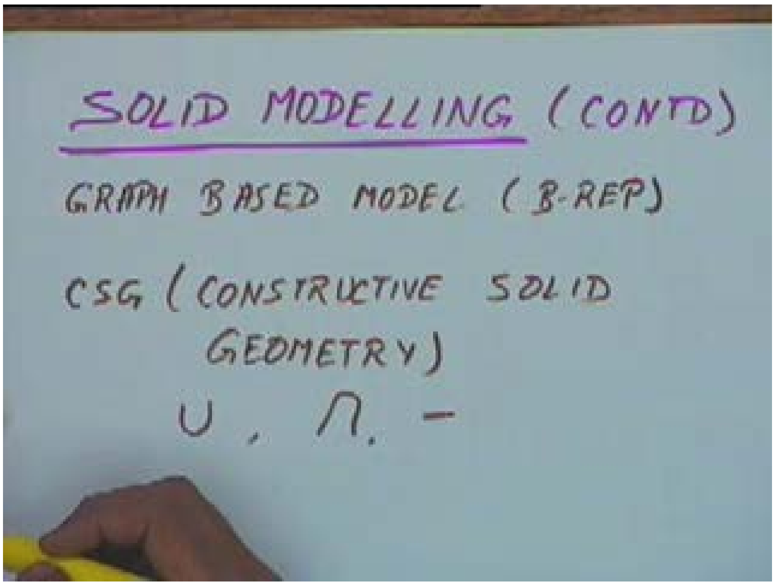
Computer Aided Design
Prof. Dr. Anoop Chawla
Department of Electrical Engineering
Indian Institute of Technology, Delhi
Lecture No. # 39
Solid Modelling (Contd.)

In the last class we had seen two different methods of modelling solids.

(Refer Slide Time: 00:01:04 min)



SOLID MODELLING (CONTD)

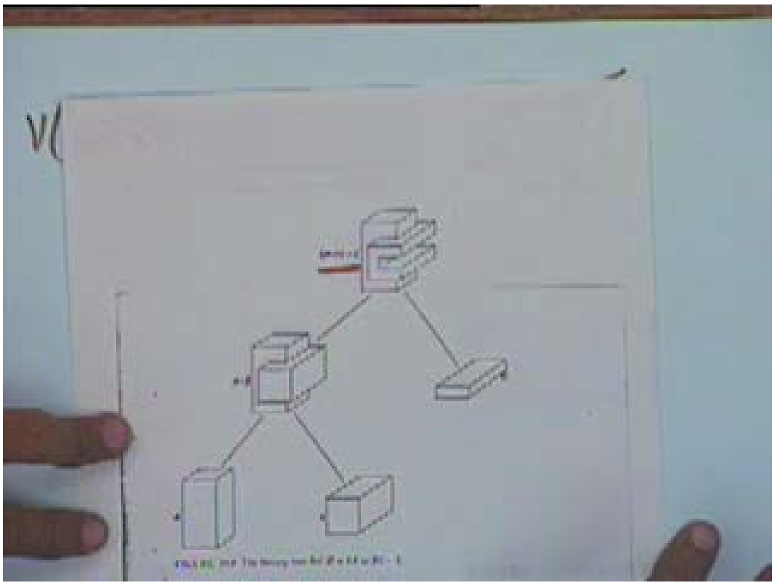
GRAPH BASED MODEL (B-REP)

CSG (CONSTRUCTIVE SOLID
GEOMETRY)

$\cup$, $\cap$, $-$

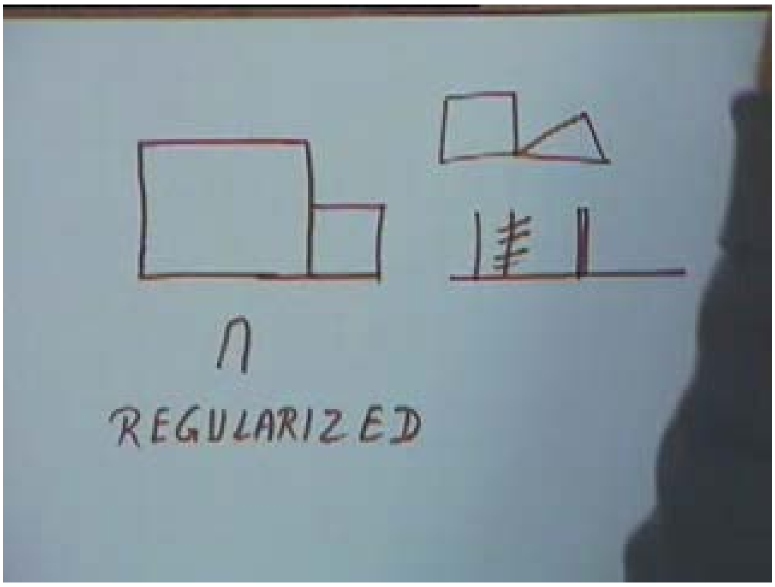
The first method we had seen was a graph based model or a B-REP model. A graph based model, one specific case of which is a B-REP model and the second type of model that we had seen was a CSG model or a constructive solid geometry model. In the CSG model we said that any solid is built by the help of a set of Boolean operations and those operations are union, intersection and the difference operation. If you take primitives and perform these Boolean operations on those primitives to get auto constructor solid. For simple example if we consider an object like this, this can be made by a difference operation between this object and this object.

(Refer Slide Time: 00:02:04 min)



And then an object like this can again be made by a union operation between this object and this object. In both these, in fact all these three objects primitives obtained from a single primitive which is a q, the transformation of a q. So this is how a CSG model is constructed but one thing that we have to be careful about in using the operations is let say if we take a simple case if we have a two dimensional square and we take another two dimensional square which is like this and we tried to find out the intersection.

