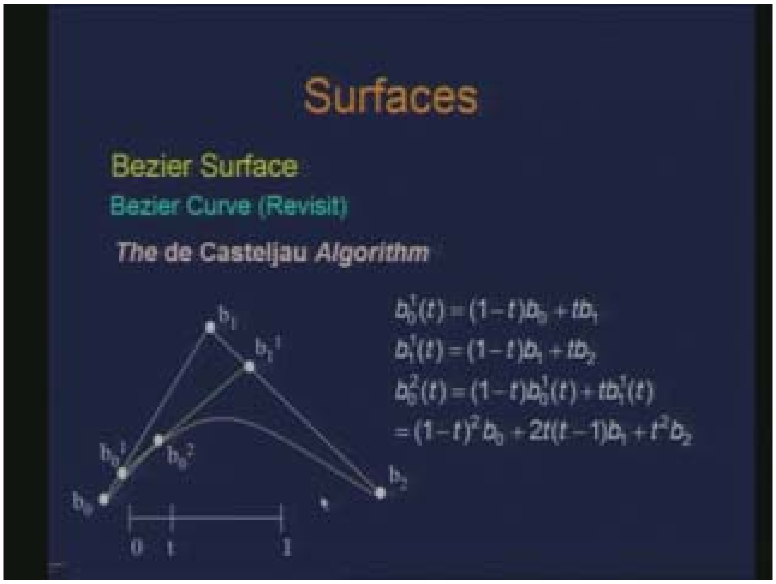
(Refer Slide Time: 05:43)



This is the Bezier curve which is constructed using these Bernstein polynomials which are defined in the parameter t. What we observe here is that the curve which is generated passes through the two end points and captures the shape according to the definition of the control points. Hence, we would like to do a similar type of construction for surfaces. That is just to give you some sort of a background for Bezier curve.

Let us go back and look at the alternate construction of Bezier curve which was to use the de Casteljau algorithm. What was de Casteljau algorithm? It was basically successive application of linear interpolation. There if I consider example where we specify the Bezier points as b_0 b_1 b_2 then what we do is for a given value of parameter t we perform a linear interpolation in the span of the control polygon. First we apply the linear interpolation for the span b_0 and b_1 and obtain a point b_{10}.

(Refer Slide Time: 07:18)



Similarly, we apply the linear interpolation in the span b_1 and b_2 and obtain the point b_{11}. Then again we perform another linear interpolation for the same parameter value t between b_{10} and b_{11} and obtain the point b_{20} which is a point on the curve. This was de Casteljau algorithm. Now, having seen this construction do you suggest the method which could possibly be used for constructing Bezier Surfaces?

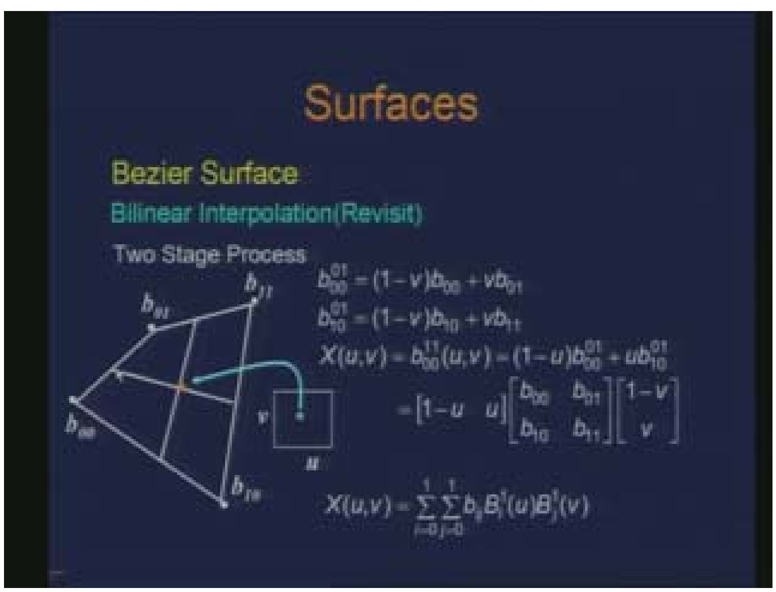
These are in the range from 0 to 1 so for any value of t I perform this and that gives me the point on the curve. What is that we are trying to achieve is, through successive application of linear interpolation I obtain a point on the curve. So the basic operation which is required for the purpose of construction is the linear interpolation. Now the question is, can some of the notion here be extended to do the construction of surfaces?

What are we trying to do is basically trying to use the simplest primitive of constructing a curve which is linear interpolation. What is the simplest parameter of constructing a surface? It is the bi linear interpolation. Now the idea is, can we use bi linear interpolation in a similar way the way we are using linear interpolation for constructing curves to generate surfaces. As Bezier curve is constructed using successive linear interpolation Bezier surface can be constructed using successive bi linear interpolation.

