S.S.D. ING-IND/15

Disegno e Metodi dell’Ingegneria Industriale

VIRTUAL PROTOTYPING

LECTURE 2

Geometric Modeling:

3D Object Modeling & Knowledge Based Modeling

Giuseppe Di Gironimo

University of Naples Federico II

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

Fraunhofer Joint-Lab

ideas

INTERACTIVE DESIGN AND ERGONOMICS APPLICATIONS IN VIRTUAL REALITY

ideainvr

Introduction

Modeling techniques are classified according to the type of ***representation*** and ***data structure***.

- **Representation** is the set of methods used to describe the geometry.

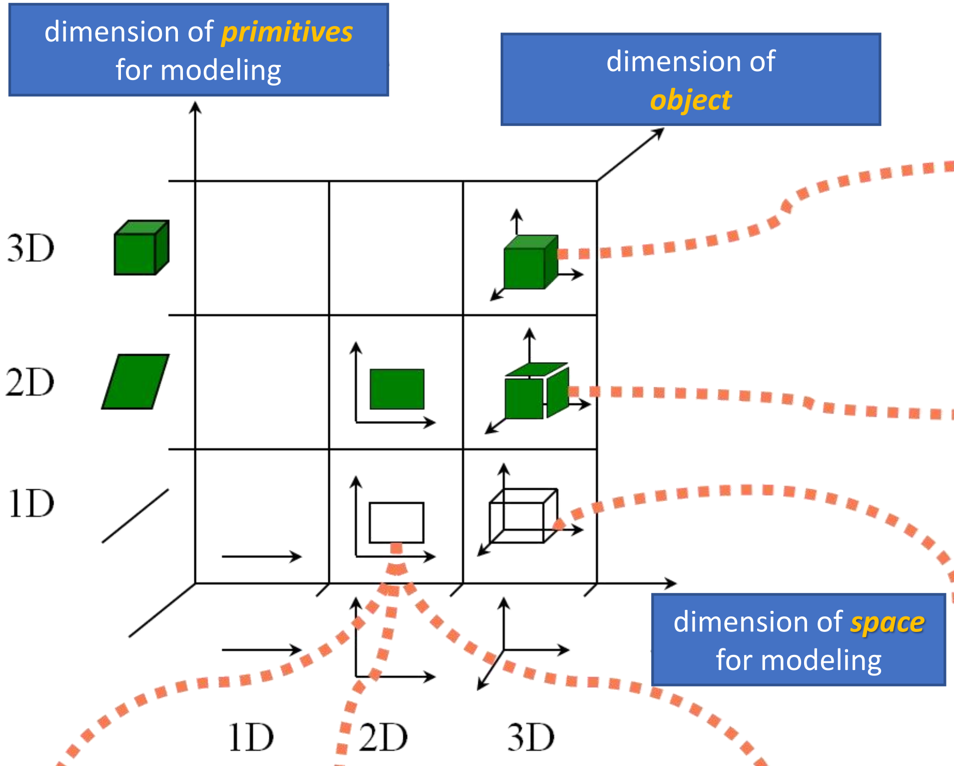
- **Data structure** is how model information are stored in computer memory and on file.

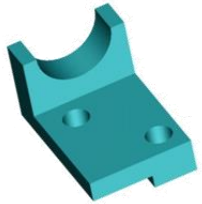
The type of representation and its data structure are a characterizing element of CAD software as it affects:

Potentiality (domain of possible geometries, modeling functions)

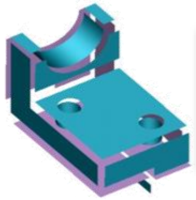
Performance (speed of execution of functions)

3D Object Modeling

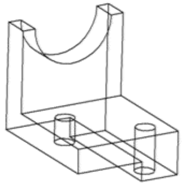




CSG (Constructive Solid Geometry)
Solid primitives combined with procedural methods

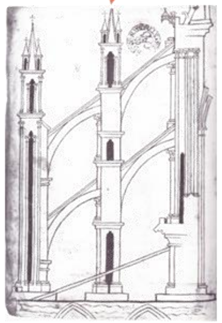


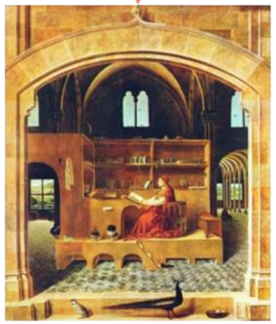
B-REP (Boundary Representation)
Representation of the external surface of the solid object

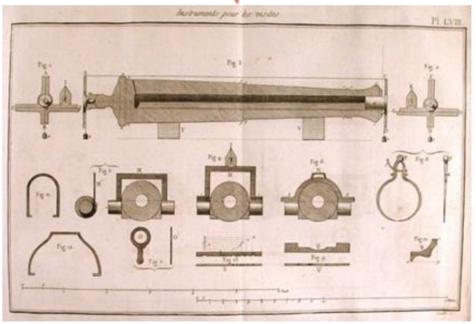


WIREFRAME
Representation of the solid object through its edges

Paintings:
- No depth
- Perspective

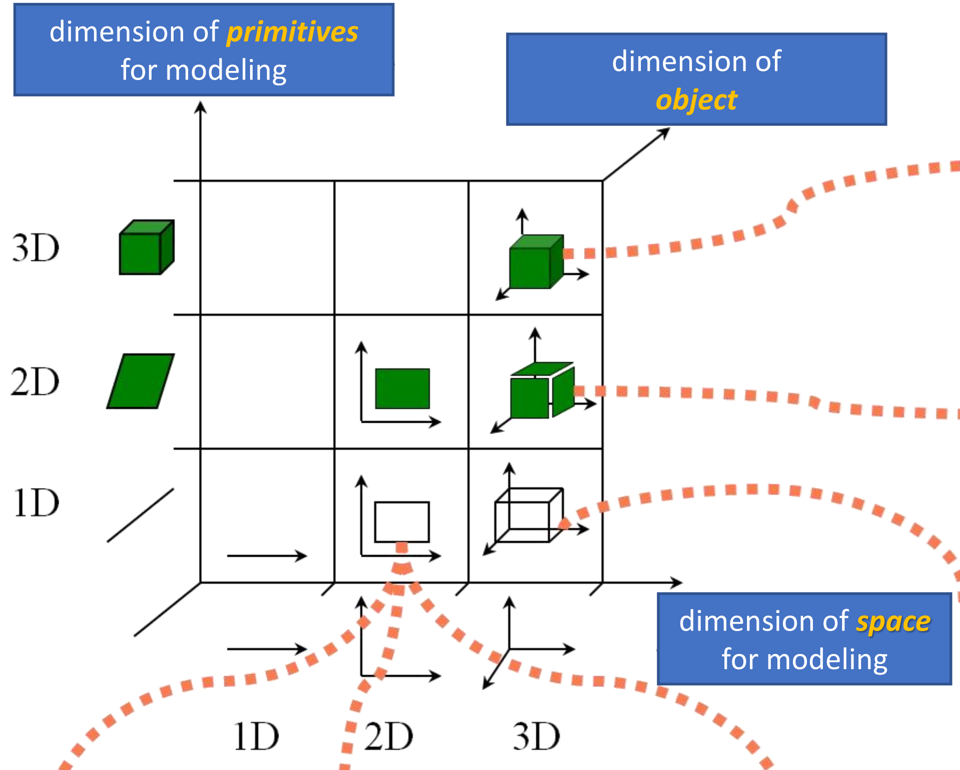


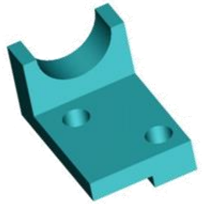




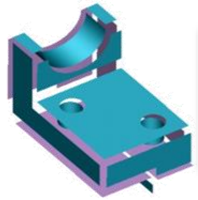
Drafting

3D Object Modeling

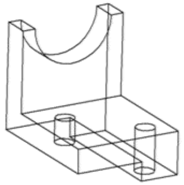




CSG (Constructive Solid Geometry)
Solid primitives combined with procedural methods

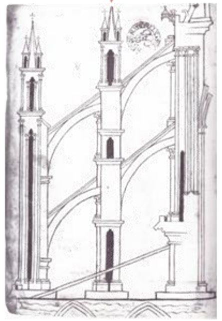


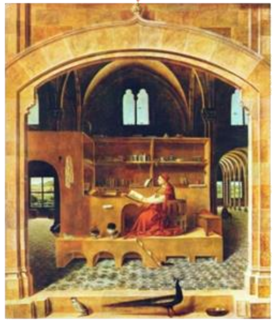
B-REP (Boundary Representation)
Representation of the external surface of the solid object

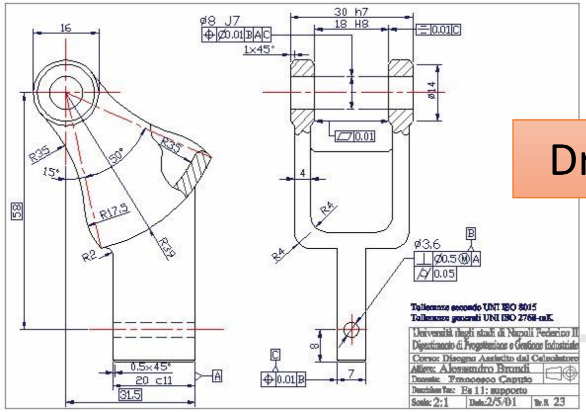


WIREFRAME
Representation of the solid object through its edges

Paintings:
- No depth
- Perspective

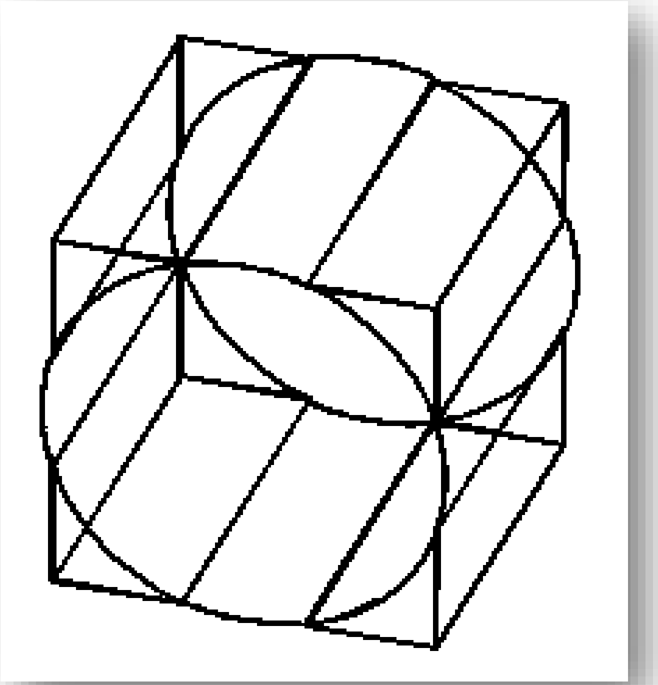






Drafting

Wireframe



Pro

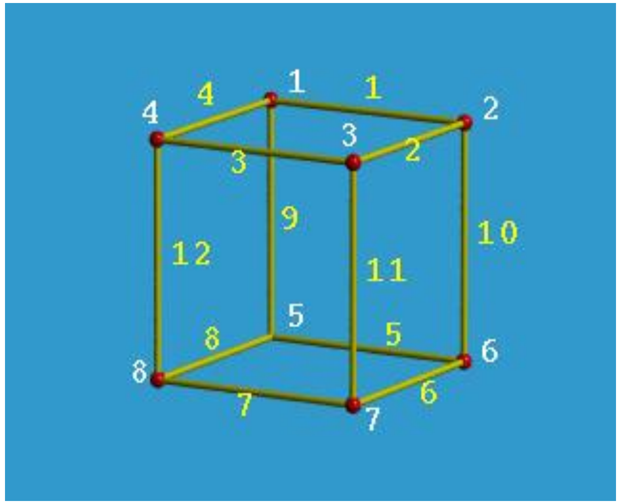
- They require few computational and graphical resources
- Easily implementable

Cons

- "Poor geometry" (only visible and / or hidden edges).
- Complicated to read and interpret
- There is no explicit description of the boundary surfaces of the models.
- They cannot be interrogated on areas and volumes.
- Ambiguity in representation and in the transition from 3D to 2D.

Wireframe

An object is specified by two tables: (1) Vertex Table (2) Edge Table.



Vertex	X	Y	Z
1	1	1	1
2	1	-1	1
3	-1	-1	1
4	-1	1	1
5	1	1	-1
6	1	-1	-1
7	-1	-1	-1
8	-1	1	-1

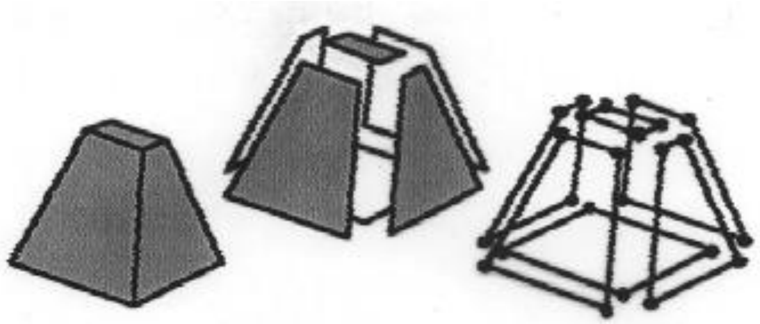
Edge	Start Vertex	End Vertex
1	1	2
2	2	3
3	3	4
4	4	1
5	5	6
6	6	7
7	7	8
8	8	5
9	1	5
10	2	6
11	3	7
12	4	8

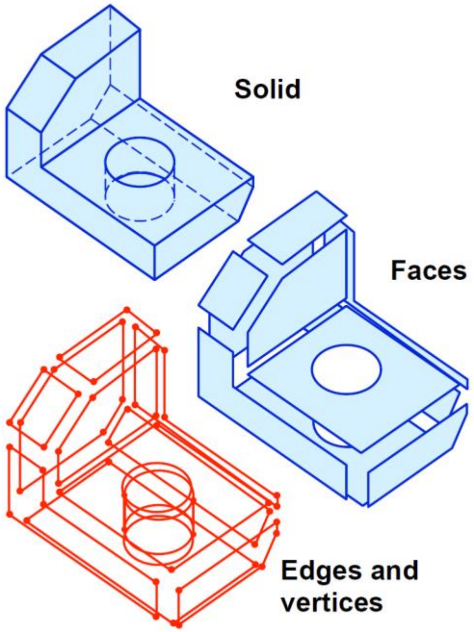
The vertex table consists of three-dimensional coordinate values for each vertex with reference to the origin.

