SOME 024:
Computer Aided Design

E. Rozos

Introduction to CAD theory part 2

Lesson structure

- Why Solid modelling?
- Solid modelling methods
  - Representation based
  - Manufacturing based
- Solid modelling storage database

Solid modelling

Wire frame modelling

+ Quick and accurate calculation of the objects position (applications including movements of very complicated objects)
– Lack of surface, volume, validity and ambiguity

Surface modelling

+ Good description of object surface properties and overall appearance (applications related with fluid dynamics)
– The objects are not solids

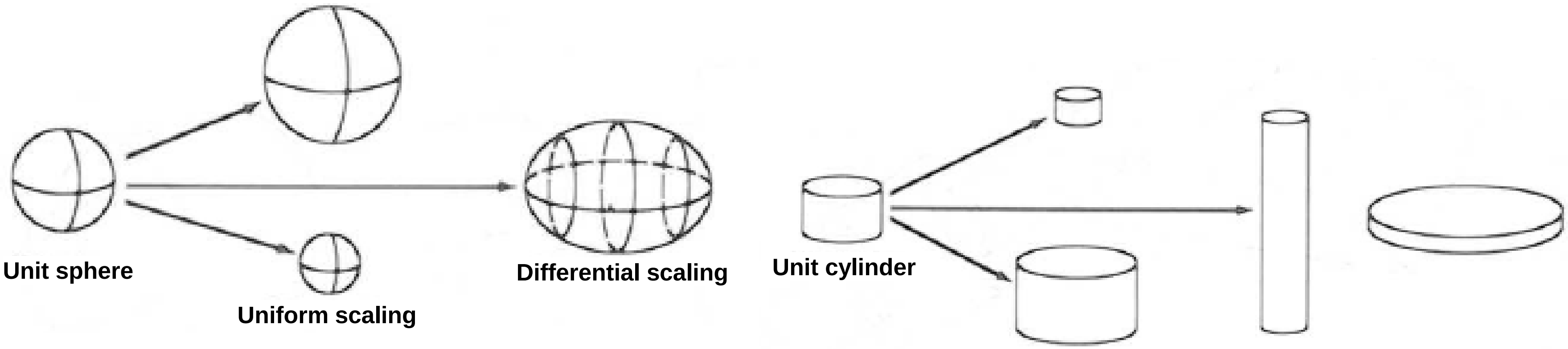
Solid modelling

+ Representation of both surface and interior properties of a solid (kinematics, stress analysis)
– Skilful and imaginative object drawing procedure

Representation based techniques

Parameterised primitive instancing

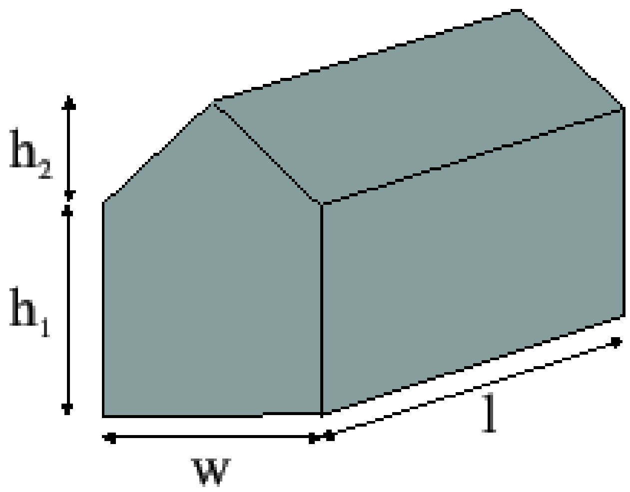
An object is specified by reference to a library of parameterized primitives.



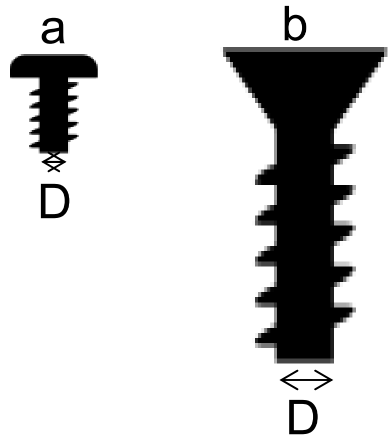
(R_1, R_2) (R, H)

Representation based techniques

Parameterised primitive instancing



Parameter vector P=(l, w, h_1, h_2)



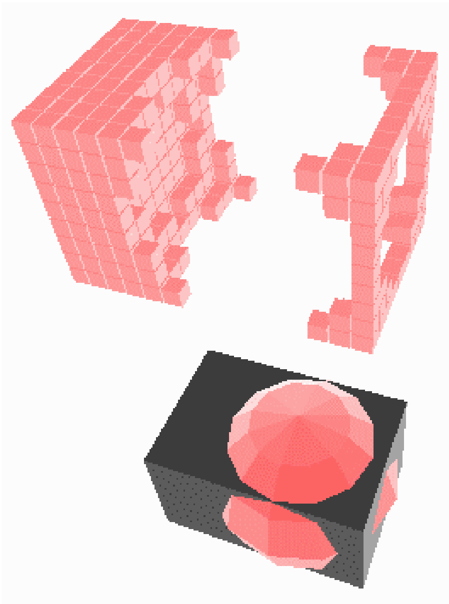
Parameter vector P=(Ht, Tp, D, L)

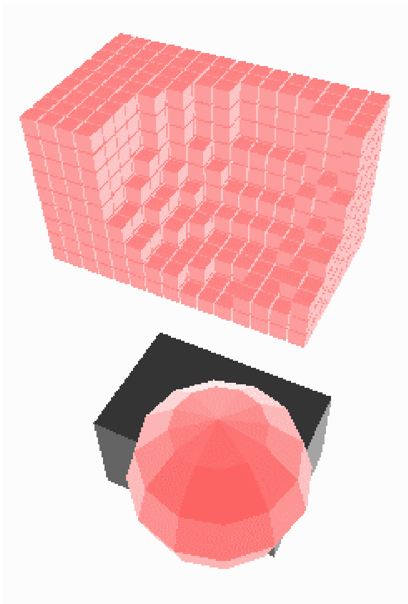
a) (1,1.8, 3, 20)
b) (2, 0.9, 9, 50)

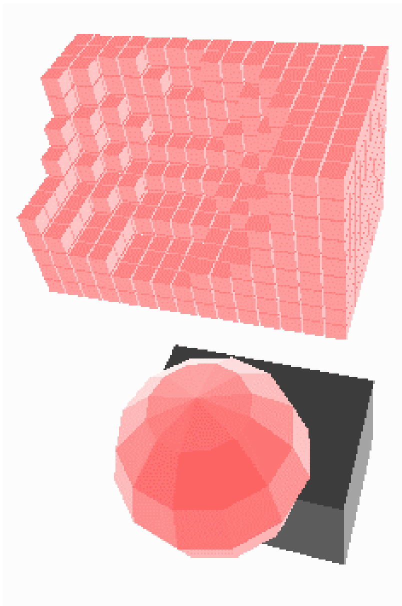
Representation based techniques

Spatial occupancy enumeration-voxel

The whole space is subdivided into regular cells (3D raster), and the object is specified by the set of cells it occupies. Models described this way are used in Finite Difference Analysis. This is usually done after a model is made, as part of automated pre-processing for analysis software.





