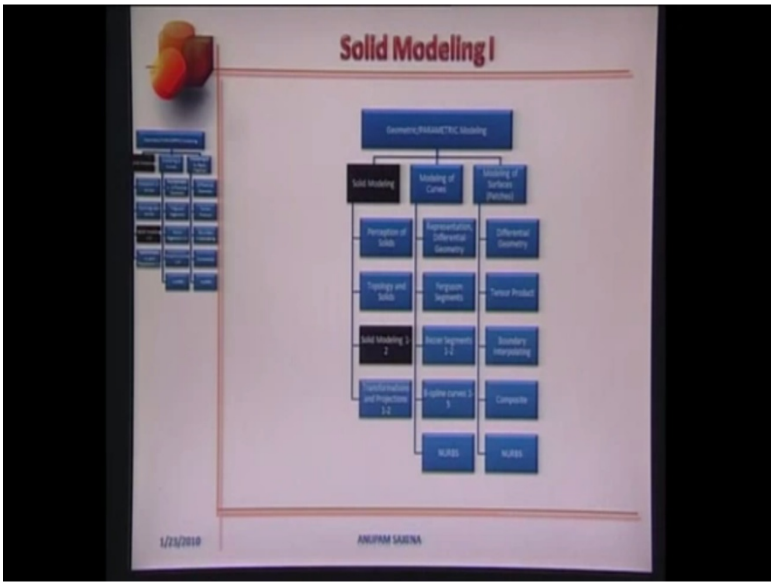
Computer Aided Engineering Design

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

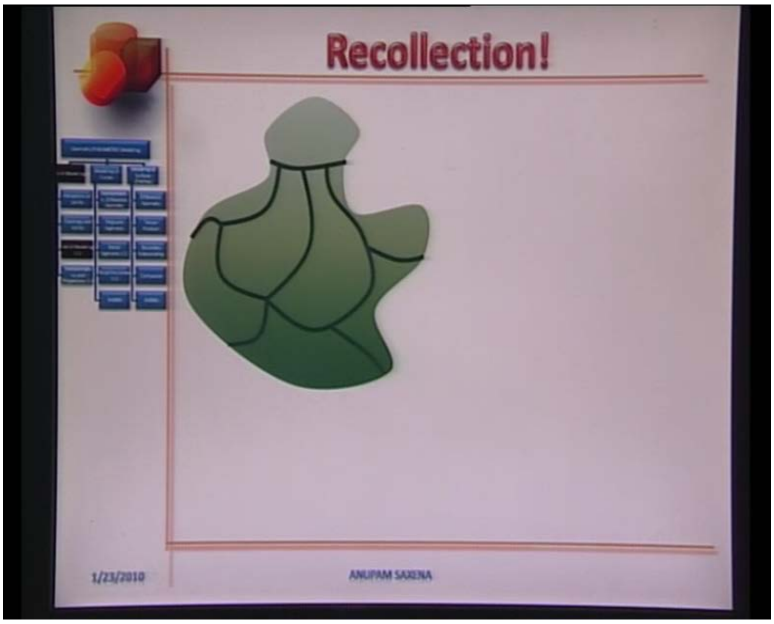
Lecture - 6

(Refer Slide Time: 00:28)



Hello, this is lecture number 6 of CAD NP-TEL video series lectures. This is the first lecture in solid modeling.

(Refer Slide Time: 00:33)



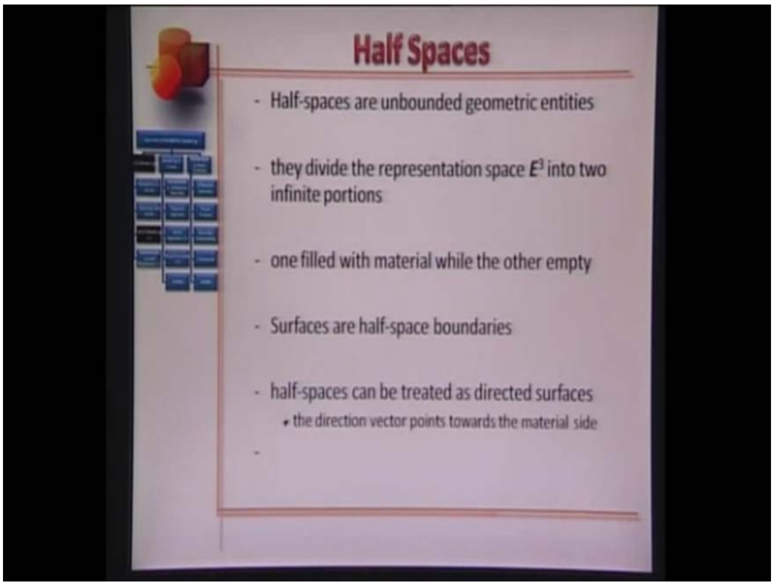
A little recollection before we start the extended Jordon's curves theorem a simple closed and orientable surface bounds a solid. You have seen in lecture 5, what orientable surfaces meant was. In a sense a simple closed orientable surface can be discretized into surface patches. In other words the bounding surface can be divided into different surface patches. Each patch has curved boundaries and each curve has end points. While representing any solid in a solid modular both geometric and connectivity information should be stored and retrieved at any time.

So what I will do is I try to peel of the banana very carefully, I carefully cut the top part of the banana and try to peel it off, I will be very careful not to damage the top part. So, I have peeled of the outer cover into four pieces, this is the crust I am going to be taking off the substances. I am a little hungry so I am going to be eating something. So, what I will do now is I will try to so this is what I call the surface patches, they are like surface patches, four patches here. So, what I will try to do now, is I will try to build them up using a cello tape here. I will try to stitch the common boundaries between the two surfaces like this. I am not sure how good I am at it; but I will try my level best. I am trying to give a very rough idea, looks like I have succeeded.

Now the common edge between the other two patches. I am going to be reinforcing the stitch. Now things are going to be a little difficult; but I will try my level best. I am not trying to go for a perfect stitch here; but something that is just workable for me to demonstrate the idea and now for the final common boundary and now for the top portion.

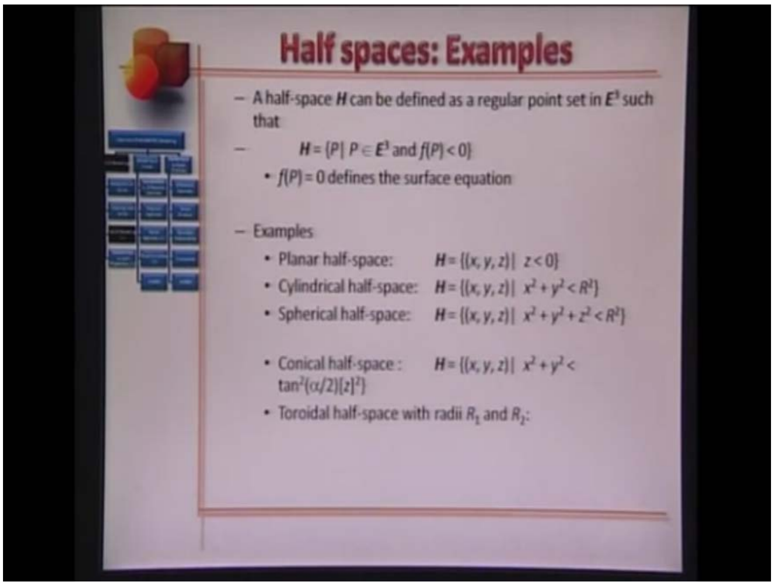
So remember what I have done, I have taken away the substance from the banana; I am still eating, I am still working on it. And I have tried my level best to stitch the common boundaries together. It is pretty much like four or five different surface patches stitched as a common boundary. This is like the bounding surface of a banana which represents the solid. So this lecture is about representation of solids, in any solid model.