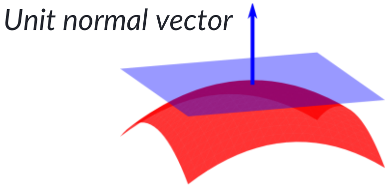
Edge table specifies the start and end vertices for each edge.

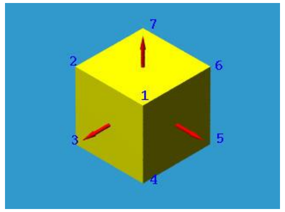
Boundary Representation

- The solid is defined indirectly through its limiting surfaces (faces) and their contiguity relations to each other
- Faces are represented as assemblies with edges and vertices
- The position of material is defined by the use of oriented geometry





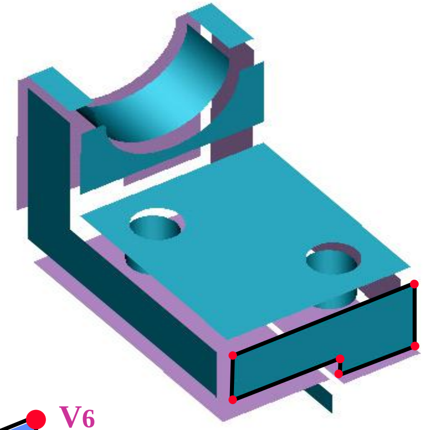




A normal to a surface at a point is the same as a normal to the tangent plane to the surface at the same point.

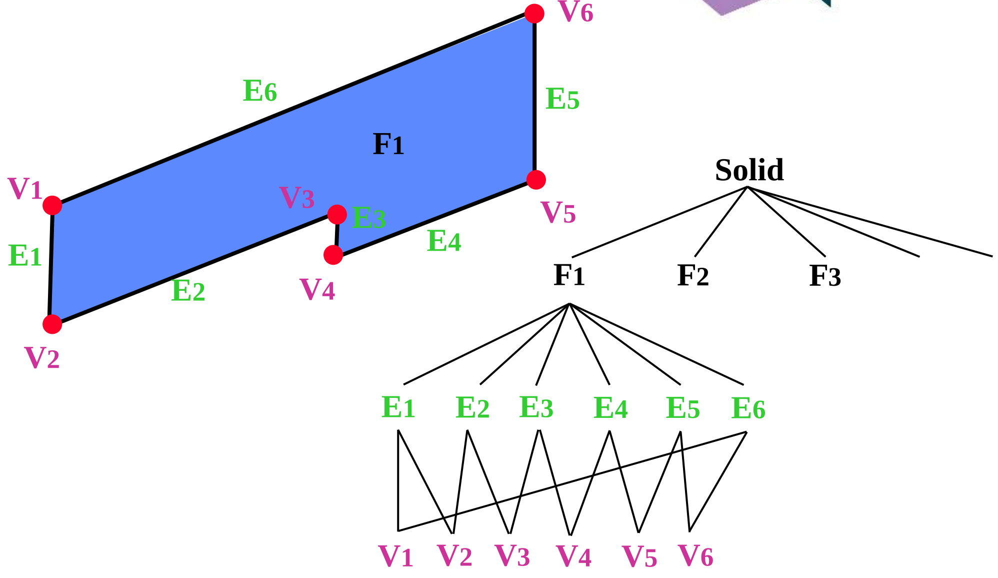
Topological and geometrical entities

The **geometry** describes, by means of parameters, the shape, the size and the position of the objects: *coordinates of the vertices, radius, lengths and angles*, etc.



The **topology** describes the connections between the various geometric entities: *Vertices, edges and faces* are topological entities

- Geometrical Entities
  - Point (X,Y,Z)
  - Line (X,Y,Z) Start (X,Y,Z) End
  - Circle (X,Y,Z) Center & Diameter
  - Arc (X,Y,Z) Center , Radius, Start & End Angle or Point
- Topological entities
  - Vertex (edge1, edge2, etc..) – Point geom.
  - Edge (vertex1, vertex2) – Curve geom.
  - Face (edge1, edge2, etc) – surface geom.
  - Solid (Face1, Face2, etc.)



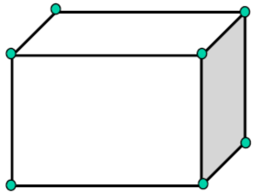
Topological entities	SOLID	SHELL	FACE	LOOP	EDGE	VERTEX
Geometrical entities			SURFACE		CURVE	POINT

Topological consistency: The Eulero-Poincaré Formula

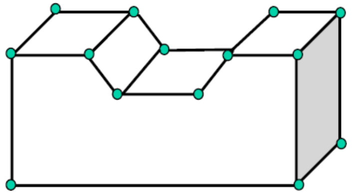
$$V - E + F = 2(S - H) + R$$

Leonhard Euler (1707 -1783) defined the fundamental rules governing the topology of solid bodies.

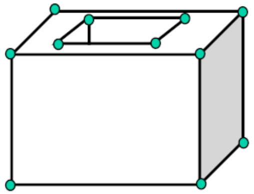
V: number of vertices
E: number of edges
F: number of faces
H: number of holes that penetrate the solid
S: number of shells (the solid is counted as a shell, so always S ≥ 1)
R: number of rings (closed loops on faces)



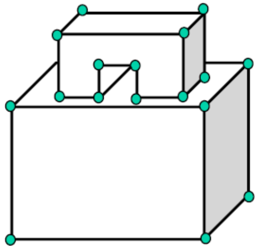
V=8, E=12, F=6, S=1, H=0, R=0



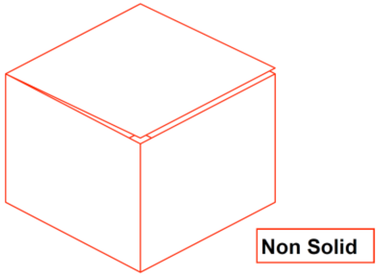
V=16, E=24, F=10, S=1, H=0, R=0



V=16, E=24, F=10, S=1, H=1, R=2



V=24, E=36, F=14, S=1, H=1, R=2



V=10, E=15, F=6, S=1, H=0, R=0
V-E+F=2(S-H)+R
10-15+6=2(1-0)+0
1≠2

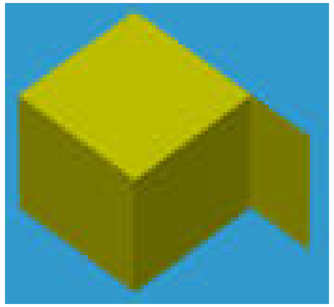
Topological consistency: The Eulero-Poincaré Formula

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

V: number of vertices
E: number of edges
F: number of faces
H: number of holes that penetrate the solid
S: number of shells (the solid is counted as a shell, so always S ≥ 1)
R: number of rings (closed loops on faces)

Leonhard Euler (1707 -1783) defined the fundamental rules governing the topology of solid bodies.

The Euler-Poincaré formula is a ***necessary but not sufficient condition***.
If the relationship is true, this does not mean that the solid is valid.

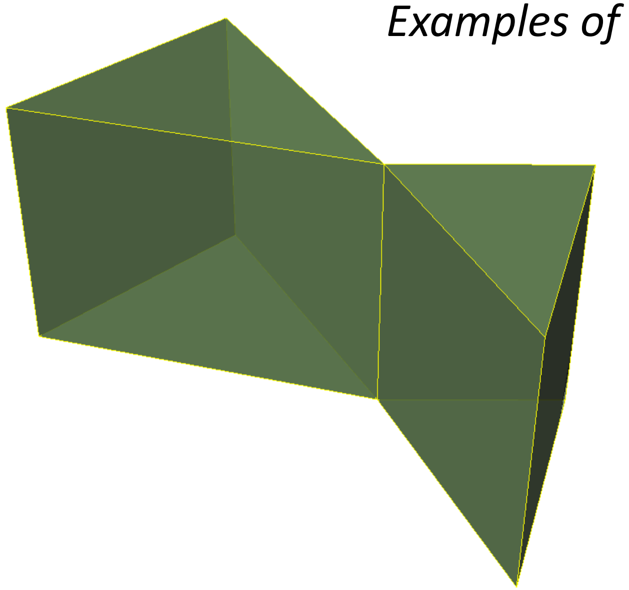


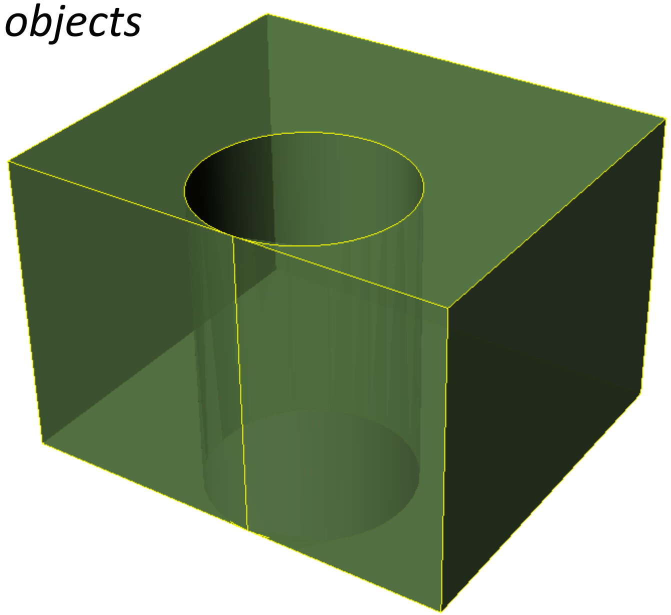
V=10, E=15, F=7, S=1, H=0, R=0
V-E+F=2(S-H)+R
10-15+7=2(1-0)+0
2=2

MANIFOLD and NON-MANIFOLD objects

In manifold objects, each boundary unambiguously separates an inner region from an outer one
In manifold objects, edges can only belong to two faces

Examples of non-manifold objects

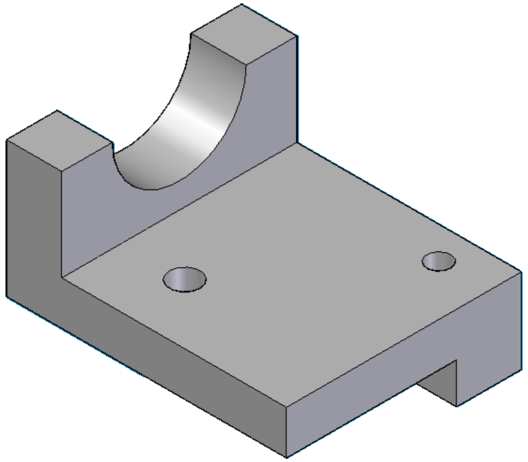


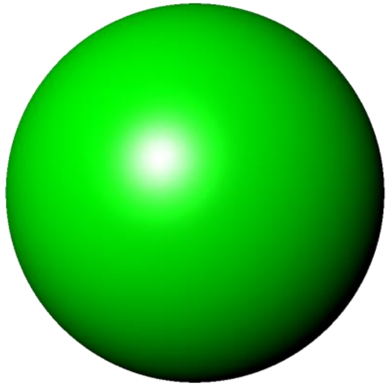


Precise vs approximated geometry

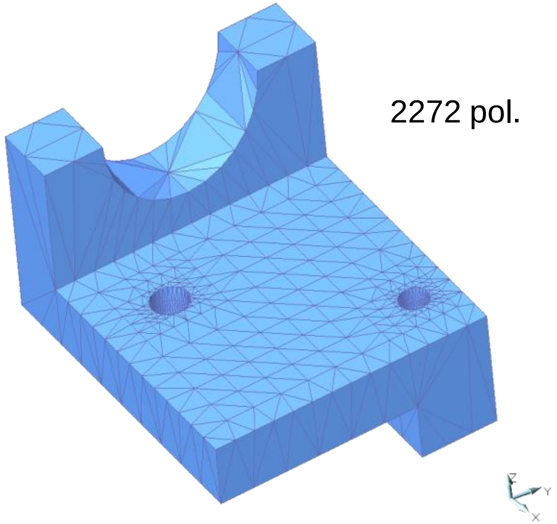
A B-rep model is a precise representation of an object. A simplified representation (always based on "surface" information) can be obtained with a "faceting" process that creates a polygon mesh.

Precise geometry

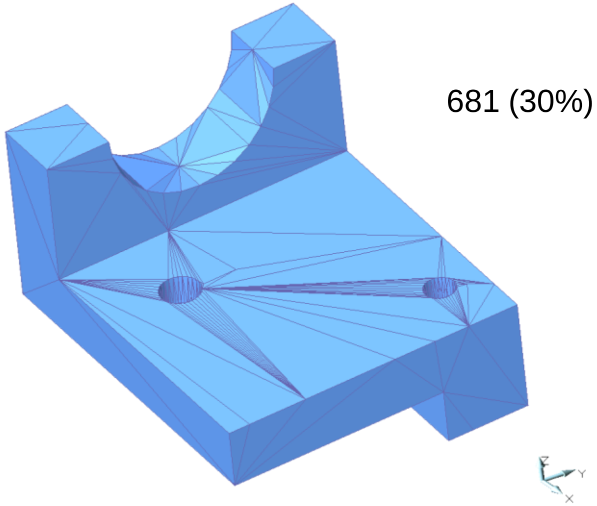




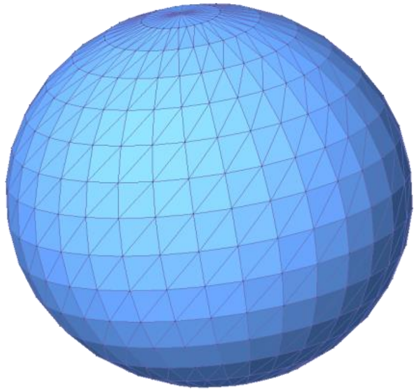
Approximated geometry



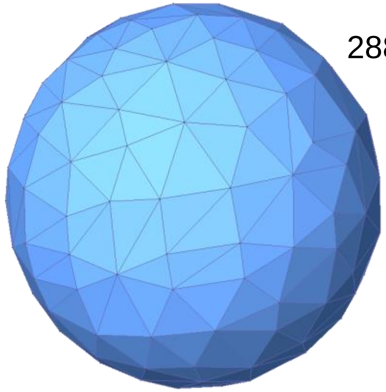
2272 pol.