(Refer Slide Time: 00:02:45 min)



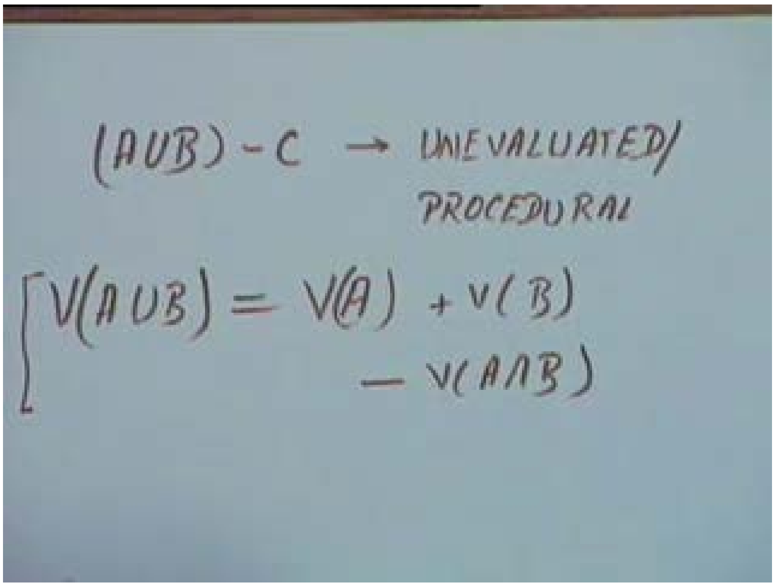
The intersection of these two strictly speaking would be a straight line because the set of points between this square and this square, the set of points which are common are just these points. So intersection of these two will just be a straight line over here. Similarly, if I want to define a difference of these two with this set of points minus this set of points, so one line will have to be subtracted. So these kinds of problems can come up.

If I just consider this single line, this is not really the result that one would normally like to have in a solid model. If we have let say a cube and another cube kept side by side, I will ideally like to say that the intersection is 0 or the intersection is a null set. So this operation is let's say if I am doing an intersection over here and I get this result. This is normally intersection operation. So whenever we are talking of solid models or the CSG model, we don't do a normal Boolean operation. We do what is called a regularized Boolean operation. And the regularized Boolean operation is an operation in which the final result is of the same dimension as the initial result.

If you are talking of a two dimensional object over here, we will not consider cases like this because this result is one dimensional. Instead of this line if it was a narrow rectangle then it is all right but if it is just a single line that will be treated as not existing. So in a regularized Boolean operation, the final result has the same dimension as the initial result. So normally whenever we are talking of union, intersection and difference operations for a CSG model, we are always talking of a regularized operation, a regularized union, a regularized intersection and a regularized deference. So this is one difference between a normal Boolean operation and the operation that we use in CSG model. This becomes critical when we are writing in or when we are writing a program which would carry out this intersection operation. We have to identify cases were such lines are being generated or we might have one rectangle like this and another triangle like this, only one point is common. We have to identify such cases and take care of that.

The other thing I mentioned last time that if I have an object being made like this, these objects are essentially primitives. The transformation of a cube but this object is not a primitive. So if I want to get some properties of this object, I cannot just use the binary expression given over here A union B minus C. I cannot use this binary expression, I have to evaluate this expression, get a graph representation of this or get the information on the surface which constitutes this body and only then I can use this.

(Refer Slide Time: 00:07:02 min)



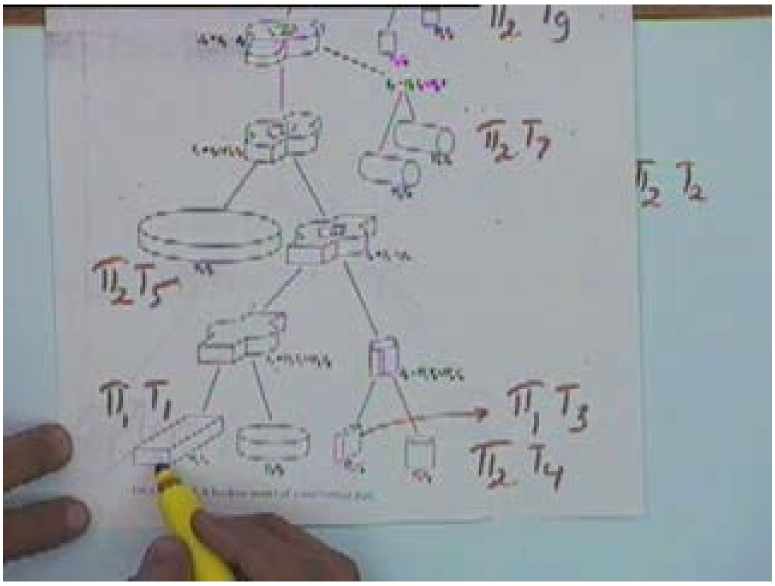
$(A \cup B) - C$ → UNEVALUATED/ PROCEDURAL

$$\left[V(A \cup B) = V(A) + V(B) - V(A \cap B) \right.$$

In that process I mentioned was a process of evaluation. When I have a simple expression like A union B minus C, this expression or this form is referred to as unevaluated or the procedural form. Procedural because it is giving the procedure of how the object is being constructed, unevaluated because it is giving an expression and not the final representation that would be needed for the solid. And only when we evaluate this model that means we get the information of this solid in terms of what are the faces which constituted and so on. Only then we will call that an evaluated form. The primitives like in this case will be represented using some other modeling technique let's say B-REP. There can be other modeling techniques that we use for that, we right now will talk only the B-REP. If we have let's say using some another technique here when you evaluate this model, this solid will also be represented in that technique not as a B-REP normally.

