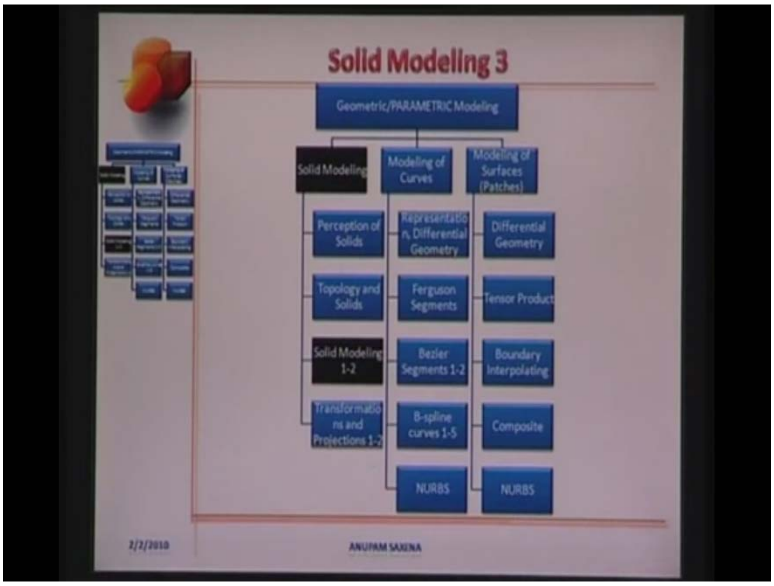
Computer Aided Engineering Design
Prof. Anupam Saxena
Department of Mechanical Engineering
Indian Institute of Technology, Kanpur

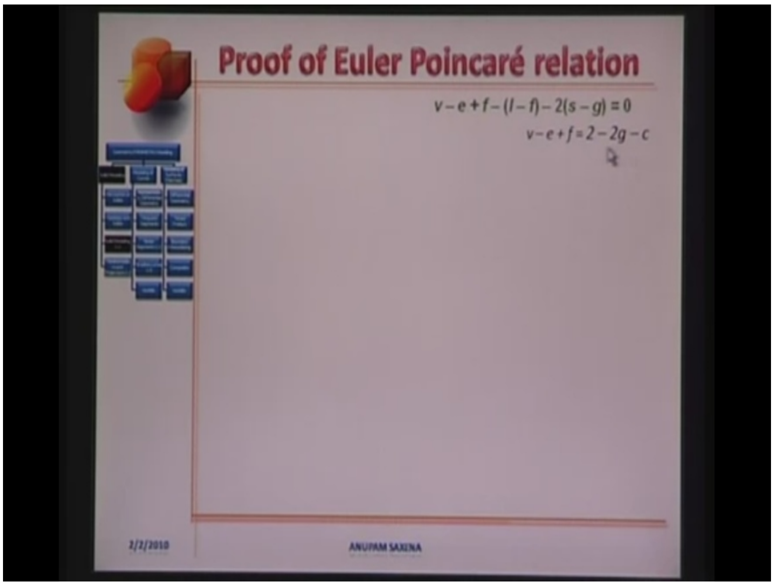
Lecture - 8
Solid Modeling

Welcome to lecture 8 of 'Computer Aided Engineering Design'. This is our last lecture in Solid Modeling.

(Refer Slide Time: 00:23)



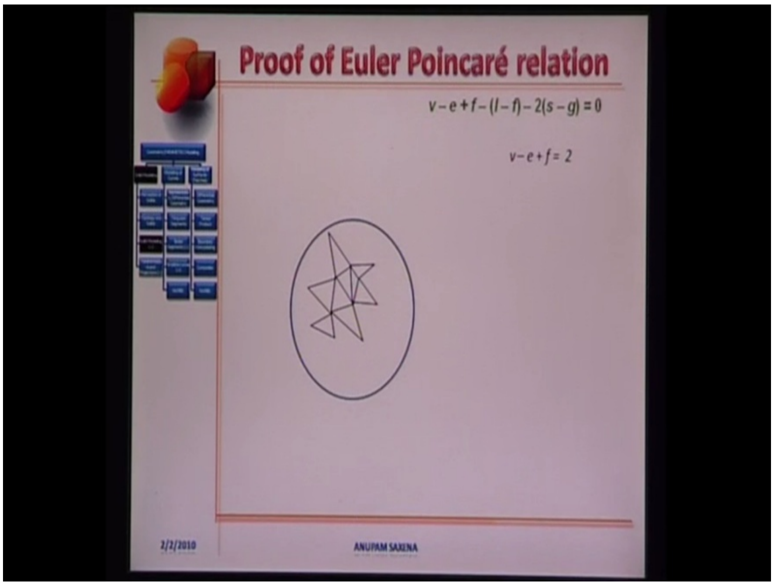
(Refer Slide Time: 00:55)



I am going be covering 3 topics here. 1) To prove the generalized Euler Poincare formula, 2) to give an example on parametric solid modeling and the last topic is going to be a little note on regularized Boolean operations. First a little in formal proof of the generalized Euler Poincare relation; v is the number vertices, e the number of edges, f the number of faces, l the number of loops, s is the total number of shares. That can have own genes value or own number of handles, g is the number of genes and the left hand side is equal to zero.

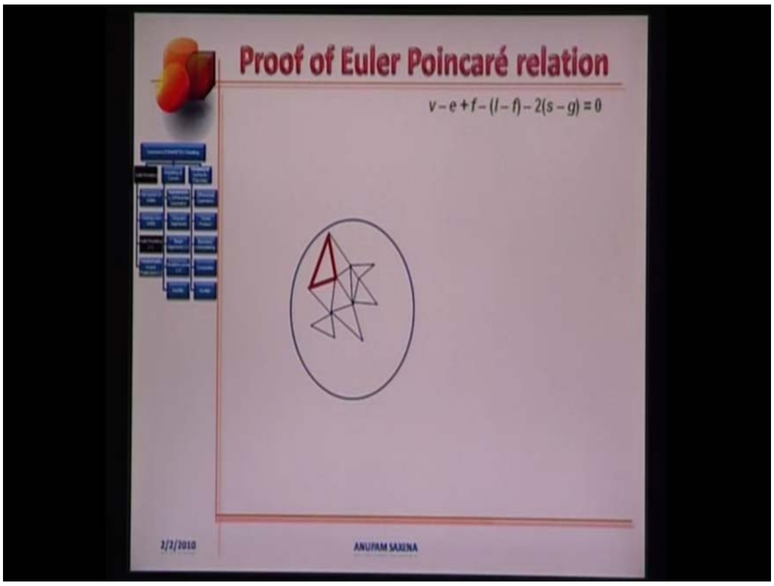
We will try to prove it in formally of course, we have seen and in formally proved is relation before, the left hand side is the Euler have respect for polyhedral solids and the right hand side is the Euler characteristic or smooth for it. Where g would be the number of handles and c would be the number of the boundary components.

(Refer Slide Time: 01:52)

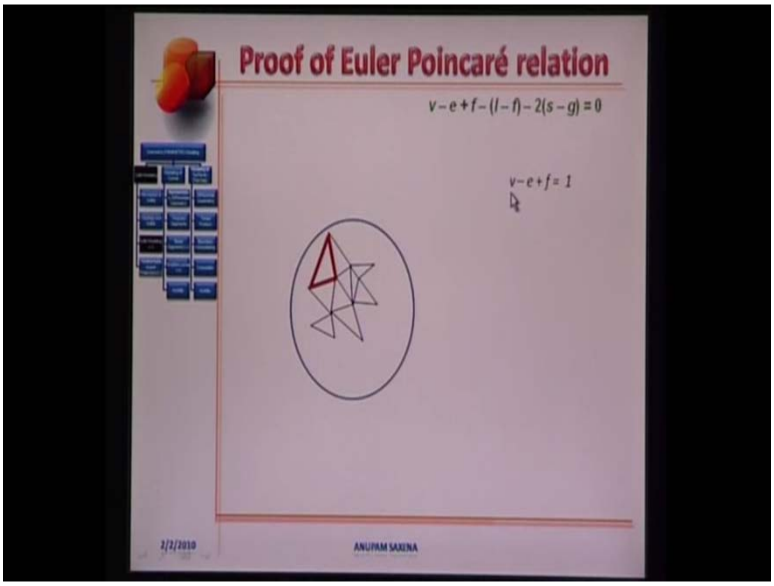


Let us start the proof now. Let us consider a spear and triangle it. We know that, this polyhedral solid is going to be having the Euler characteristic of 2. Let us take off a triangle of face from this spherical surface, to introduce a boundary component. When we do that, the number of faces get reduce by 1 and so a number vertices minus the number of verges plus the number of faces equals 1.

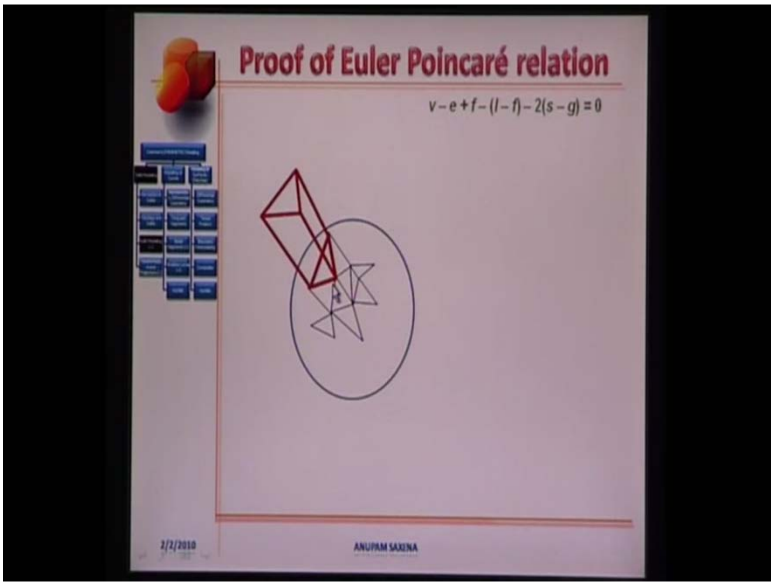
(Refer Slide Time: 02:13)



(Refer Slide Time: 02:23)

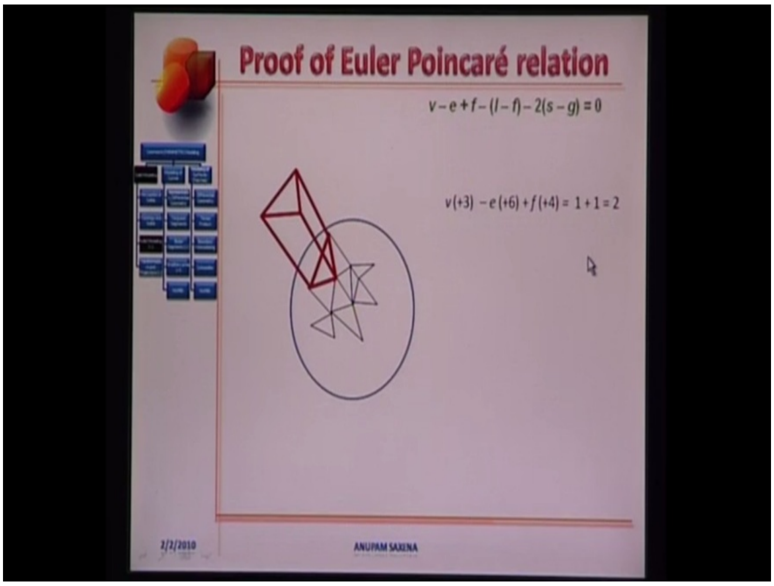


(Refer Slide Time: 02:41)



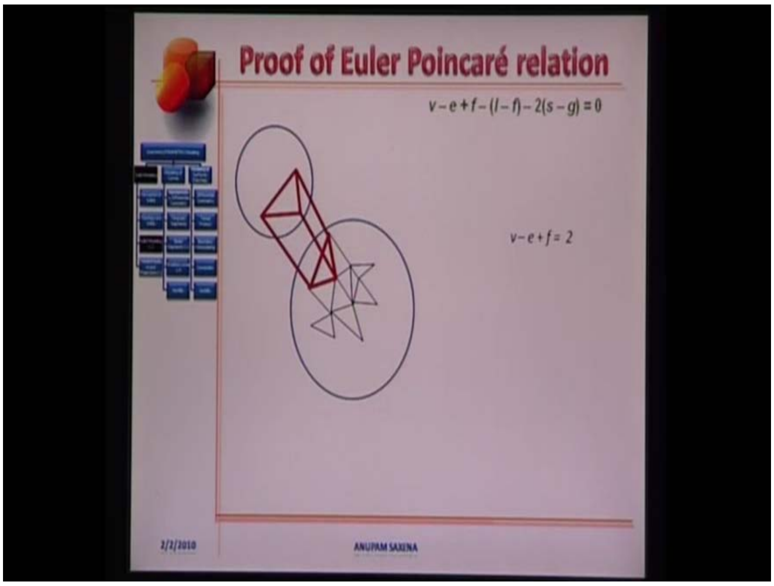
What we do now is we try to build up a prison, a triangle prison on this opened up surface. What have we done, we have introduced 3 plus 3 edges. We have introduced 4 faces; first face, the second face, the face on the back side the third and tuff face and we have additionally introduced 3 vertices.