681 (30%)



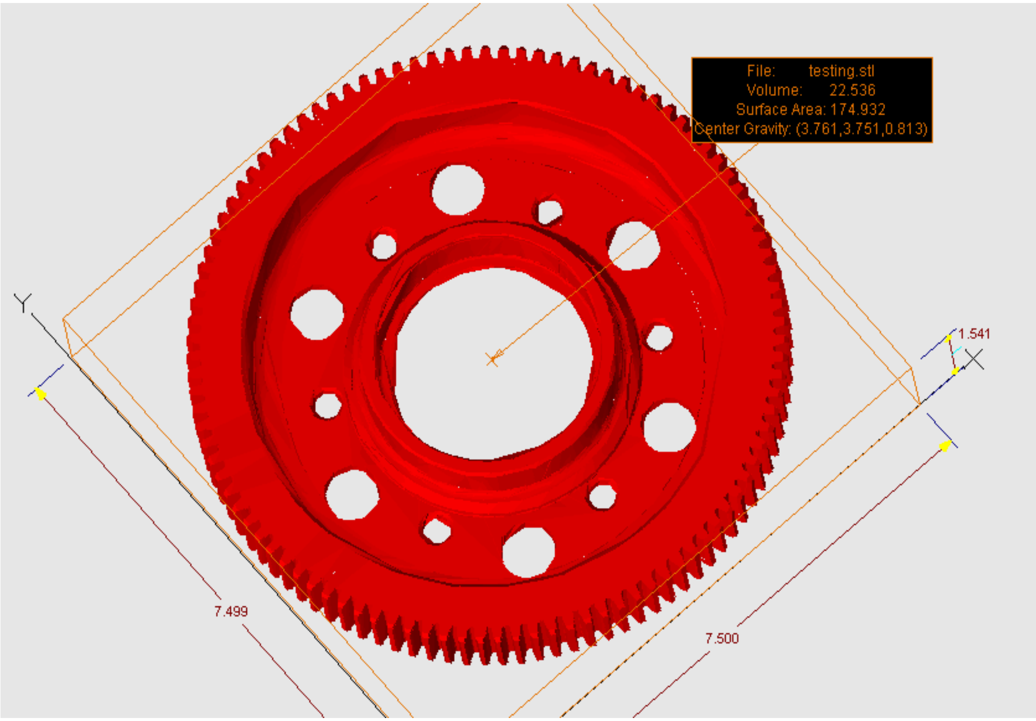
960 pol.



288 (30%)

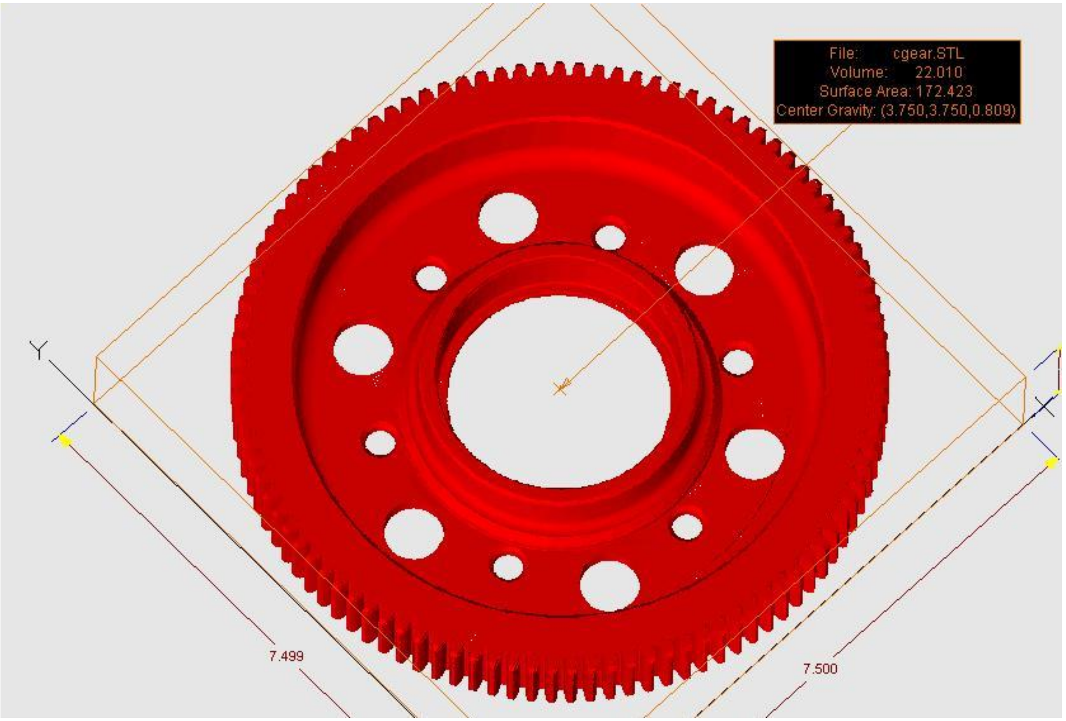
Precise vs approximated geometry

The file size grows more than linearly with the number of triangles !!



Error = 0.01 mm

Triangles: 38,000
File Size: 1.9 MB



Error = 0.001 mm

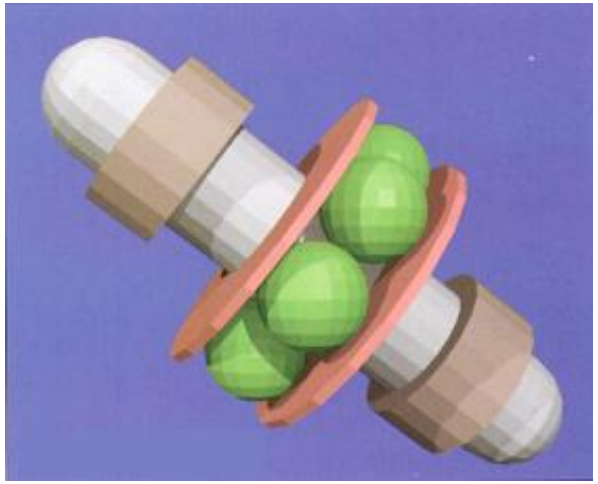
Triangles: 195,000
File Size: 19.5 MB

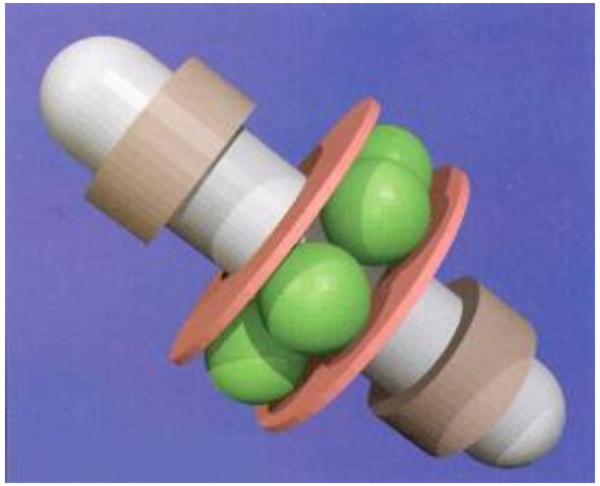
DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

Level of Detail - LoD

Less detail

More detail





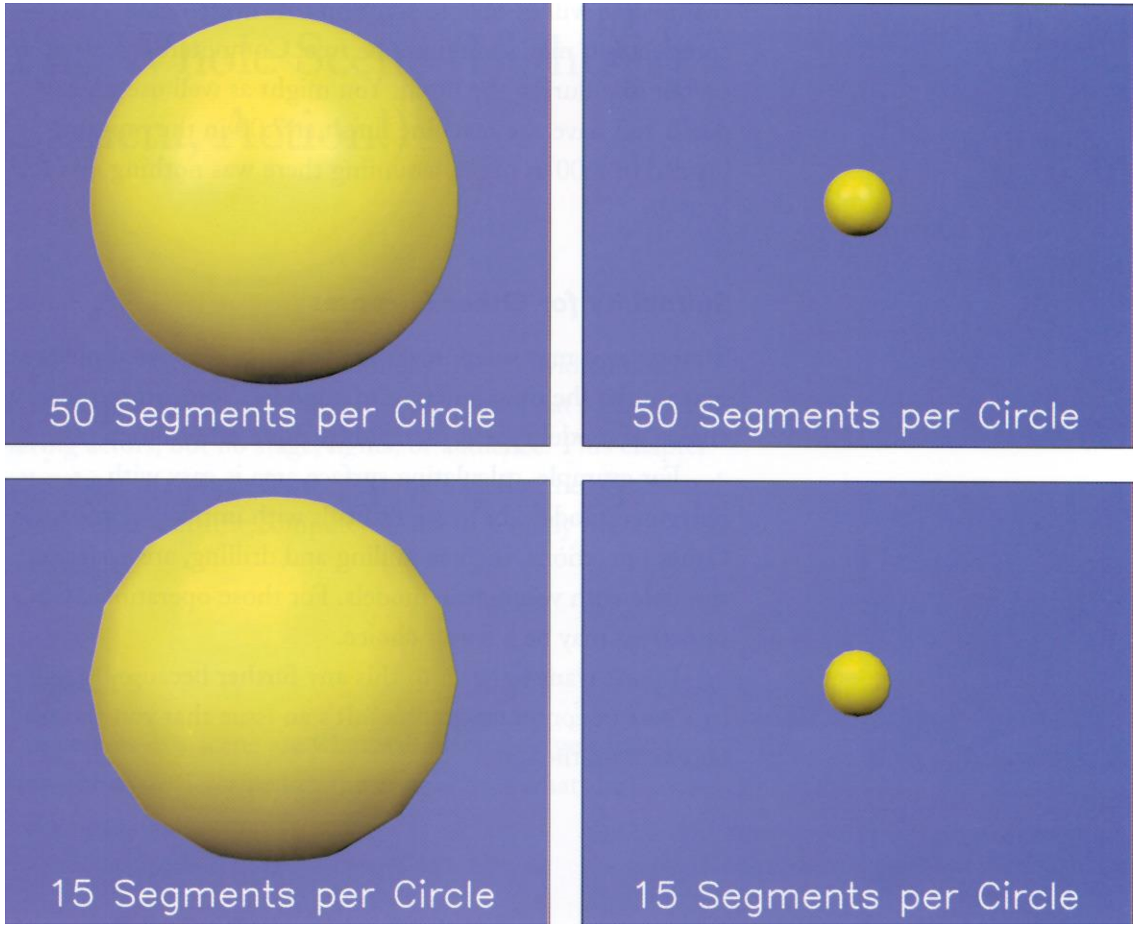
But also:

- + time for rendering
- + memory space

Level of Detail - LoD

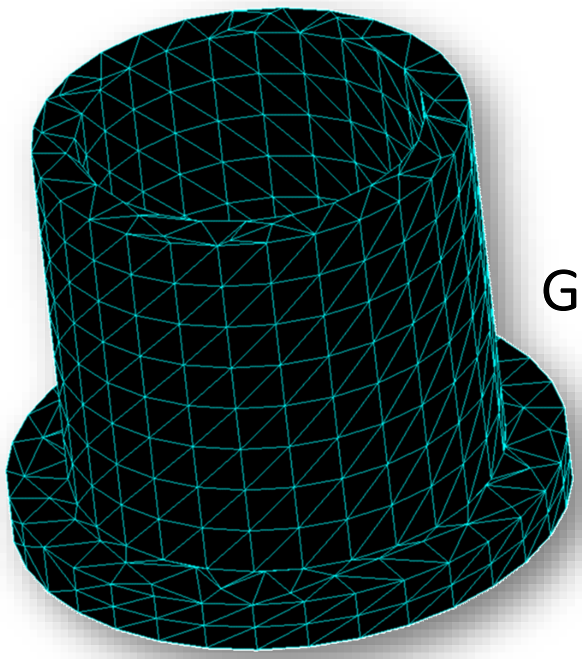
Distant or small objects on the screen require less detail

More detail | Less detail

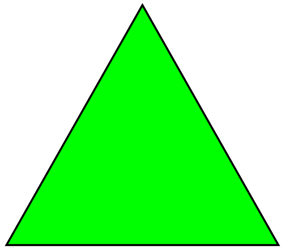


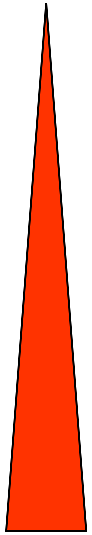
Quality of tessellation

Nearly equilateral triangles indicate an optimized tessellation.

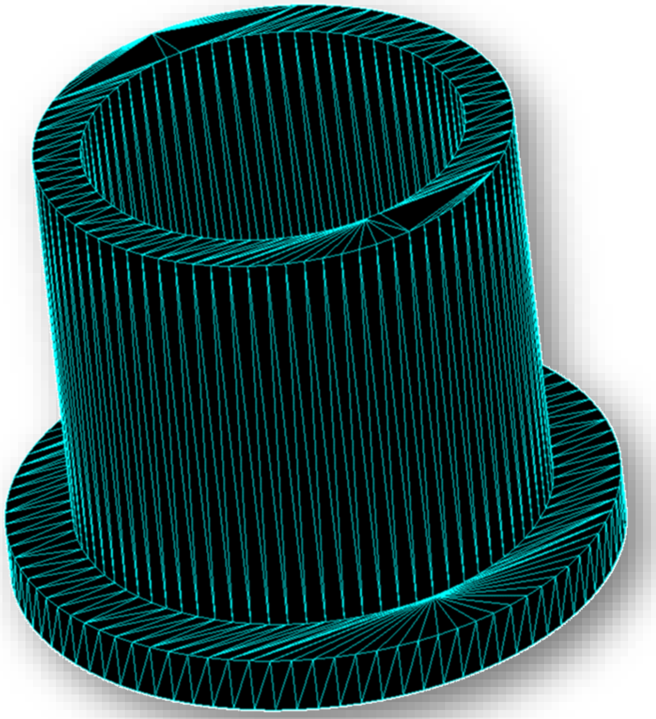


Good!





No Good!