(Refer Slide Time: 06:13)



Before we start, let us first look at the concept of half spaces. Half-spaces are unbounded geometric entities, this is keyword that works against the concept, but nevertheless we continue with understanding, what half-spaces meant was? Half-spaces divide the representation space or the Euclidean space E three into two infinite portions. One portion is filled with material while the other is empty, surfaces are all half-space boundaries in a way, half-spaces can be treated as directed surfaces: the direction vector would point towards the material side; and that is the reason why.

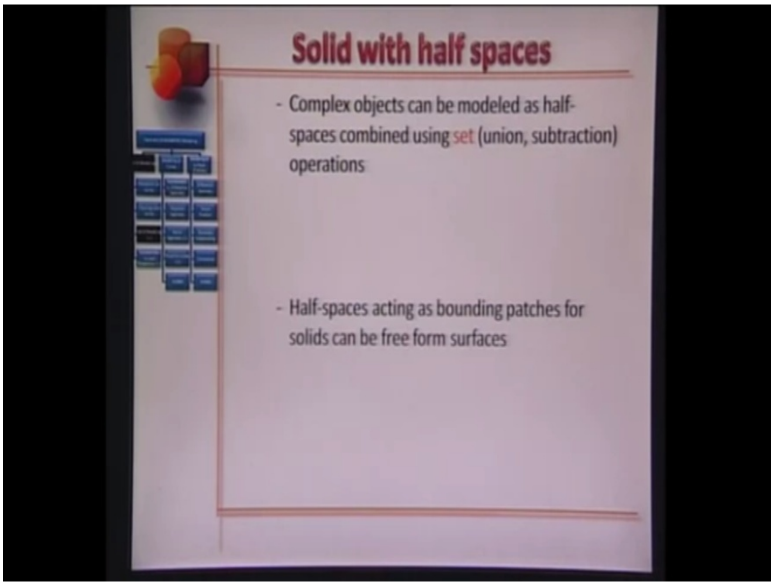
(Refer Slide Time: 07:17)



Let us consider a few examples; a half-space H can be defined as a regular point set in the Euclidean space. Such that the points at P belongs to the Euclidean space and f of P is smaller than 0; f of P equals 0 defines the surface equation. For example, a planar half-space; H equals a set x y z, so that all z values are smaller than 0. This example corresponds to the explain, it is a set of all points in the four lower quadrants of the Euclidean space for which all the z values or the z values are smaller than 0.

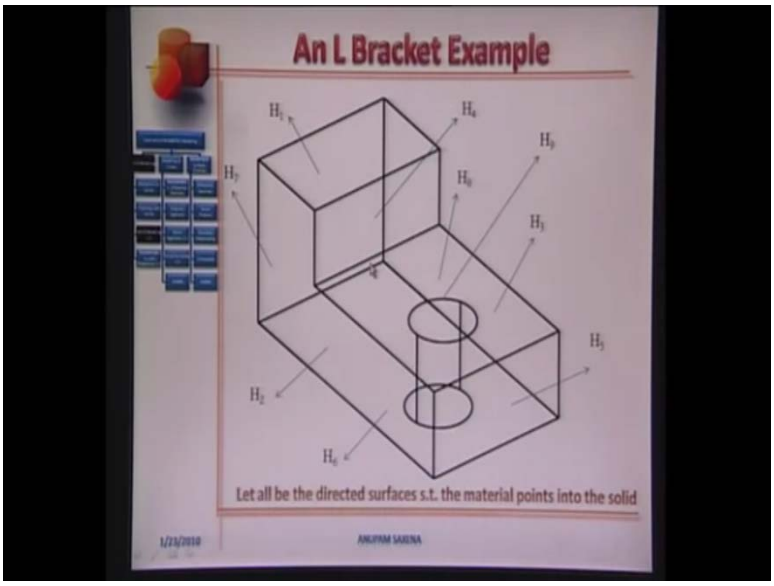
Example two a cylindrical half-space, a set of all points which lie within the cylinder defined by the equation x square plus y square is smaller than radius of the cylinder square. A spherical half-space, it is a point set of all points that lie within the spherical surface; a conical half-space again a set of all points that lie within the cone; a toroidal half-space, a radius rather the radii of the torus are R1 and R2. It is a set of all points that lie within the toroidal surface.

(Refer Slide Time: 09:36)



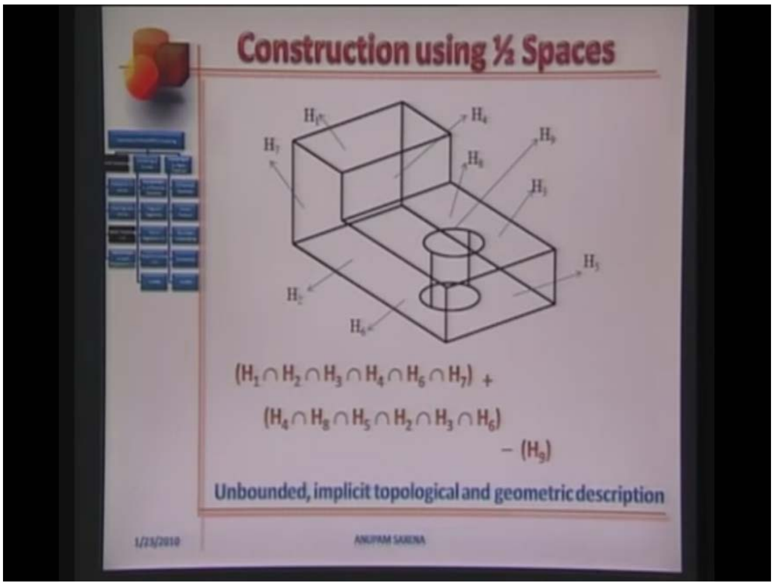
How can solids be seen via the concept of half-spaces. Complex objects can be modeled as half-spaces combined using set operations. Set operations can be like union, subtraction, addition or intersection. Half-spaces that act as bounding patches for solids can be free from surfaces; they need not all be analytical equations.

(Refer Slide Time: 10:07)



Let us take an L bracket example with a cylindrical hole, can we identify different half-spaces. Let see the top face here is identify as H 1, the side face as H 2, the other side face as H 3, the front plane as H 4, this plane here as H 5, the base of the bracket as H 6, the back plane here as H 7 and the top plane here as H 8, the cylindrical surface is identified as a cylindrical of plane H 9. How can we think of constructing this L bracket using the concept of half planes; but first let us imagine that all the directed surfaces are such that the material points into the solid.

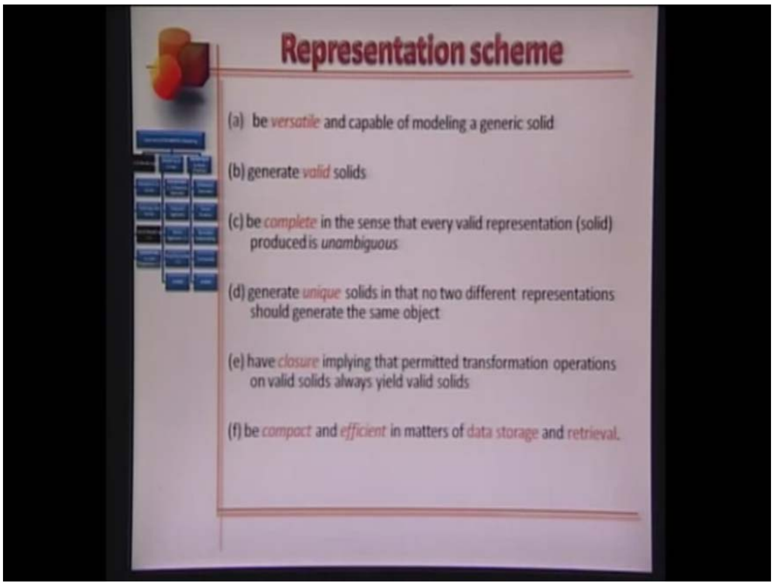
(Refer Slide Time: 12.01)



Let us try to construct this L bracket using the concept of half-spaces. We can think of constructing this L bracket using a union of two blocks; 1 block would be this one, the second block would be this. Let us try to get the first block; so different set operations operating on half-spaces. The first block can be thought of being composed of the intersection between the half-spaces; H1, H2, the back side face H3, this front plane H4, the base plane H6 and the back plane H7.