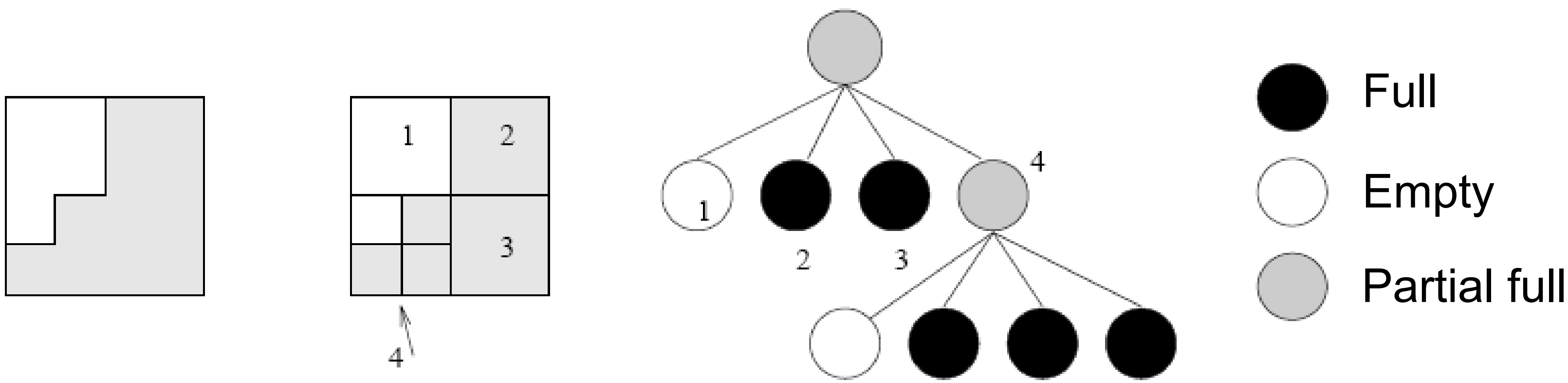


Representation based techniques

Spatial occupancy enumeration-octree

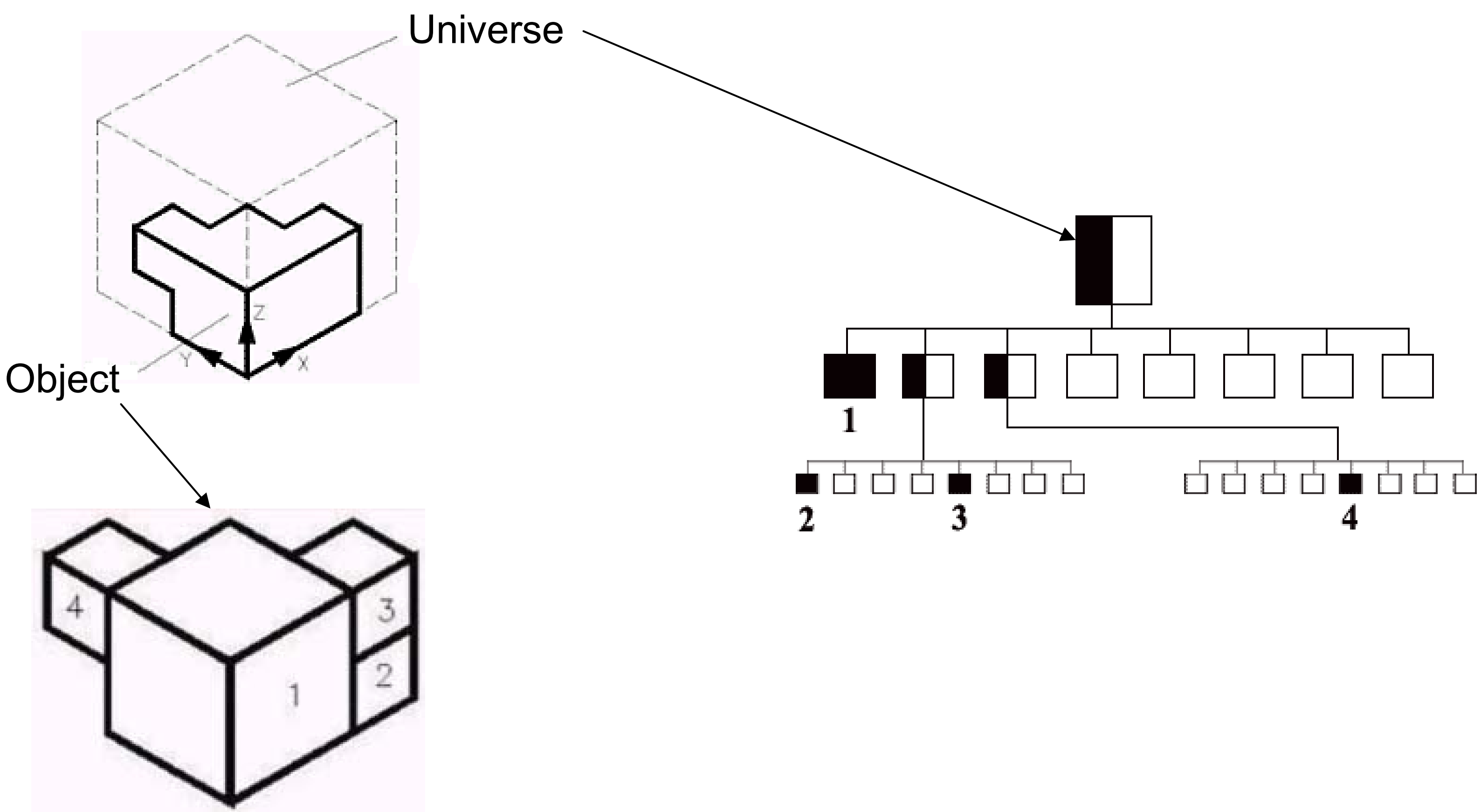
The whole space is represented by layers of nodes, each layer is twice denser (2×2×2=8) than the parent layer. Each node is the centre of a cube containing 8 sub-cubes. This tree data structure facilitates the local refinement of a 3D grid.