(Refer Slide Time: 10:00)

Bi-linear interpolation is actually a simplest form of a surface which is obtained interpolating the four corner points here. The way we do the construction of a point on the surface is basically the two pass linear interpolation. Once we run a linear interpolation in v to obtain a point here similarly a point here on this leg of the polygon and then we perform another interpolation in u linear interpolation and we obtain the point here and that is a point on the surface. This is the way we perform bi linear interpolation. Now we would like to use this concept successive bi linear interpolation to generate Bezier surfaces.

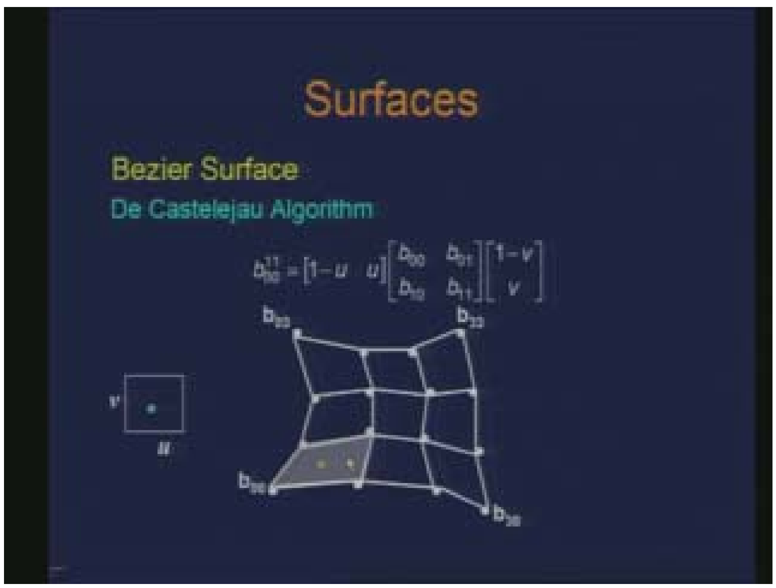
(Refer Slide Time: 10:38)



The algorithm is a straight forward extension of the de Casteljau algorithm using linear interpolation in the case of curves. Again if you define the control points for the surface as these points, these are nothing but the three dimensional control points and this is our parameter domain (u, v) so for a given value of (u, v) we would like to locate or generate a point on the surface. So we actually apply bi linear interpolation in the combination of four points which get generated in this control net the lattice of the control points.

For instance, this point is a consequence of bi linear interpolation of the four corner points this one, this one, this one and this one which are listed here b_{00} b_{01} b_{10} and b_{11}. This is the input. These are the inputs just the way you had control points for the curves. If I had to just construct a curve so I would have taken this point, this point, this point and this point to construct a curve for these four control points. These are just the last so I am considering a bi cubic case cubic in this and cubic in that so it is cubic in u and cubic in v so this is my fourth control point for (u) v is equal to 0.

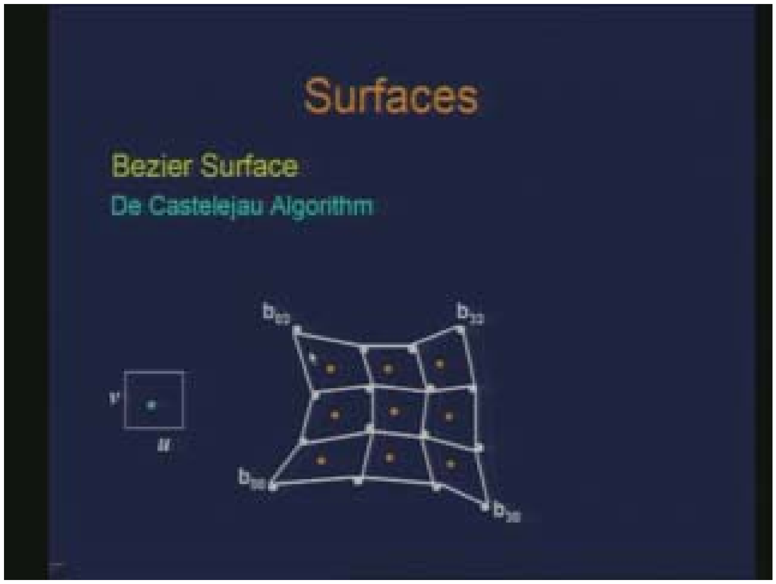
(Refer Slide Time: 12:16)



So just from one dimensional grid of control points now I am defining a two dimensional grid of control points. So, that is why these indices are used. What I have constructed here is a point located here as b power 11_{00} which is the point constructed using bi-linear interpolation of these four corner points defined as this.