So whatever method we are using for representing these primitives, after evaluation we will try to get this model also in the same representation because if have a model like this and we want to get some solid properties from this model, it's normally not possible. For instance if I just take A union B and I want to find the volume of A union B, volume of A union B, I cannot say that volume of A union B will be equal to volume of A plus volume of B. It will be volume of A plus volume B minus volume of A intersection B. So if I want to find the volume of A union B, I have to evaluate A intersection B anyway. Without that I cannot get the volume of A union B. So we will just comeback to how to compute the volumes, we will come back to that in a minute.

(Refer Slide Time: 00:09:30 min)



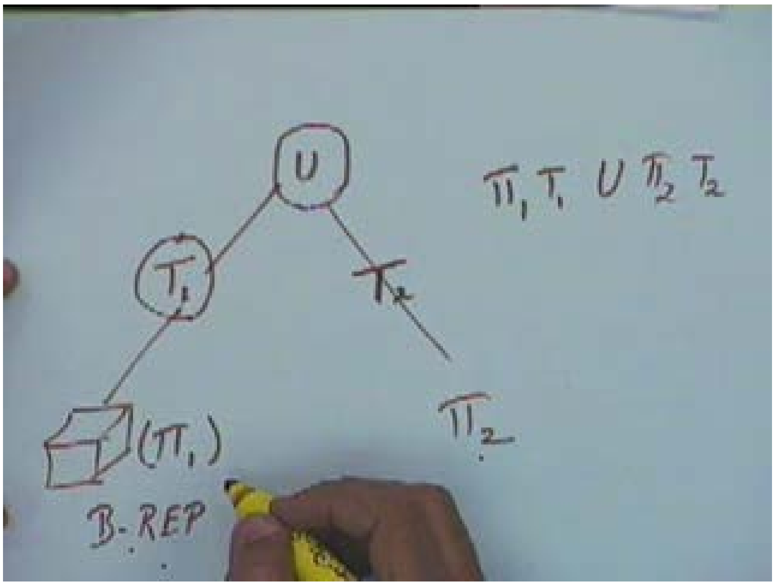
Let's see one more example of how a solid can be constructed. If we have a solid like this, we can construct this solid by taking this solid and these two cylinders and carrying out the difference. I will get these two holes, one hole here and the one hole at the other side. This solid I can construct by taking this solid and then these two cylinders and again carrying out the difference to get these two curves.

Now this cylinder is being written here as pi$_2$ T$_9$. I'll write it larger pi$_2$ T$_9$, pi$_2$ is a primitive and T$_9$ is a transformation which has been given to that primitive. So pi$_2$ is been taken as a unit cylinder in this case and T$_9$ is a transformation which transform the unit cylinder into this cylinder which has some specific dimension and specific location and so on.

Similarly when I consider these two cylinders these are referred to as pi$_2$ T$_7$, pi$_2$ is the same primitive, a unit cylinder transformed through a transformation T$_7$ that will give us this cylinder which we will be using in a difference operation with this object to get this and so on. This object is obtained by taking this object and this disk and carrying out their intersection, so as to get a curve over here. And this disk is again pi$_2$ T$_5$ same primitive with a different transformation and we will continue further. This object is made by taking this object and its difference with this.

Again this object is made by taking this block and a disk and the union. This block is written as pi$_1$ T$_1$, pi$_1$ is a primitive which is cube, unit cube, T$_1$ is a transformation which transformed it into this block. Similarly here I have this as pi$_1$ T$_3$, pi$_1$ is a unit cube, T$_3$ is a transformation which converts into this block and pi$_2$ T$_4$ which is again a unit cylinder transformed into this cylinder.

(Refer Slide Time: 0:12:37 min)

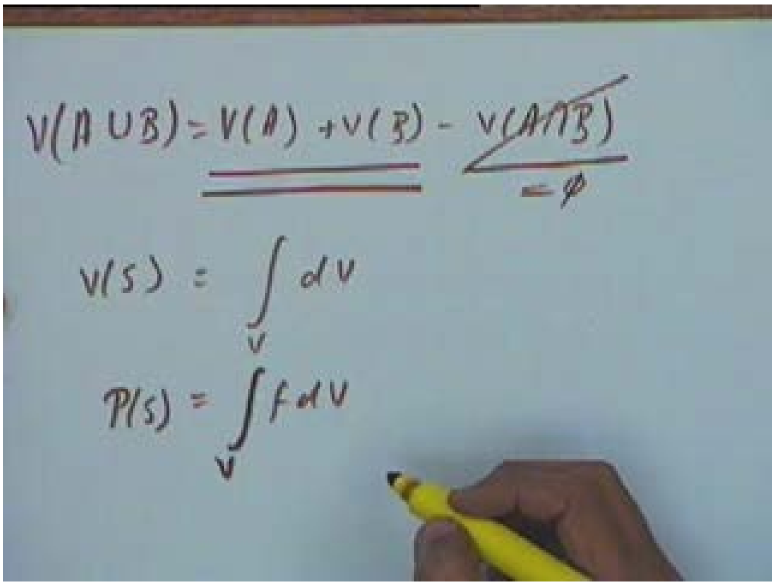


The point that I want to make out now is that let's say we have any cube, unit cube this is a primitive let's say pi$_1$ in this case. This I am transforming through some transformation T$_1$ and then I am getting another object and let's say I have taken another primitive pi$_2$ which is being transformed through let's say T$_2$ and then let's say I am taking some Boolean operation of that maybe their union. When I have to evaluate this model, see this is only a procedural model right now. That means this expression pi$_1$ T$_1$ union with pi$_2$ T$_2$, this is only a procedural model right now.