Quatree representation in 2D



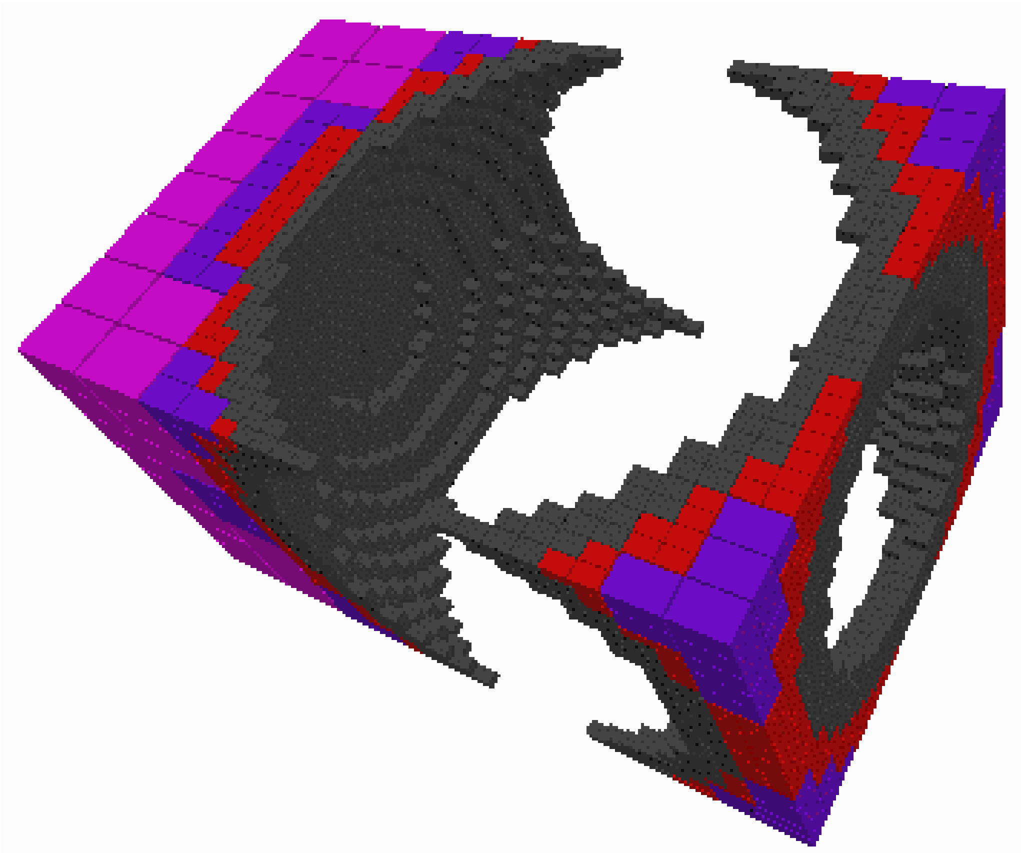
Representation based techniques

Spatial occupancy enumeration-octree



Representation based techniques

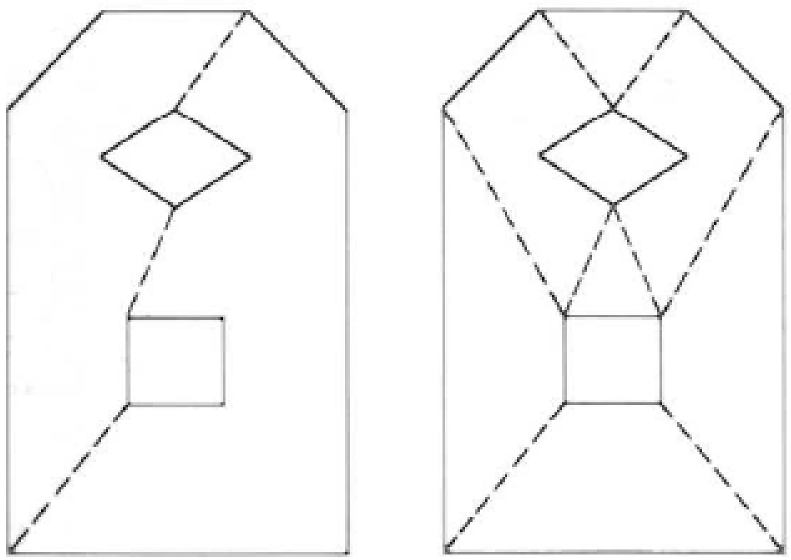
Spatial occupancy enumeration-octree



Representation based techniques

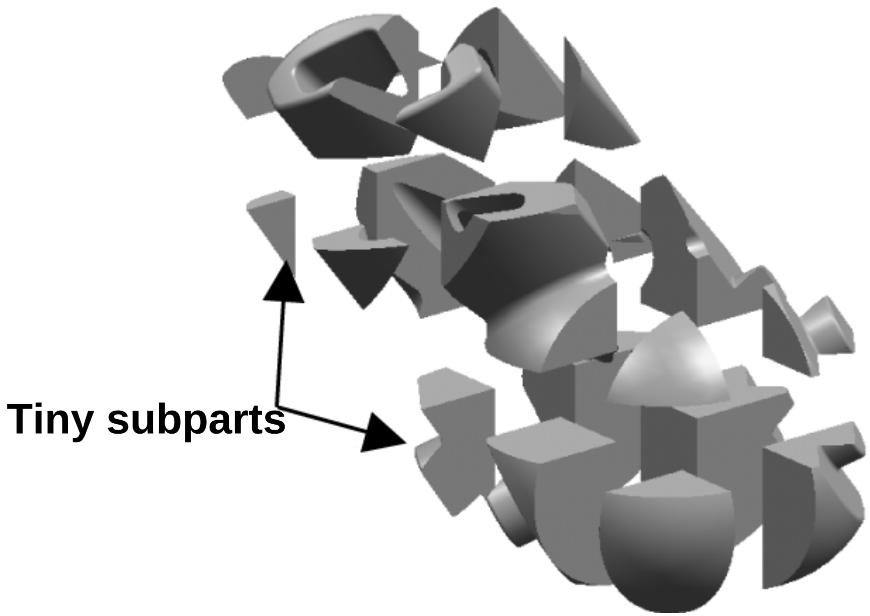
Cell decomposition

The object is subdivided into small volumes. Models described this way are used in Finite Elements Analysis. This is usually done after a model is made, as part of automated pre-processing for analysis software.