(Refer Slide Time: 03:12)



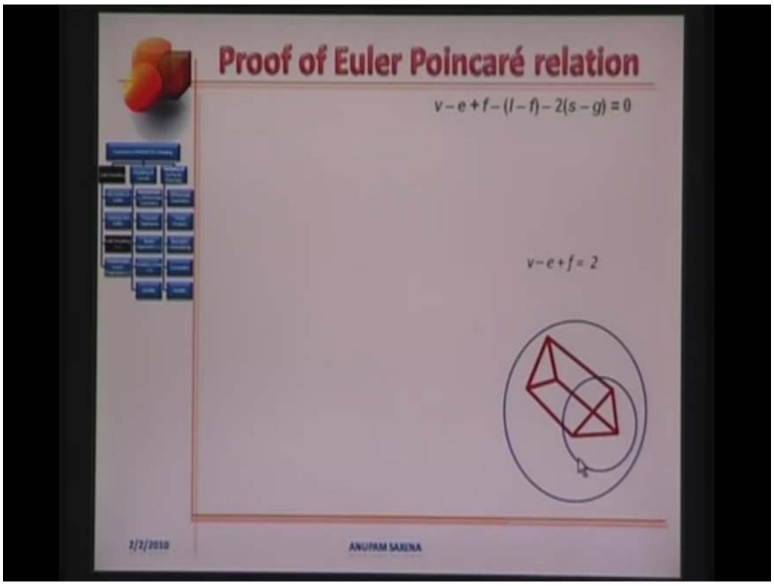
So in this Euler relation we have three additional vertices, six additional edges and four additional faces; because of which, the right hand side becomes 2.

(Refer Slide Time: 03:36)



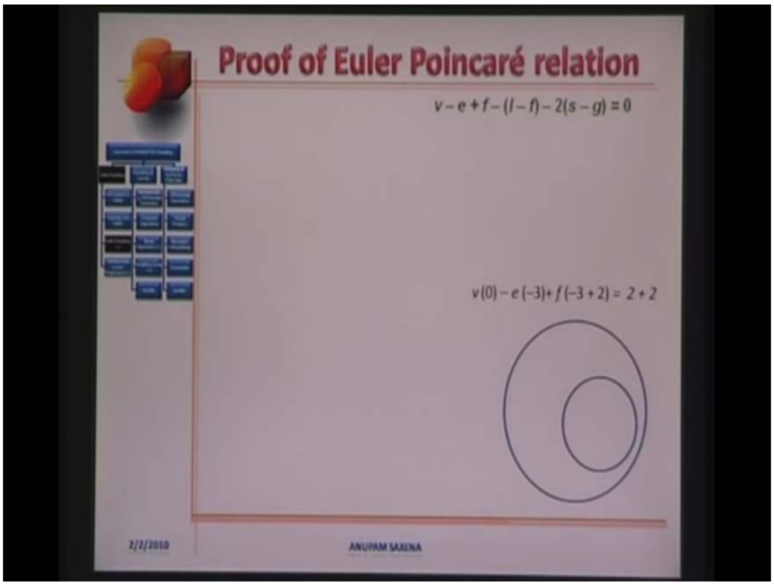
In the next step what we may want to do is we may want to blush out this triangular you know face in two spiracle space. In the Euler relation nothing changes because this thing is a valid solid and in fact this solid is homeomorphic to the spear.

(Refer Slide Time: 04:17)

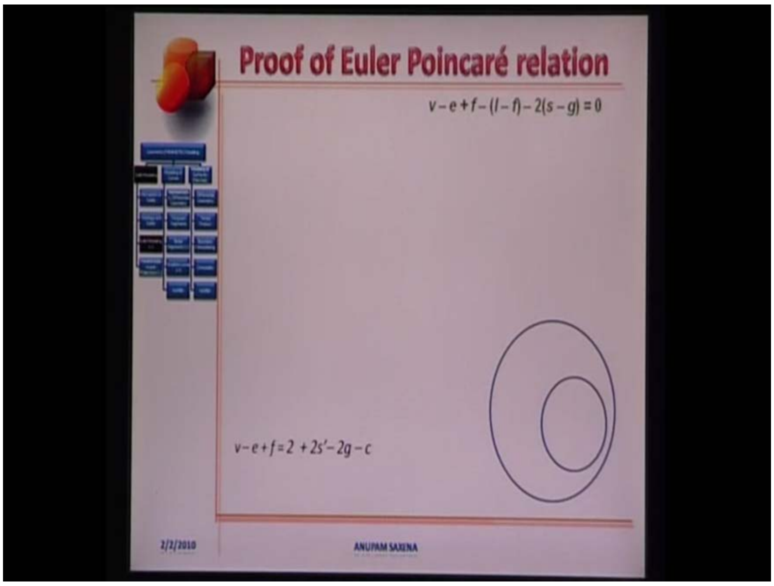


I could as well have constricted a triangle prison inside the parents fear, an acre of bushed out this triangle of peace into spherical surface. Again nothing would have changed; we would still have a solid which is homeomorphic to as pear clearly because, the Euler characteristic tells us where it is values 2. What we might want do now, as let go of this triangle of prison and cover up the outer spear and the Innes here, to leave out spherical wide within spherical solid. Let us see how? When we let go of this triangle of prison, we are deleting three edges; we are not deleting any vertex and we are deleting three phases. In addition we are re-introducing two phases, one phase on the outer spear and one on the Innes spear.

(Refer Slide Time: 05:42)

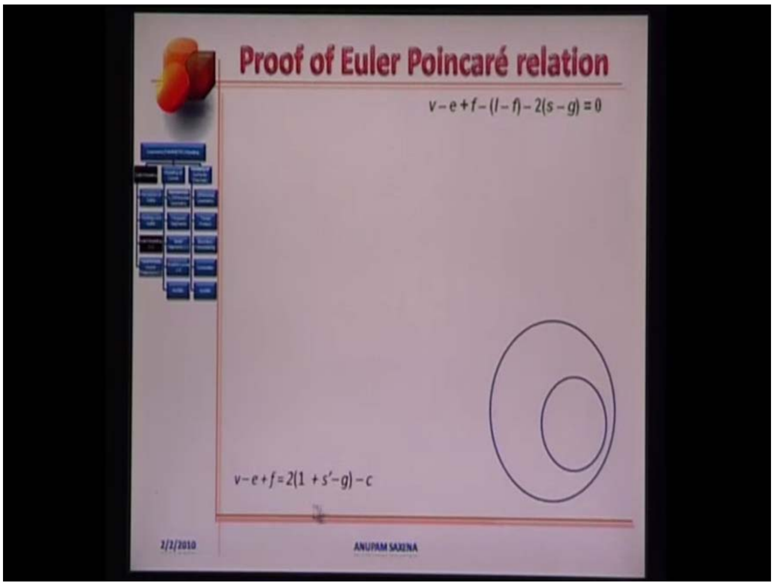


(Refer Slide Time: 06:50)

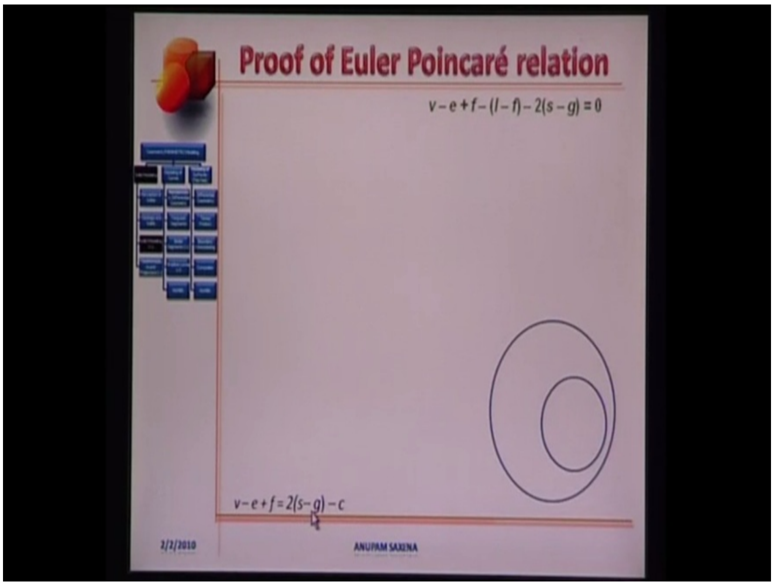


So the left hand side v minus e plus f, that we will have no change in the number vertices, in number of edges will get decremented by 3 and the number of phases will get decremented by 1; so the right hand side will be 2 plus 2. I can keep on introducing spherical shells; spherical wides inside sprint fear, which would help me, generalize your right hand side of Euler relation v minus e plus f, as 2 plus 2 s prime. S prime would be the number of shells that I have introduced into the sprint fear. I can combine the previous relation with this one and have a more general formula, as v minus e plus f equals 2 plus 2 s prime minus 2 times g minus c. These are the terms that I get from previous smooth solids. I can manipulate this right hand side further.

(Refer Slide Time: 07:27)

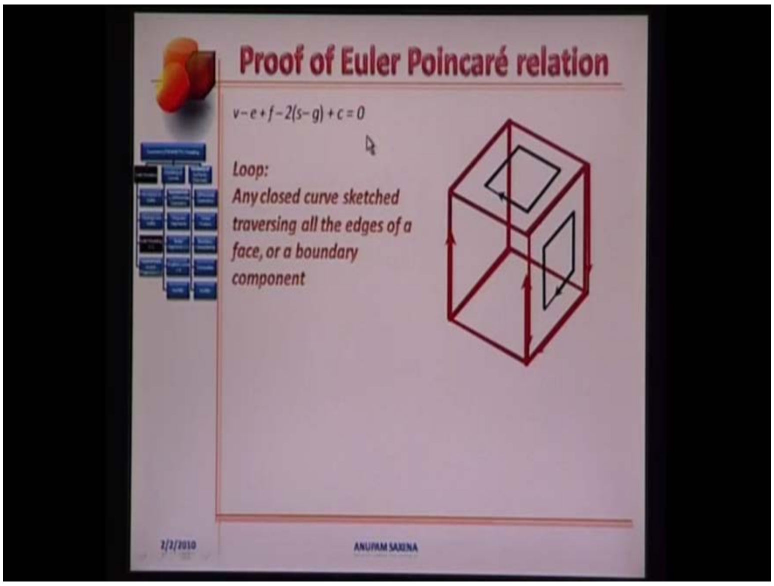


(Refer Slide Time: 07:56)



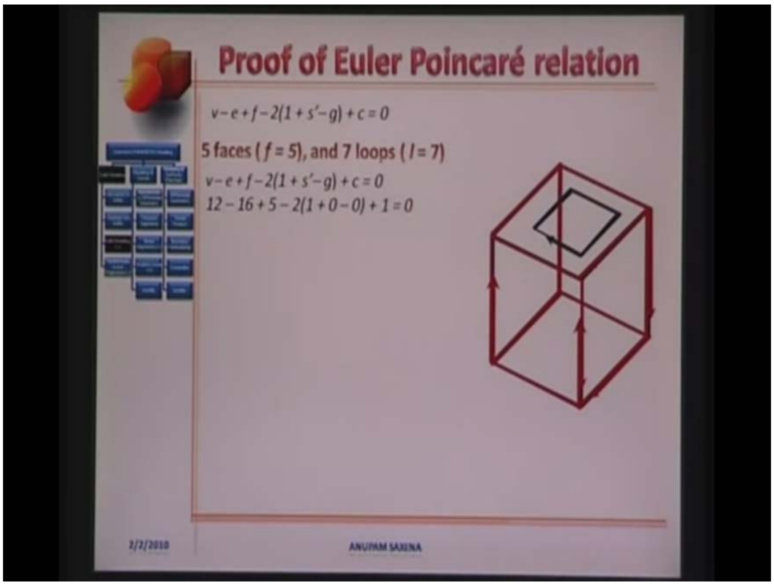
I can faceted to outside and I can rewrite the right hand side as two times within the parentheses, 1 plus s prime minus g minus c outside the parentheses. Once again s prime is the number of additional shells that I have introduced within the pincher. I can combine one plus s prime together and write it as s. So just thing that s would be the total number of shells. If I compare this relation with the Euler Poincare formula; I have the number vertices, the number of edges, the number of faces, the number of shells and the number of handles in place. All I need to worry about is to introduce this term here, in the Euler Poincare relation.