When I want to evaluate this, the process of union is going to be very expensive whereas I mentioned earlier whenever I have to find out, whenever I have to carry out Boolean operation I have to find out the intersection of all surfaces in one with all the surfaces of the other body. To find out which are the new surfaces which are generated and which are the new lines which are generated, which are the new edges which are generated and so on. So union can be a complex operation but carrying out this transformation that is relatively a simple operation and these transformations they do not affect the boundary model. Let's say this solid is represented using either B-REP or some graph based representation. This transformation is not going to alter this representation.

The transformations are not going to alter the, what do you say the topology of this object. The transformations are not going to alter this topology. Only the union operation is going to alter the topology. All Boolean operations can alter the interaction between the different surfaces. New surfaces can be generated, new edges can be generated and new vertices can be generated by a Boolean operation but that cannot happen by a transformation. So these transformations are normally relatively simple to use. It's only the Boolean operations which are expensive, in terms of evaluation. If we are talking of a transformation that does not change the topology of the object, only the Boolean operation would do that. Transformations would change the properties of the solid. Properties meaning, if I want to compute the volume of this object a transformation can change the volume of this object. Therefore I can change this cube into maybe a block or rod or anything. So the volume of this object can get altered but the topology or the interaction between the surfaces that will not get altered.

(Refer Slide Time: 00:16:08 min)



$$V(A \cup B) = \underline{\underline{V(A) + V(B)}} - \underset{= \phi}{\underline{\cancel{V(A \cap B)}}}$$

$$V(S) = \int_V dV$$

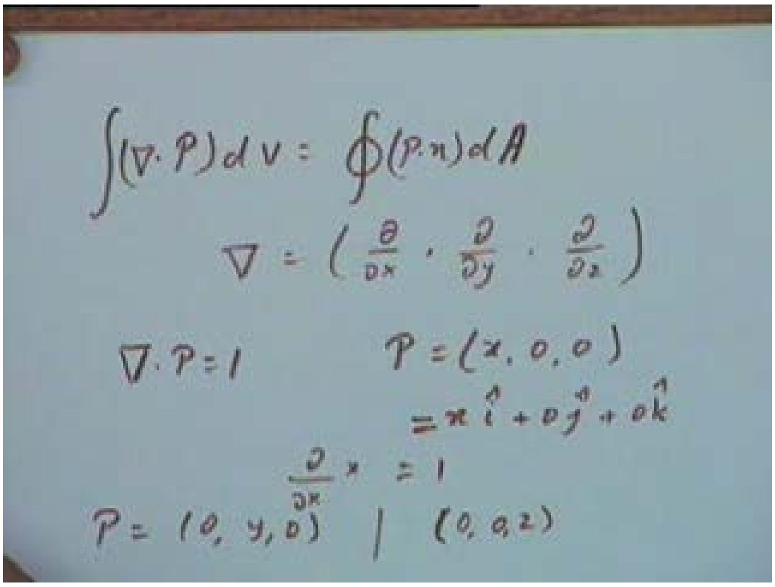
$$P(S) = \int_V f\,dV$$

Now let's see if I want to compute the volume of an object, how can we compute the volume of an object represented in this manner. For instance as I said if I have A union B, I want to find the volume of this. This will be volume of A plus volume of B and minus volume of A intersection B. In this case if I can ensure that A intersection B is equal to 0, that is equal to a null set. Then the volume of A intersection B will always be 0. So there are modelling techniques which ensure that A intersection B will always be 0. The intersection of all solids will always be 0 in that case. We will see some of those modelling techniques.

So in that case we can easily use this operation or this method in finding out the volumes because A and B will be primitives or will be derived from primitives. So volume of the objects can be easily computed and we know that the intersection will always be 0, so we will always ignore this term. But if that does not happen then we have to use the boundary representation of the solid. So if we have any arbitrary solid, let say this solid and we have its boundary representation, we have the equations of all its faces and the interaction with the other faces and so on. We can use that representation to get the volume of this solid. How we can do that? See the volume of a solid is a volume integral of dv.

Similarly, if let’s say instead of a volume, if we want to find out some other property maybe the moment of inertia about a particular axis or something like that then that will be equal to integral of f times dv where f is some function of x y and z. So we are interested in evaluating this integral and this is we want to find out the volume integral. This is equal to some property of the solid. Therefore evaluating this volume integral, we know the volume in terms of its bounding surfaces. So we will try to breakdown this volume into a surface integral. For breaking down a volume into the surface integral, we will use what is called the Gauss’s theorem, integral of del dot p with dv is equal to close surface integral of p dot n with dA where the del operator is del by del x, del by del y, del by del z and P is any field, P is vector field and space in three dimensions.

(Refer Slide Time: 00:18:49 min)



$$\int(\nabla\cdot P)\,dv = \oint(P\cdot n)\,dA$$

$$\nabla = \left(\frac{\partial}{\partial x}\cdot\frac{\partial}{\partial y}\cdot\frac{\partial}{\partial z}\right)$$