Representation based techniques

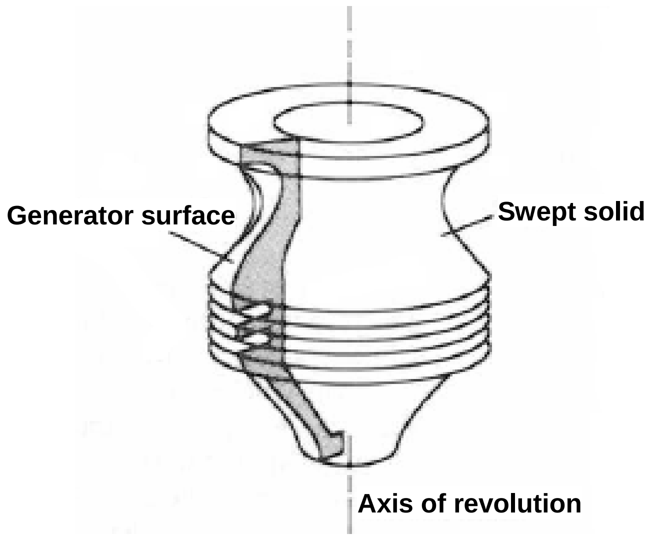
Cell decomposition



Manufacturing based techniques

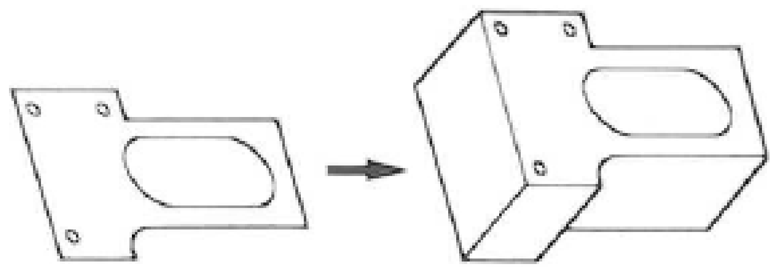
Sweeping

An area feature is "swept out" by moving a primitive along a path to form a solid feature. These volumes either add to the object ("extrusion") or remove material ("cutter path"). Analogous to various manufacturing techniques such as extrusion, milling, lathe and others.



Manufacturing: Lathe

Google SketchUp almost completely relies on extrusion to create solids.



Manufacturing: Extrusion

Manufacturing based techniques-CSG

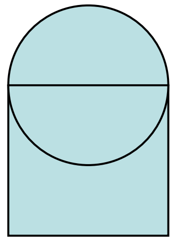
Constructive solid geometry (CSG)

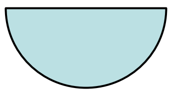
Complex surfaces or objects are create by using Boolean operators to combine primitive objects.

Boolean operators

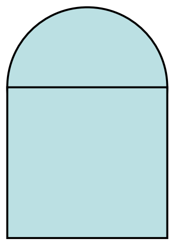
- Intersection (∩)
- Union (U)
- Difference (-)

Primitive objects

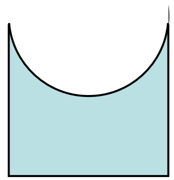




Intersection (∩)

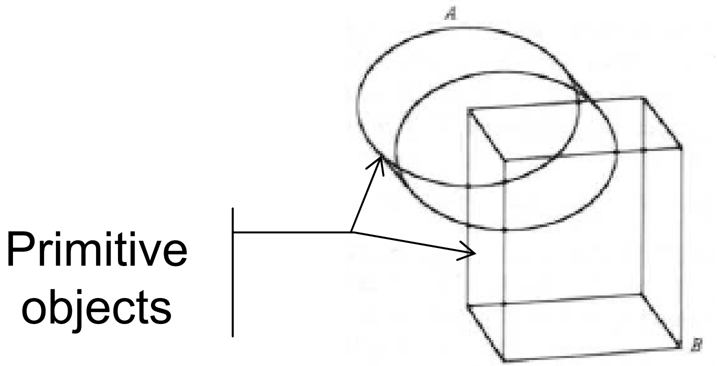


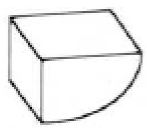
Union (U)

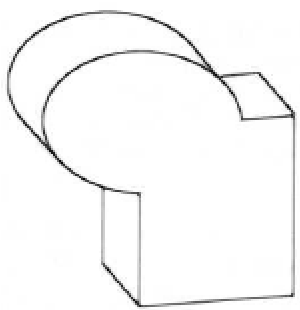


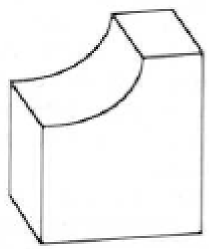
Difference (-)

Manufacturing based techniques-CSG







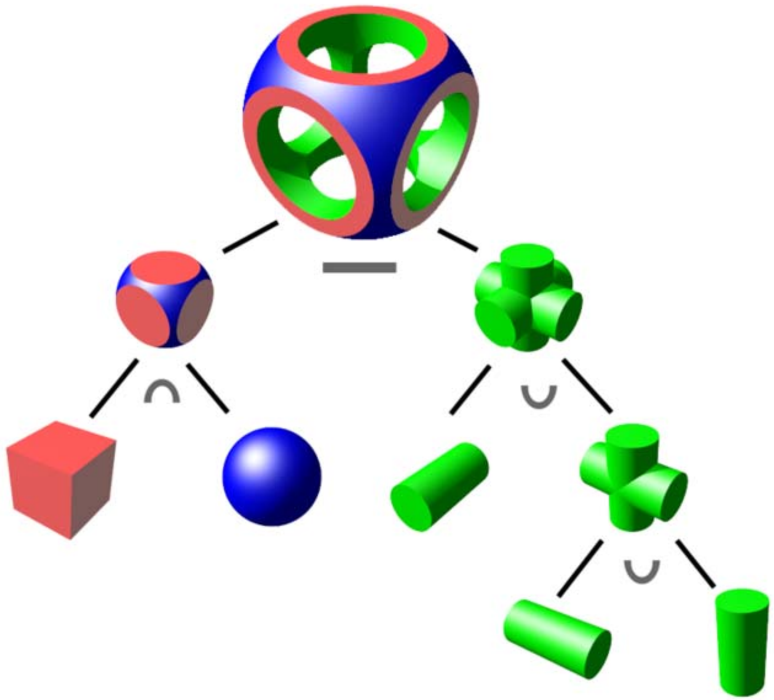


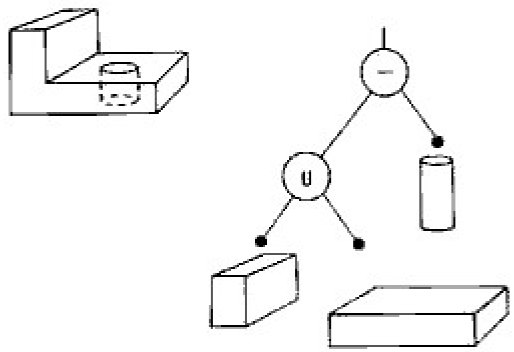
Intersection (∩)
Manufacturing: Cut

Union (U)
Manufacturing: Drill and weld

Difference (-)
Manufacturing: Drill

Manufacturing based techniques-CSG





The CSG method under normal conditions creates valid solids.