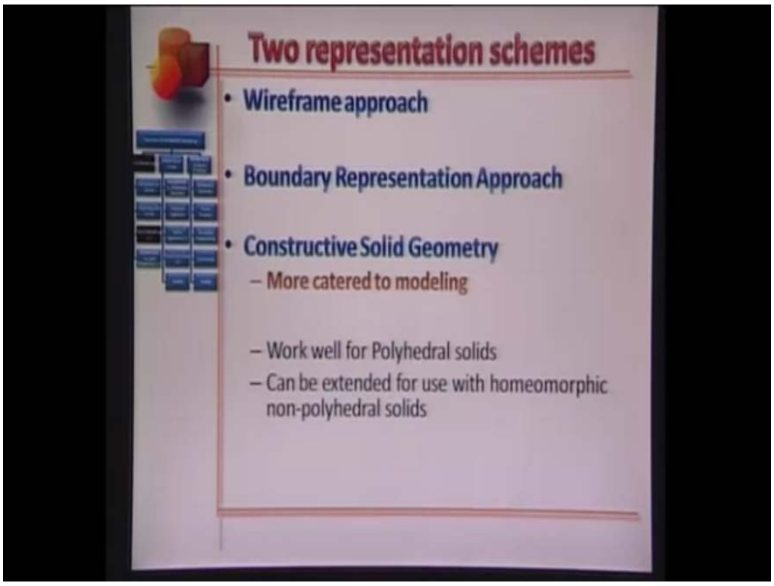
Likewise the second block can be thought of being composed of the intersection between H4 this plane here, H8 which is the top plane here, H5 which is the front plane here, H2 the front side plane, H3 the back side plane and H6 the base plane; H9 of course, would be the cylindrical surface. So the L bracket is a union of these two blocks or in addition of these two blocks and the cylindrical half plane subtracted from the result. A few disadvantages of this concept are that the half-spaces can be unbounded and these set operations would only implicitly store the topological and geometric description of a solid. Data storage would not be explicit there are other alternative representation techniques which are used.

(Refer Slide Time: 14:54)

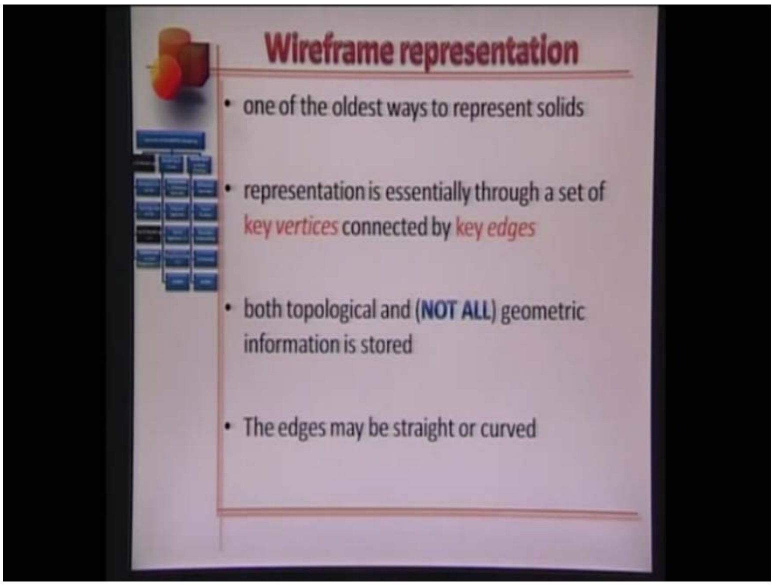


We will look at them. But first what is expected from a representation scheme? For one, it should be versatile and should be capable of modeling any generic solid. One should remember the main classification theorem for surfaces here, that any solid can either be a sphere or a connected sum of different toroidal. A representation scheme should be able to generate valid solids. It should be complete in the sense that every valid representation produced is unambiguous. It should generate unique solids such that no two different representations should come up with the same object. It should encourage or it should have closure rather implying that permitted transformation operations, transformation set operations in particular on valid solids always generates valid solids. Of foremost importance is compactness and efficiency in matters of data storage and retrieval.

(Refer Slide Time: 16:35)



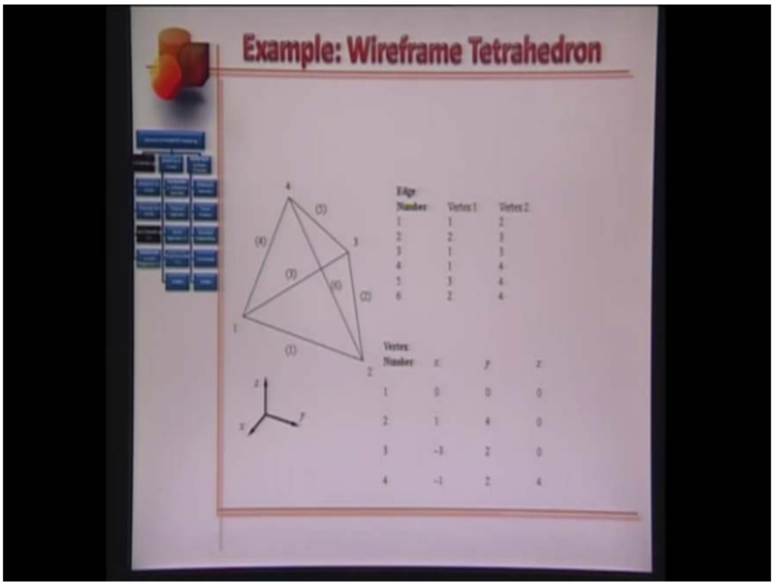
(Refer Slide Time: 17:18)



There are mainly two representation schemes; the wireframe approach and the boundary representation approach. The third one constructive solid geometry; caters more to solid modeling. Both these representation schemes work well for polyhedral solids, and for all solids non-polyhedral ones that can be homeomorpically obtained from corresponding polyhedral solids.

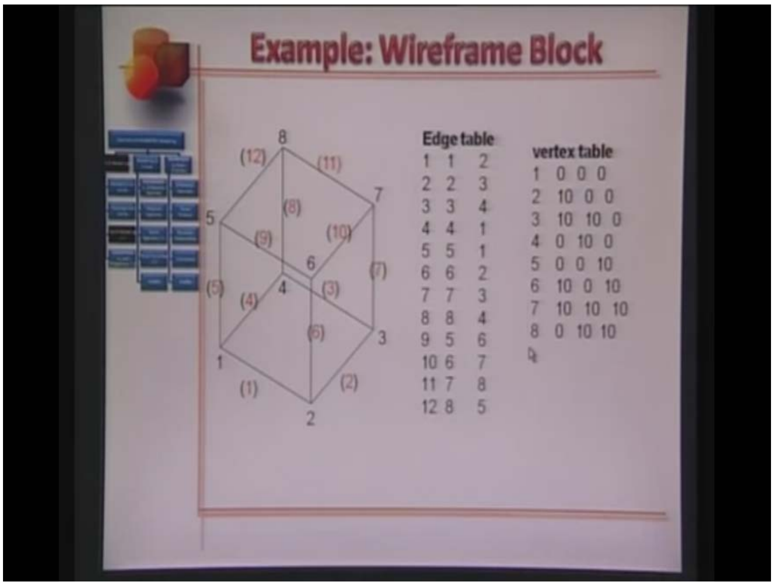
First the wireframe representation, this is one of the oldest methods to represent solids. As the name suggests the representation is primarily through a set of key vertices connected by key edges. Remember here we are not talking about faces, only vertices and edges. Both topological and some and not all geometric information is stored. As I mention before faces are not considered in the wireframe representation. The edges may be straight or curved.