Classification of tessellation

Based on the type of data structure:

Without connectivity
(e.g. reverse engineering or 3D scanning models)

- *For each polygon (triangles) the coordinates of the vertices and the normal (towards the outside) are stored*
- *The resulting data structure is very simple, but not compact (many vertices are repeated thousands of times !!)*
- *Universal as it is used by the standard .STL (Stereolitography by 3D labs) - ASCII or binary*

With connectivity (shared vertices)
usually result from direct CAD modeling.

- *Objects are represented with vertices and their connectivity.*
- *This information is stored in tables:*
  - *Vertex table*
  - *Edge table*
  - *Polygon table*
- *Tables are saved in memory and on file.*

Classification of tessellation

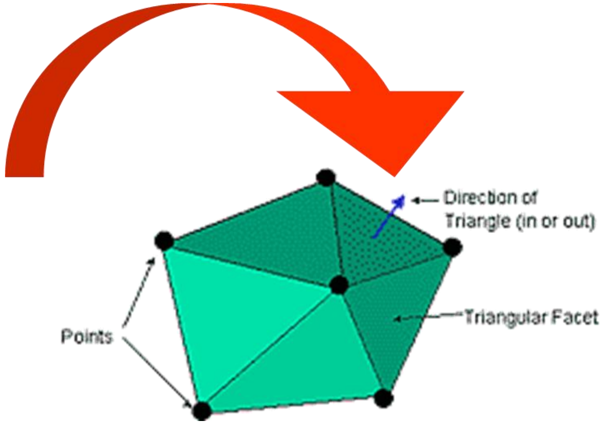
Based on the type of data structure:

Without connectivity
(e.g. reverse engineering or 3D scanning models)

- *For each polygon (triangles) the coordinates of the vertices and the normal (towards the outside) are stored*
- *The resulting data structure is very simple, but not compact (many vertices are repeated thousands of times !!)*
- *Universal as it is used by the standard .STL (Stereolitography by 3D labs) - ASCII or binary*

STL standard

```
solid <part name>
 facet normal -1 0 0
   outer loop
     vertex 1 1 1
     vertex 1 1 2
     vertex 1 2 1
   endloop
    endfacet
 facet normal 0 -1 0
   outer loop
     vertex …
   endloop
 endfacet
endsolid <part name>
```



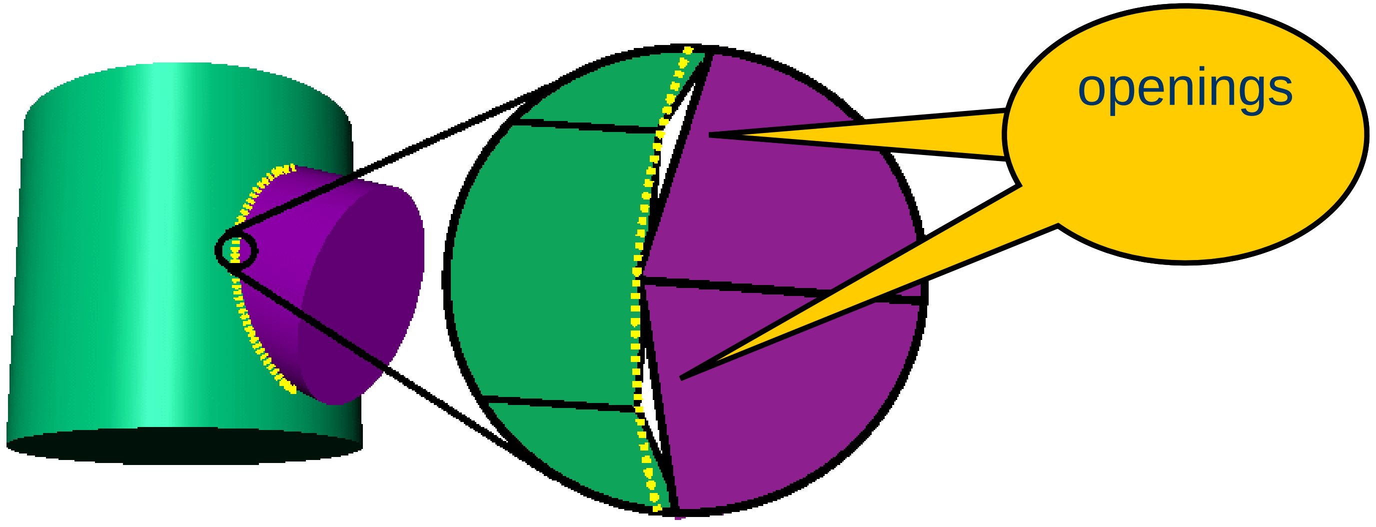
Repeated for each triangle -> large files !!

Classification of tessellation

Based on the type of data structure:

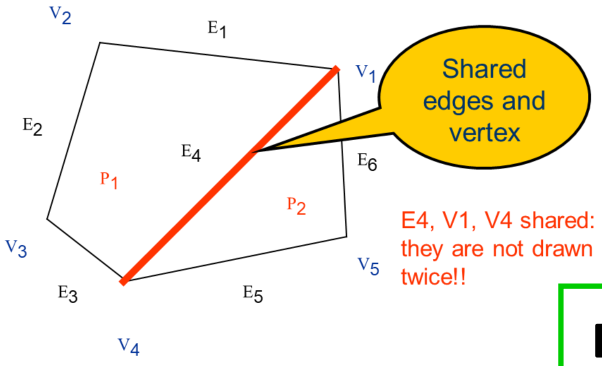
Without connectivity
(e.g. reverse engineering or 3D scanning models)

- *For each polygon (triangles) the coordinates of the vertices and the normal (towards the outside) are stored*
- *The resulting data structure is very simple, but not compact (many vertices are repeated thousands of times !!)*
- *Universal as it is used by the standard .STL (Stereolitography by 3D labs) - ASCII or binary*



Classification of tessellation

Based on the type of data structure:



Vertex Table

$V_1 : x_1, y_1, z_1$

$V_2 : x_2, y_2, z_2$

$V_3 : x_3, y_3, z_3$

$V_4 : x_4, y_4, z_4$

$V_5 : x_5, y_5, z_5$

Edge Table

$E_1 : V_1, V_2$

$E_2 : V_2, V_3$

$E_3 : V_3, V_4$

$E_4 : V_4, V_1$

$E_5 : V_4, V_5$

$E_6 : V_5, V_1$

Polygon table

$P_1 : E_1, E_2, E_3, E_4$

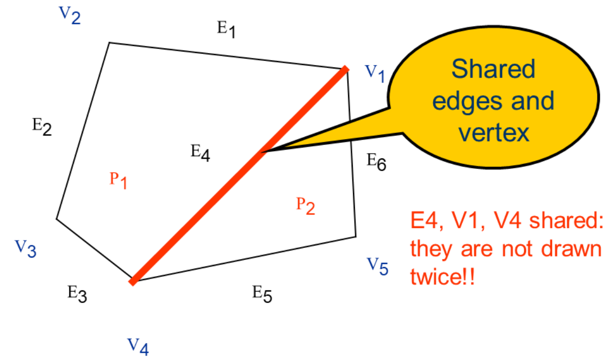
$P_2 : E_4, E_5, E_6$

With connectivity (shared vertices) usually result from direct CAD modeling.

- *Objects are represented with vertices and their connectivity.*
- *This information is stored in tables:*
  - *Vertex table*
  - *Edge table*
  - *Polygon table*
- *Tables are saved in memory and on file.*

Classification of tessellation

Based on the type of data structure:



- The vertices must belong to the end point of at least 2 edges
- Each edge must belong to at least one polygon
- All polygons are closed
- All polygons have at least one shared edge

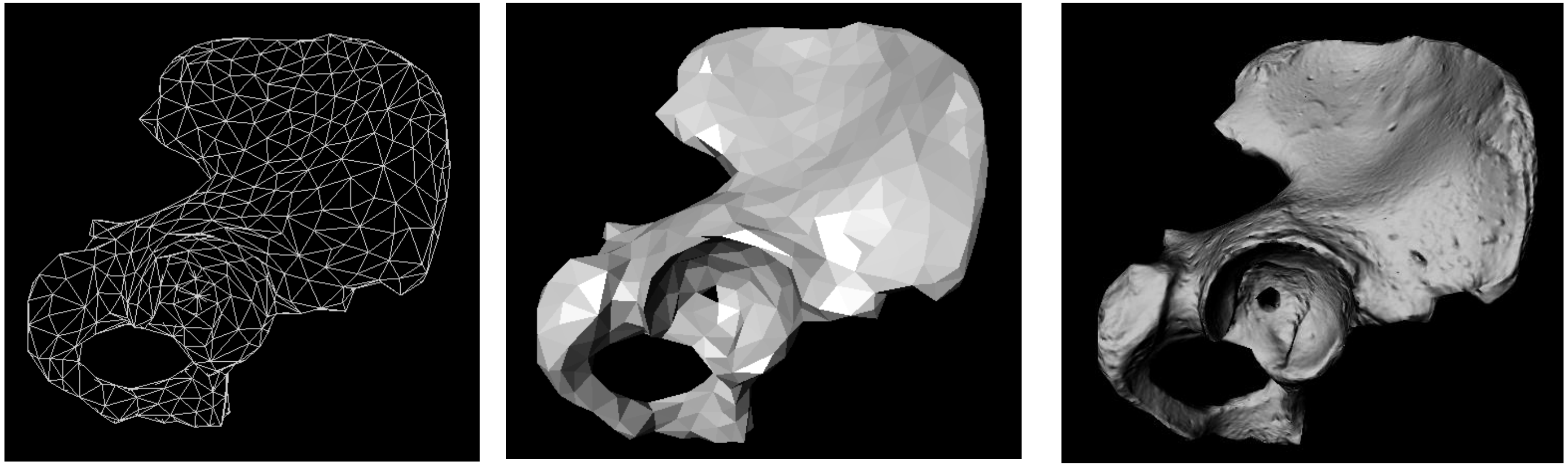
With connectivity (shared vertices) usually result from direct CAD modeling.

- *Objects are represented with vertices and their connectivity.*
- *This information is stored in tables:*
  - *Vertex table*
  - *Edge table*
  - *Polygon table*
- *Tables are saved in memory and on file.*

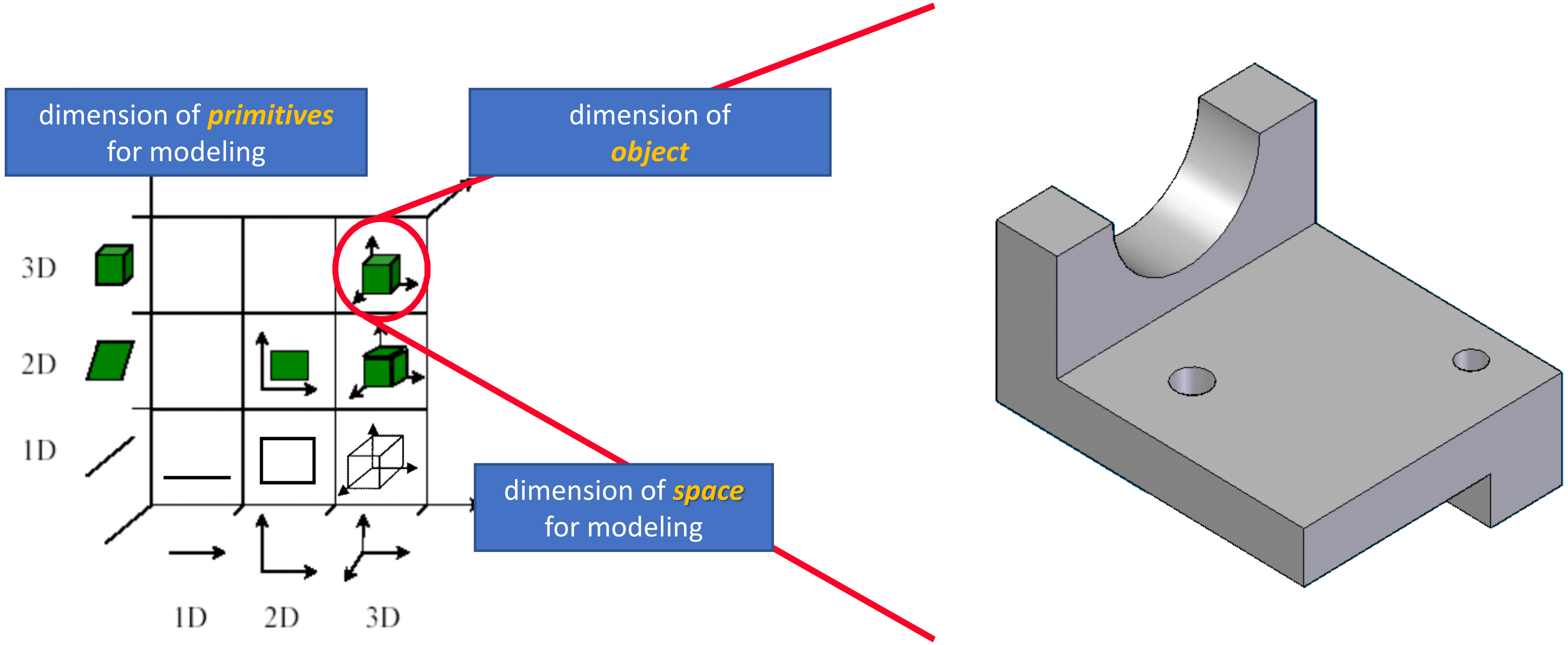
Engineering applications of tessellated models

- Visualization: everything we see on a monitor is transformed into triangles (tessellation)
- Reverse engineering systems output (STL format)
- Input of rapid prototyping systems (eg stereolithography).
- Compact visualization of 3D models on the web (vrml, 3dxml, java3D)
- Input for VR software (jt, iv, 3ds, vrml)

Research in the field of **video games** has led to the improvement of the aesthetic side, starting from very simple polygonal models (**texture**, **bump mapping**).
The result sounds accurate but it isn't!