To determine the faces, edges or vertices the CGS tree must be evaluated (boundary evaluation).

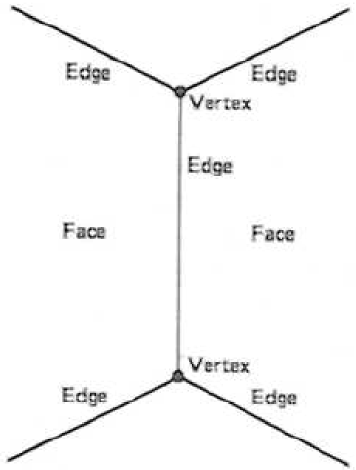
Manufacturing based techniques-BREP

Boundary representation (BREP)

A solid is represented as a collection of connected surface elements. Analogous to various manufacturing techniques: Injection moulding, casting, forging, thermoforming, etc.

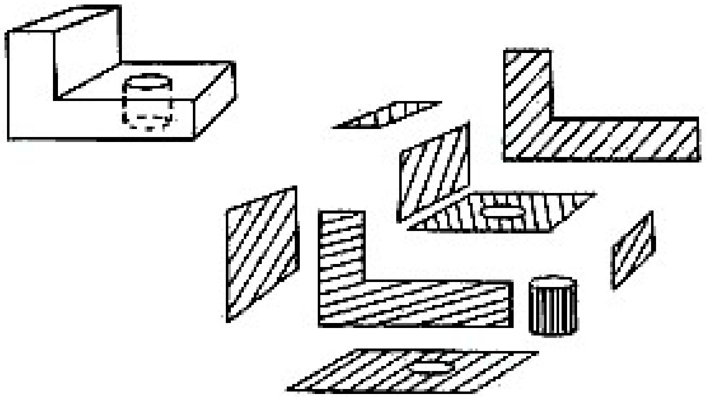
Topology is described by face-edge-vertex graphs. Geometric information is stored separately. The geometric elements are linked to the appropriate topologic elements.

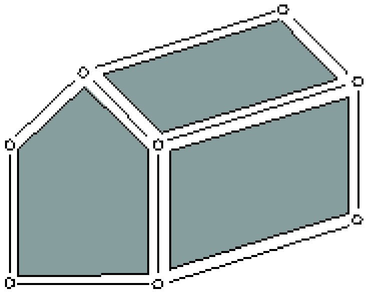
Topological element	Geometric element
Face	Surface
Edge	Curve
Vertex	Point

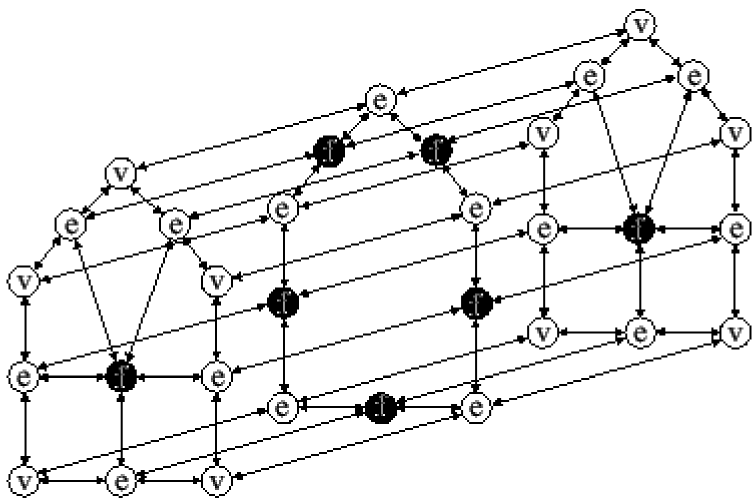


Manufacturing based techniques-BREP

Boundary representation (BREP)







Manufacturing based techniques

Euler-Poincare formula

The geometric data includes:

1. Coordinates for the vertices
2. Curve geometry for the edges
3. Surface geometry for the faces

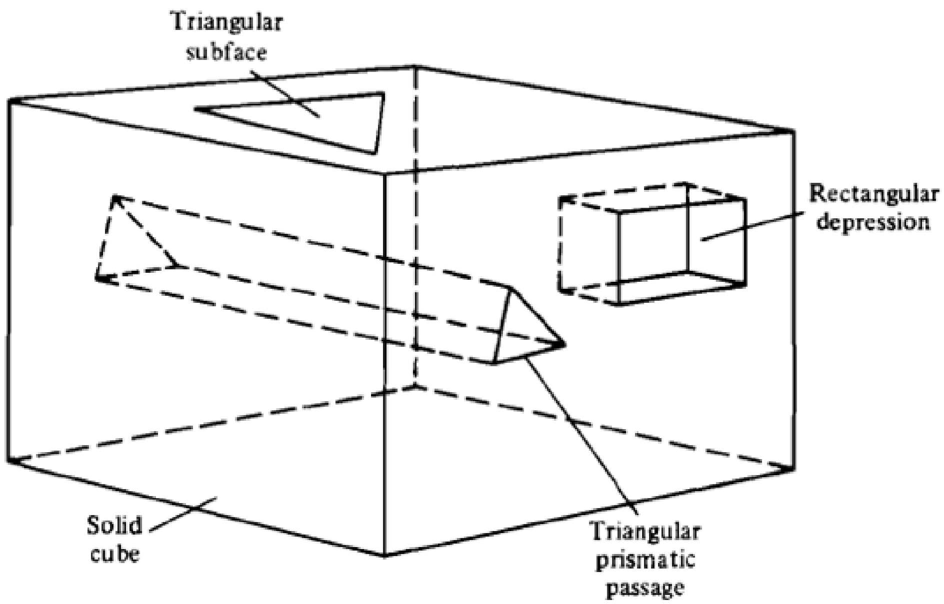
Euler-Poincare formula necessary but not sufficient condition for solid validity:
$V - E + F - H = 2(S - P)$

where F the number of faces, E the number of edges, V the number of vertices, H the number of hole loops in the object, S multiplicity of the object (number of disjoint pieces), P passageways.