$$\nabla\cdot P = 1 \qquad P = (x, 0, 0)$$

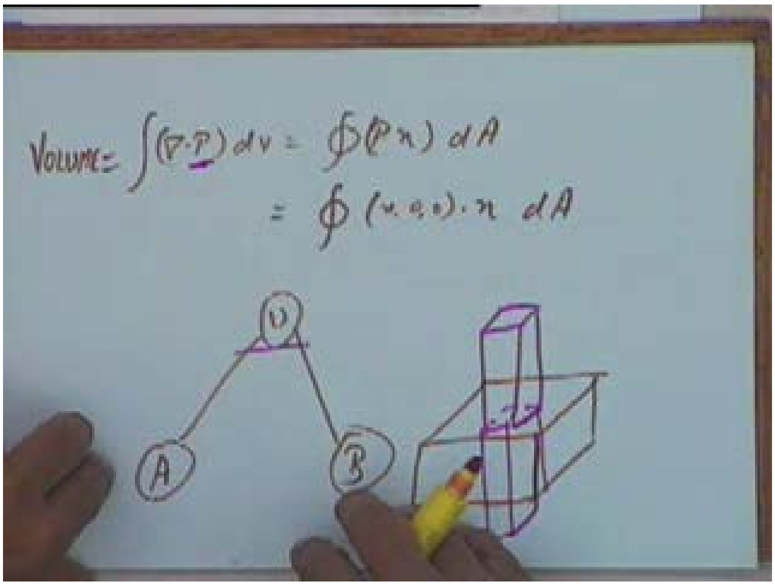
$$= x\hat{i} + 0\hat{j} + 0\hat{k}$$

$$\frac{\partial}{\partial x}x = 1$$

$$P = (0, y, 0) \;\Big|\; (0, 0, z)$$

So let’s say if I want to find out the volume, what I want to do is I want to ensure that del.P should be equal to 1. So for ensuring this we will take P to be equal to x, 0, 0 that means it will be x_i plus 0 j plus 0 k. If I take P to be this, my del.P will be equal to 1 because I will take the dot with this operator and we will get del by del x of x plus the other two terms are 0 which is equal to 1. So del.P will be equal to 1 in that case or x dot of x 0 0, I can also take P to be equal to 0, y, 0 or I can take it to be 0, 0, z. If I take del.P to be equal to 1 let’s say I take P to be x, 0, 0 and I want to find out this integral that means surface integral of P.n into dA.

(Refer Slide Time: 00:20:59 min)



Volume $= \int (\nabla \cdot \underline{P})\, dv = \oint (P \cdot n)\, dA$

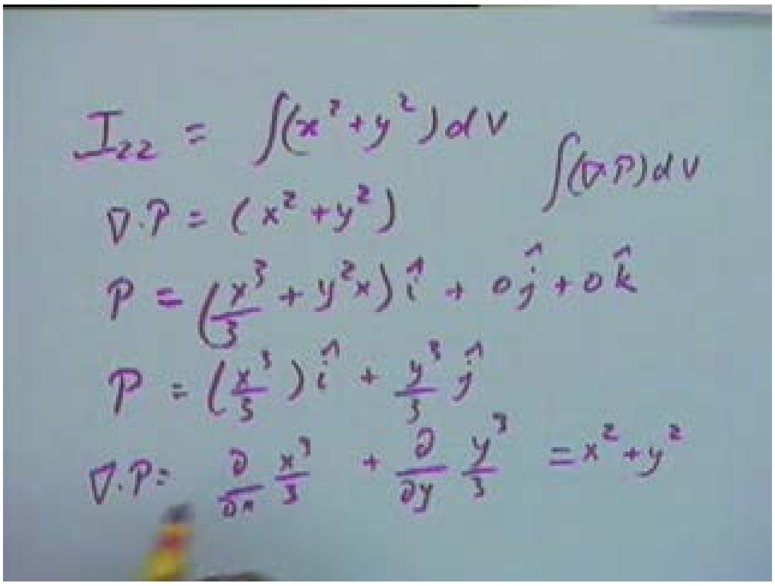
$= \oint (x,0,0) \cdot n \; dA$

So this will become a surface integral of x, 0, 0 dot with n and integral over all surfaces. This means that if I have any boundary or any solid, at each of the surfaces I can find out the direction of the unit normal vector. And if I know n for this surface, I can get an expression for this integral. Since I know n, I can take as dot product with x, 0, 0 and then integrate it over this surface because I know the equation of this surface, I know the details of this surface. And this I can repeat for all the surfaces which are defined in this solid and the details of the surfaces are known because we have a boundary representation model.

So we can evaluate this integral and this will be equal to the volume. Yeah, (Refer Slide Time: 22:30). See, let's say if I have two solids A and B, I have done the union and now I have a solid A union B. I know the boundary representation model for A, I know it for B but if I have to find the volume of A union B, I have to evaluate A union B first. That means I have to get the B-REP model for A union B. That means let's say if I have one block like this and a second block. Let's say which is something like this, let's say initially it is longer. Once I do the union, I get this block with the handle on top, I will have to get the B-REP model for this complete solid which will be this top surface with the cut in between these vertical surfaces only up to this point and so on.

Just by looking at the B-REP of A and B-REP of B, I cannot find the volume of A union B, I have to the get the B-REP of union B. That is why in order to find the volume or in order to compute any properties of A union B, I have to evaluate A union B first. Unless I evaluate this model it is not going to be possible for me to the compute the volume. So my most of the operations, even if I want to let's say display this, it is normally not possible to do it unless you evaluate the model (Refer Slide Time: 24:22). So evaluation has to be done. Only once the evaluation is done we have a complete B-REP model of A union B only then you can apply this one. And this function that we take P, the P that we are choosing P that will control what property we are trying to analyze.