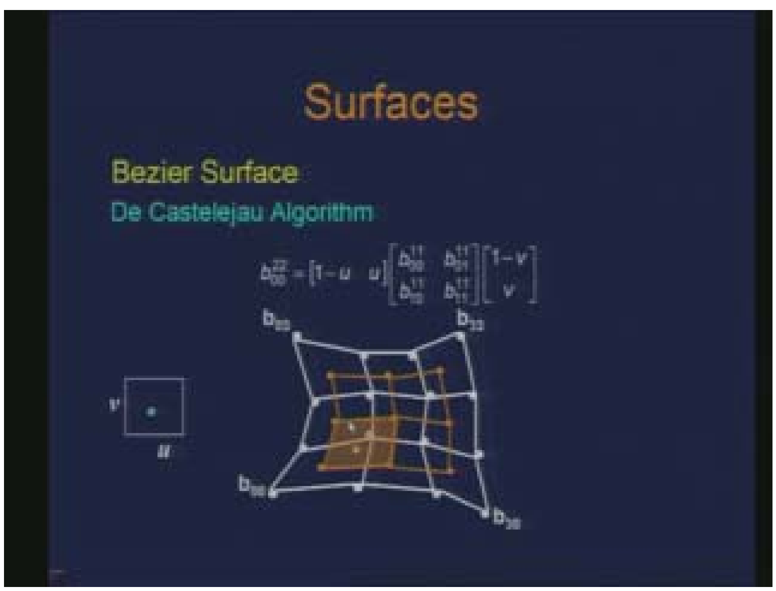
Now I can repeat this process for the rest of these four corner points which would mean I get these points. So this point is obtained from these four points, this point is obtained from these four points and so on. So all I am doing is taking the four corner points which are obtained from the grid of these control points and obtain the point corresponding to bi linear interpolation.

(Refer Slide Time: 14:50)



This is one level of bi-linear interpolation. Now I do the same thing just the way I had done in the case of the curve. I apply this bi linear interpolation repetitively which is that I obtain this point here as a bi linear interpolation of these four points which there constructed earlier step.

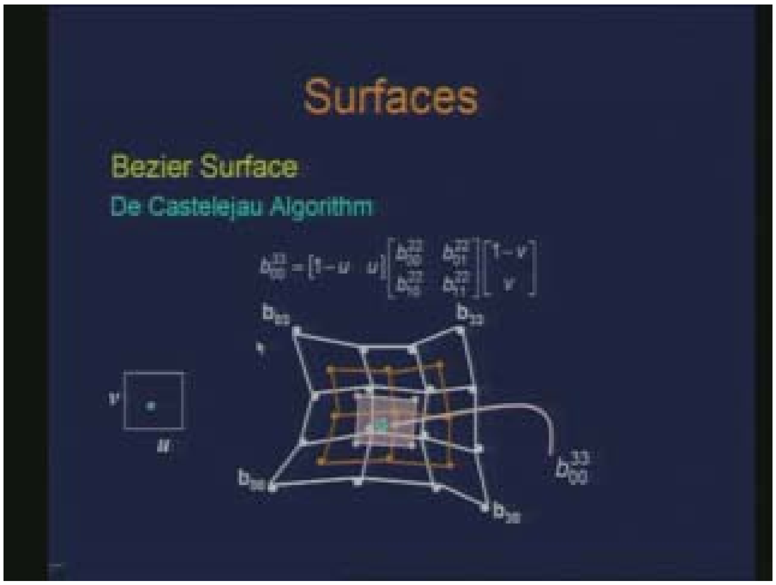
(Refer Slide Time: 16:24)



So I am basically doing a bilinear interpolation of the points like b power 11_{00} b power 11_{01} b power 11_{10} and b power 11_{11} which are these four points. And the corresponding point I get I call that as b power 22_{00}. So this superscript here is indicting me the level or the iteration of the column.

In a similar manner I can locate the point here and I can locate the point here and here. And now I can again apply once a bi linear interpolation corresponding to these as four corner points which is getting the point b power 33_{00} as a bi-linear interpolation of point b power 22_{00} b power 22_{01} b power 22_{10} and b power 22_{11}. So this is the point on the surface. It is exactly the same way as we had done in the case of curves.

(Refer Slide Time: 18:04)



Here we use this primitive as bi linear interpolation. I have the notion of doing this construction of Bezier surfaces through the grid of control points and I have used here the grid to be a square grid meaning I have four points in u and four points in v that I have a total of sixteen points which gives me the degree of the surface to be the same in u and same in v. So I could apply this de Casteljau algorithm in a direct way. Once again as in the case of curves the degree of the surface which you generate is directly related to the number of control points you specify.

Here if I specify four in this direction and four in that direction I have the degree to be three in both u and v. Now the question is, can I have these two degrees to be different? Again I apply de Casteljau algorithm. If I take the example here (Refer Slide Time: 20:29) what I have is 1 2 3 control points in u direction and 1 2 3 4 in v direction.