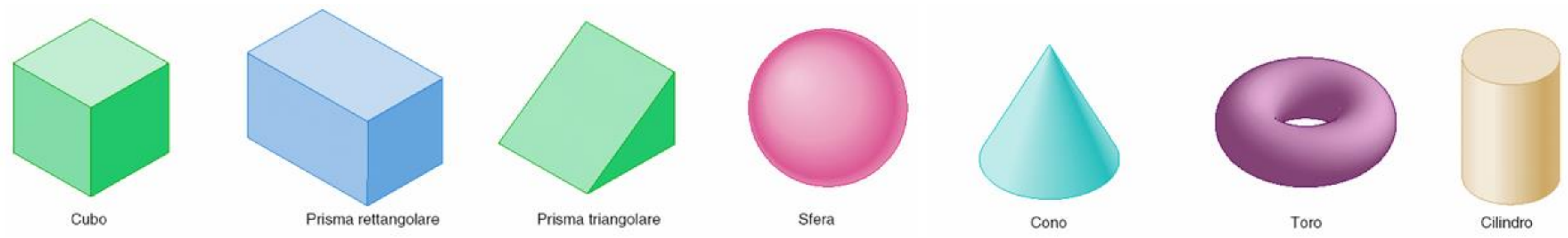
3D Modeling with solid elements

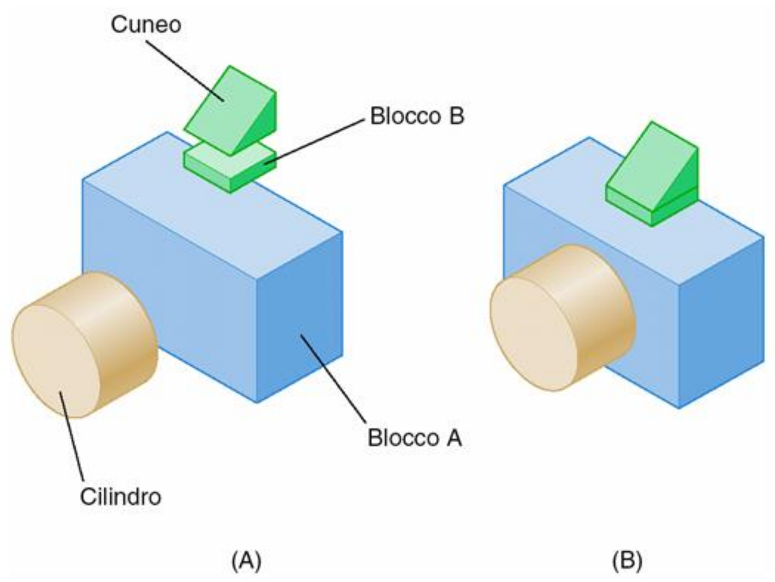


- **Primitive Instancing**
- **Systems based on solid primitives and construction methods (C.S.G.)**
- **Decomposition based systems:**
  - **- by spatial enumeration**
  - **- Octree**

Primitive Instancing

- Starting from simple solid primitives it is possible to obtain some solid objects of not complex shape.
- By combining the primitives in different ways
- The items obtained are very limited



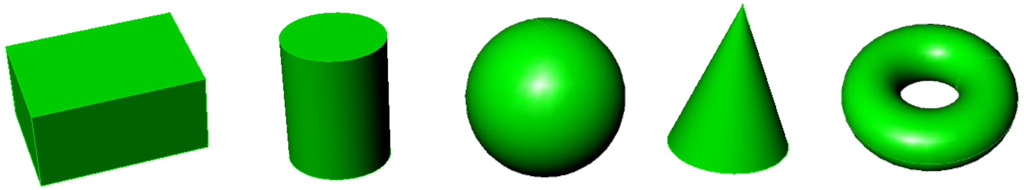


C.S.G. Modeling

C.S.G. = Constructive Solid Geometry

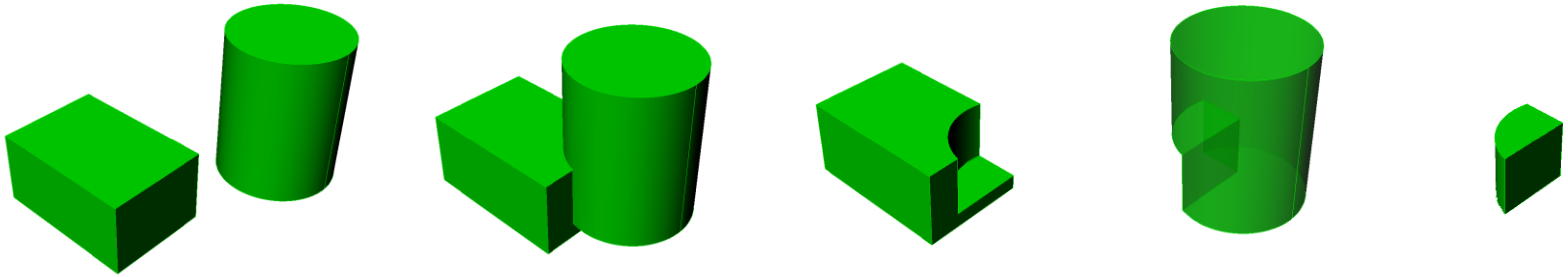
The solid is represented as a composition of parameterized instances of primitives of solids by means of Boolean operations and rigid motions.

Parallelepiped - Cylinder - Sphere - Cone - Torus

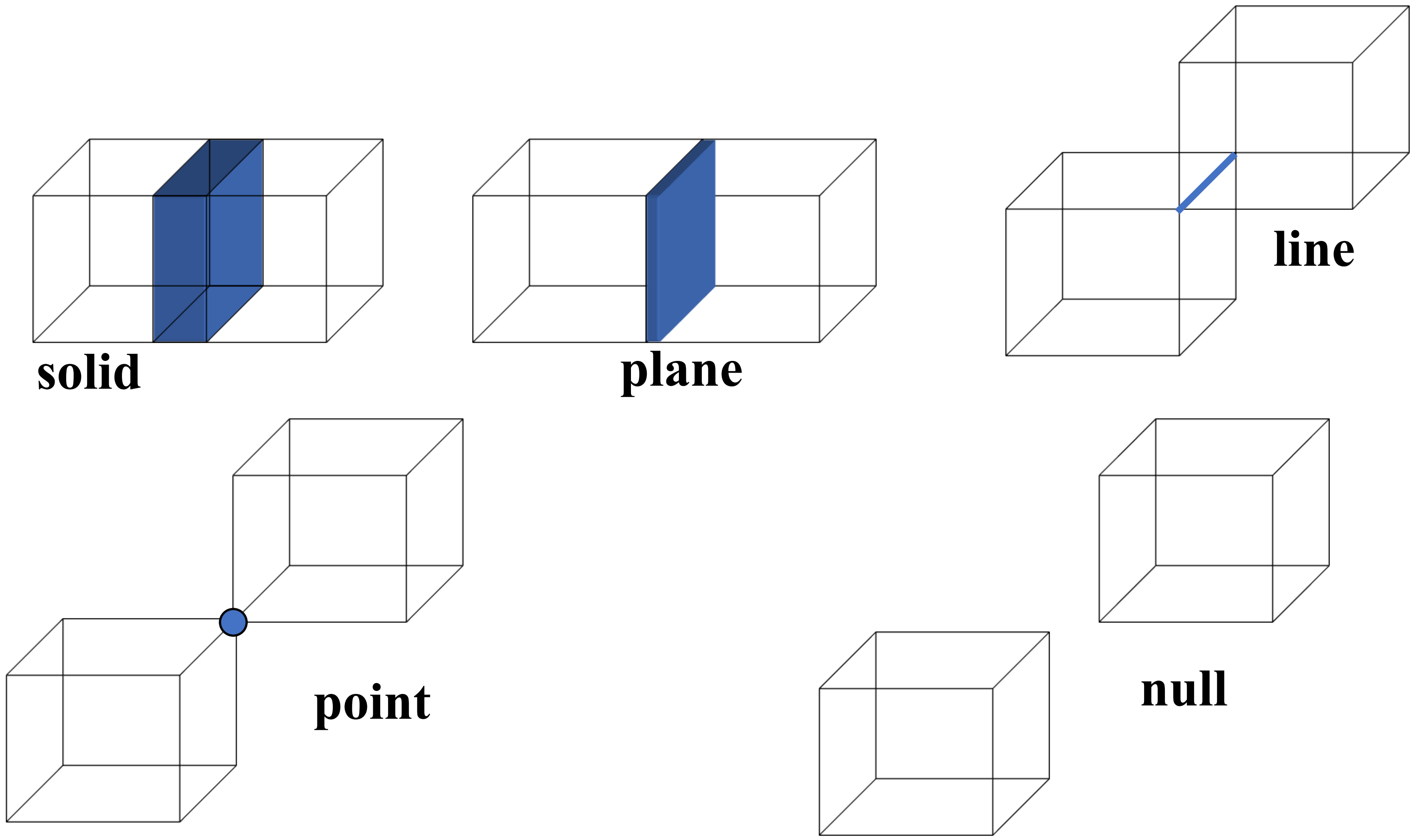


Primitives can usually be scaled, rotated, translated before being combined with the Boolean operators (Union - Subtraction - Intersection).

A B | A + B | A - B | B - A | A $\cap$ B

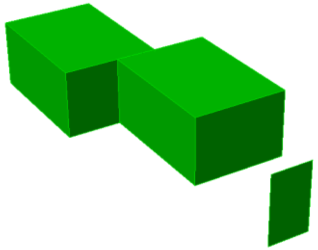


Boolean operations



Boolean operations in C.S.G.

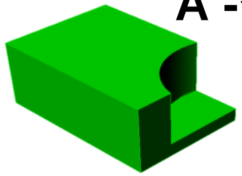
Boolean operations on solids do not always result in a "solid"



$A \cap B$ is a surface!

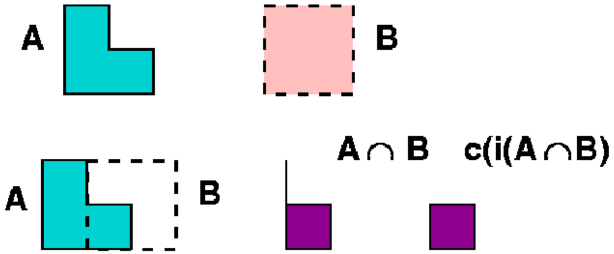
Need for "regularized" Boolean operations

1) Carry out the "traditional" operation, which can generate a non-solid element (surface or curve).
2) Calculate the interior of the object obtained. All lower level entities are removed. The result is a solid without its boundary.
3) Calculate the closure of the object obtained in the previous step. This operation adds all the outline to the object.



$x^2 + y^2 + z^2 = 1$
Internal => $x^2 + y^2 + z^2 < 1$
Closure => $x^2 + y^2 + z^2 <= 1$
External => $x^2 + y^2 + z^2 > 1$

$A +^* B$ = closure (int ($A + B$))
$A \cap^* B$ = closur (int ($A \cap B$))
$A -^* B$ = closure (int ($A - B$))