(Refer Slide Time: 18:14)

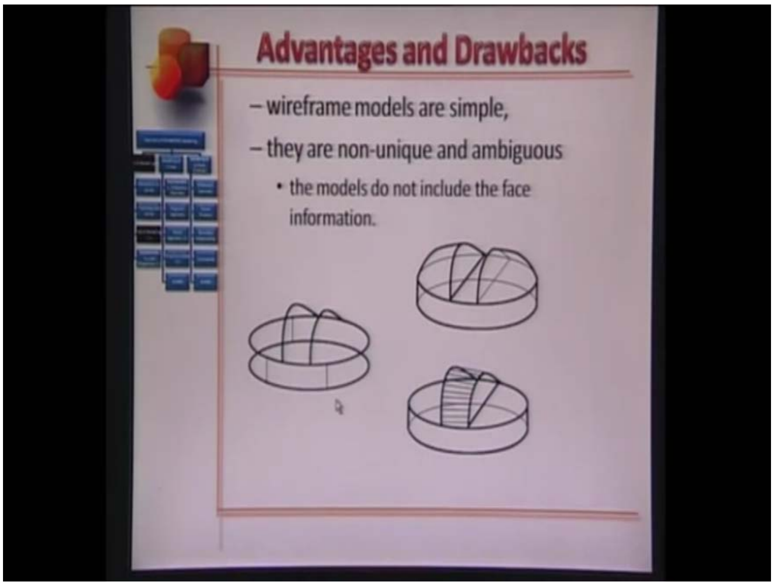


A few examples here: a Wireframe Tetrahedron. You can mark or identify four vertices vertex 1, 2, 3 and 4. You can also identify edges with numbers within parenthesis; edge 1, edge 2, edge 3, edge 4, edge 5 and edge 6; and this time faces are not identified. One can come up with an edge table that records how edges are connected to different vertices. For example edge 1 is connected to vertex 1 and 2, edge 2 is connected to vertex 2 and 3, edge 3 is connected to vertex 1 and 3, edge 4 to vertices 1 and 4; and so on. This table here corresponds to the topological connectivity. How do we store geometric information? We use another table called the vertex table. The coordinates of each vertex are recorded in this table. Once again we emphasize that we do not consider faces in the wireframe approach.

(Refer Slide Time: 19:59)



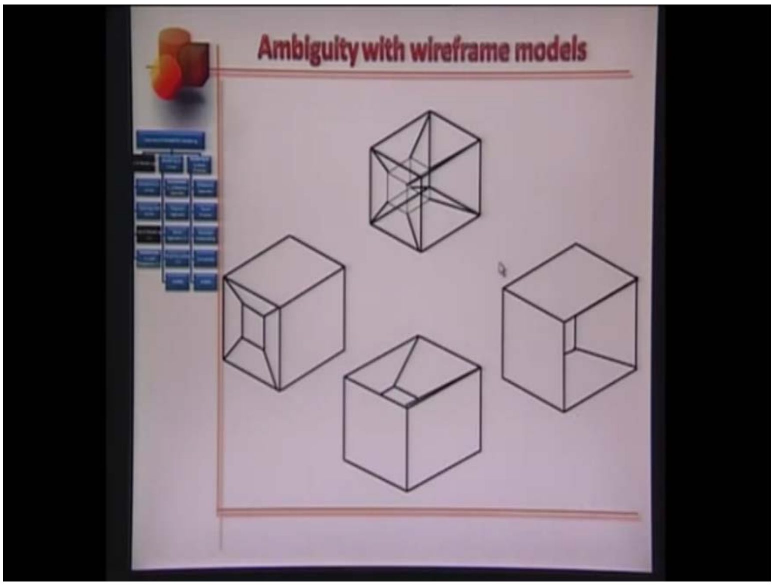
(Refer Slide Time: 21:13)



Another example: of a wireframe block. The edges are numbered within parenthesis and the vertices are numbered without them like in the previous example; we will have two tables, the edge table that would record the topological connectivity between different vertices and edges, and the other the vertex table that would record all the coordinates of all the vertices. First the edge table that suggests that edge 1 is connected to vertices 1 and 2, edge 2 is connected to vertices 2 and 3, edge 4 is connected to vertices one and four and so on: and the vertex table that would record the x y and z coordinates of all the vertices or nodes.

There are a few advantages and drawbacks of the wireframe approach. Wireframe models are very simple; but they are non unique and ambiguous. This is because the models do not include the face information. Consider this wireframe model, one may not find it difficult to identify that the base is cylindrical; but one would find it hard to identify what is laid overhead. Will this wireframe correspond to this solid or will this wireframe correspond to this solid, it is not very clear.

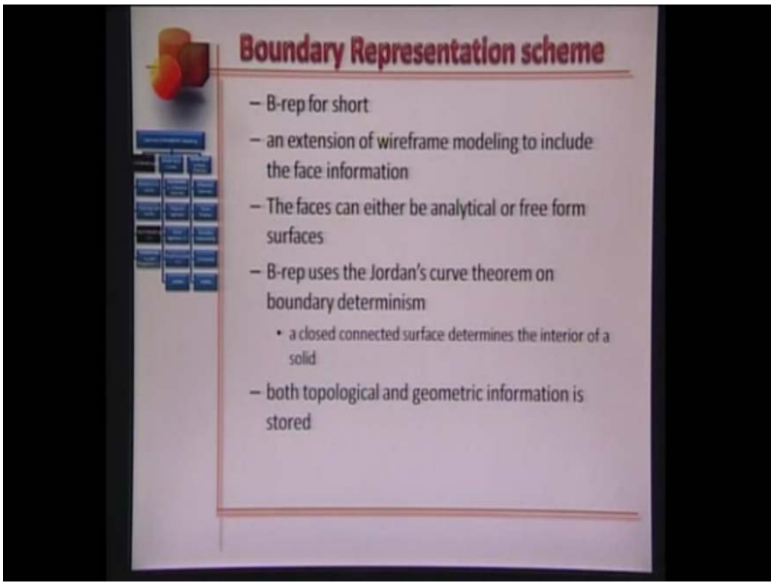
(Refer Slide Time: 22:12)



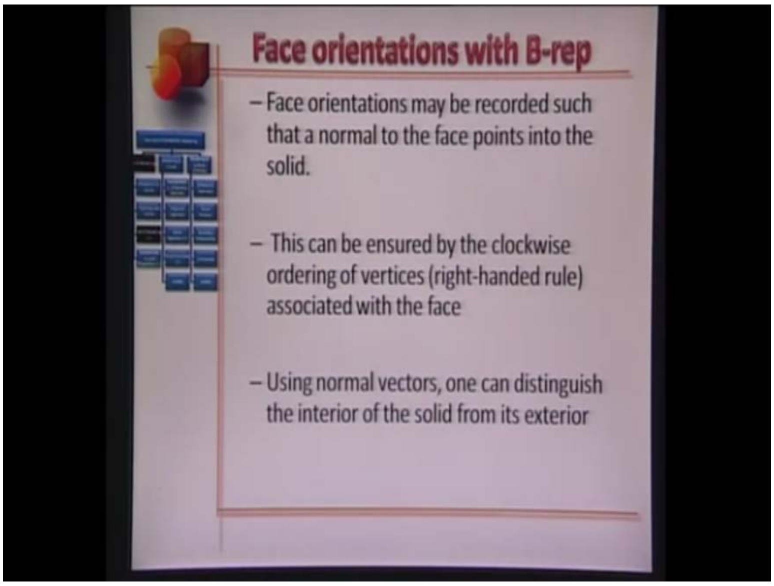
The second example: consider a block within a block. This represents a through hole in a block. What is not clear from this model is, whether the opening of the hole is from left to right like this or whether it is from top to bottom or whether it is from right to left. The boundary representation scheme takes care of the anomalies in the wireframe approach. This is called b-rep for short and the boundary representation scheme is an extension of the wireframe modeling to include the face information.