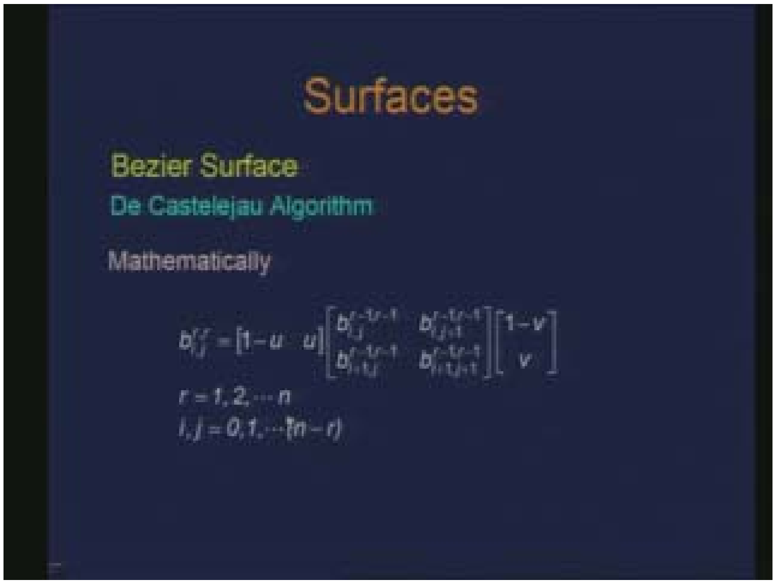
Now if I apply my de Casteljau algorithm I can apply for these four points to get this point similarly for these four points get this point similarly for these four point get this point and so on. So I can locate 1 2 3 4 5 6 points here. And again I can apply de Casteljau algorithm at this level to get this point using these four corner points and I can get this point using these four as corner points.

(Refer Slide Time: 21:28)

Now I am left with two points. So I cannot directly apply de Casteljau algorithm beyond this level. The way you can handle is that now you have to do a univariate de Casteljau algorithm. You cannot just repeat the de Casteljau algorithm at this level. There is a sort of special case which gets generated. So you can apply the de Casteljau algorithm up to the minimum of the levels or the degrees defined in u and v and beyond that you take that as a univariate case.

As far as the algorithm is concerned we can mathematically see just the way it was in the case of curves. So you can sort of represent this as just a simple bi linear interpolation using the levels r gives you a recursive definition of formulation of de Casteljau algorithm where r could change from 1 to n and these indices could change from 0 to n minus r. Then r becomes n and that is where you terminate.

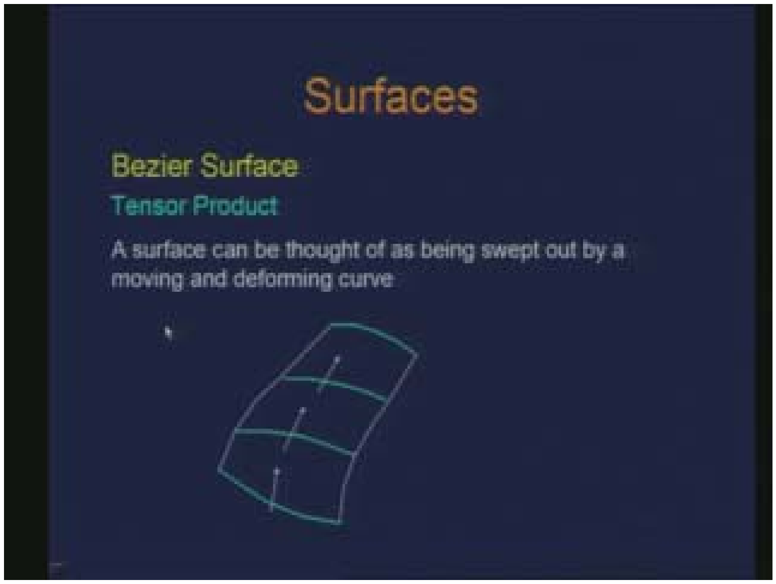
(Refer Slide Time: 23:11)



This is not really a limitation but this has some awkwardness in applying the de Casteljau algorithm when we have the degree in u and degree v not to be the same.

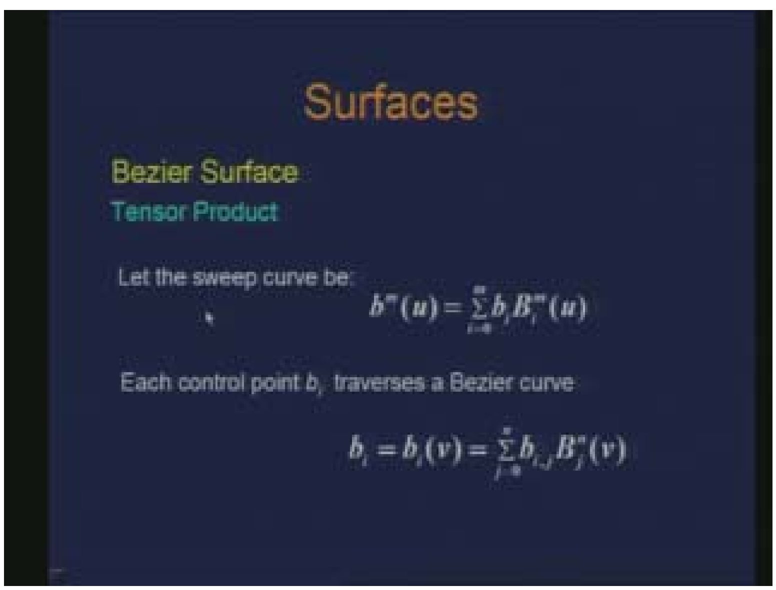
Now the question is that can we devise some way or some better way of defining the surfaces where it is considering these degrees in u and degrees in v in a very natural way? You do not have to do anything extra or special as a special case. In fact what you can think is the Bezier surface can be thought of as being a surface swept out by a moving and deforming curve.
Therefore here again the method has four curves. So the idea here is that I can think a surface being generated where a curve is being swapped by moving and it is also deforming.