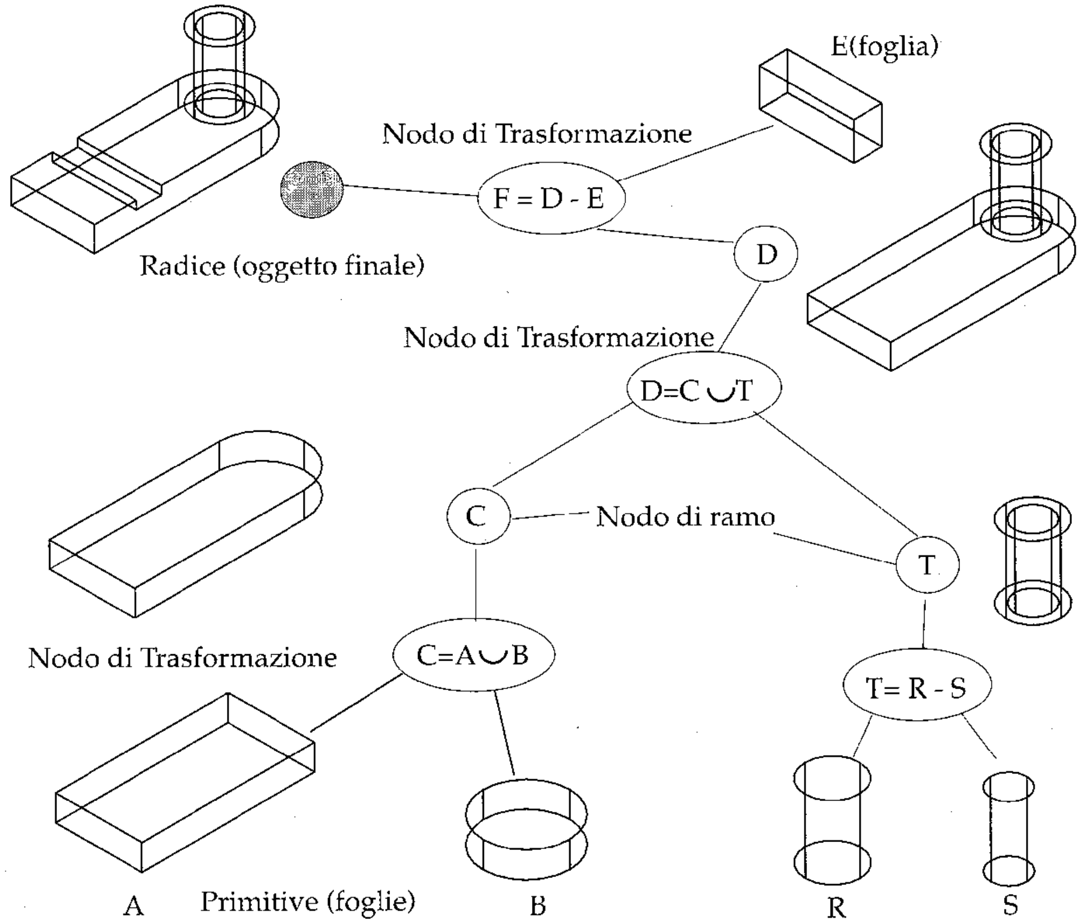


C.S.G. Tree

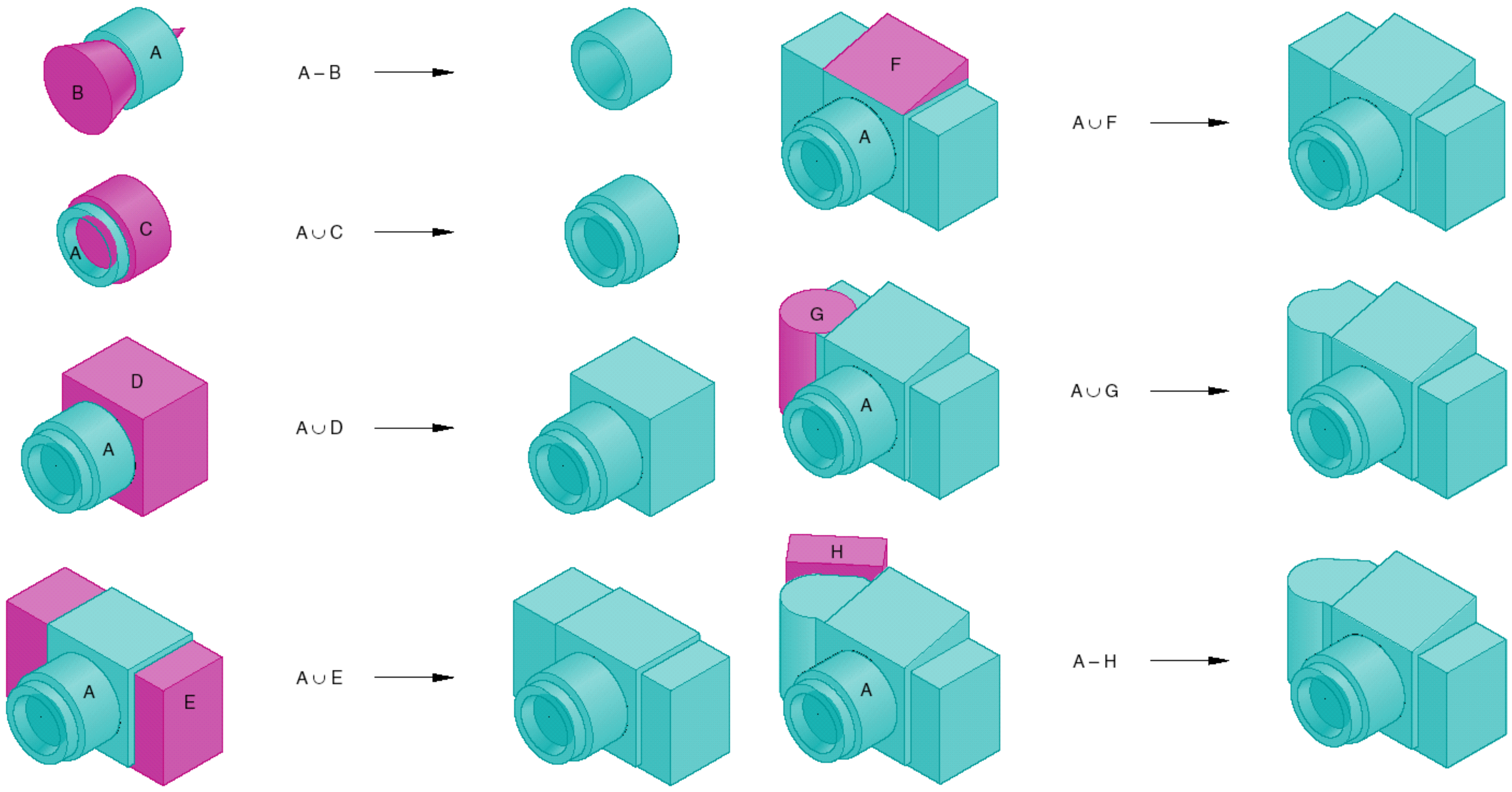
The solid model is implicitly represented through a tree structure that reports the combination of Boolean operations and geometric transformations (translation, rotation, scale) on primitive solids.
The solid is described as a sequence of operations.

The leaves of the CSG tree are the solid primitives and the nodes indicate the Boolean transformations.

Before carrying out a Boolean operation, it is necessary to position the primitives in space.



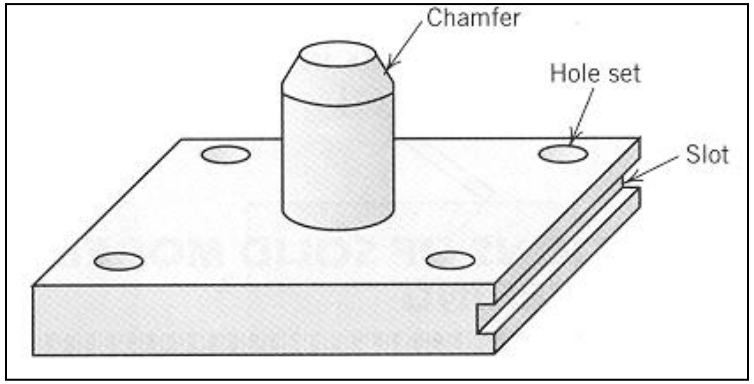
C.S.G. Modeling exemple



3D Knowledge Based Modeling

Feature Based Parametric Modeling

Strong link between CAD and CAM
The manufacturing characteristics (features) are “embedded” in the CAD modeler

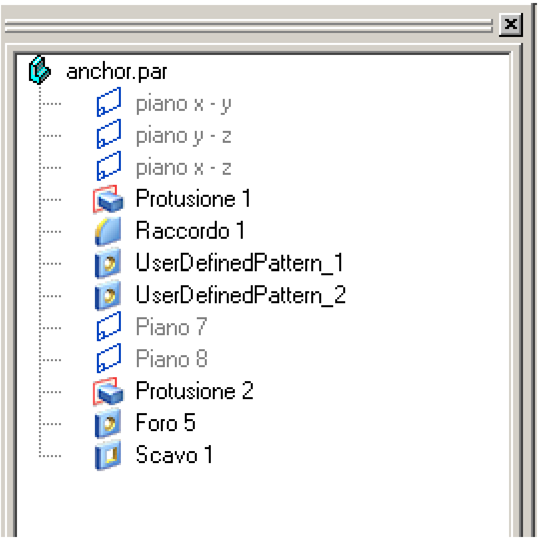


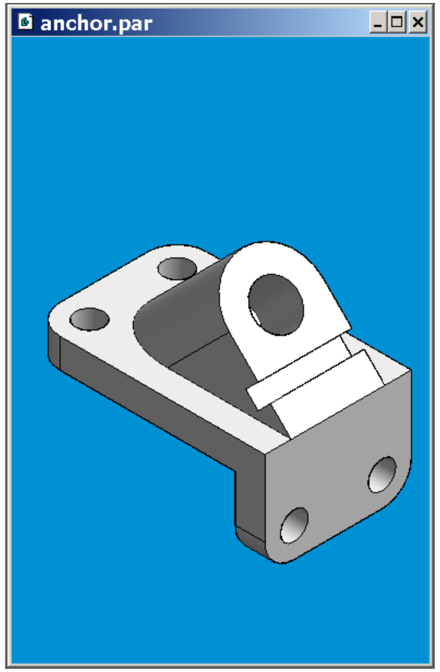
Feature control by means of parameters: Parametric approach

The features are organized in a tree that represents the history of modeling

Most 3D CAD systems are feature-based

Feature-based modelers use some aspects of CSG and B-rep in a combined way





3D Knowledge Based Modeling

Feature Based Parametric Modeling

- You start with a simple basic model and add features
- Features represent manufacturing “operations” such as holes, ribs, fillets, chamfers, cutouts, etc.
- The material can be added or subtracted, as in the CSG
- Features are not limited to just simple primitives but can be generated by extrusion, sliding, revolving, etc.
- The modeling history is stored in the form of a feature tree like the Boolean tree in the CSG

3D Knowledge Based Modeling

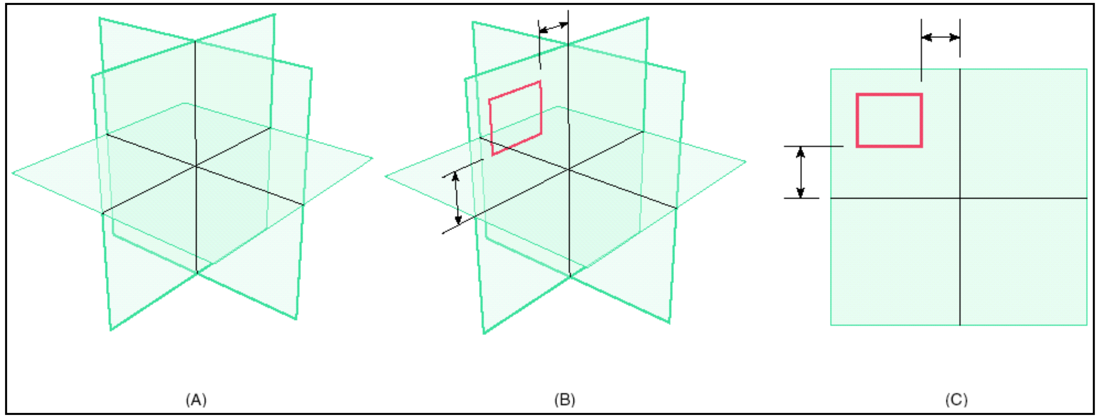
Feature Based Parametric Modeling

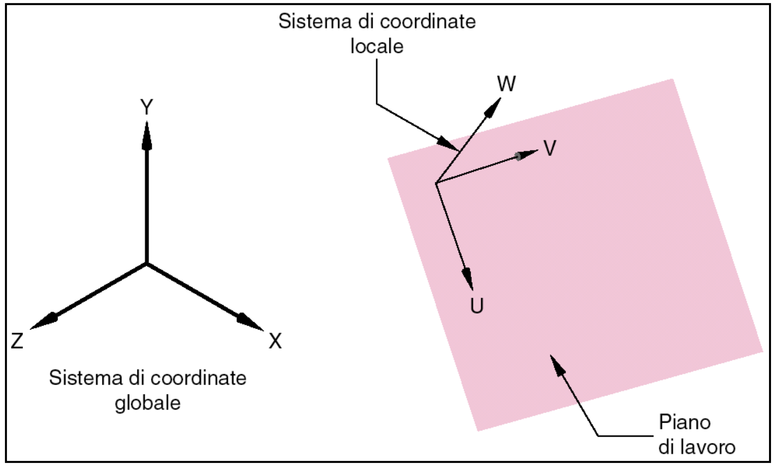
The geometry is defined by first creating a 2D profile on a sketching plane which is then “extended” into 3D
This 3D "extension" can take place by means of:

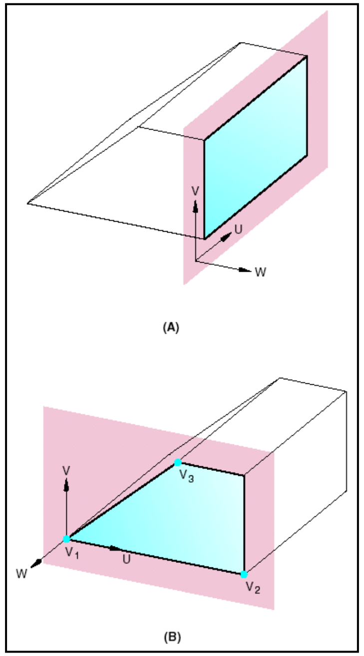
- *Extrusion (protrusion)*
- *Revolution*
- *Sweeping*
- *Lofting (blending) (passing through multiple profiles)*

These operations can add or subtract material

Feature Based Parametric Modeling



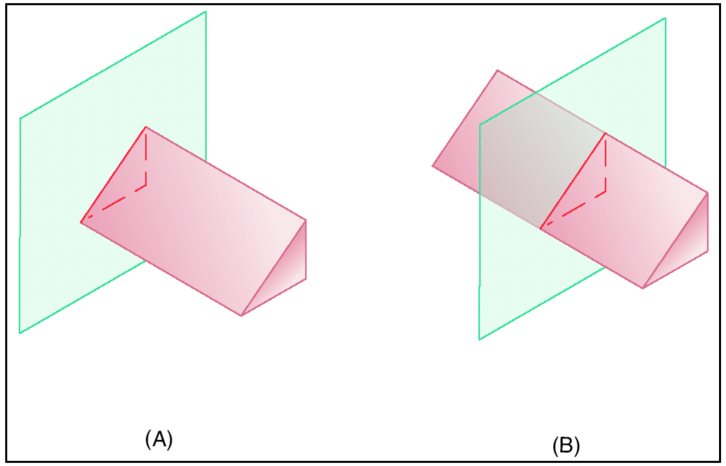




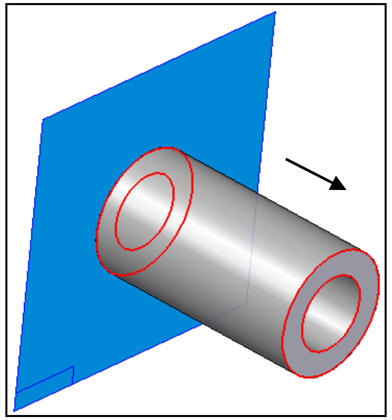
Choice of sketch plane:

- on a flat face
- for 3 points
- normal to a curve
- offset
- tangent
- angled
- ...