Manufacturing based techniques

Euler-Poincare formula

Euler-Poincare formula ensures that an object is a valid polyhedron.



$$F - E + V - H = 2(S - P)$$

	F	E	V	H	S	P
Cube	6	12	8	0	1	0
Triangular passage	3	9	6	2	0	1
Rectangular depression	5	12	8	1	0	0
Triangular subface	1	3	3	1	0	0
Total	15	36	25	4	1	1

Manufacturing based techniques

Euler operations

Euler-Poincare formula defines the permissible steps in constructing a solid

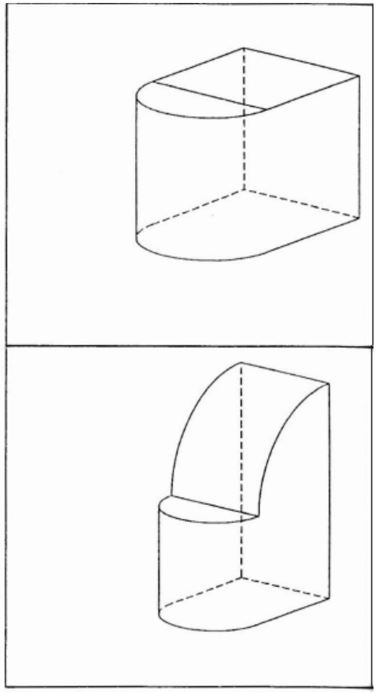
Remains to be defined

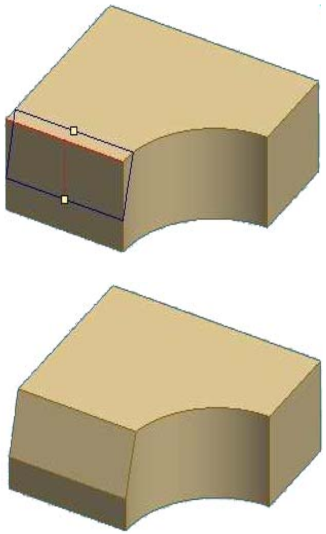
Defined in each step

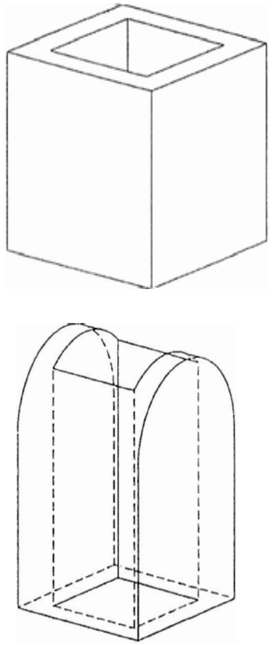
		V	E	F	H	2S	2P	
		4	6	4	0	2	0	
1.	mbfv	1	0	1	0	2	0	Make a body, a face and a vertex
		3	6	3	0	0	0	
2.	mev	1	1	0	0	0	0	Make an edge and a vertex
3.	mev	1	1	0	0	0	0	Make an edge and a vertex
4.	mev	1	1	0	0	0	0	Make an edge and a vertex
		0	3	3	0	0	0	
5.	mfe	0	1	1	0	0	0	Make a face and an edge
6.	mfe	0	1	1	0	0	0	Make a face and an edge
7.	mfe	0	1	1	0	0	0	Make a face and an edge
		0	0	0	0	0	0	

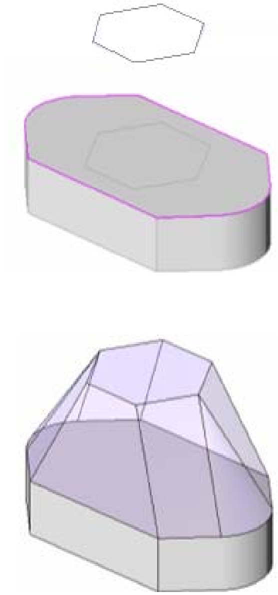
Manufacturing based techniques

Adjustments to an existing shape.









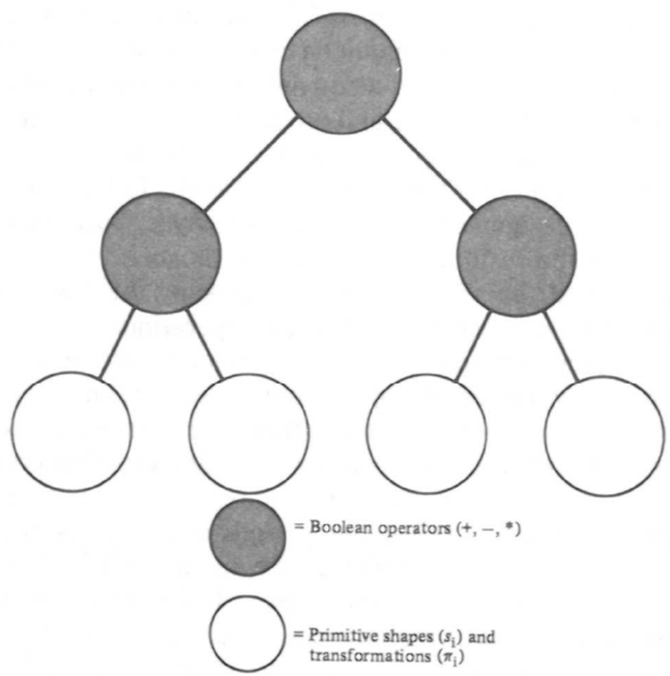
Swinging Chamfering Tweaking Blending

Solid modeller storage database

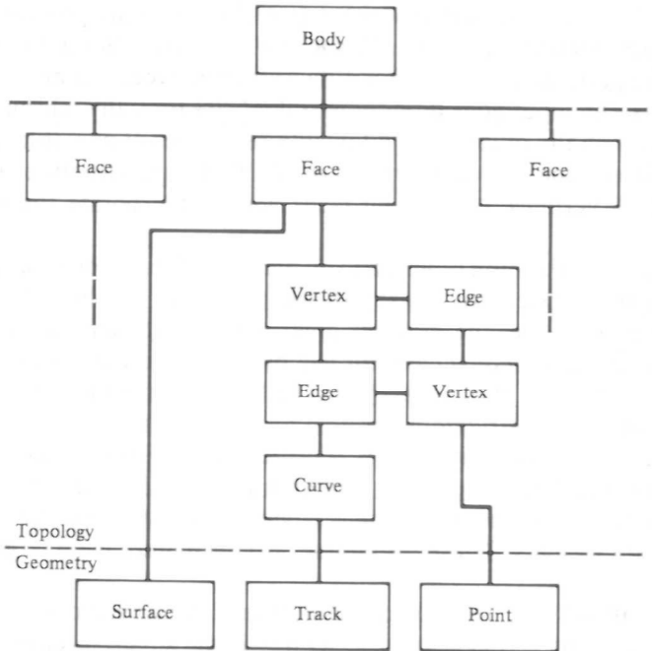
Major forms of storage:

1. CSG stores the part as a tree, where the leaves are primitives and the Interior nodes are Boolean operators i.e. CSG stores the instructions to make a part.
2. BREP stores the part as an explicit description of the solid bounds i.e. BREP stores the actual part.