(Refer Slide Time: 24:25)



So the swept surface or the envelope which I get is the Bezier surface. What do we mean by this? Again I can define this curve which I am using for getting the swept surface. I define this curve in u so I can sort of do this construction in a univariate way just in u where I have these Bernstein polynomials defined and these are the associated control points Bi's. This is just a simple Bezier curve.

(Refer Slide Time: 25:22)

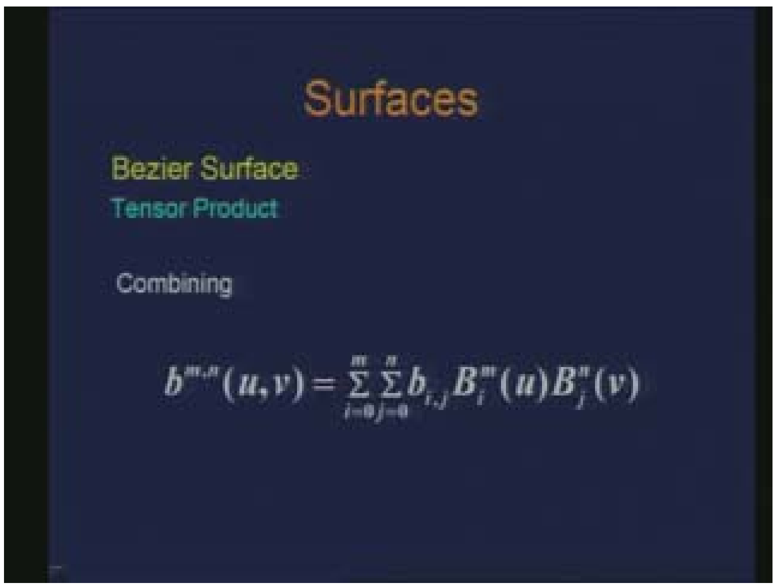


Now I can consider the corresponding control points I have here which are these Bi's as traversing a Bezier curve. I am constructing a Bezier curve here using these as the control points Bi's.

Now imagine that these Bi's actually traverses or moves along a Bezier curve which is to say that Bi's are nothing but again some sort of a Bezier curve defined through the parameter v in the other dimension where I could use the Bezier control points as Bij's and the corresponding Bernstein polynomials. What I mean is, when I use a Bezier curve traversing along another curve, actually the control points of the Bezier curves are now traversing along another curve so that is where I get the movement as well as the deformation of the curve.

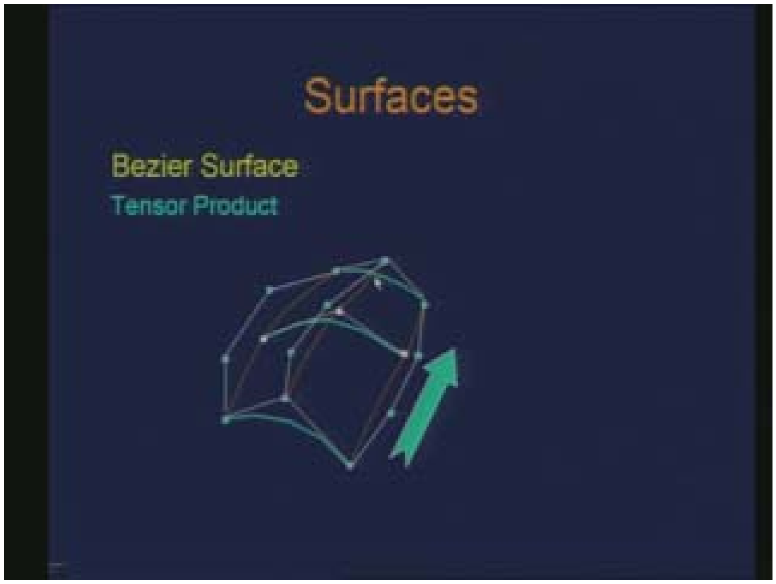
Now if I just combine these two, I have this as the univariate case to define the Bezier curve in u and the control points here are themselves lying on a Bezier curve defined through the control points defined for v in the orthogonal direction for the parameter.

(Refer Slide Time: 28:28)



Now if I just substitute the Bi's what I have obtained and combine it I get this. That is nothing but my Bezier surface which combines the definition of these Bezier curves a curve which is being swept and the control points of this curve move along another Bezier curve which is defined by another set of control points. So geometrically if you want to see this is what is happening.