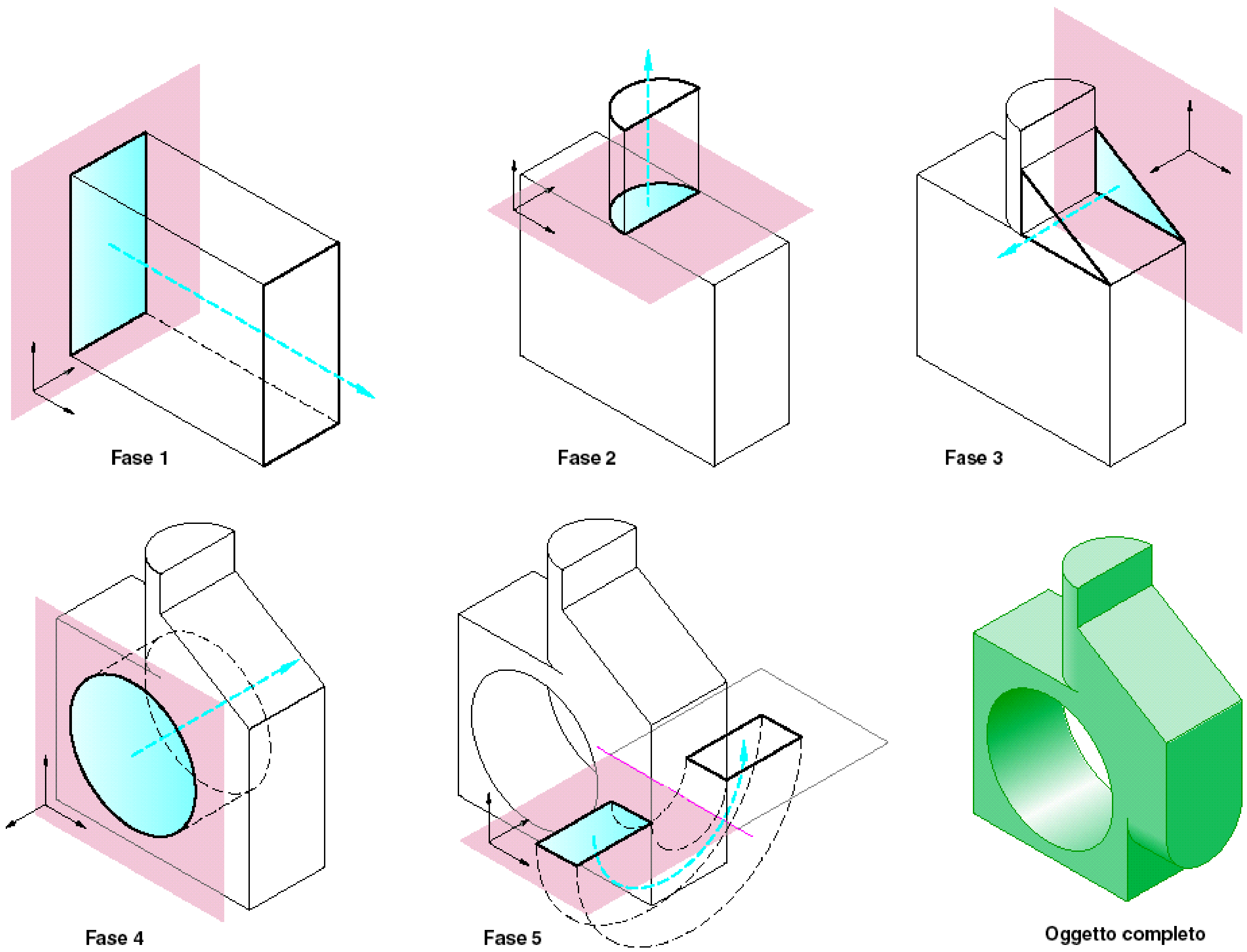
Unidirectional and bidirectional extrusion



Multiple profile

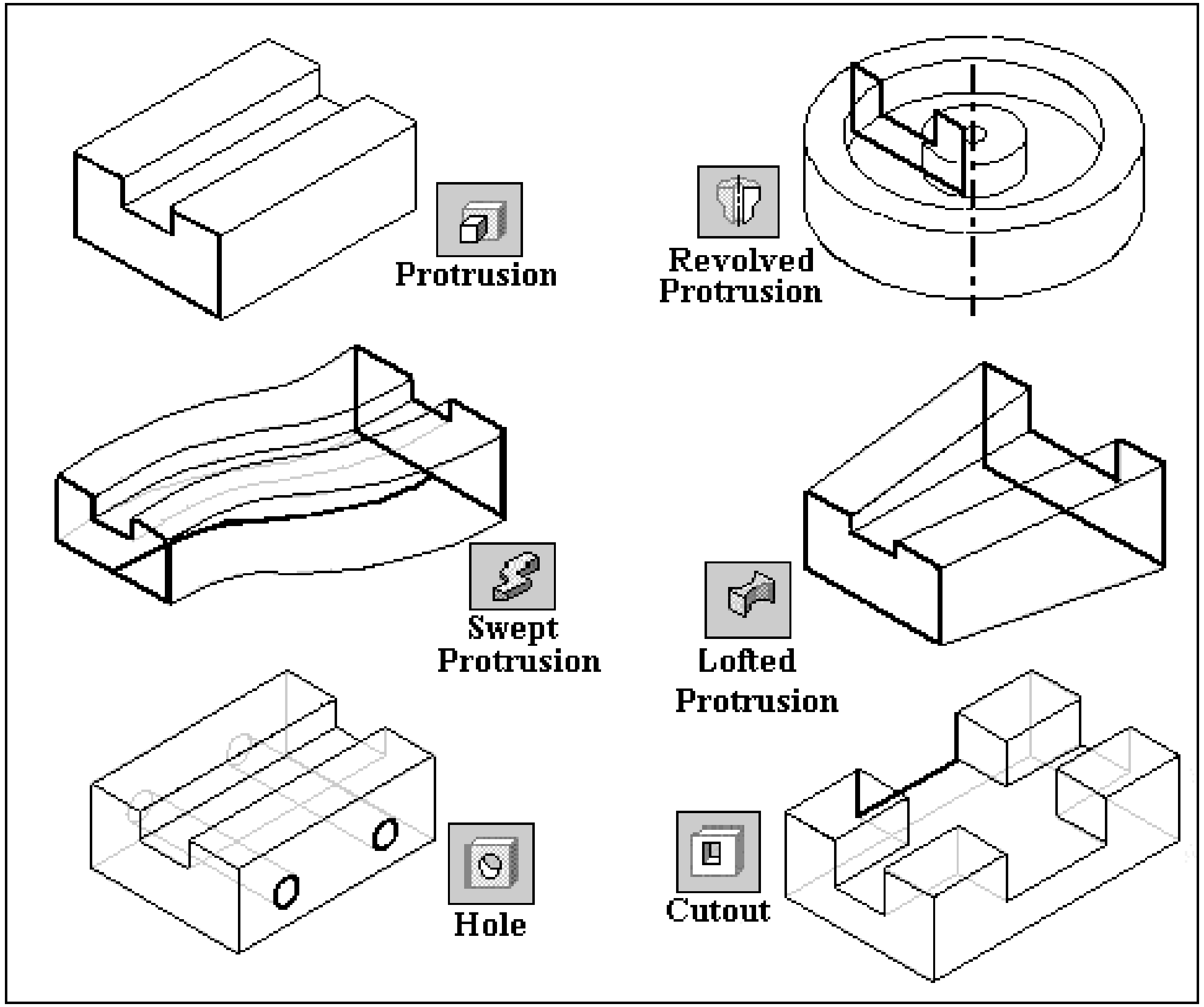


Feature Based Parametric Modeling



Feature Based Parametric Modeling

Main Modeling Tools

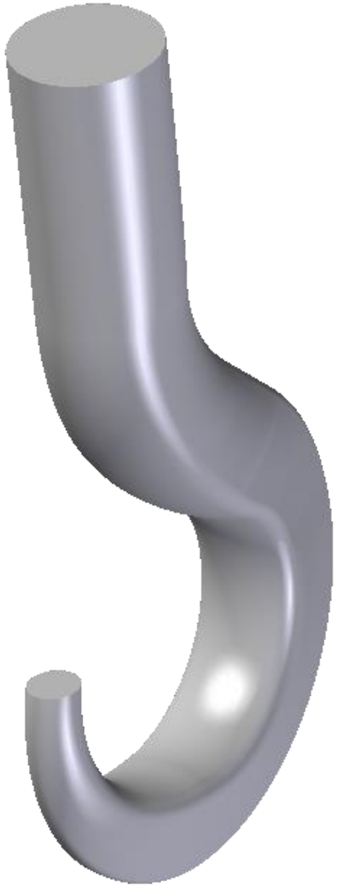


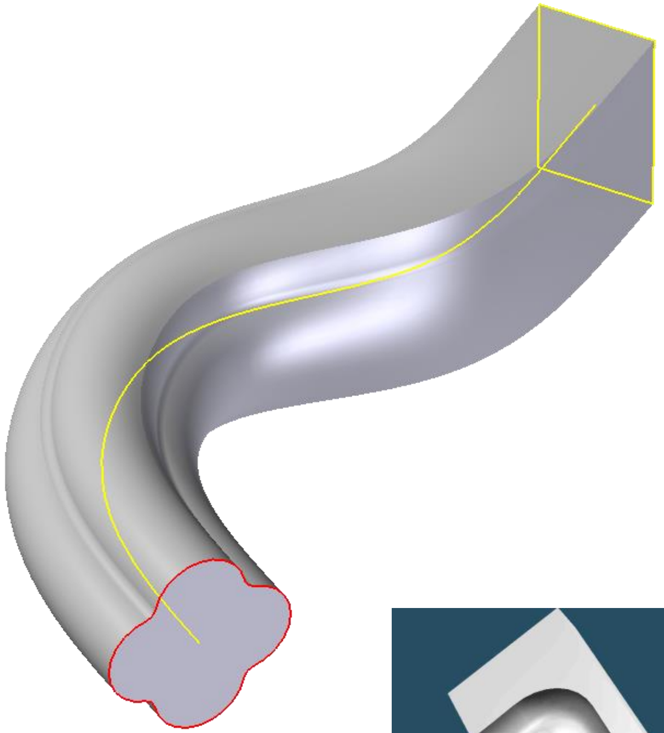
Feature Based Parametric Modeling

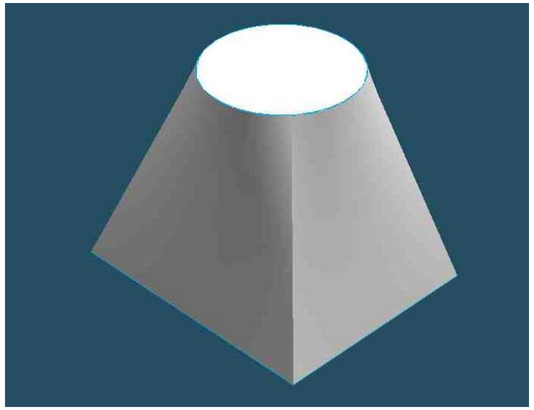
Main Modeling Tools

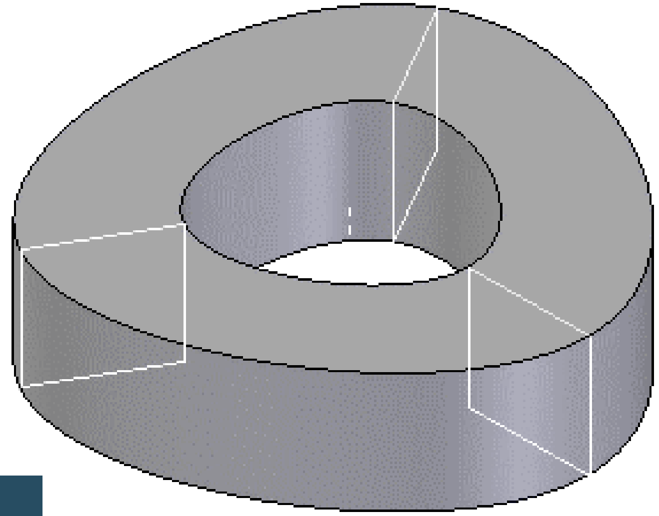
SWEEP

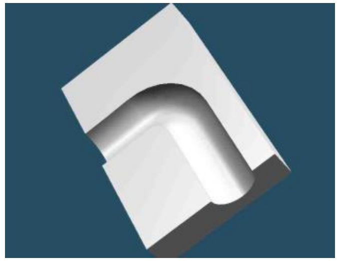
- Sliding profile
- Path curve (spines)
- Eventual guide curve
- Any additional passage profiles







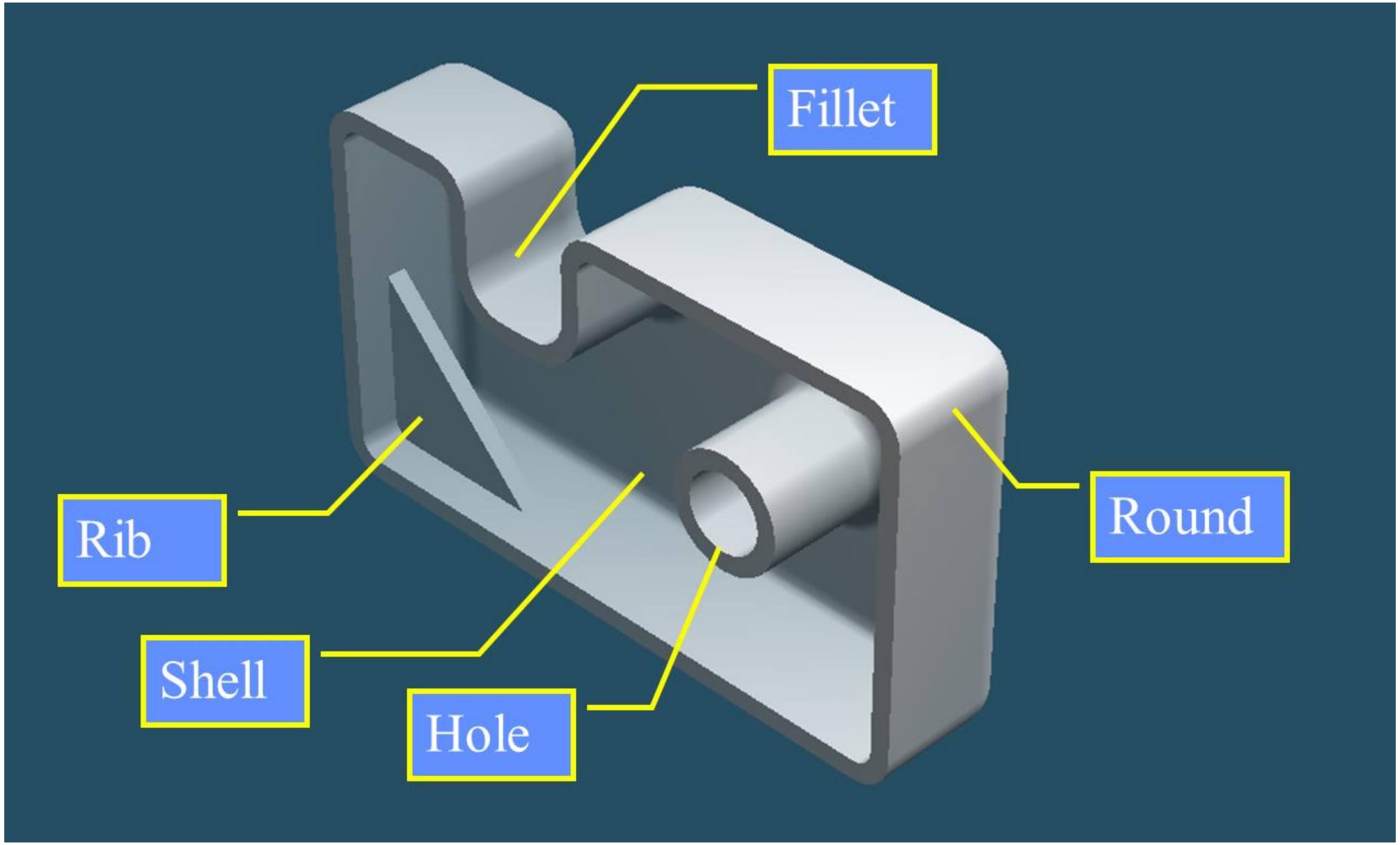




Feature Based Parametric Modeling

Main Modeling Tools

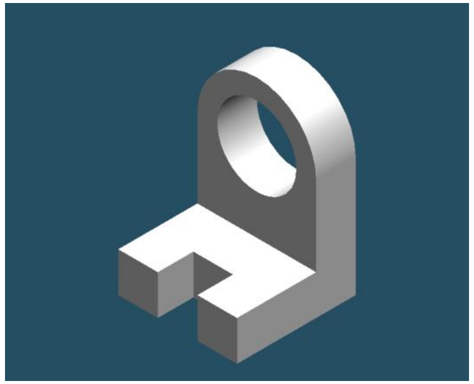
- Fillets, chamfers
- Shells (emptying)
- Holes
- Ribs