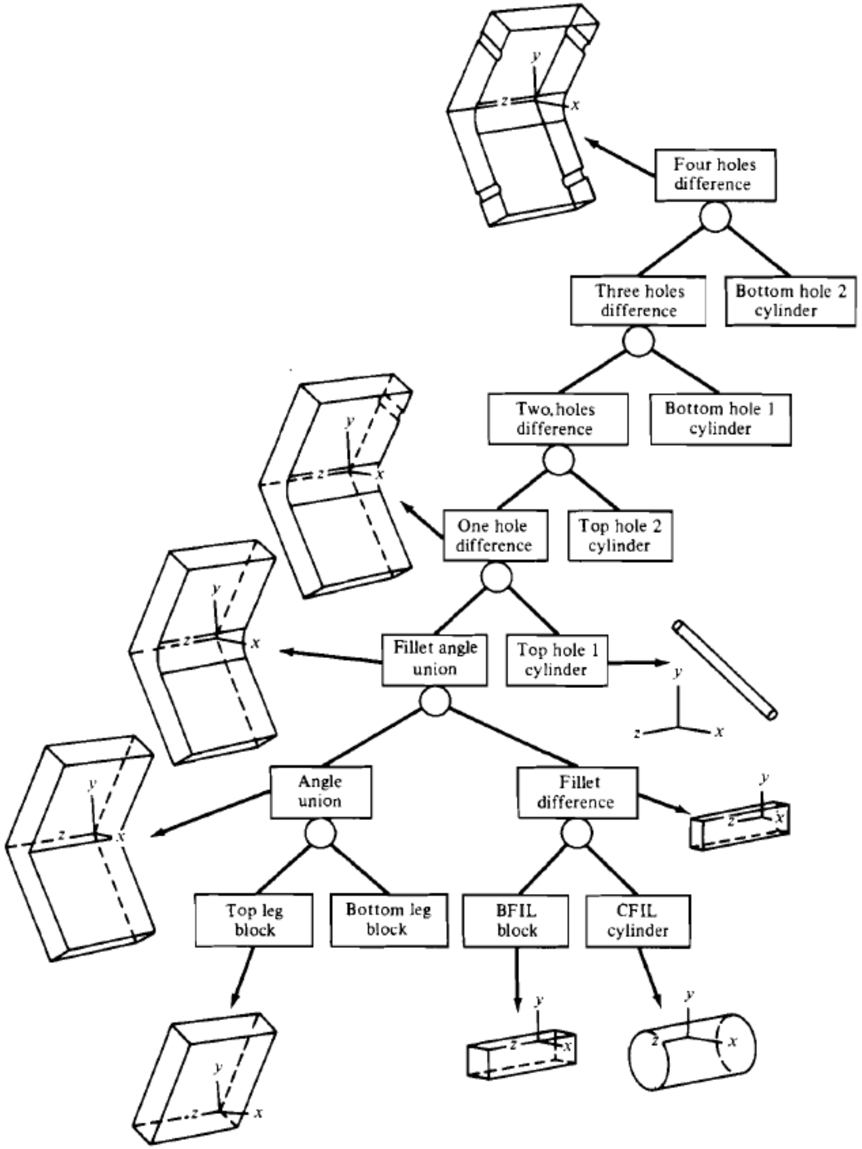
CSG



BREP



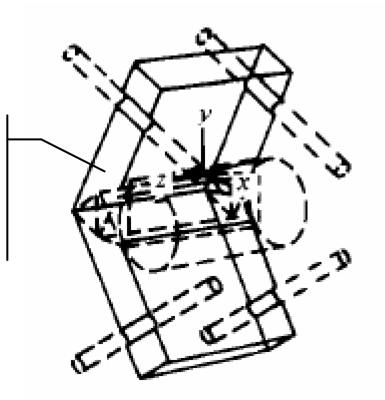
Solid modeller storage database CSG



The tree of CSG:

+ smaller than BREP
– the tree evaluation is more computational demanding
– can store only models created with CSG method

Final object and objects used in Boolean operations

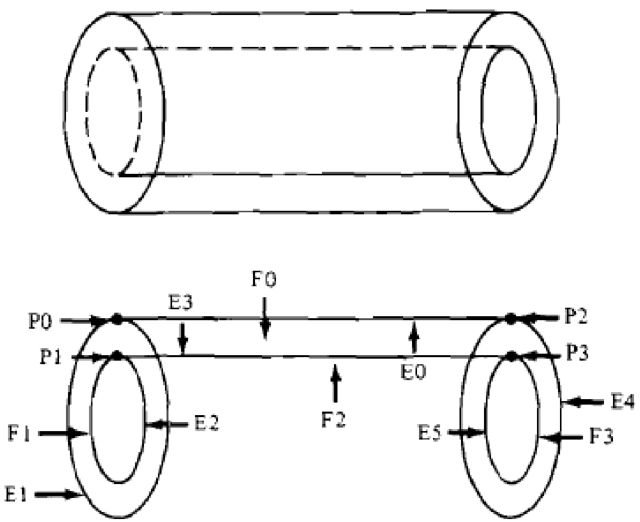


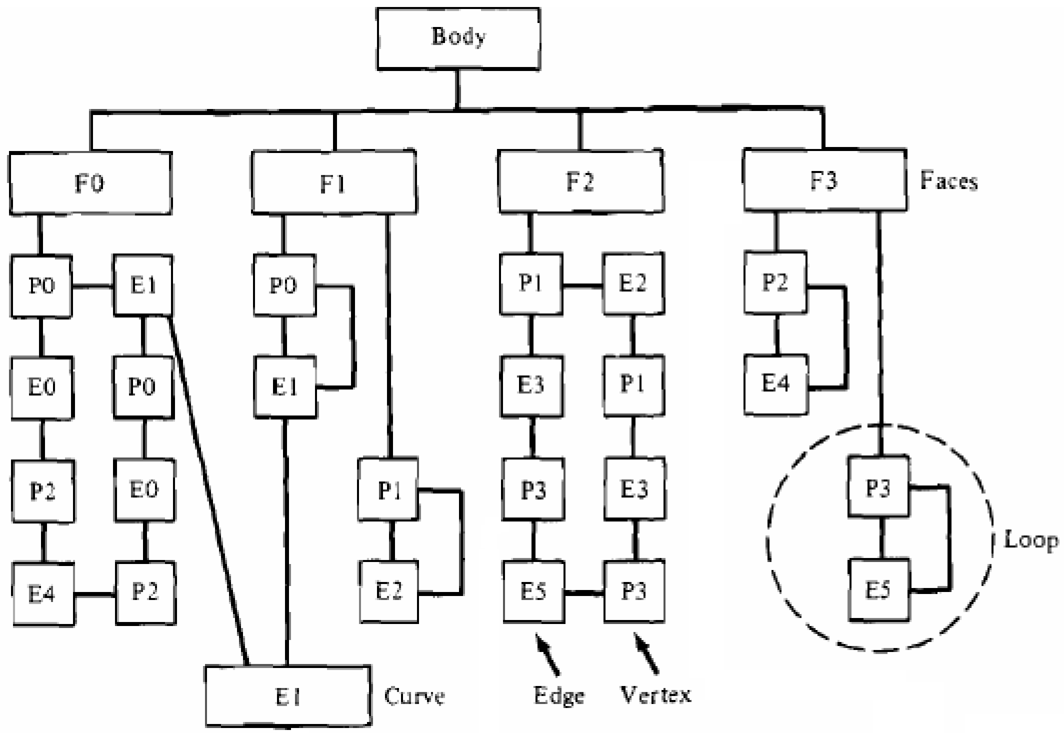
Solid modeller storage database BREP

The tree of BREP is:

+ larger than the CGS.