Faces can either be analytical or free form surfaces. B-rep uses the concept from the extended Jordan's curve theorem on boundary determinism; identical to what we discussed at the start of this lecture. To repeat a closed connected orientable surface determines the interior of a solid both topological and geometric information is stored in the b-rep approach to emphasize the face information is stored in particular.

(Refer Slide Time: 22:58)

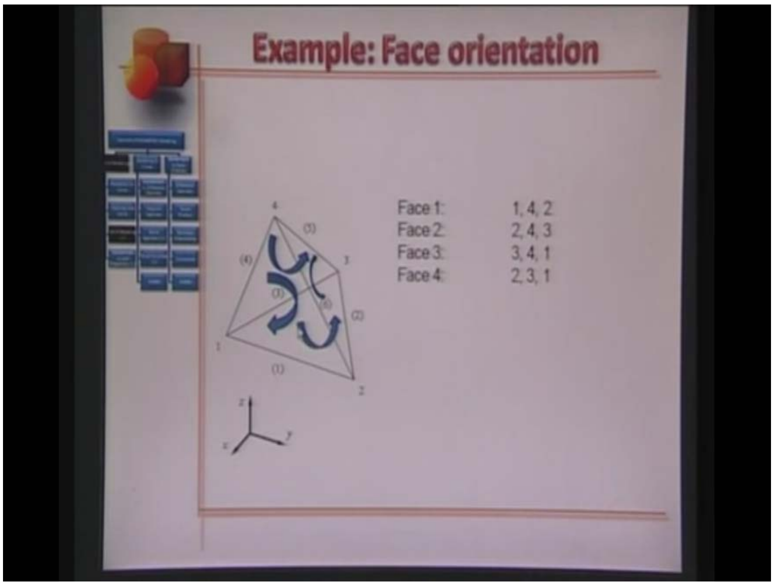


(Refer Slide Time: 24:04)



How do we orient faces with the b-rep approach, face orientations may be recorded such that the normal to the face points into the solid. This can be ensured by the clockwise ordering of vertices using the right-handed rule associated with the face. We will consider a few examples in a few minutes. Using normal vectors it is possible to distinguish the interior of the solid from its exterior.

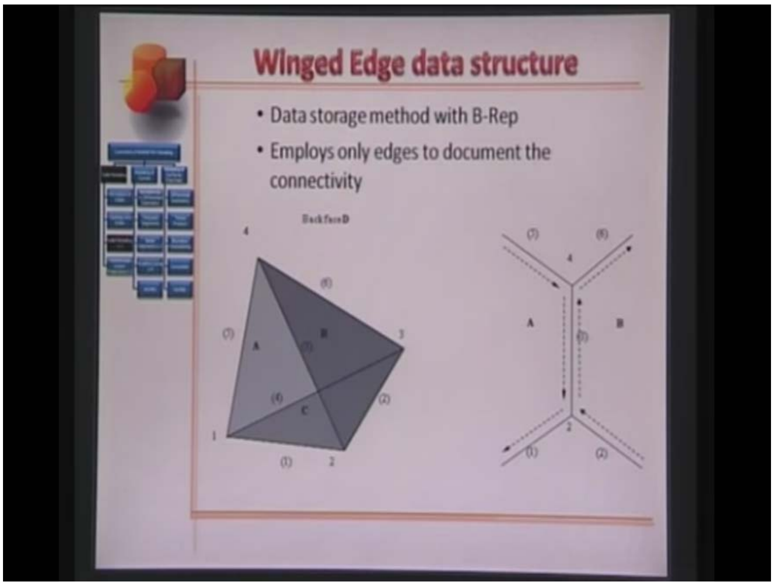
(Refer Slide Time: 24:40)



For example: consider this tetrahedron the vertices are numbered as 1, 2, 3 and 4; the edges are numbered in parenthesis as 1, 2, 3, 4, 5 and 6. Face 1 is composed of vertices 1, 4 and 2. Here if we consider the clockwise orientation we can imagine using your right hand and curling the fingers along this arrow with your thumb pointing outwards. If you think of that you would notice that the thumb pointing into the tetrahedron.

Face 2 likewise is composed of vertices 2, 4 and 3. 2, 4 and 3would give you a clockwise order. Once again if you use your right hand with your thumb pointing outwards and if you curl the other 4 fingers along this arrow you would notice that your thumb is pointing into the tetrahedron. How about face 3 we have two choices here. Choice one is we can go back to face this plane here and order the vertices, or we can stand or set right here as we are consider face 3, 4, 1 to be the rear face and oriented counter clockwise; like this. Once again if you use your right hand as a thumb pointing outputs, curl your fingers, you would notice that the thumb would be pointing into the tetrahedron. Likewise the bottom face can also be ordered counter clockwise, to have the thumb of your right hand pointing into the tetrahedron.

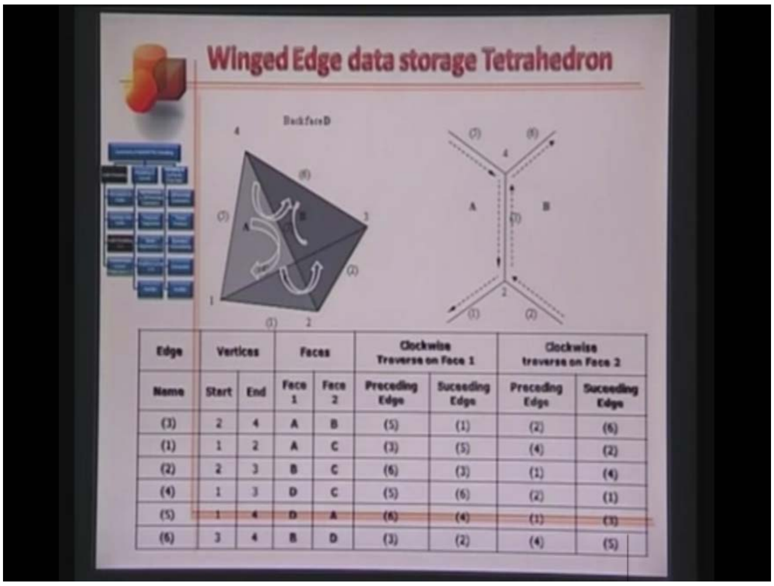
(Refer Slide Time: 27:15)



This is an important concept in the b-rep approach, the winged edge data structure; it is a data storage technique, employs only edges to document the entire topology or connectivity in a polyhedral solid. Let us take an example of a tetrahedron and in particular let us consider edge 3. This edge connects to vertices; Vertex 2 and vertex 4. On one of the sides there is space A and on the other side there is space B. Edge 3 here connects vertex 2 with vertex 4 and one of the sides we have face A and the other side we have face B.

Recall the manner in which we had ordered the vertices for each face, for face A we went clockwise, for face B we went clockwise with respect to the order. There will be an edge proceeding edge 3 on face A and there will be an edge that succeeds edge 3 on face A. In this figure edge 5 would be the edge on face A that precedes edge 3 and edge 1 is the edge on face A, that succeeds edge 3. Likewise on face B edge 2 precedes edge 3, edge 2 precedes edge 3 and edge 6 succeeds edge 3. Let us look at the structure here it pretty much resembles a butterfly structure, with these extensions forming the wing of a butterfly for which reason this data structure is called the winged edge data structure.