(Refer Slide Time: 00:24:54 min)



$$I_{zz} = \int (x^2+y^2)\, dV \qquad \int (\nabla\cdot P)\, dV$$

$$\nabla\cdot P = (x^2+y^2)$$

$$P = \left(\frac{x^3}{3}+y^2x\right)\hat{i} + 0\hat{j} + 0\hat{k}$$

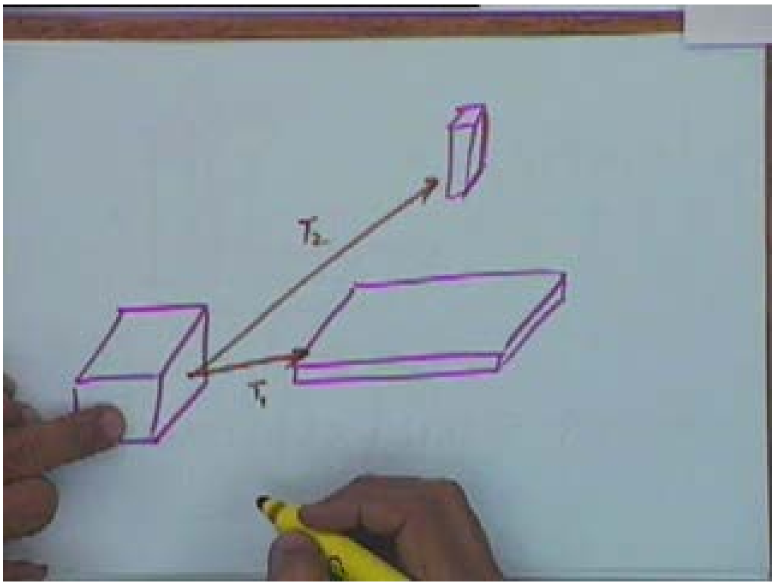
$$P = \left(\frac{x^3}{3}\right)\hat{i} + \frac{y^3}{3}\hat{j}$$

$$\nabla\cdot P = \frac{\partial}{\partial x}\frac{x^3}{3} + \frac{\partial}{\partial y}\frac{y^3}{3} = x^2+y^2$$

Let’s say if we want to find out the moment of inertia, the moment of inertia we can, let’s say if I want out the moment of inertia about the z, I_{zz} which is equal to integral of x squared plus y squared dv. So we want that del.P should be equal to x squared plus y squared. So what should be the P that we take? We want del.P should be equal to x square plus y square, x cube by 3 plus y squared x times i plus 0 j plus 0 k. All alternatively I can take x cube by 3 i plus y cube by 3 j then del by del x of this will give me x square, del by del y of this will give me y square. I can take this to be expression for P where del.P will be equal to x square plus y square and then again I can find out the integral of P.n. I will take a dot product of this where all the normal vectors of the surfaces and compute the surface integral. That way I will be able to evaluate the moment of inertia.

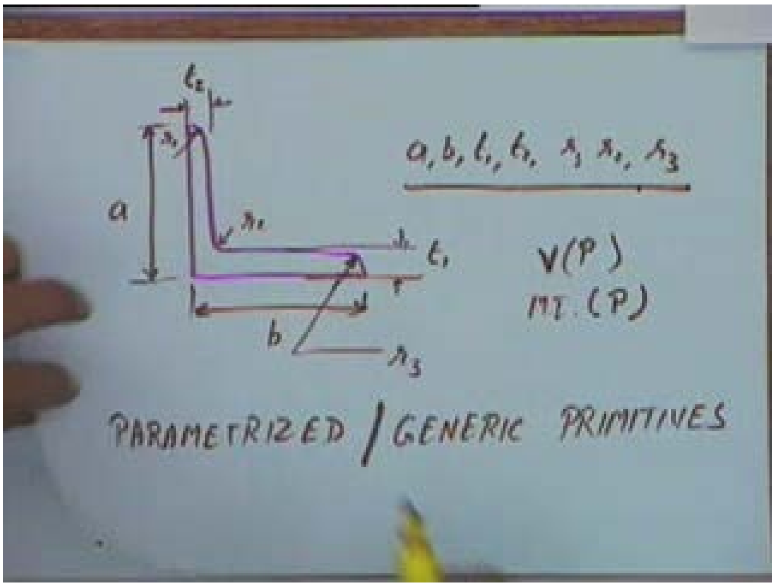
Let’s say if I am taking P to be equal to this, my del.P is equal to del by del x of x cube by 3 plus del by del y of y cube by 3 which is equal to x square plus y square. Whether I take this P or whether I take this P, evaluating the integral del.P dv, it will give me the same moment of inertia I_{zz}. The choice of this function P of this field P is not unique. In this you have to differentiate this term with respect to x, so this will give you y square, this will you give x square. Here you differentiate this with respect to y and this with respect to z, so you will still get x square plus y square. So the choice of P is not unique and you have now different sets of P are giving you the same moment of inertia. Now depending of the moment of inertia we need or the depending on property that we need will give the different value for P and carryout this integral, carryout this surface integral.

(Refer Slide Time: 00:28:32 min)



Now the next thing that we can see is that let's see if I am taking cube or a unit cube as a primitive, I can transfer the unit cube and get even a plate and even a thin vertical rod. This is simply by an operation of scaling but quite often we can have complex object.

(Refer Slide Time: 00:29:01 min)

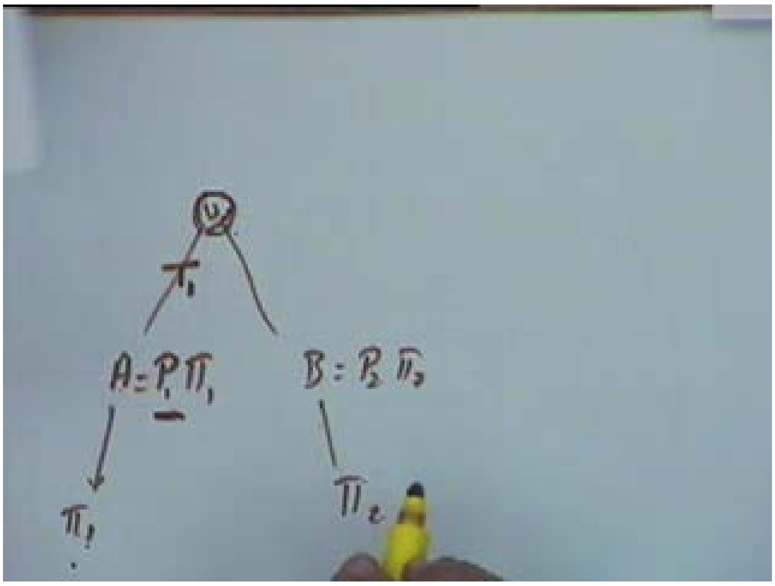

t_2
r_2
a
r_1
t_1
b
r_3
$a, b, t_1, t_2, r_1, r_2, r_3$
V(P)
MI. (P)
PARAMETRIZED / GENERIC PRIMITIVES