(Refer Slide Time: 08:35)



We come back to polyhedral solids with the same relation, that we had derived or we had proved in fumier of course last time and we try to understand what a loop is more familiar. A loop would be any closed curve sketched by traversing all the edges of a face or a boundary component. If you look at this figure here of a block, we had the top phase opened up by a boundary component. So this is the first loop we traverse the four edges of the phase, which is nearest was the second loop; the third loop traversing the bottom phase; the fourth loop traversing the top phase; fifth loop and the sixth loop and then will have a loop corresponding to the boundary component at the top. If I introduce any other boundary component, I will introduce a corresponding loop with that.

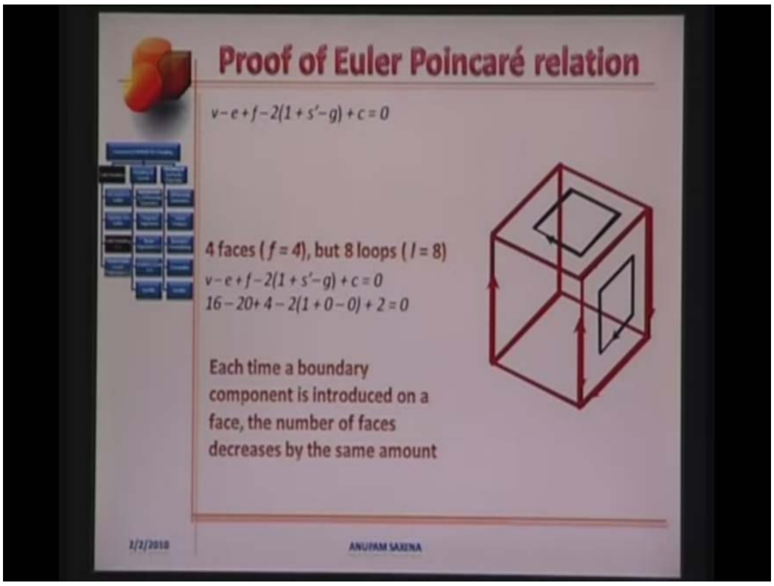
(Refer Slide Time: 10:19)



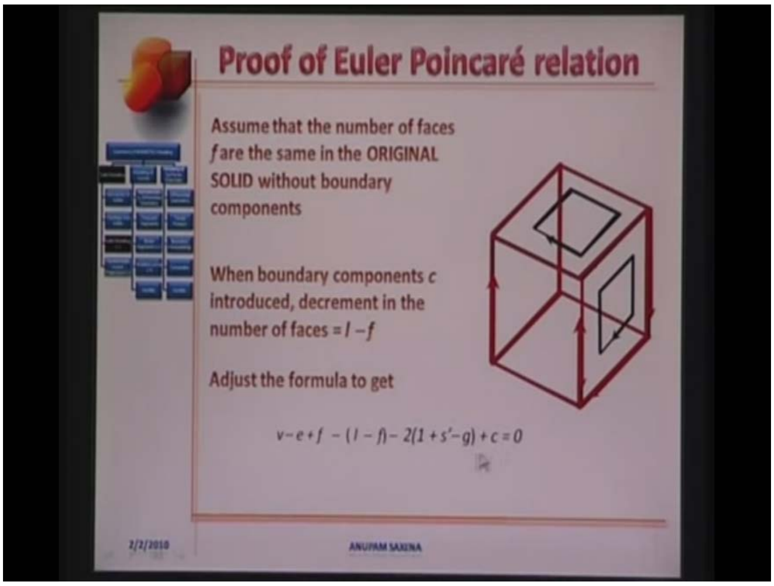
Now let us see how introducing different boundary components would affect the Euler Poincare formula. We start from the formula and now we segregate of the divide s into two parts. We say s 1 plus s prime for certain reasons, which I tell you about later. If I introduce a boundary component, I open up this top phase because of which and of course, I will have a loop here; because of which the number of phases will decrease by 1. So that the total number of phases for this polyhedral solid will be fire but, the number of loops will be 7. Let us verify the above Euler Poincare relation; we have 12 vertices, now 16 edges, 5 phases. We have no interior shell s prime of which s prime is 0, we have no handles, because of which g is 0 and we have one boundary component c, we see all the above Poincare relation is satisfied.

If, I introduce another boundary component and a loop whit it, we are opening up one more face because of which, we actually have 4 faces but, 8 loops. Let us verified the Euler Poincare relation again; we now have 16 vertices, 20 edges, 4 faces, again no interior. Shall I explain no handles and now we have two boundary components. So c is 2, where we have noticed each time a boundary component is introduce on a face; the number of faces will decreases by the same amount. If I have introduced let us say 2 or 3 boundary components the number of faces will decreases by 2 and 3 respective.

(Refer Slide Time: 11:56)

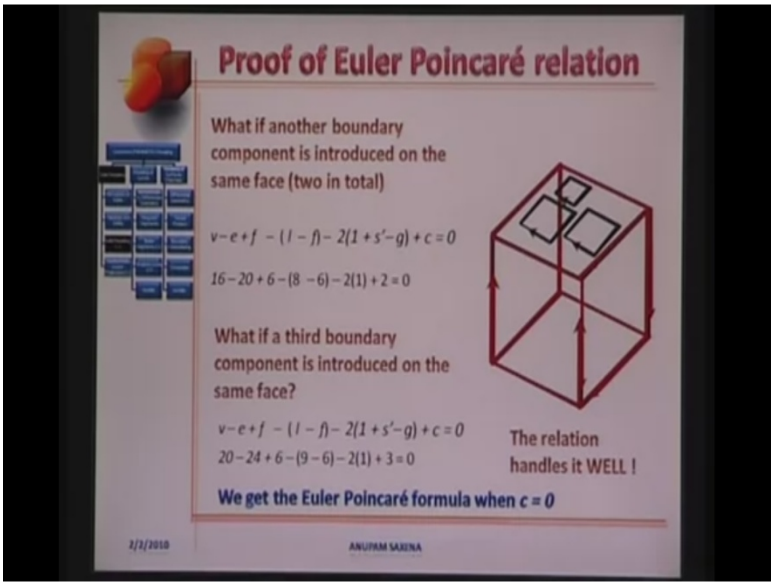


(Refer Slide Time: 13:03)



We can use this observation in this manner. We first assume that, the numbers of faces f in the Euler Poincare relation are the same in the original solid, without boundary components. We assume that, the original solid is a perfectly valid polyhedral solid, which does not have any boundary. Whenever a boundary component c is introduced or whenever a number of boundaries c is introduced, there is a corresponding decrease in the number of faces. In a sense, decrement in the number of faces will be equal to the number of loops minus the number of original faces and this difference l minus f will be the same as c. Once we know that, we can adjust the Euler Poincare relation assess, v minus e plus f minus within parenthesis l minus f minus two times within parentheses 1 plus s prime minus g plus c. Once again each time I increase c by say 1, I decrease the number of original phases by the same amount 1. I would repeat here, that c is the same as the number of loops minus the number of phases.

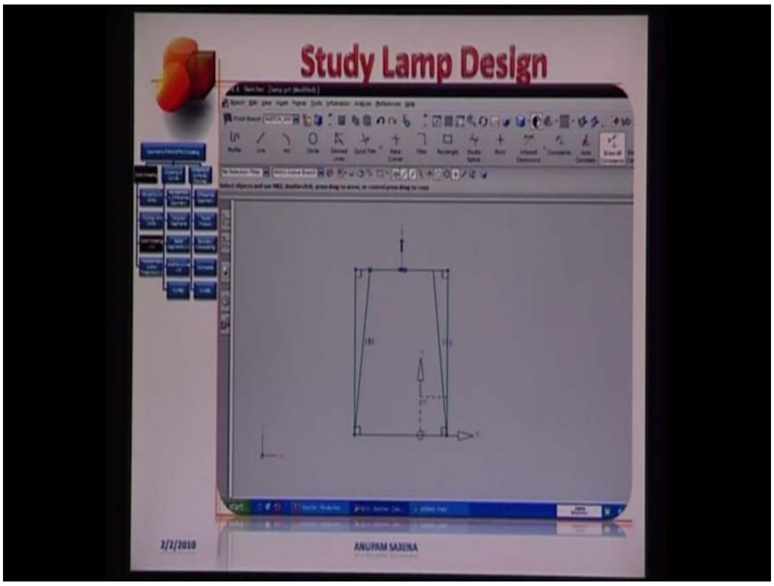
(Refer Slide Time: 15:04)



Question, what if another boundary component is introduced on the same phase that is a face will have two boundary components in total. This is the first one the loop and this is the second one the loop. Let see how the previous relation affects, well we have this relation here. We compute in number vertices as 16, 8 from the original solid; 4 here and 4 here. The number of edges will be 20; 12 from the original solid, 4 from this boundary component and 4 from this boundary component. The number of faces in the original lock will be 6; the number of loops will be 8. Now, 6 corresponding to the original solid and 2 corresponding to these two boundaries; so this is 8 minus 6. Well s prime is the number of shells within this solid, since there is no shell when this solid s prime is 0. So this is minus of 2 and the new number boundary components introduced is 2, 2 boundaries. We see that the left hand side concerned to be 0 and so the Euler Poincare relation hopes.

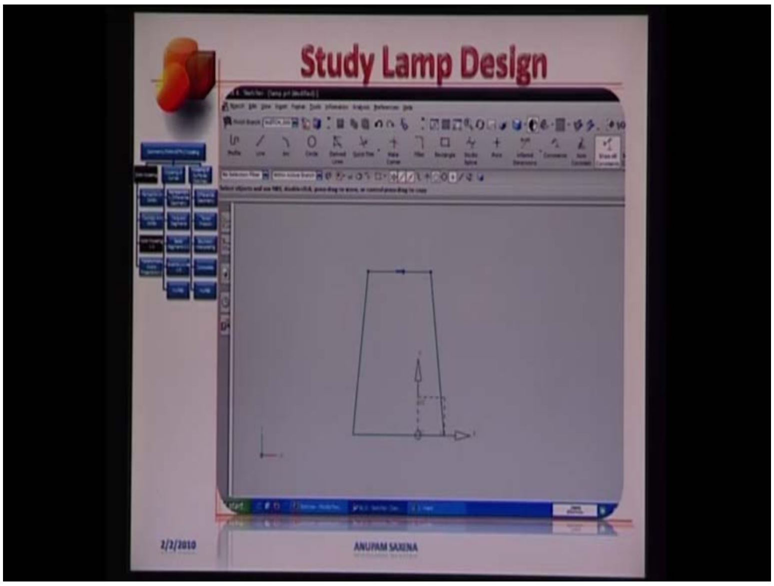
What if? Now a third boundary component is introduced on the same face. The number of vertices well now we train for additional the number of edges, well again before additional the number of faces in the original solid remained same. The number of loops will now incremented by 1, so this is 9 and the number of boundary components c will again the 3; because the 3 boundaries. Again c that the Euler Poincare relation hole introducing a boundary component on the same face, is equal to borrowing a boundary from a different face and that is possible. The reason why? This relation as good, finally we get the Euler Poincare formula for valid solids, without any boundaries; when c equals zero.

(Refer Slide Time: 18:25)



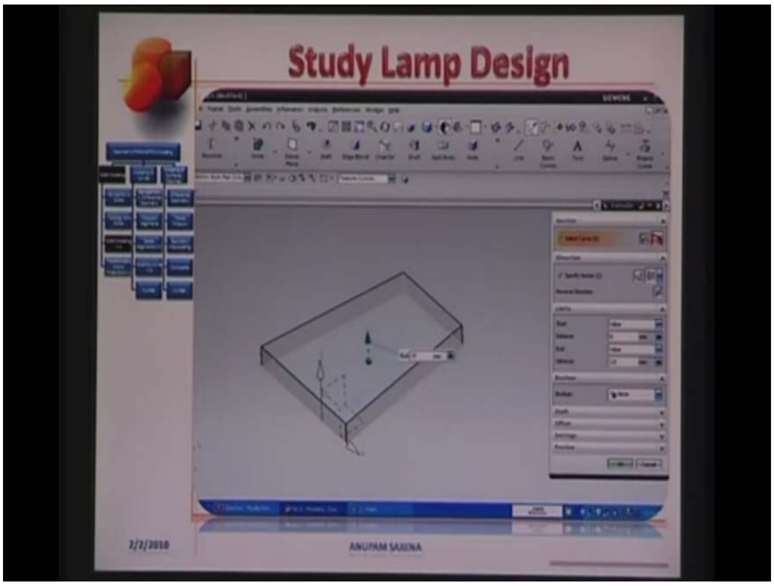
We now come to the second part of the lecture. This is an example on parametric solid modeling that, Mr. Shailesh Pandey has helped me to prepare. Here I would like to show case different features, that many solid modelers have; since we have done a work shop on the design of a study lamp or table lamp. I will use the same example, we are going to be trying to design a study lamp, where all prismatic features, first the design of a base. I would like to have a trofosoidol base we call this troposide from this rectangle here.