– no need for any extra computation to query the object.





Solid modeller storage database BREP

The Winged-Edge Data Structure

Object is described with 3 tables.

1. the edges table holds the topology
2. the vertex table holds the geometric information of points
3. the face table holds the geometric information of surfaces

The **edges** table defines for each edge:

- its vertices (start-end)
- its left and right faces
- the predecessor and successor of this edge when traversing its left face
- the predecessor and successor of this edge when traversing its right face

The **vertex** table defines:

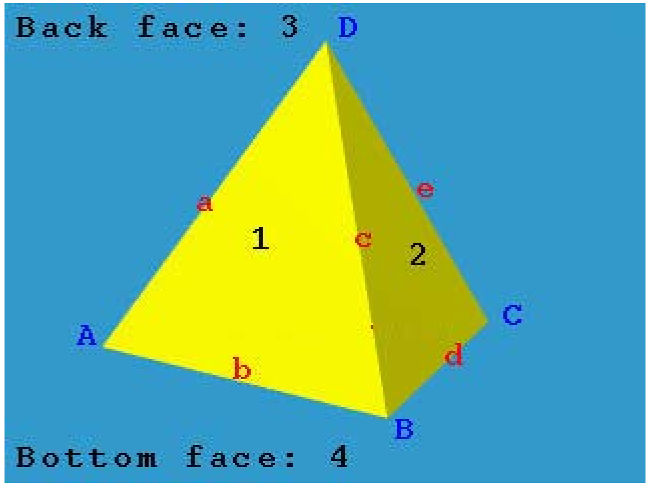
1. vertices geometry
2. one edge connected to that vertex

The **face** table defines:

1. surfaces geometry
2. one edge connected to that vertex

Solid modeller storage database BREP

The Winged-Edge Data Structure



Edges table

Edge	Vertices		Faces		Left Traverse		Right Traverse	
Name	Start	End	Left	Right	Pred	Succ	Pred	Succ
a	A	D	3	1	e	f	b	c
b	A	B	1	4	c	a	f	d
c	B	D	1	2	a	b	d	e
d	B	C	2	4	e	c	b	f
e	?	?	?	?	?	?	?	?
f	?	?	?	?	?	?	?	?

Vertex table

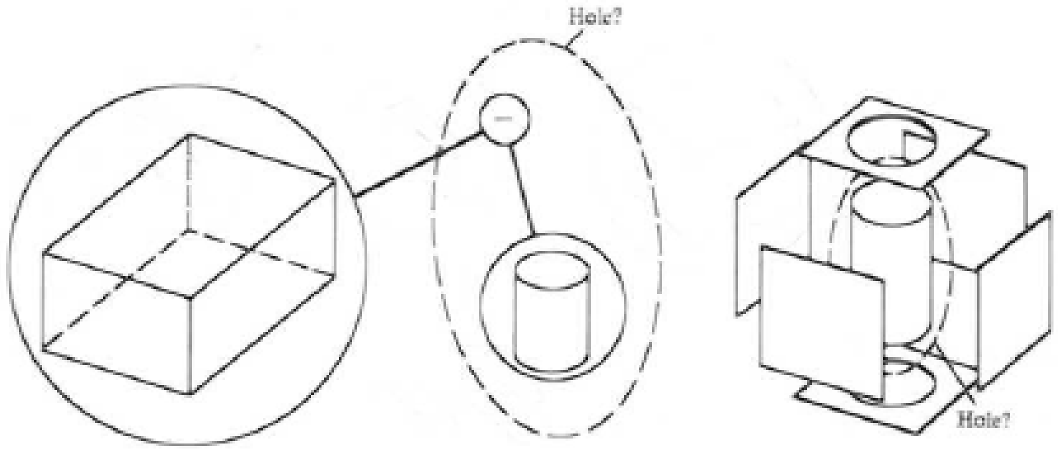
Vertex Name	Incident Edge
A	a
B	b
C	d
D	e

Face table

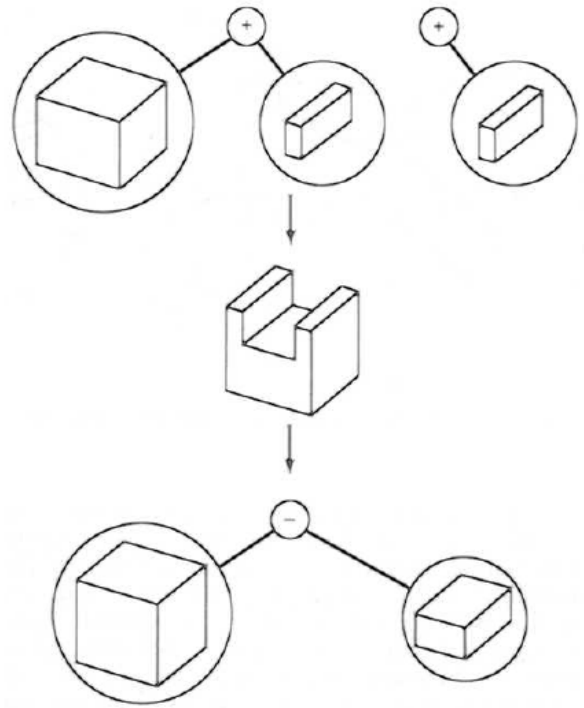
Face Name	Incident Edge
1	a
2	c
3	a
4	b

Feature recognition

The figures 2.33 and 2.34 of the Computer-Integrated Design and Manufacturing should go here



Feature recognition



Feature recognition

Not practical in CSG because of

- Non unique feature construction
- Need of complete tree examination to ensure no overlap

BREP is the structure of choice. Example pseudo code that searches the stored data to find a hole.

hole(Facelist) IF

> *entrance_face(Face1) AND*
>
> *valid_hole_face(Face2) AND*
>
> *adjacent(Face1, Face2) AND*
>
> *valid_hole_face(Face2) AND*
>
> *more_hole(Face2, Facelist)*