(Refer Slide Time: 29:38)

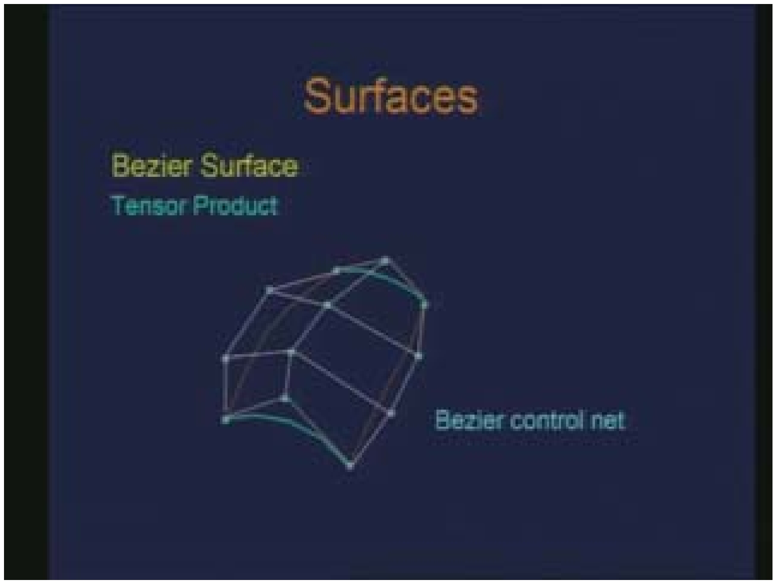


I have this to be the direction u, this to be the direction v so here I have the degree for u is 2 and the degree for v is 3 then this curve is actually moving in this direction. So the control points with respect to these curves are actually lying on another Bezier curve which is constructed using these as the control points. So this particular curve is another Bezier curve generated using these Bezier points. Therefore this is what I mean by movement of the Bezier curves such as the control points of these Bezier curves move along another Bezier curve. So this particular control point you see in an intermediate curve is actually lying on a Bezier curve obtained by these four points.

So envelope which gets constructed using this is nothing but the Bezier surface. This is such a very natural way of combining univariate cases which are nothing but construction of Bezier curves. So what do we have as the control lattice for the surface are nothing but these.

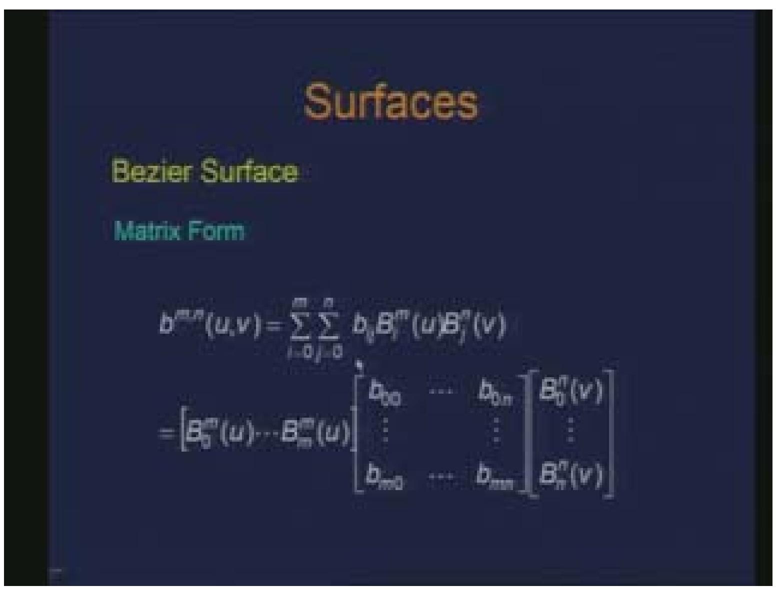
These are all points here the twelve points I have and they are nothing but the control net or control lattice of the Bezier surface. Once again it is very easy to establish that this representation or this way of constructing Bezier surfaces is equivalent to de Casteljau's construction just the way we had done in the case of curves.

(Refer Slide Time: 31:21)



If you want to see in a matrix form this is the way we can have the combinations when we combine the univariate cases. Again I can rewrite this expression in a matrix form where I can decouple the basis of the Bernstein polynomials in this case in u and in v. And this is nothing but the matrix for control points.

(Refer Slide Time: 32:55)



Since these control points are nothing but 3D vectors they sort of form as a tensor product because these are matrix of matrix where in this case this matrix is a vector. This is nothing but a vector and that is why we call these as tensor products. Therefore you are using these tensors.

Now we can look at some properties of these Bezier surfaces. Just the way we have the properties in the case of Bezier curves many of those properties can straightaway be extended in case of Bezier surfaces. For example these are affine invariance. What do we mean by affine invariance? If I apply affine transformation to the control points so it will be equivalent if I had applied the affine transformation to the surface itself.