(Refer Slide Time: 18:55)



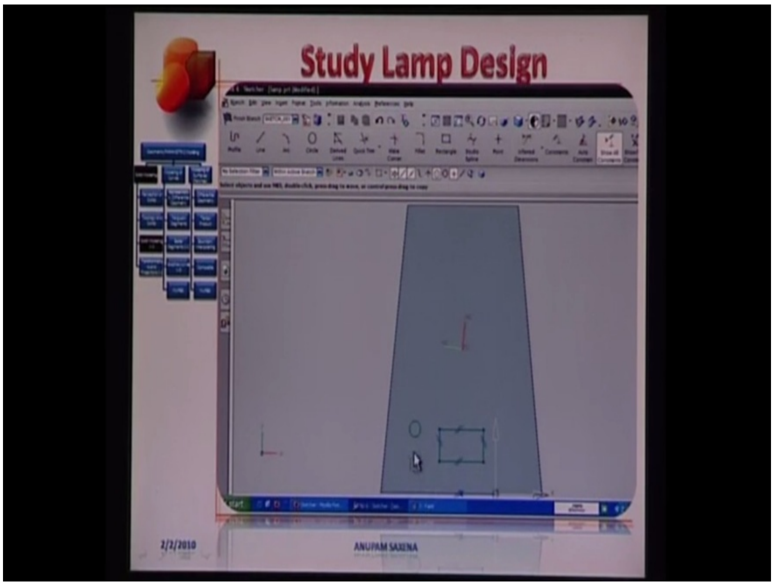
We have a trofoside, now we are going to be striving this cross section in to the third dimension; that is the dimension towards here.

(Refer Slide Time: 19:07)



To get a solid feature, the height of this proposal base is about ten units. Next we will try to cover out certain features on this space, so these circle of features might correspond to say (( )) tends is rectangular tactful, would correspond to say holding raisers a fence. We are just trying to play with the design here. We are specify the relations between these two edges, we ensure that these two edges remain (( )) we change the dimensions of this rectangular.

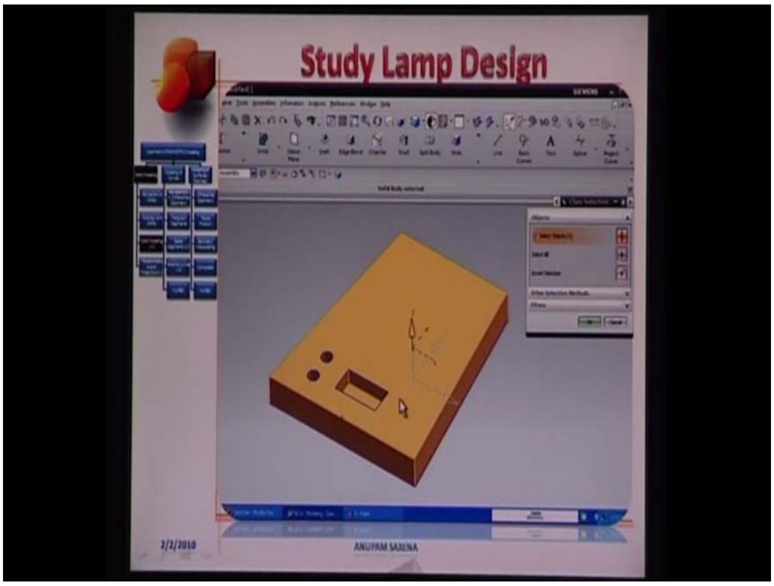
(Refer Slide Time: 19:22)



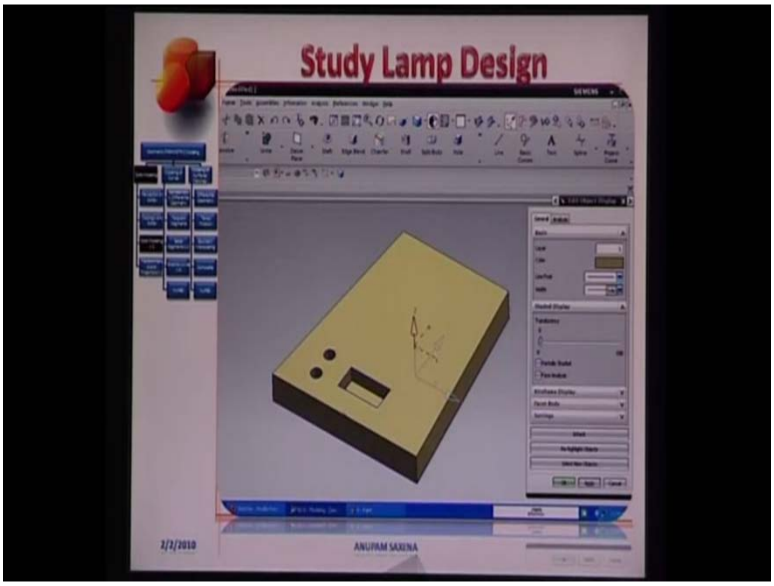
(Refer Slide Time: 20:06)

Once peace patch out these cross sections on the base, we try to intrude is cross sections into the base. To get circle of dot holes and a rectangular (( )), the death of these part hole is about five units.

(Refer Slide Time: 20:28)

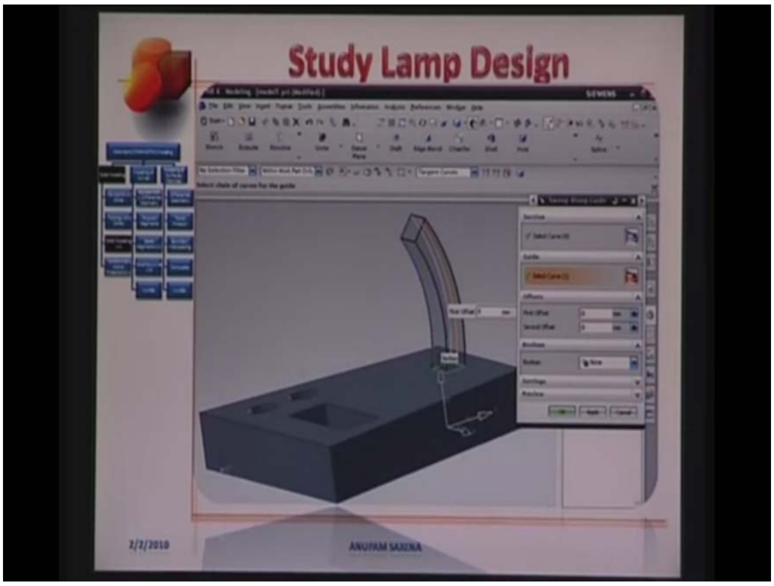


(Refer Slide Time: 20:33)

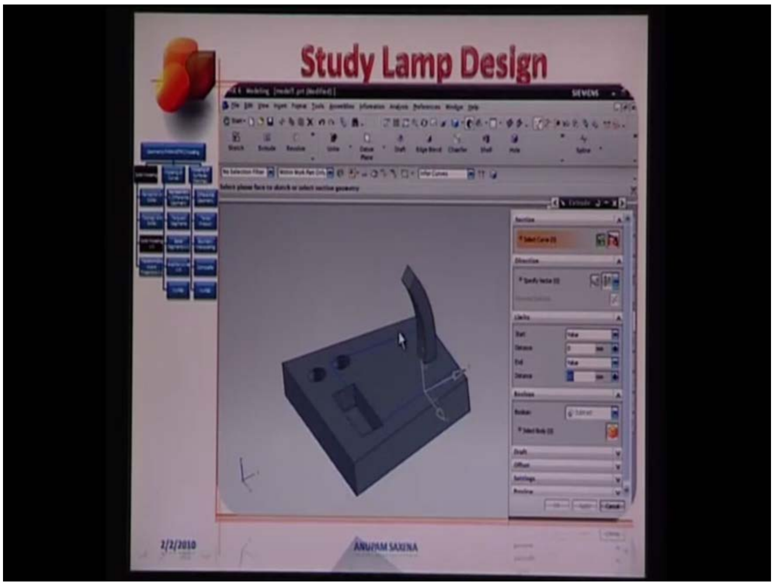


This is how we see different part holes. We can actually change the colors of different parts if we want. Next we will try to design a prismatic connector that connects the base to the powerful. We had this rectangular feature on the top phase of the base and we have this circle art. The idea is to get the connected again a prismatic shave, through this non-leaner speed; this rectangular cross section is going to be swept the along circle of art.

(Refer Slide Time: 20:42)

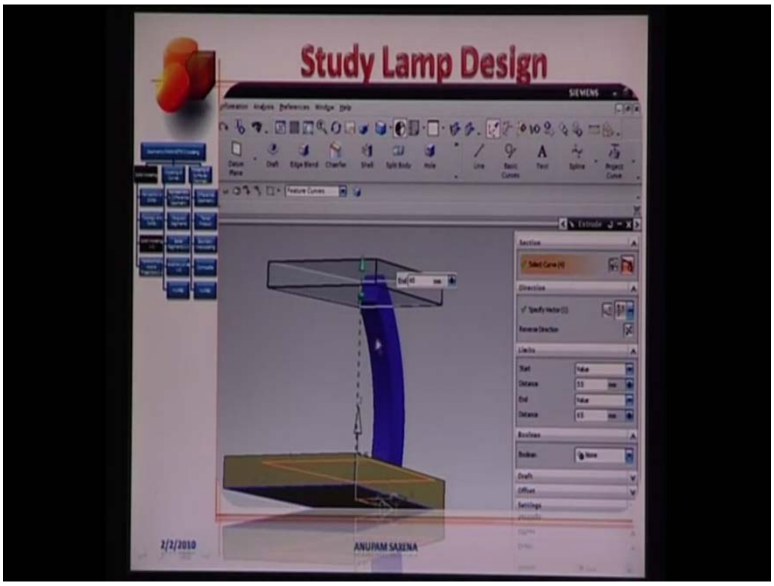


(Refer Slide Time: 21:19)

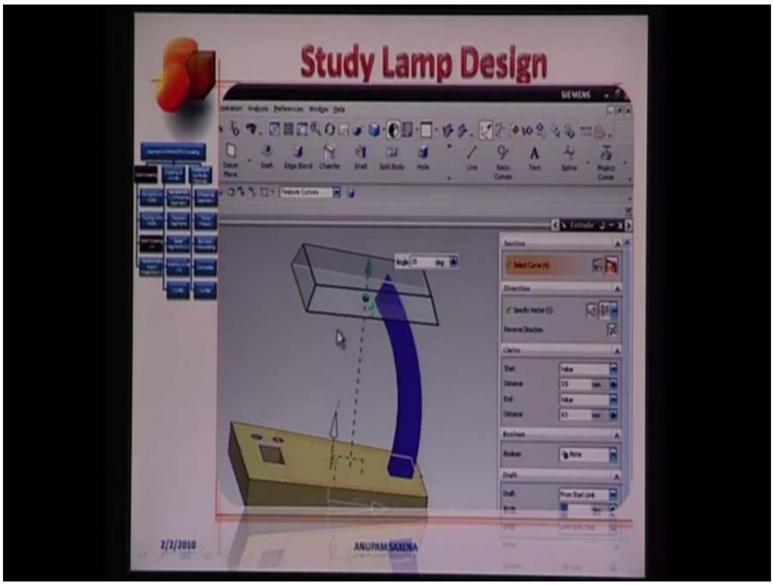


We do that and we get the connector of this shape, next we try to design a bulb holder. We have a tropasoral cross section here, which we have excluded along this vertical direction by some amount. This is another view the lamp and we are going to be shacking up this bulb holder like a different. We introduce a taper, along this direction at the base.