(Refer Slide Time: 29:48)



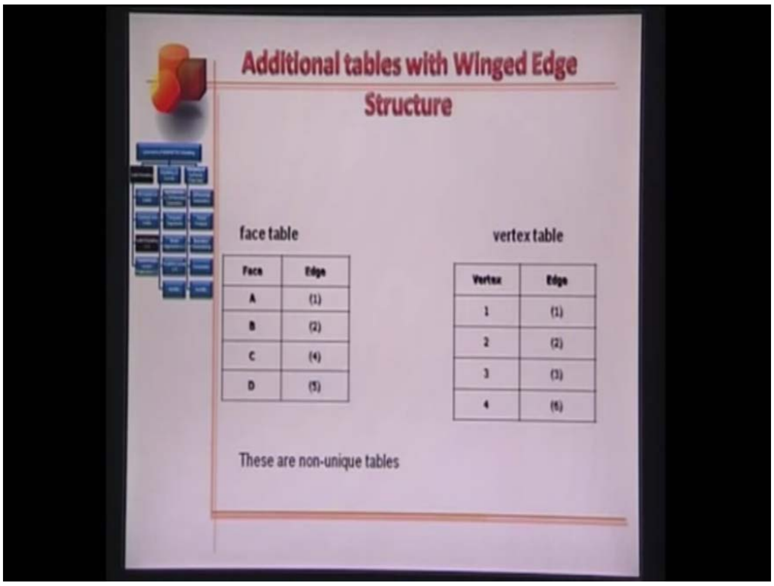
Let us now see how the winged edge data structure works in case of a tetrahedron. These are the pictures from the previous slide, as I mentioned earlier the winged edge structure uses edges as bases to record topological connectivity. In the first column therefore, we have the edge number as the reference. For each edge we will have a start vertex and an end vertex there will be two faces incident on the edge face one and face two. If you perform a clockwise traverse on face 1 what is the preceding edge, what is the succeeding edge. And likewise if you perform clockwise traverse on face 2 again what the preceding and succeeding edges are we will record all these.

Let us start with the first edge, edge 3 we know that the start an end vertices are 2 and 4, face A and B are two faces incident on edge 3. For face A the preceding edge is 5 and the succeeding edge is 1; for face B the preceding edge is 2 and the succeeding edge is 6. Let us try now for edge 1 the start an end vertices will be 1 and 2, once again two faces are going to be incident on this edge face A and face C, face C is the bottom face for this tetrahedron.

On face A the edge that precedes edge 1 is 3, note this clockwise orientation once again the edge that precedes edge 1 is edge 3, the edge that succeeds edge 1 is edge 5. On the bottom face C note this counterclockwise orientation because this is the rear face the preceding edge is 4 and the succeeding edge is 2. Now for edge 2, the vertices are 2 and 3, two faces incident on edge 2 are B and C respectively. On face B if you again know the clockwise orientation, the preceding edge is 6 and the succeeding edge is 3. On face C the bottom face, the preceding edge is 1 and the succeeding edge is 4. For edge4 here start an end vertices are 1 and 3, two faces incident are D and C. C is the bottom face, D is the back face or the rear face.

On D the preceding edge is 5, note this orientation here for the face D, the preceding edge is 5 and the succeeding edge is 6. For face C, the preceding edge is 2 and the succeeding edge is 1. For edge 5, two vertices are 1 and 4, two faces incident are D and A respectively. On face D the preceding edge is 6 the succeeding edge is 4, on face A the preceding edge is 1 and the succeeding edge is 3 and for the last edge which is edge 6 start an end vertices are 3 and 4. 2 faces are B and D on face B the preceding edge is 3 succeeding edge is 2 on face D the preceding edge is 4 and the succeeding edge is 5. So this table here which is based on the winged edge data structure stores pretty much the entire topological connectivity; that includes the face information as well. What is left now is to store the geometry of the tetrahedron. We can use a few more tables for that.

(Refer Slide Time: 35:44)

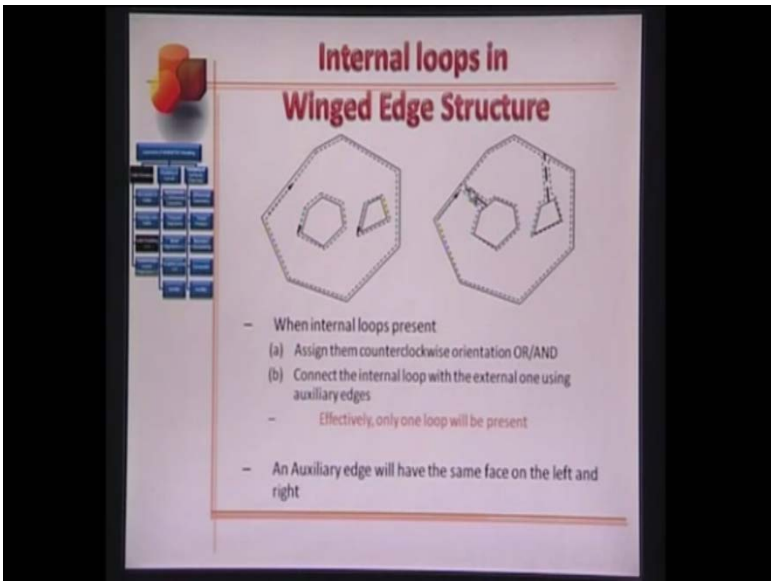


One would be the face table, and the other would be the vertex table. The face table will contain information about the four faces A, B, C and D. We will note down the edges which are incident on these four faces and we can use this table to also store geometric information for each face. We will have to use additional columns for that.

In the vertex table again we can store the four vertices, what edges are incident on each of them respectively. We can store the coordinates of vertices using additional columns which are not shown here therefore, we can actually store the geometry using additional tables. Note that these are non-unique tables, what I mean here is if you look at the second column in both tables, the entries corresponding to the incident edges can be different.

There may be other ways to construct the face and vertex tables. In real life there can be many polyhedral solids, that can have pot holes and through holes. Let me sketch a figure on the board to show you this. Look at this figure for example; a block with a pot hole, this pot hole is going to leave an internal loop on the top face of the block. How would the winged edge get a structure work in case, when these inter loops are present on faces.

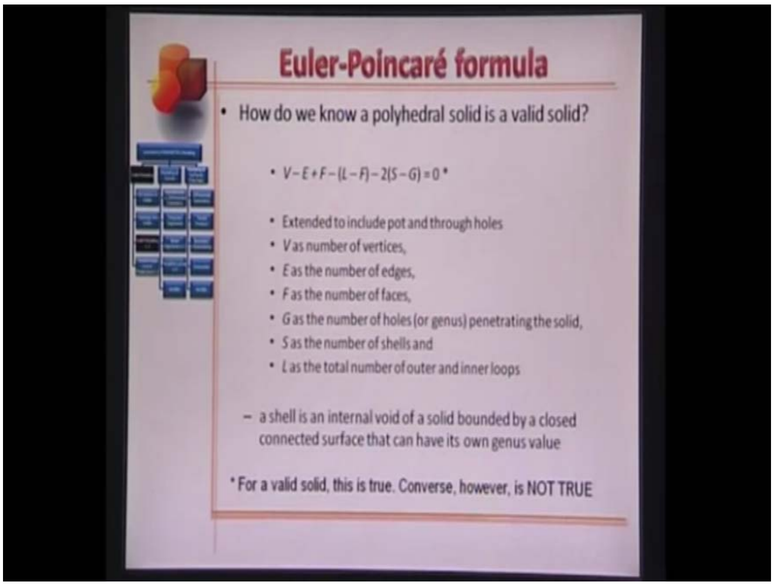
(Refer Slide Time: 38:04)



These are some candidate faces with a few internal loops, when internal loops are present, we assign them counterclockwise orientation and we connect the internal loops with the external loop using auxiliary edges; in this figure for example, these are the two auxiliary edges. Once again the outer loop the vertices on the outer loop are ordered clockwise and the vertices in the internal loops are ordered counterclockwise. And we have used auxiliary edges to connect the internal loop with the external one, what happens in effect is that only one loop remains.