(Refer Slide Time: 34:05)

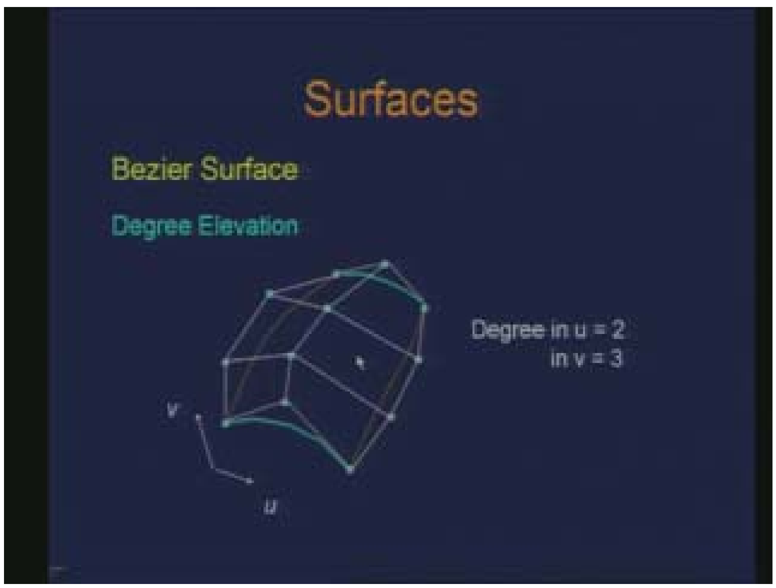
Similarly they also exhibit the convex hull property. So the surface which is generated resides in the convex hull of the control points. And we also observe that the surface interpolates the four corner Bezier points, also the boundary curves which we have are nothing but the Bezier curves obtained by the respective boundary control points. Therefore this Bezier curve is nothing but obtained by these three control points, this Bezier curve is nothing but obtained by these four Bezier points. So the boundary curves are nothing but the Bezier curves which you would have obtained by using the boundary control points of the Bezier net.

In other words, the surface which gets generated interpolates these boundary curves. So the other property which is a very desirable feature of these Bezier surfaces is that if I change the position of the control point then the surface changes accordingly just as we observed in the case of curves. So these surfaces are not interpolating surfaces except that they have boundary curve and end point interpolation but they capture the shape which is provided through the control points of the Bezier net. Therefore these are the control handles for the shape of the surface. Now the next question we had observed is the degree elevation in the case of curves.

What we basically mean by degree elevation is that for the same curve you want to elevate the degree. And since in the case of Bezier curves the degree of the curve is related to the number of control points so elevating the degree by 1 requires you to add an additional control point. That is what degree elevation in Bezier curve means. Now the question is, can we have the same thing for Bezier surfaces? This could be a desirable feature particularly, as a pre processing to the direct application of de Casteljau algorithm I may actually elevate the degree in one direction before I apply de Casteljau algorithm.

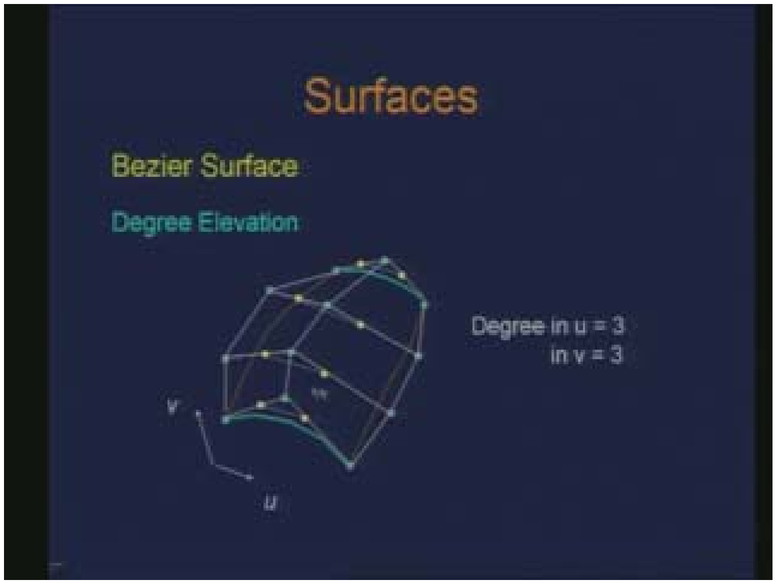
What is the process of generation? It is actually that the curve moves such that the control points for that curve moves along a Bezier curve. So it would not be adequate just for the boundary curves. If I take the example where for the surface defined through these control points has got degree in u as 2 and degree in v as 3 then if I want to elevate the degree in u to 3, then clearly just by counting the number of control points how many control points should I have if I have to define a surface which is bi cubic means cubic in u and cubic in v? It is sixteen points and here I have a curve so it requires four additional points. What we can do is we can actually make use of the information from the univariate cases as Bezier curves.

(Refer Slide Time: 39:17)



If I were to increase the degree of the curve here I would have added a point such that the points which are generated have a linear combination of these two points and here another point which has got a linear combination of these two points that would have located two points here instead of one point there. Similarly, I could do this here for this set of three points, similarly for this and similarly for that and that takes care.