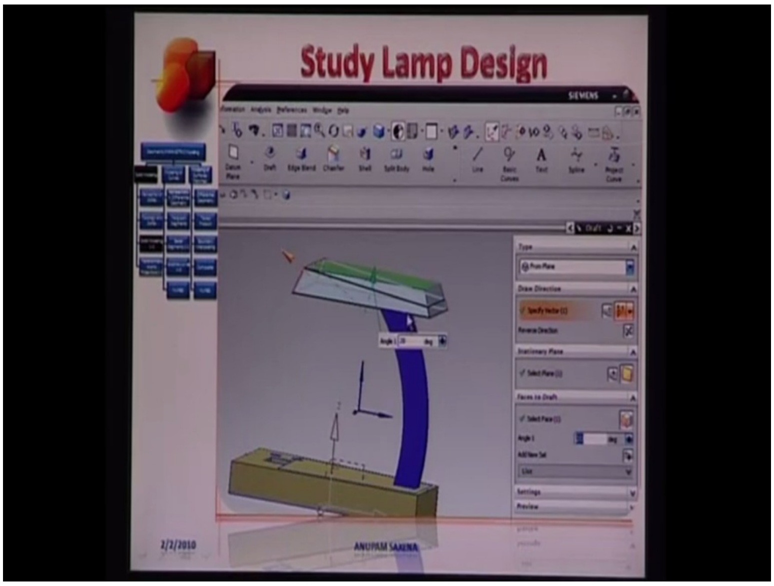
(Refer Slide Time: 21:30)



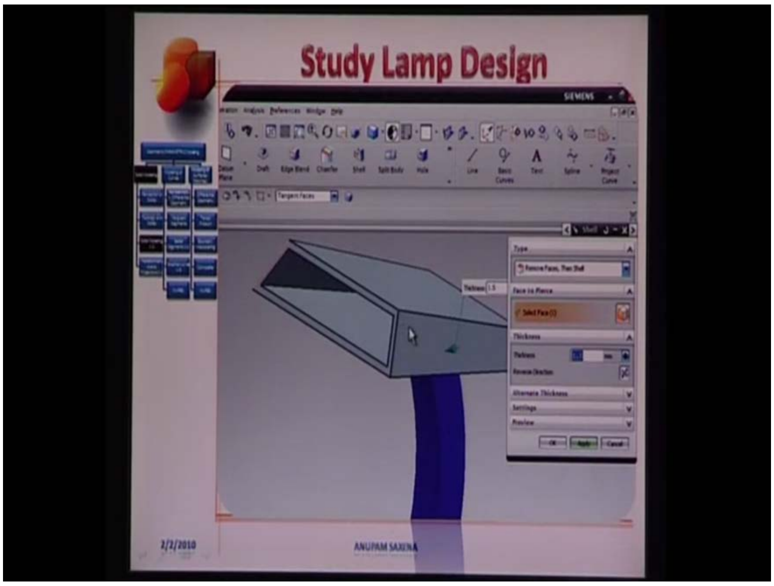
(Refer Slide Time: 21:50)



(Refer Slide Time: 22:04)

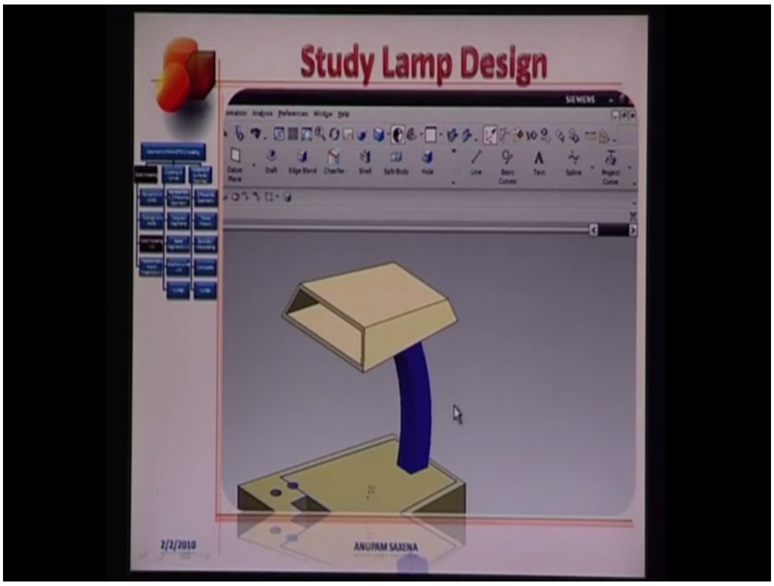


(Refer Slide Time: 22:17)

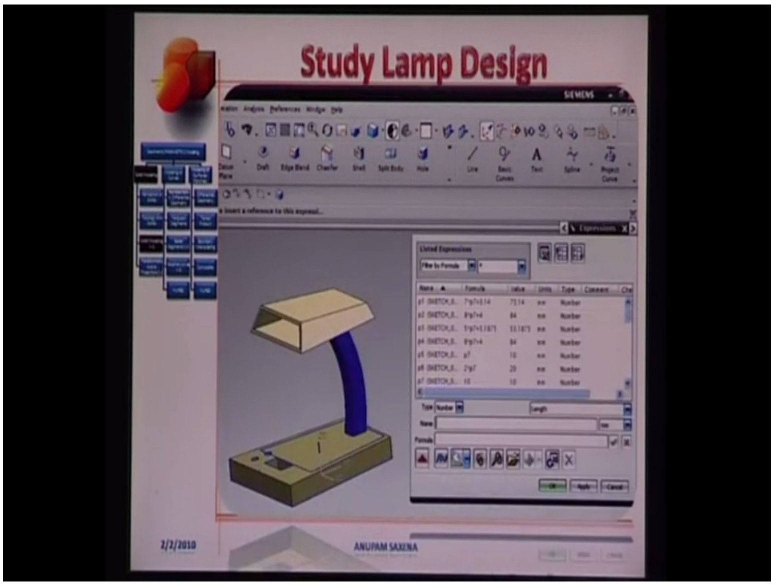


After we do that, we will introduce a shell like structure to represent the bulb holder; the thickness is about 1.5 units. We have a basic lamp design ready. These are different relations that Salish at used to design this lamp. By a metric solving modeling allows us to choose different features in the design and modified them locally. So that, we do not have to undergo the entire planes taking design procedure.

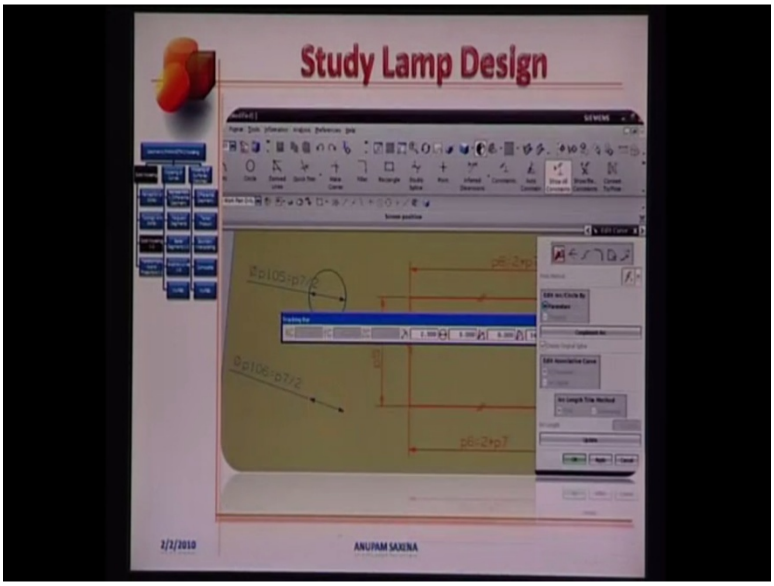
(Refer Slide Time: 22:37)



(Refer Slide Time: 22:43)



(Refer Slide Time: 23:16)



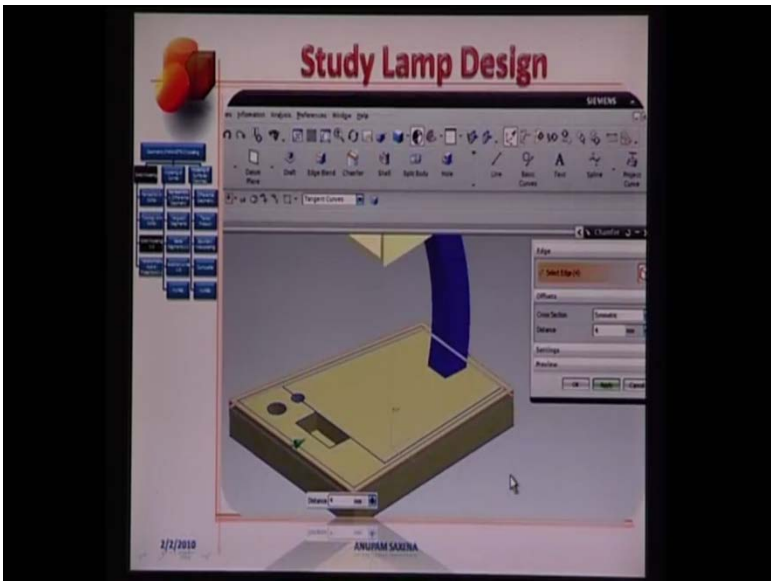
Let us try to understand a metric solving modeling, show the following slides. First we choose this circular feature and we play with the sights of the spot hole.

(Refer Slide Time: 23:30)

We increase the size when we do that, we get this spot hole of slightly larger in size. This window here represents the history tree, when creating this design in the text form; not in the graphics form. One can choose any feature from this history tree, for double click the corresponding feature in the graphics window here; to implement whatever changes he or she has in mind.

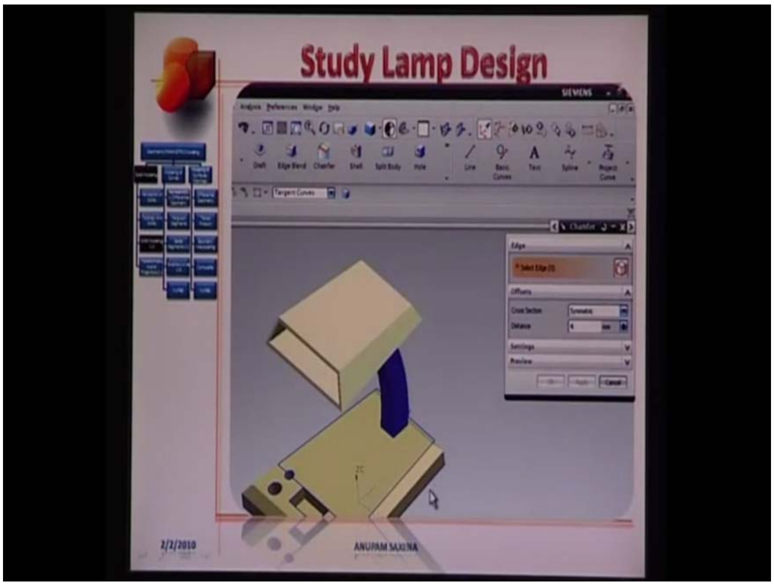
(Refer Slide Time: 23:37)

(Refer Slide Time: 24:09)

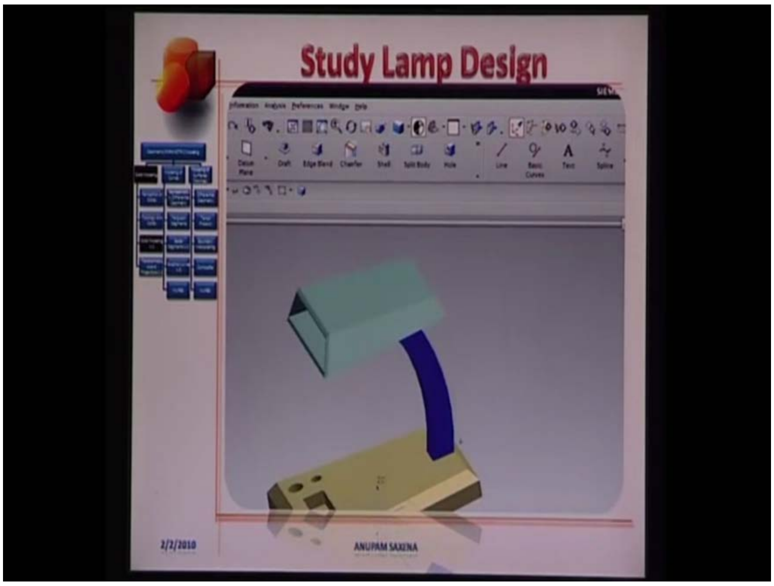


Now, we are trying to introduce the shamper on the base. After we do that our design appears like this. In change colors, we want.