Look at this figure for instance if I start traversing from this vertex clockwise, I go from here to here in traverse towards internal loop. I traverse the internal loop in the counterclockwise manner, go back to the external loop traverse set clockwise, come back to the second internal loop traverse set anti clockwise. Once again go back to the external loop and end up coming to the same vertex, where I started. In the winged edge data structure one should note that an auxiliary edge will have the same face, on the left as well as on the right. In other words, it will be the same face as two faces which will be incident on the auxiliary edge.

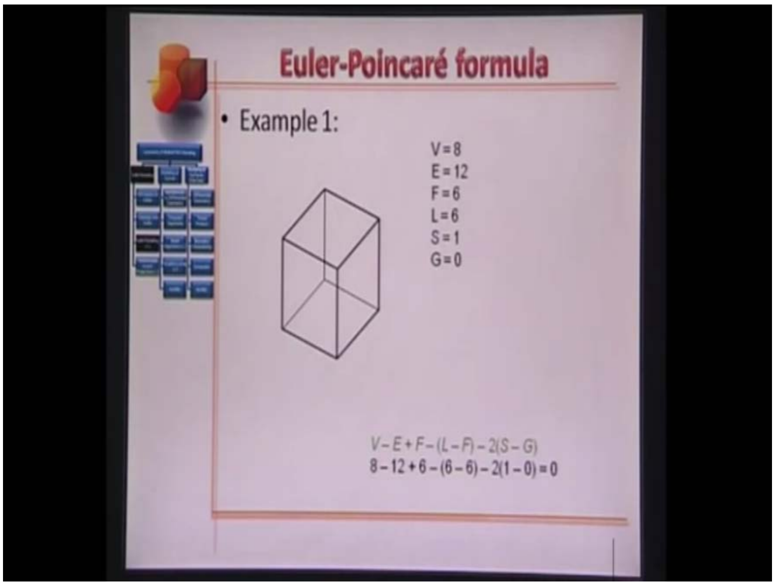
(Refer Slide Time: 40:35)



Euler-Poincare formula; this is used to answer whether a polyhedral solid is a valid solid. The formula goes like V minus E plus F minus within parenthesis L minus F minus 2 times S minus G equals 0. I will tell you what these alphabets mean but first let me say here that this relation is an extension of the Euler characteristic given by V minus E plus F. We will try to prove this relation later or it can be given as an exercise to the students.

Note that for a valid polyhedral solid the relation holds true however, the converse is not true that is if the relation holds true for any polyhedron; that may not be a valid solid. The Euler-Poincare formula is extended to include pot holes and through holes, V is the number of vertices, E is the number of edges, F is the number of faces, G is the number of holes or genus penetrating the solid; it is the number of handles in case of connected some of toroid S is the number of shells and L is the total number of outer and inner loops. A little comment about the number of shells S, a shell can be identified as an internal void of a solid bounded by a closed connected surface; that can have its own genus value. In the sense, whenever we see a closed simple and orientable bounding surface, the shell value S is 1.

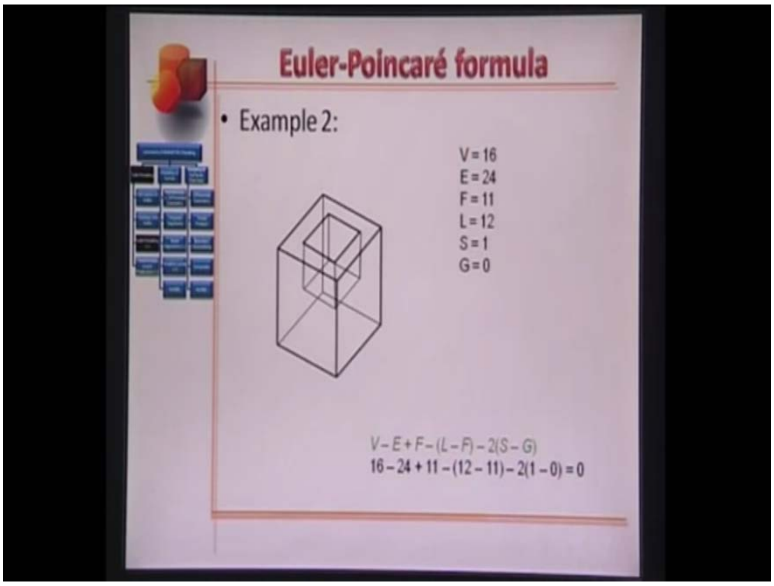
(Refer Slide Time: 43:30)



We will take an example to elaborate a little more on the value S, let us consider the first example. This is a cube or a block, the number of vertices as we all know is 8, the number of edges is 12, number of faces 6, number of loops 6. This is because with each face a loop is associated, the number of shells will be 1 here because a cube will be a bounding surface comprising or including of 6 planer faces. It does not have a handle for which reason the genus value is 0. If you plug in all these values in the Euler-Poincare formula, we see that the right hand side turns out to be 0, suggesting that this cube is a valid solid.

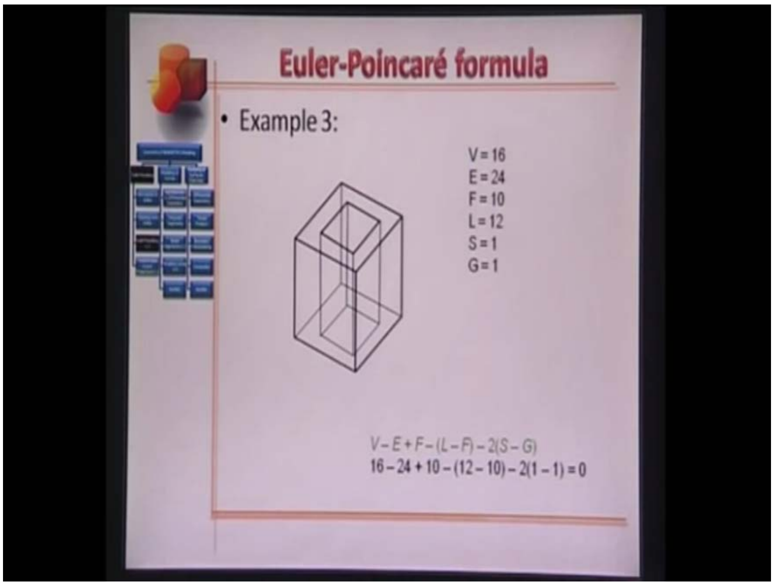
The second example, this is a cube with a pot hole remember that the number of handles here will not be equal to 1 but will be equal to 0. Imagine that you have indented the pothole using a thumb on the outer face of the cube; the solid is still homeomorphic to sphere. If you look at the number of vertices it will be 16, 8 times 2, the number of edges will just be double of the number of edges in a cube 24, the number of faces will be 11.

(Refer Slide Time: 44:43)



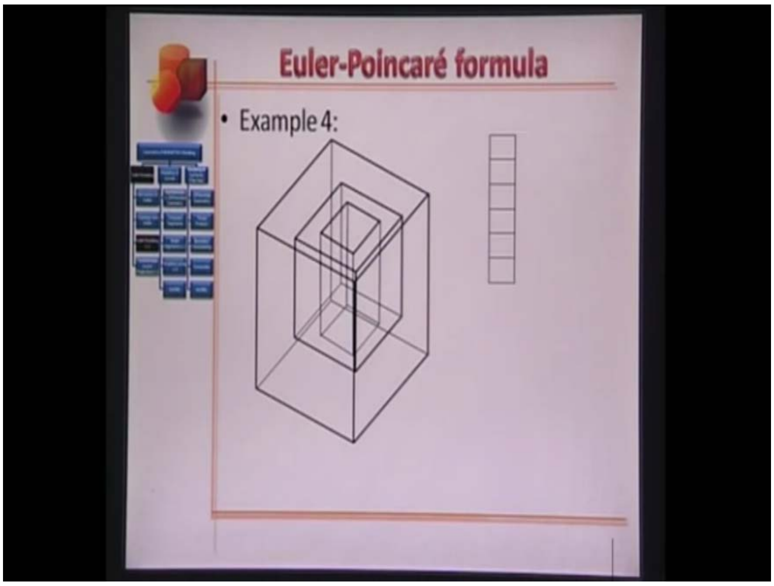
This is because the top face here includes an internal loop, the number of loops will be 12, how? 6 loops corresponding to the outer cube and 6 loops corresponding to the pothole, the shell value is still 1 because we still have one bounding surface that we see and the genus value is 0. If you plug in these values in the Euler-Poincare formula, we again see that the right hand side is 0.