(Refer Slide Time: 41:26)

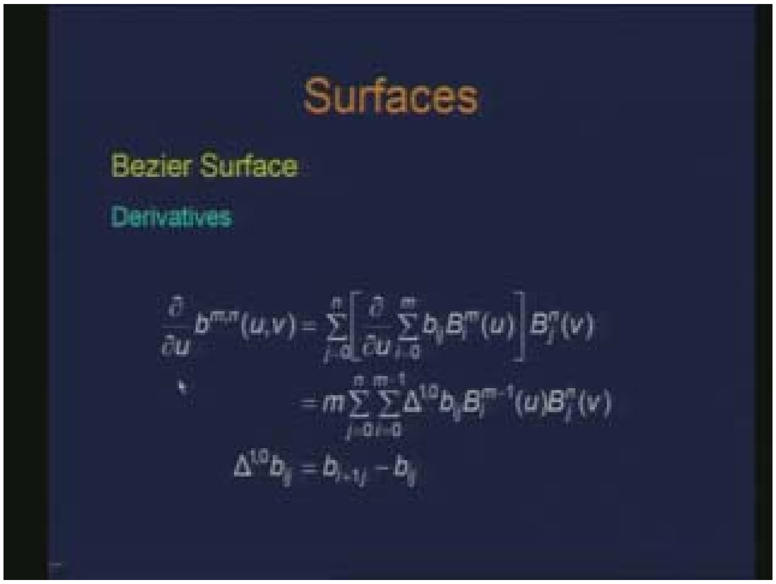


I have these yellow points as the points which are added to degree elevation instead of these points so I would use these yellow points. Now I have both in u and both in v four each as control points. This particular combination which I am trying to locate would use a parameter in u. So I am going to use similar ratios here. So, similar ratios would be used for this and this and the other points.

What I am doing is if you had to just do degree elevation of the Bezier curve I would have obtained the necessary ratios or the linear combination which I require just by comparing that the resulting curve is the same as the previous curve and compare the coefficients of the two because they are basically the same thing. Therefore I will do the same thing here and it does not change anything. And it is the same thing here if I were to elevate the degree in v I would have done for these control points.

Next thing is how we get the derivatives. Now I am talking about partial derivatives in u and partial derivatives in v because I have these bi parametric functions. So if I have to take the partial derivative in u then all I can do is take this inside so that this is only associated to the blending functions in u and then I can perform the derivative of these Bernstein polynomials which would give me from Bmi to B power m minus 1i. Therefore if you recall this it is nothing but actually taking a derivative of Bezier curves, it is actually the same thing. This is what i have done. I just basically decoupled these u's and v's and I obtained the derivative here and this operator delta 1, 0 operator just gives me the necessary first difference of points I require.

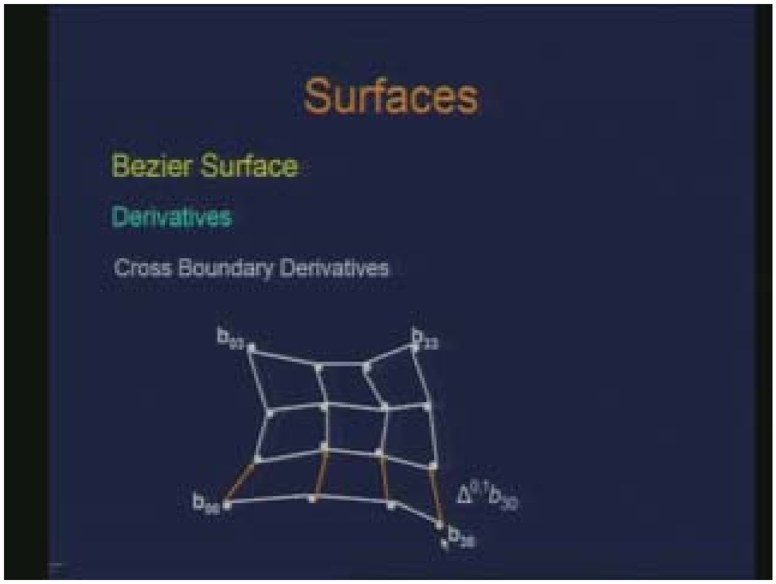
(Refer Slide Time: 44:11)



So delta 1, 0 is nothing but bi plus 1j minus bij. And again if you look at this and if I were to perform the partial derivative in v it is the same thing. Now instead of taking for u I will just take this for v and this delta operator as 0, 1 gives me the first difference of the points bij plus 1 and bij. And in fact when you look at the expression here this is again just as in the case of Bezier curves where the derivatives were Bezier curves themselves with less degree.

Similarly, the derivatives of Bezier surfaces are another surface where we use the control points as the first difference. So these are vectors they are not point values but vector values. So I can see that the cross boundary derivatives are again defined through the first span of the control lattice or the control grid in the respective directions. For instance, in this direction the derivatives are basically defined through the first difference obtained from this point and this point and the boundary.

(Refer Slide Time: 47:44)



So if you remember, in the case of curves the first derivative at the first point and the last point were defined through the respective spans of the control points in the polygon which is to say that the tangent vectors of the curve in direction could be obtained by the spans of the Bezier polygons. Similarly, the boundary derivatives are indicated by the respective spans which are there in the control grid of points. Hence this is essential when we are talking about compositing or joining two Bezier surfaces.

(Refer Slide Time: 49:38)