(Refer Slide Time: 24:20)

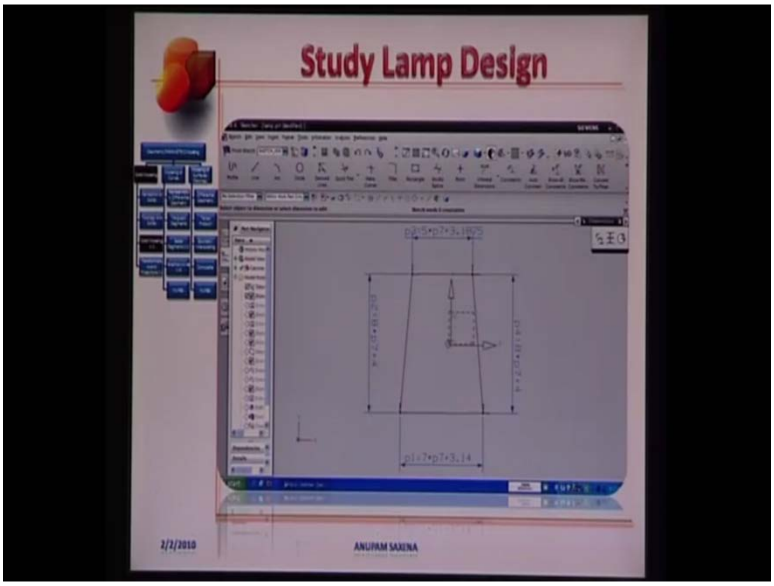


(Refer Slide Time: 24:30)

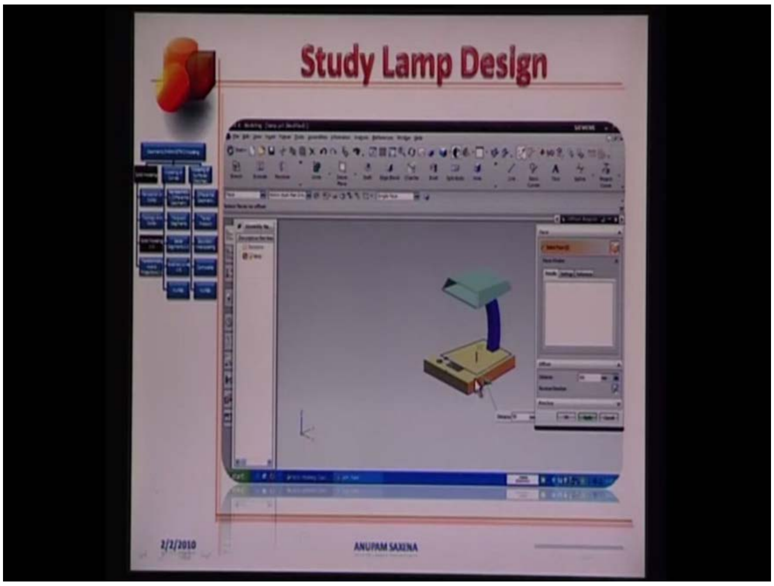


Let us now try to enlarge the base of the lamp and absorb how the design looks. We had different dimensions here, which are relating each other. If we change any one or two of these dimensions are base, we get a large this is one key feature of parametric solid model. We had to have relations between different features and face.

(Refer Slide Time: 24:36)

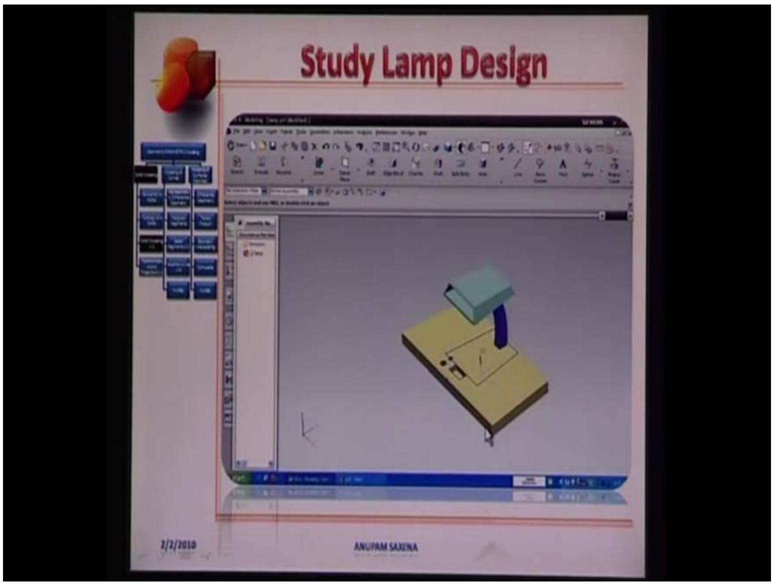


(Refer Slide Time: 25:04)

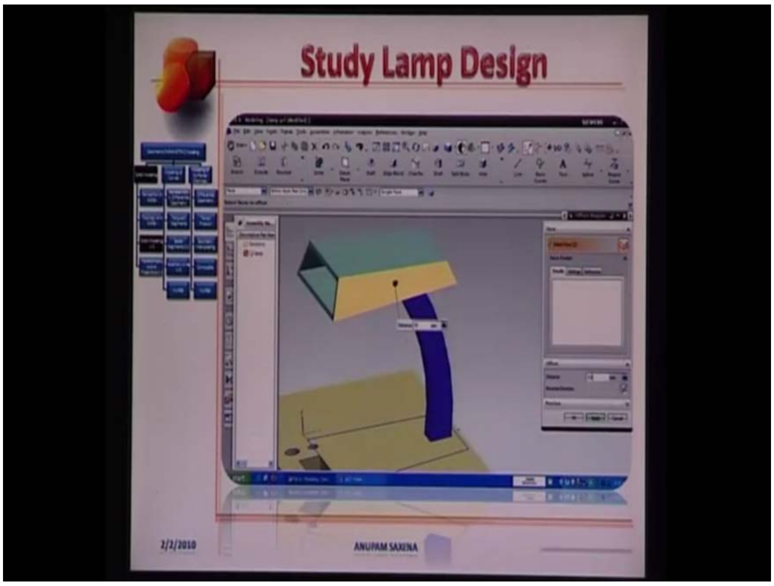


We can choose this face as well as the face beheaded and we can increase the distance between the faces. When we do that, this is how our design looks. Next let us try to play with the size of the bulb holder; once again we choose the face nearest was in this part. We choose the face right opposite and then we enlarge the distance between two phases.

(Refer Slide Time: 25:15)

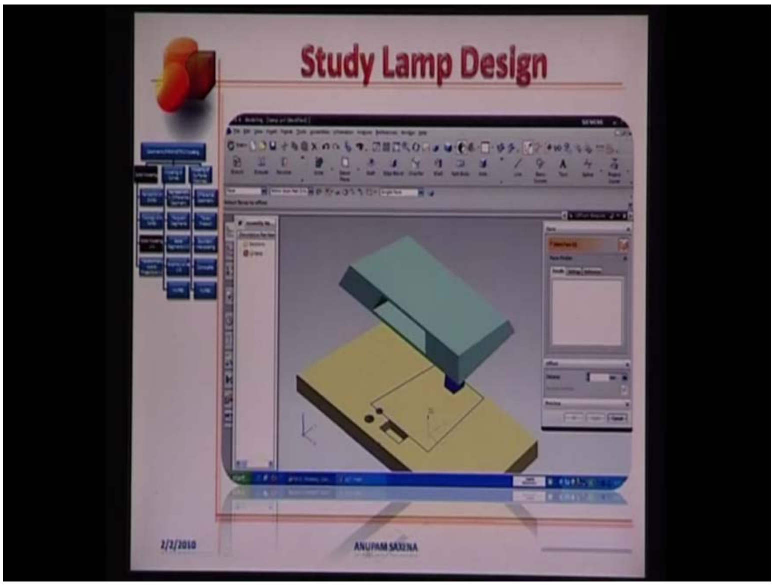


(Refer Slide Time: 25:26)

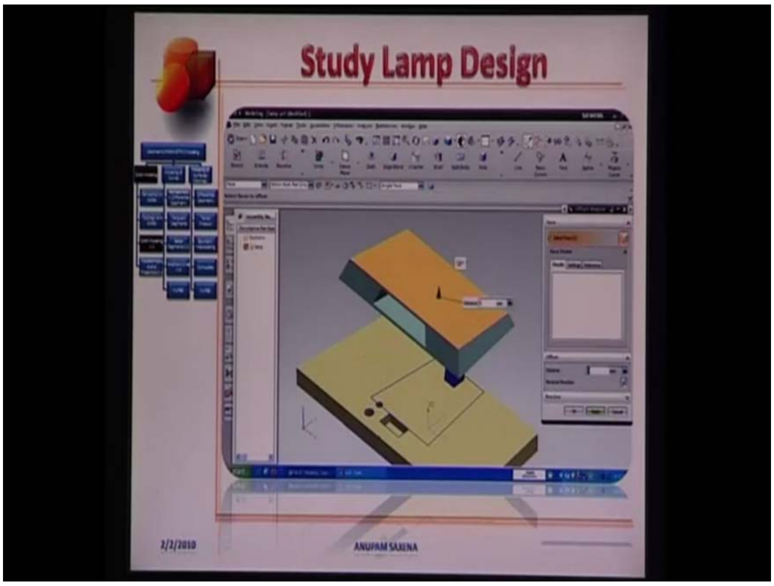


Let us try to play with the bulb holder a little more, the increase the vertical dimension. Again in a similar passion; choose the top face, choose the bottom face and increase the distance between.

(Refer Slide Time: 25:44)

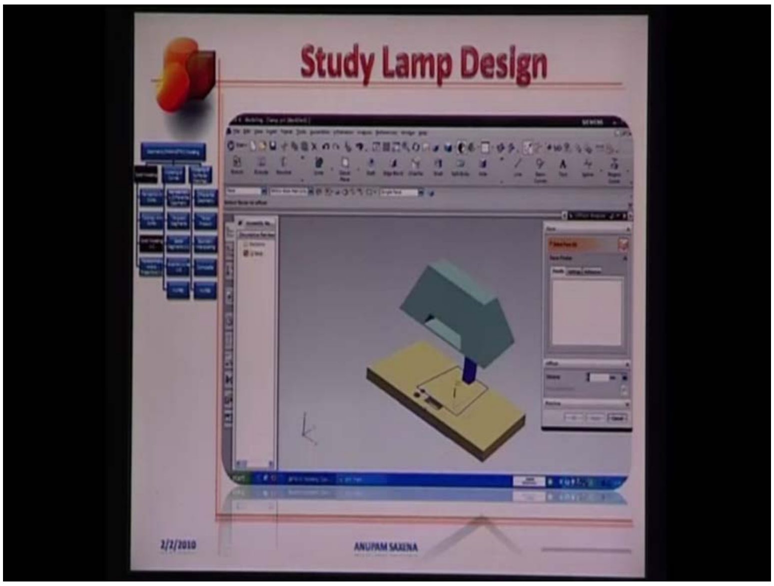


(Refer Slide Time: 25:56)

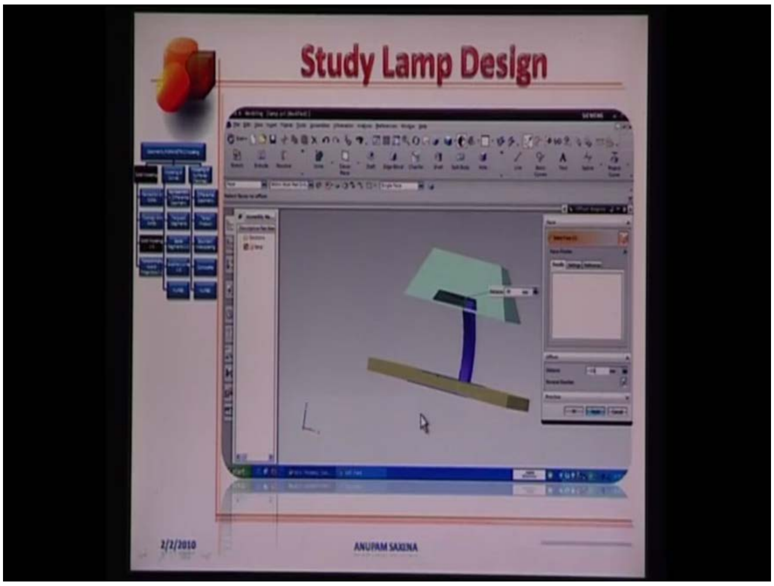


We have left this hole right there, we can bring this hole a little upward along the vertical direction. Very similar procedure, we do that. We have now got the hole at the right place.