For instance if I take a angle, we have an angle like this which has let's say certain radius of curvature here, certain radius here, certain radius here, certain thickness here, some thickness here, certain height over here and certain height over here. Now let's say in some application we are using angle of difference sizes in difference places. Every time I want to make a model of this angle, it can be even called complex curve. Making an angle like this will mean that I will first take a plate horizontal or a vertical plate. I will take the union then I will take and fill it over here then we will take another fill it here, another fill it here and so on. Angle can also have a draft over here. So it will complicate matter even further.

Every time I have to make an angle, one would normally not like to go through a sequence of steps like that. So what is often done is we make primitive which are complex object like this and these primitive will have certain parameter. Let's say this is a, I call this b, I call this as say t_1, this is t_2. Let say this is r_1, r_2 and let say this is r_3. So, we define these parameters and this primitive is defined by the help of these parameter a, b, t_1, t_2, r_1, r_2 and r_3. So, all these parameters together will be defining a primitive like this. So whenever we use this primitive in any application, we will have to specify these parameters. So this is what is referred to as a parameterized primitive, parameterized or generic primitive.

In addition to specifying a transformation, in addition to specifying a transformation, we shall convert this unit cube into either a plate or a rod. Let say this is a transformation t_1, this is a transformation t_2 in addition to that for the case of parameterized primitives, we also have to specify the values of these parameters.

(Refer Slide Time: 00:32:00 min)



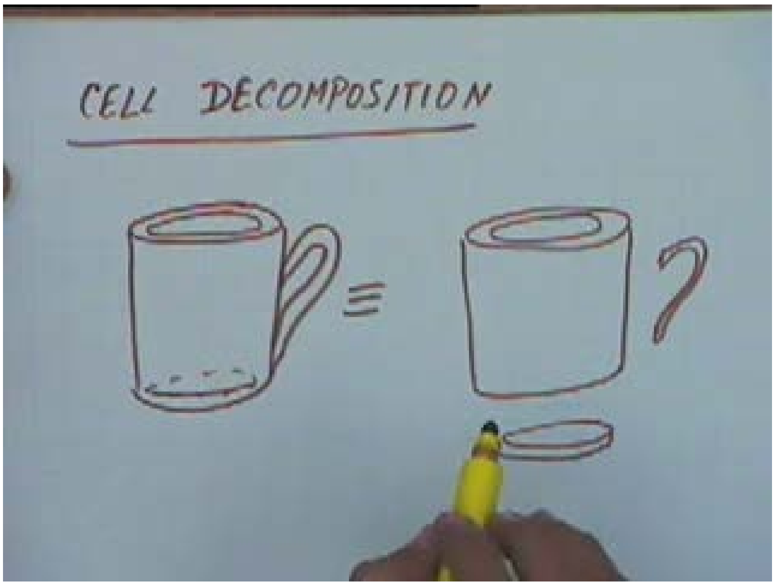
So in a particular situation I might have a solid A which is actually let's say I have some primitive pi$_1$ which I am instantiating (Refer Slide Time: 32:11) through a set a parameters P, A is equal to P of pi$_1$ let's say, pi$_1$ with a set of values attached to it which are P and this A is referred to as an instant of pi$_1$. This is a generic primitive when I give some set of values to it, this is referred to as an instance of this generic primitive and I say I can have a number of such parameterized primitive and then I can perform Boolean operation on that. So this process of instantiating parameterized primitive gives us in some sense a family of primitives which have the same topology which has the same surfaces, surfaces are the same.

Even if I take a different value of t_2 or t_1, my surface interaction is not going to change. So, this process of parameterization is also like, is a similar to the process of transformation in the sense that it does not alter the boundary model. It does not alter the topology of the object, so this process is relatively inexpensive and after taking an instance of this primitive then we can carry out a Boolean operation. The Boolean operation cannot be performed unless you take an instance of this primitive. The Boolean operation can only be performed if we take an instance because unless we give some values to these parameters, we don't know the exact sizes of these surfaces and location and so on. So typically if you are talking of CSG model using parametric primitives in that case we will have a process of parameterization. Then again of course they can be a transformation also. This parameterized primitive can be transformed and then used in a Boolean operation because just a single transformation cannot help us to get the different sizes of the angles because if I give it in a non-uniform scaling in the two direction, the different parameters will not change proportionally. So we have a of what is referred to as parametric or generic primitives which help us give us in some sense a family of primitives and in a particular application, we give values to these parameters through a process of what is referred to as instantiation. And then we have instance of these generic primitive being used in any particular application. And if we want to find out let's say any property of this, this primitive either the volume or the moment of inertia, the volume of this primitive that we can compute with a help of some formula which can be known to us.

For finding out the volume of this I need not carry out the complex integration. The volume will be some function of these parameters the P's, even the moment of inertia will be some function of these parameters P and these properties, the formula for this volume and the moment of inertia etcetera these can be stored along with the model of these primitives. So that, one doesn't have to carry out a complex integration to find out the volume of these primitives. And as I mentioned the process of this instantiation that does not offer, that does not alter the boundary model for these primitives. That does not alter the topology and the B-REP model will remain unchanged.