(Refer Slide Time: 46:31)

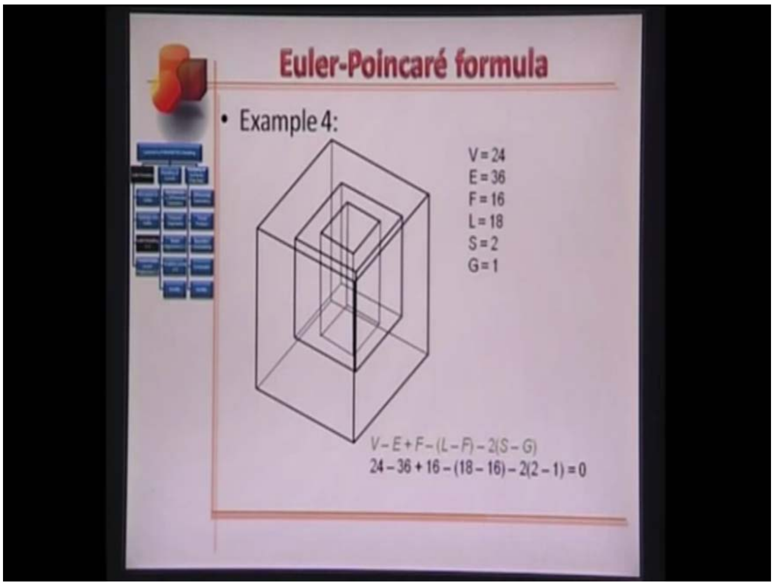


The third example, a cube with a through hole; now we have a handle, this solid is homeomorphic to tutors. We have an internal loop on the top surface of the outer cube and we have an internal loop at the bottom surface of the outer cube. The number of vertices is 16, the number of edges is 24, the number of faces will be 10. Why? Because there will be six faces on the outer cube and four faces corresponding to the polyhedron inside it. The number of loops will still be 12, the shell value is 1 and the genus value is again 1, remember that this solid is again homeomorphic tutors. You plug in these values into the Euler-Poincare Relation and again see that the relation is satisfied.

(Refer Slide Time: 47:49)



(Refer Slide Time: 48:55)

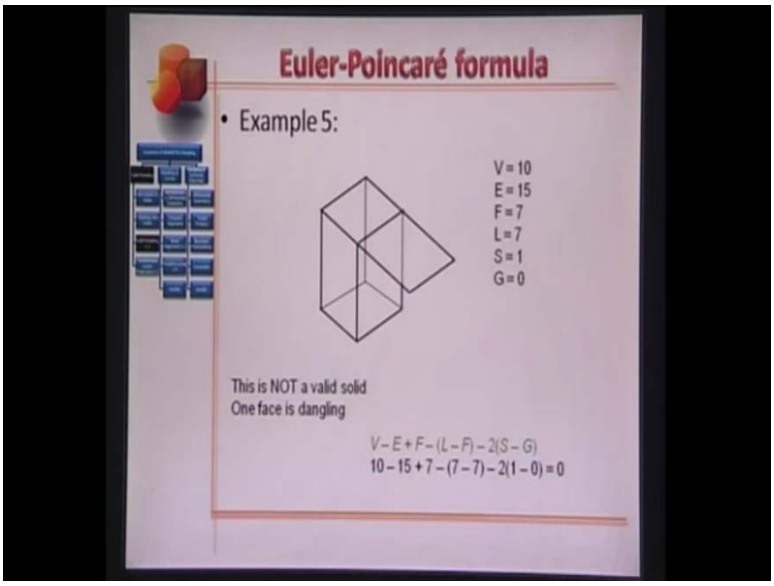


The fourth example, this looks a little complicated. We have a block within a cube and this is within the outer cube, what kind of a solid is this? Let me explain the solid looks pretty much like this; imagine that these two blocks have some thickness. If I insert a duster in one of these blocks and if I try to cover the duster using the other block, this is pretty much how the solid in the slide looks.

In other words, there will be materiel corresponding to the inner most block and its neighborhood is going to be void and there will be material between the outer most block and the middle block. Is this a valid solid? I would say it is. Let us try to see what the Euler-Poincare formula has to predict; the number of vertices 24, the number of edges 36, number of faces there would be 16, number of loops there would be 18. What would be the shell value?

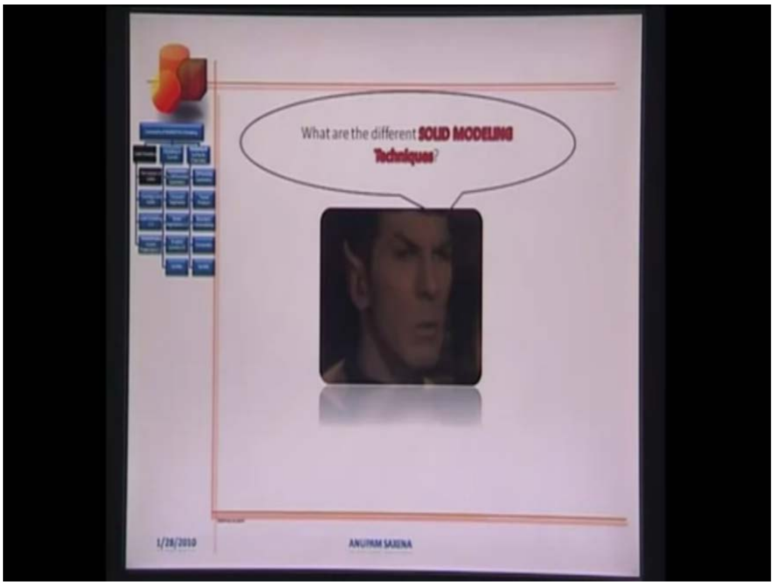
Now if you look at this polyhedral solid closely there are two bounding surfaces; one that corresponds to the outer cube and the one that corresponds to the 2 inner blocks, which looks pretty much like a torus. So the S value for this figure will be 2, what is the genus value? Note that the bounding surface corresponding to the two inner blocks looks like a torus; it has a handle so the genus value will be 1. If we look at the Euler-Poincare formula and feed in all these values, we see that the relation holds, the right hand side of the formula turns out to be 0.

(Refer Slide Time: 51:04)



Example 5, a cube with a dangling face the total number of vertices for this figure is 10, 8 for the cube, 9 and10 the total number of edges is 15, 12 for this block 13, 14, 15 the number of faces is 7, 6 for the block and one dangling the number of loops is also 7. And if we assume that this is a valid solid, the shell value the number of shells is 1 with no handles and therefore, the genus is 0. Euler-Poincare formula predicts that this is a valid solid, but we know that this polyhedron violates the homogenous three dimensionality property of solids. This refers to the converse that even if, Euler-Poincare formula is satisfied, the polyhedral solid need not be a valid solid; in this case we see that one face is dangling.

(Refer Slide Time: 52:40)



Mr. Spock would want to ask now, what would be the different techniques for solid modeling? We will cover them in the next lecture.