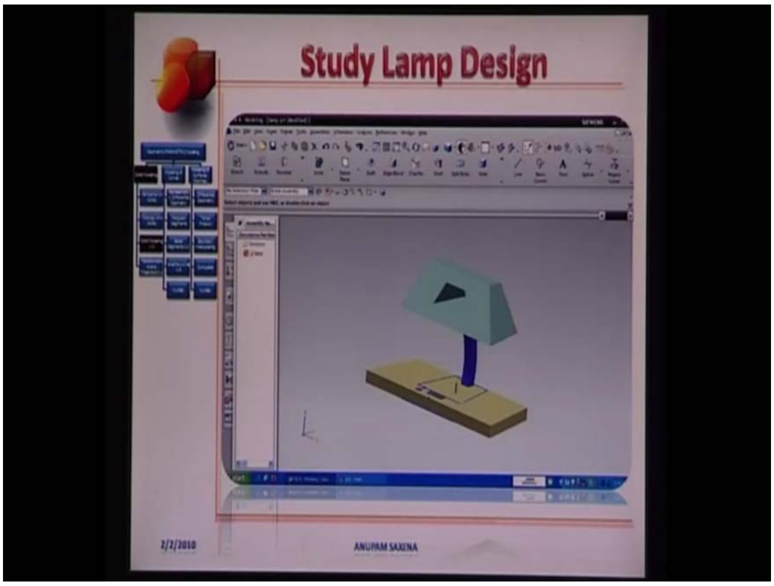
(Refer Slide Time: 26:03)



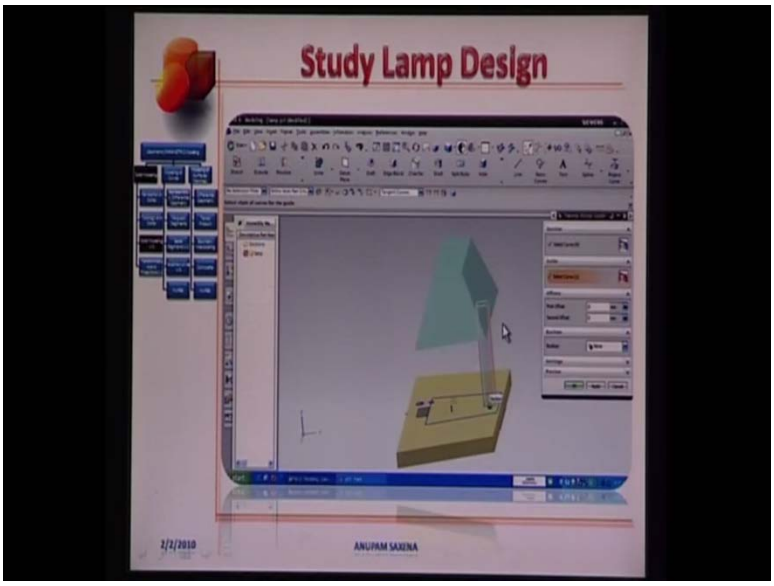
(Refer Slide Time: 26:17)



(Refer Slide Time: 26:21)

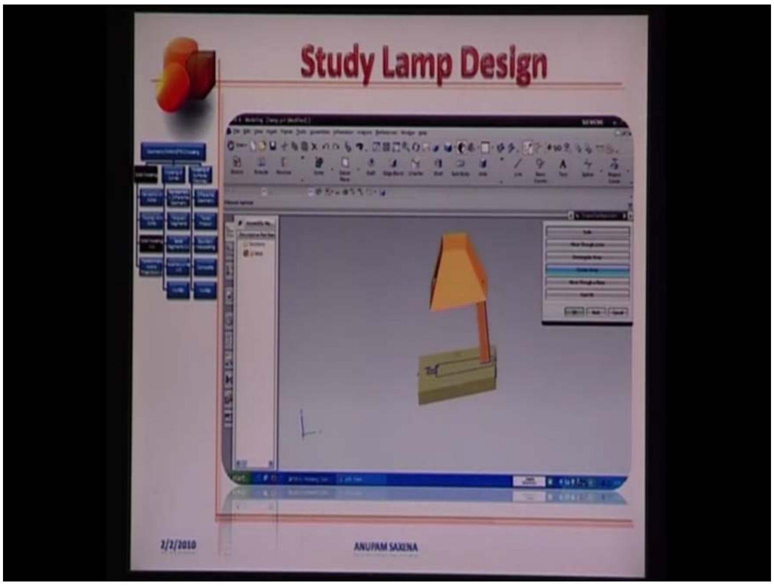


(Refer Slide Time: 26:34)

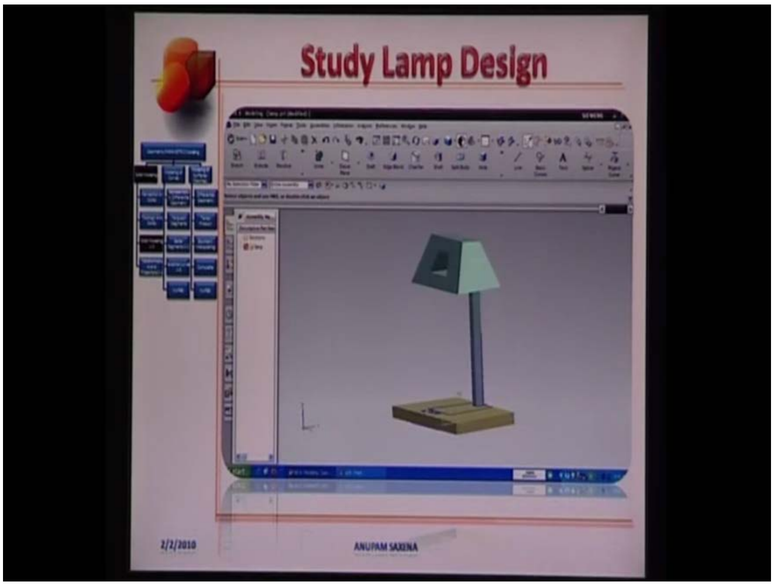


Let us now play with the neck of the lamp. So in the previous example, what we had was we had rectangular cross section, swept along a circular r. Here will have the same cross section but, swept along a line, a straight line. When we do that, we get a straighter neck.

(Refer Slide Time: 27:05)

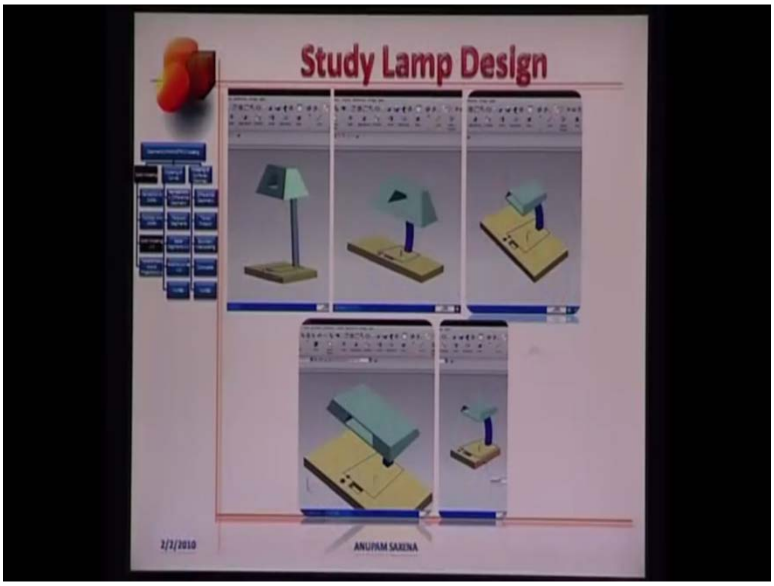


(Refer Slide Time: 27:12)



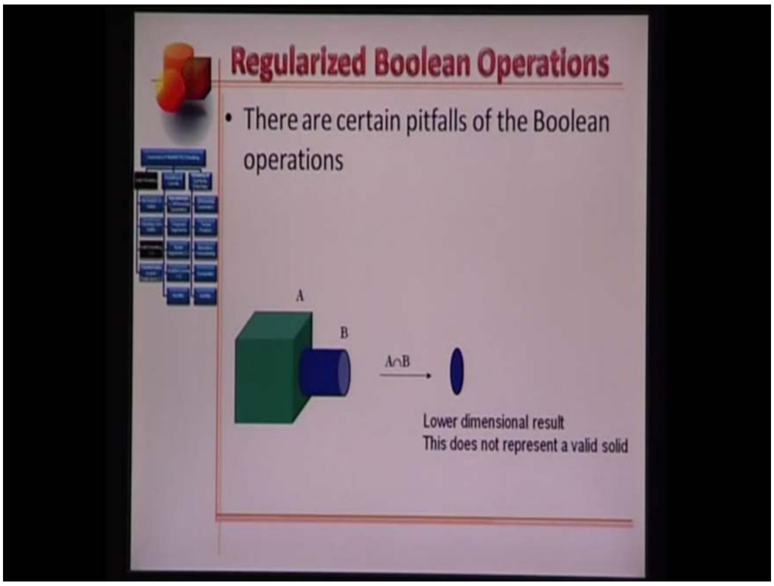
We can elongate the neck, if we want to get a new design. In a sense we are parametrically trying to change different geometric features of this design, without actually having to redesign the entire lamp from scratch and it is this feature, that is very nicely employed in complex solve modeling.

(Refer Slide Time: 27:46)



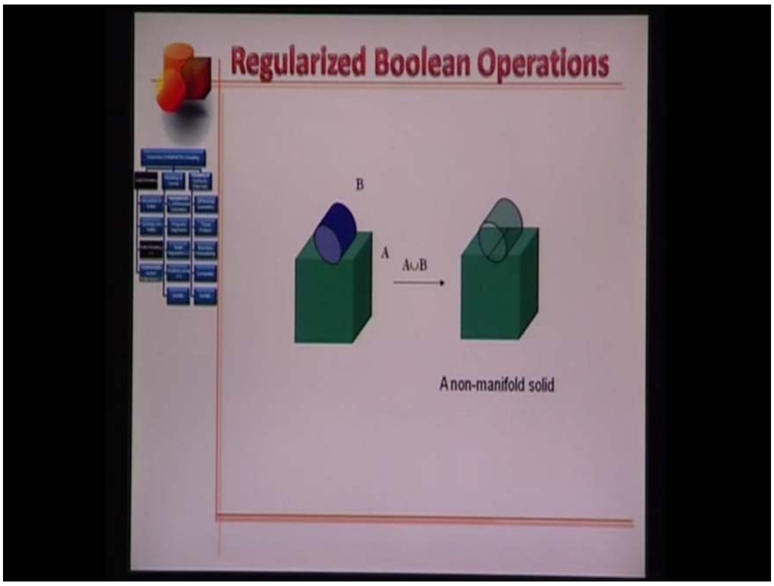
We can have a group of designs, two parametric some model. Once again if you look at these five designs, the basic topology or connectivity of these designs remains identical. It is only the shapes and sizes are different features, which are changing as we intend. We can present these designs to the customers and let them choose which of them, they would like.

(Refer Slide Time: 28:31)



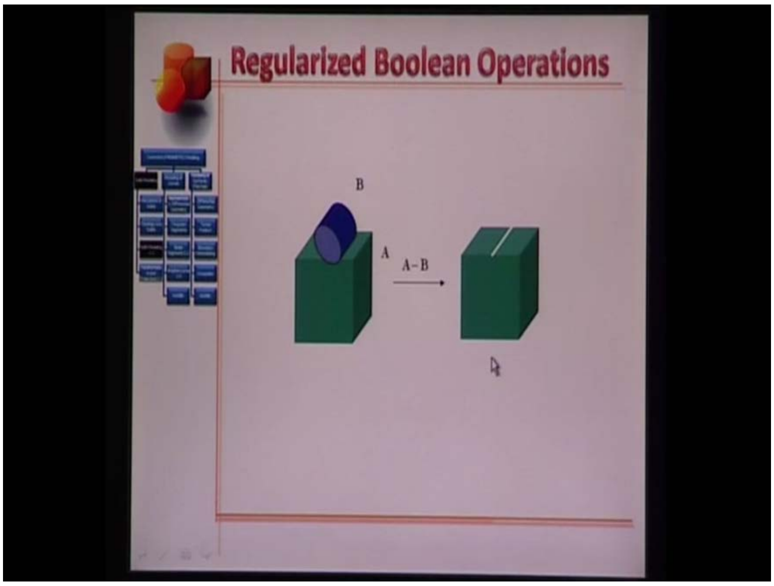
This is the third and the final part of this lecture, a little note on regularized Boolean operations. This is the slide that I had covered in the previous lecture. I emphasized that, they were certain fit falls or drawbacks of the raw Boolean operations. We take a block or a cube, we take a cylinder; we align them or transform them in such a way, that one of the faces of the cylinder sits on one of the vertical faces of the cube. The two solids are A and B and if you try to compute the intersection, we get a test. This is a lower dimensional feature to dimensional nature and it does not represent a valid solid and it was best example that I have used to introduce regularized Boolean operations.