So when I am evaluating this union, the B-REP model for pi_1 will be the same as that for A and will be the same after transformation T_1. So the extensive process will still be the union. Any question on this? Same topology, the adjacency relationship between the different surfaces will remain unchanged. Any question on this?

(Refer Slide Time: 00:37:45 min)

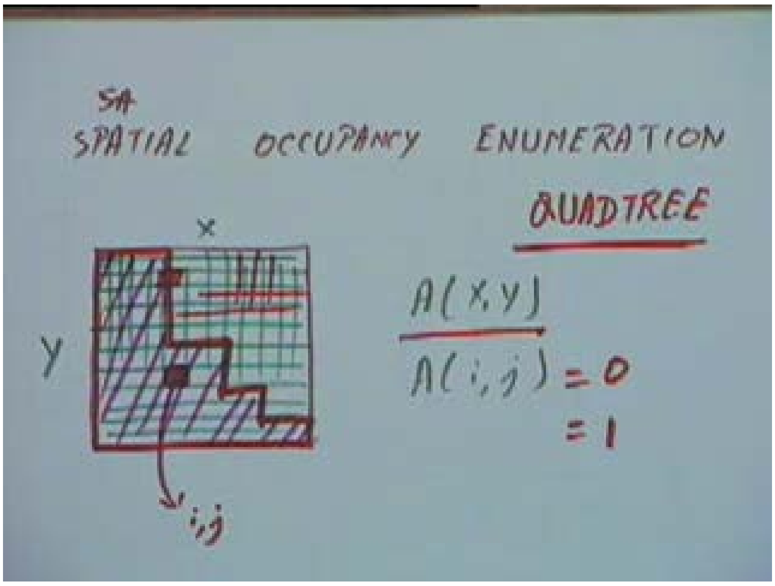


Now let's see the other method that I mentioned that whether intersection of the different solids is always 0. And that is let's say one method is this method, what is referred to a cell decomposition and this is a set of methods all of which come under this category. What is generally done is if you have any solid, we will decompose them we will decompose that solid into a different set of part. So let's say if I have a cylinder like this with maybe a handle over here. Let’s take a realistic thing. It’s a hollow thing and it’s got a disk at the bottom. So this solid I can represent that by one simple hollow cylinder. It’s union with a disk and its union with this handle. So what I am effectively doing is I am taking the solid like this and decomposing it into three different parts. Each of these parts can now be represented relatively easily.

Now I will say this solid is the union of these three parts. In this method, the intersection between any of these solids will always be 0. I ensure that there is no intersection between these, we are just decomposing this solid into a set of smaller or simpler solid and each of these solids can again be further decomposed. But we will not have any intersection between two primitives. So creating a solid like this, we are in effect starting from the initial solid and then decomposing it. Yeah, what is meant by approximate? I am coming to that, so we do that later. The cell decomposition is a set of generic method; this is one way of decomposing it. We will talk of other methods in which any complex object will be broken down into a set of small cubes; I will be coming to that topic.

So this method is referred to cell decomposition or these kinds of methods are referred to as cell decomposition method. The advantage that we have is that since there is no intersection, the properties can be computed very easily. But they are times when this kind of method can become slightly complex in creating an object because you have to know the final object, what its shape will be before you decide what will be the primitive.

(Refer Slide Time: 00:41:12 min)



In other type of cell decomposition method is what is referred to as spatial occupancy enumeration. We just consider two dimensional examples to start with. For instance, if we have let’s say this is a square and my object is let’s say something like this. That means the surface that I want to represent is just this surface. The outer square that I have drawn that is just the limit of the surface. What we can do is we can divide this square, we can divide the limits of this surface into a set of very small squares, we can divide it into an array of squares. And then for each of these squares, we can say whether the surface exists there or it does not exist. So we will get an array of zero’s and one. These are the complete area, this complete area will be divided into an array let’s say x in this direction and y in this direction.

And, if I say an array A of x y such that A of ij will tell us whether the surface exists at location ij or not. If ij is this particular square, let's say if ij is this square, this square is ij. A_{ij} will be equal to 0 if it is outside the surface, it will be equal to 1 if it is inside the surface because surface defined in this case in this boundary. We obviously have a problem if my square is partly inside and partly outside, if it is something like that. So in that case what we will do is we can either decide as to what is the resolution that we want. We do not want to consider squares which are smaller than a particular given side or we can subdivide this further into a smaller area.

We will see that also but right now what I am saying is if you just want to show it as a simple array then that will not be possible because if it is a uniform array, you will like your squares to be uniform. So if you want to have a non-uniform size of the square, that will obviously save some space for the area which is not there or the area which is totally inside. So there you can have a large size and then you can have a small size at the boundary. For this you have to keep track of the size of every, we will see methods for doing that also. So if I just keeping an uniform array and for every cell we say it is either a 0 or a 1 depending on whether it is outside or inside in that case we will like to have uniform size of the cells. And then we can enumerate each cell being either a 0 or a 1 starting from one corner and going up to the other end.

The disadvantage is obviously that if you want to have a very a high resolution, you end up using a very large amount space because we want the same size being used all the way throughout, the space required will become very large. And then if you are going to, if you want to extend this on to three dimensions it can become even more complex. The space required will be really very large for a three dimensional object. So in order to take care of that we use a method which is normally referred to as either a quadtree in two dimensions or an octree in three dimensions. So what we will do is in the next class we will see what is the quadtree and what is an octree and how they use for representing solids. We see that in the next class. Any question on what I have covered today, especially the last part spatial occupancy enumeration? In that case in the next class time, we will go on to quadtree and octree.