(Refer Slide Time: 29:25)



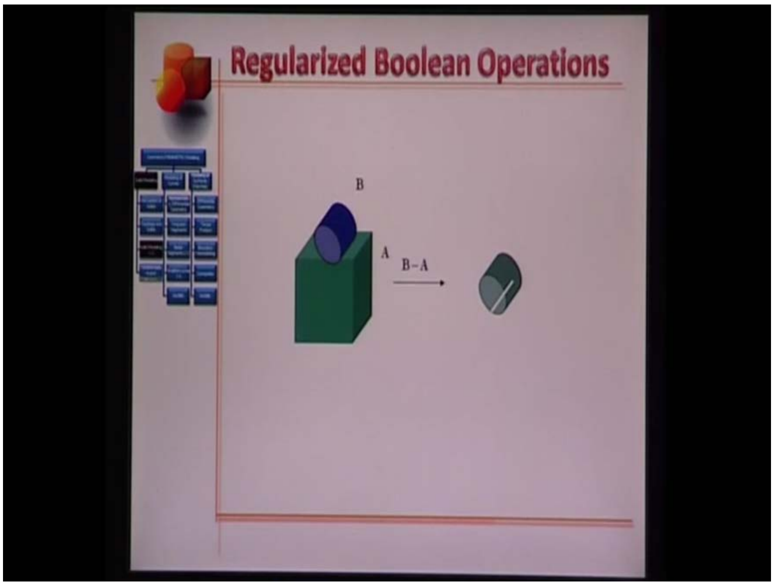
Let us look at a few more examples, we have a cube and now we have a cylindrical material setting over the top face of the cube. The cube is solid A, the cylinder solid B; if we try to compute the union between two solids, we get this solid with common edge between the two (( )). This is a non-manifold solid because, if you look at this common edge it is shearing whole faces; this is face 1, this is face 2, this is the third face and this is fourth face. If we call the definition of manifold solids, there we have used an open wall. If we place an open wall here anywhere on this edge, will see that the intersection between the open wall and the solid will result into inter-primed desks, which cannot be defund into a single desk for which reason, it is non- manifold solid.

(Refer Slide Time: 31:06)



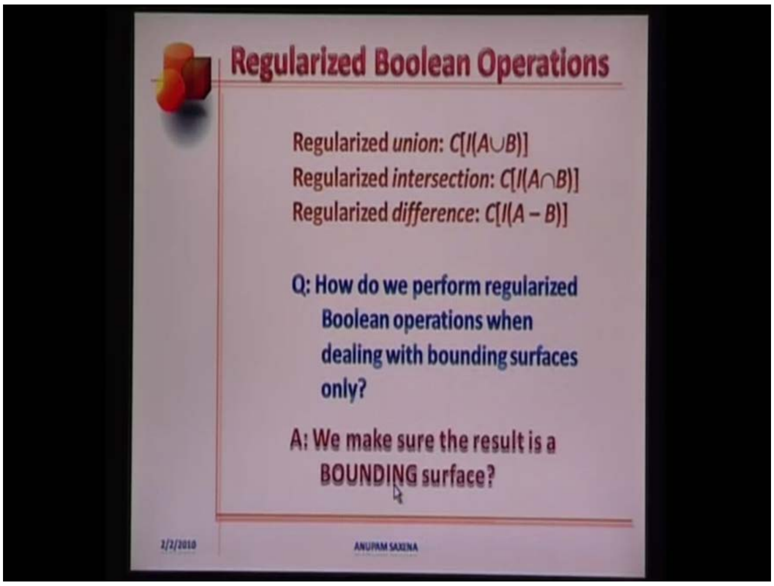
Third example, of a block and of the same cylinder, solid A solid B, we now compute A minus B. We will get a block with a slight singularity on the top face. The common edge between the cylindrical (( )) and the block will not be present on the block. If you see this, they would be a very hen slit, (( )) open up the top face of the block.

(Refer Slide Time: 31:56)



Same example, now a block and a cylinder sitting over it; we try to compute for this B minus A. Once again will possibly have a cylindrical surface with a little slit opened up right there. These are certain singularities or normalizes, which are associated with raw Boolean operations.

(Refer Slide Time: 32:39)

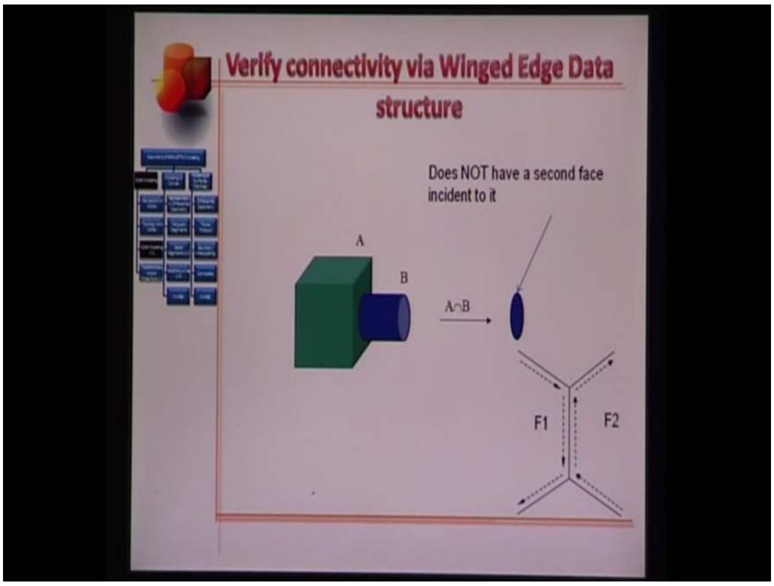


To eliminate these normalizes we had proposed regularization of Boolean operations port, so a regularized union will be like this. It compute the raw union, it compute the interior of the result and then you perform the closure operation. In the closure operation we actually compute the union of the interior as well as the boundary of a solid. Likewise regularized intersection we compute the raw intersection, we compute the interior of the result and again you perform the closure operation taking the union of the interior and the boundary.

The regularized difference is performed in a similar manner, we compute the raw difference take the interior of that and then perform closer. All these operations make lot of sense, if we think about solids as a set of points occupied of finite volume in the ((fluidounce)) space but, I have emphasize, that solids a seen by virtually all solid models by their corresponding bounding surfaces as per the extended Jordan's ((afford)). This is the question that I asked myself last time, how do we perform regularized Boolean operations when dealing with only bounding surfaces? When we do not have a (( )) because, only then these operations will make sense with only bonding surfaces possibly, these operations will be difficult to be perform. I can only conjecture at this time but, this is what my conjecture is. All we need to do as we need to sure at the result of raw Boolean operations is a bounding surface by itself, in other words the result of the Boolean operations is perfectly valid solid.

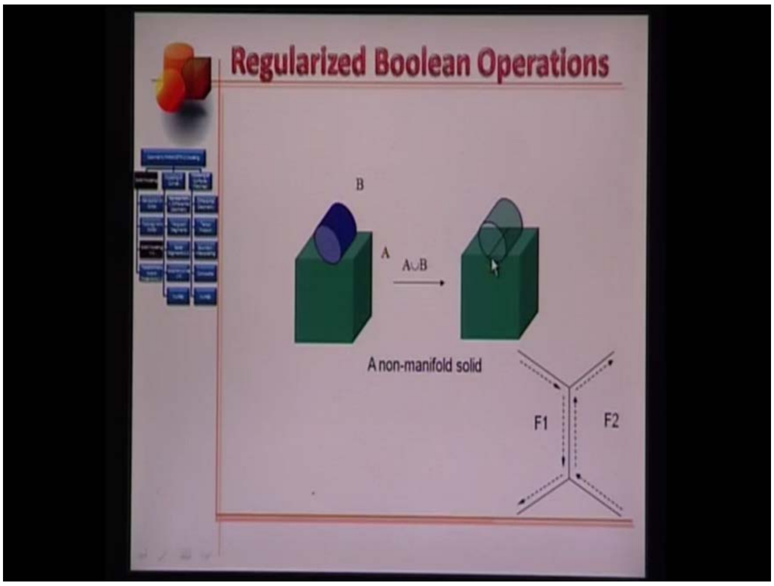
How do we do that? Well we have a tool we call ((went)) edge data structure, all we need to do is ensure that the ((went)) edge data structure this properly structured. Let us revisit is for example, and see how the data structure can have a normalizes which can be detected and then we can conclude that the results may or may not be valid solids.

(Refer Slide Time: 36:08)



Let us take a look at the intersection example first, block A cylinder B intersection give us a desk. Now f you recall for any valid solid polyhedral or non-polyhedral with difference of this patches constituting a solid and different surface patches stitched at the common boundaries. A valid solid will be 1 of which each boundary edge will have not more and not less than two faces incidental to it, this example here if this a boundary curve will have exactly two faces incidental. Now if we look at the result here it is an open desk this outer boundary has only one face incidental. It does not have a second face associated with it and this thing can be detected in the winged edge is structure and correspondingly the result can be eliminate or can be categorized as non-valid solid.

(Refer Slide Time: 37:46)

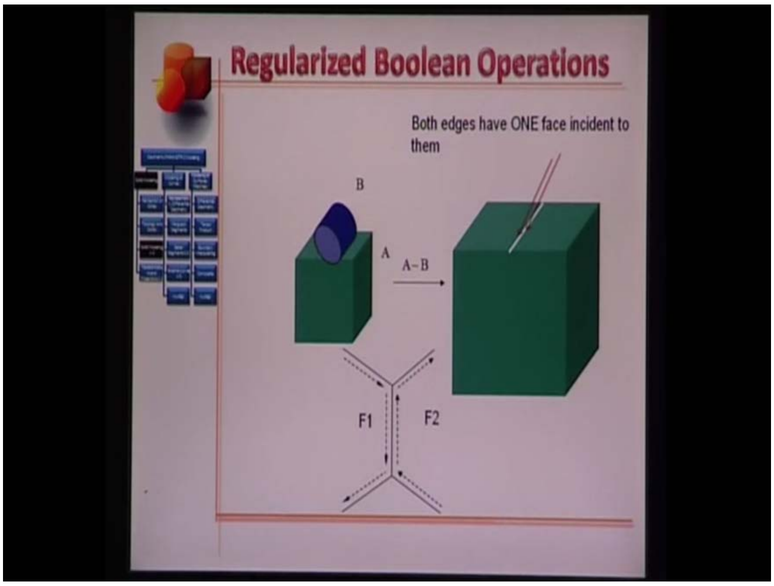


Let us take a look as the union example here. Block A, cylinder B if we compute the union will have a common edge between cylinder and the block. We know that it is a non-manifold solid, if we look at this common edge this will have as a mentioned hole four faces and not two faces associated with it. So corresponding to this edge, the winged edge data structured will complained and we can detect that complained and we can say that this solid it is not a valid solid. Just in case some of you are having visualization difficulty, let me try to sketch features around that common edge on the board.

(Refer Slide Time: 38:47)

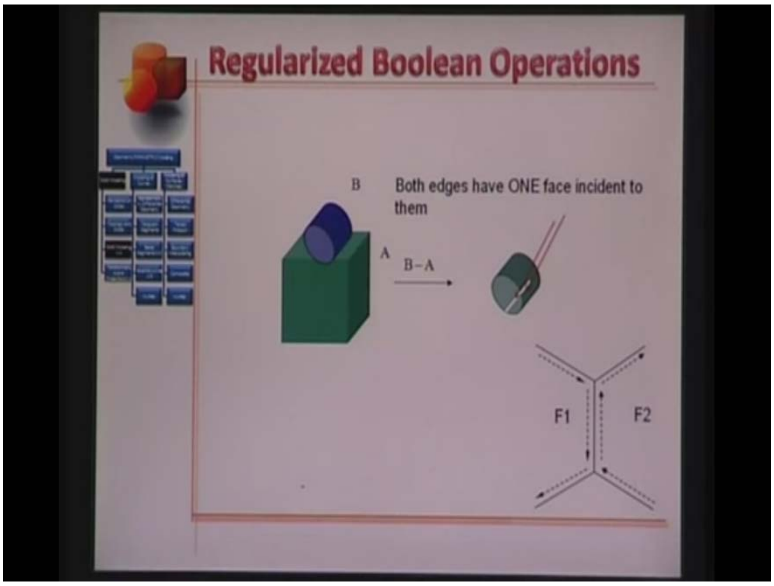
This is the common edge, this is the first face on the block, this the second face on the block, this would be third face on the cylinder and this would be the fourth face from this cylinder four faces.

(Refer Slide Time: 39:35)



Let us take a look at third example again, the difference between the block A minus the cylinder B look at this (( )) here which opens up the top face of the block, this corresponds to the common line between the cylinder and the block. The two kind of similarity 1) we can think of this slit as composed of two edges which are very close each other. And second of course, this is opening up the top face because of which block will not be bounded (( )). So that two edges and if you look at any one of the edges that edge will have only one face incidental. If you look at edge on the left will have only the left face which is incidental and likewise for the right edge. So the winged edge get a structured can detect both the similarities, if the two edges are very close to each other will complain accordingly and if there are less than or more than two faces incidental to single edge will complained as well.

(Refer Slide Time: 41:25)



Likewise the fourth example, if we subtract the block from the cylinder B minus A will have this let like here at the lower region on this length. Once again very similar problems, two edges very close to each other and each of the edges will have only one face incidental. Both the similarities can be detected then computing the winged edge get data structure.