Feature Based Parametric Modeling

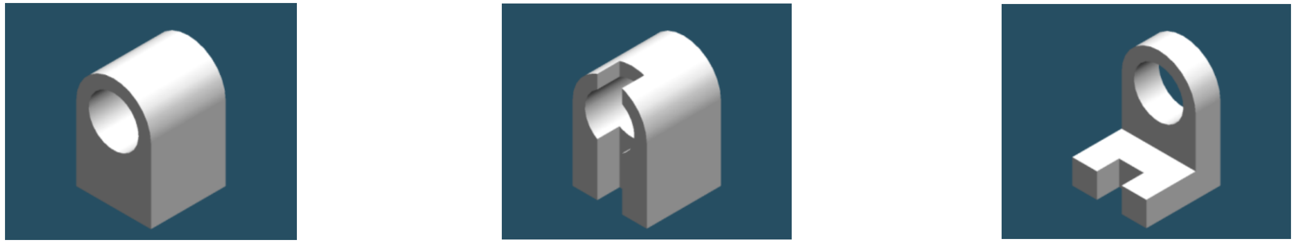
Alternative modeling approaches

- There are usually many ways of modeling the same object
- A good approach takes a few steps and is easy to modify if necessary
- Think about which steps to use in modeling before starting!

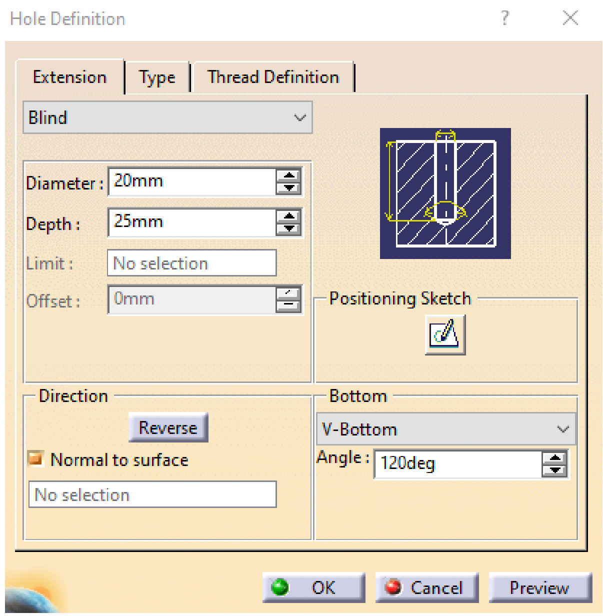


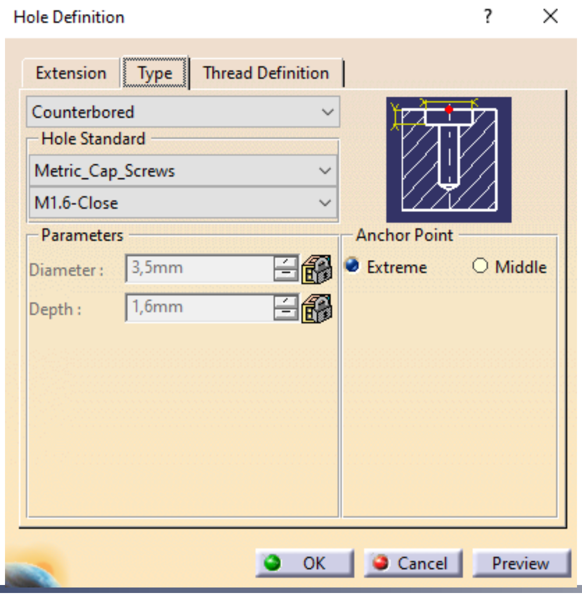
1st method

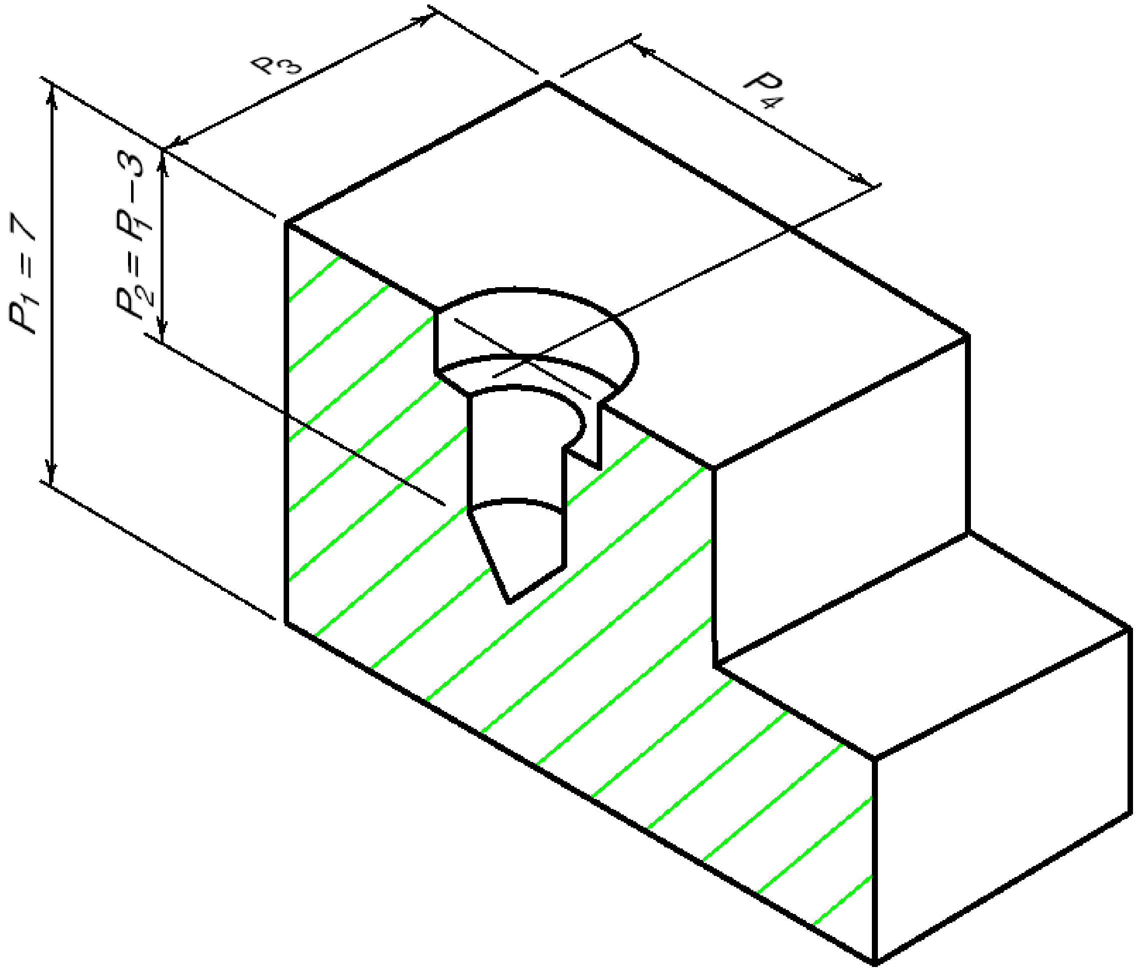
2nd method



Example of feature : counterbored hole

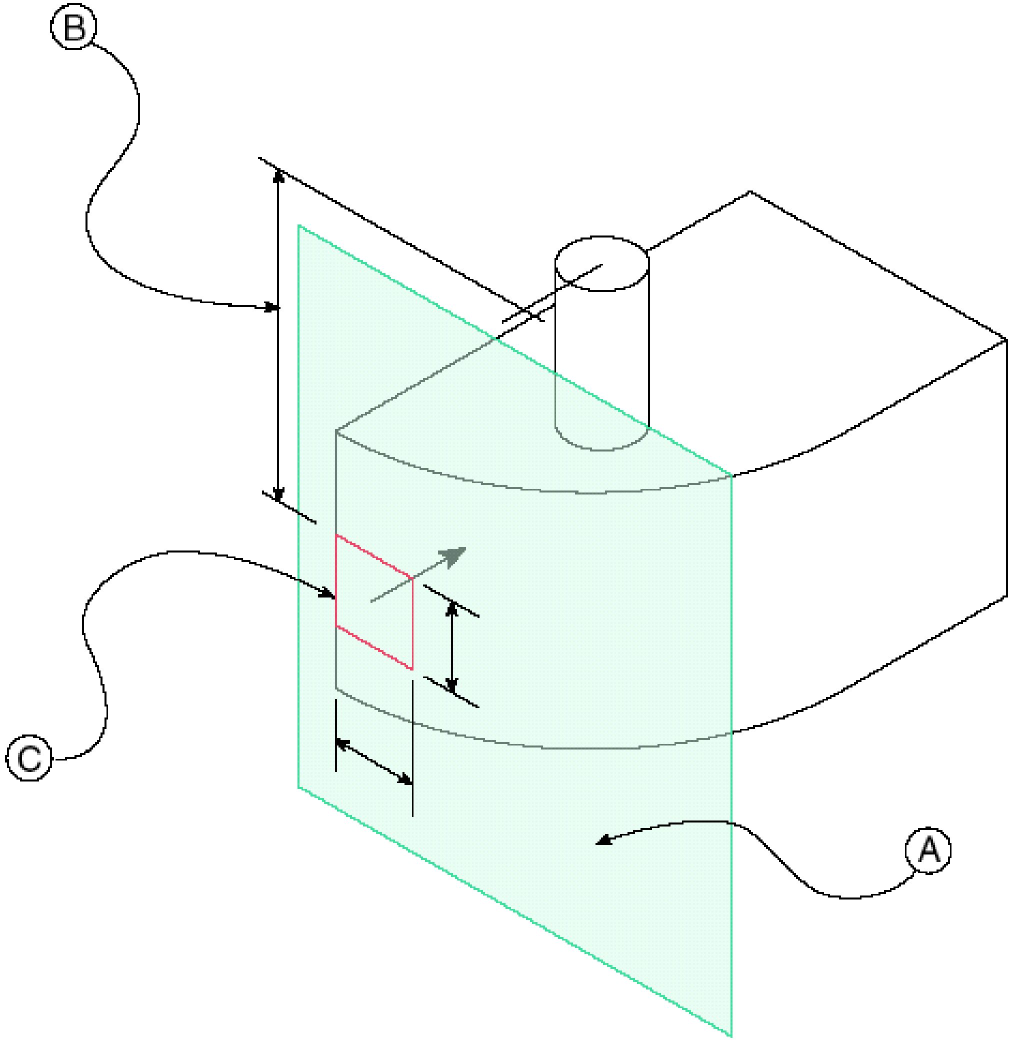






Parent-child relationship

The "parent-child" relationship creates a close dependence of a feature (child) on those (parent) with which it has been geometrically constrained



Parents of the new feature:

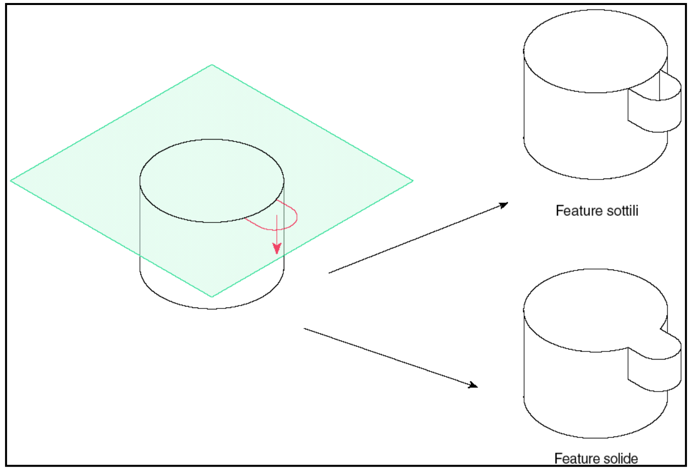
A) the sketch plane

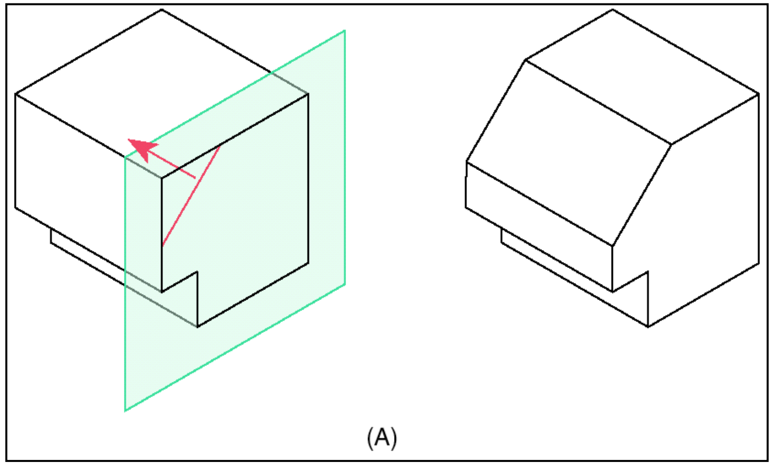
B) The dimensional constraint with respect to an existing feature

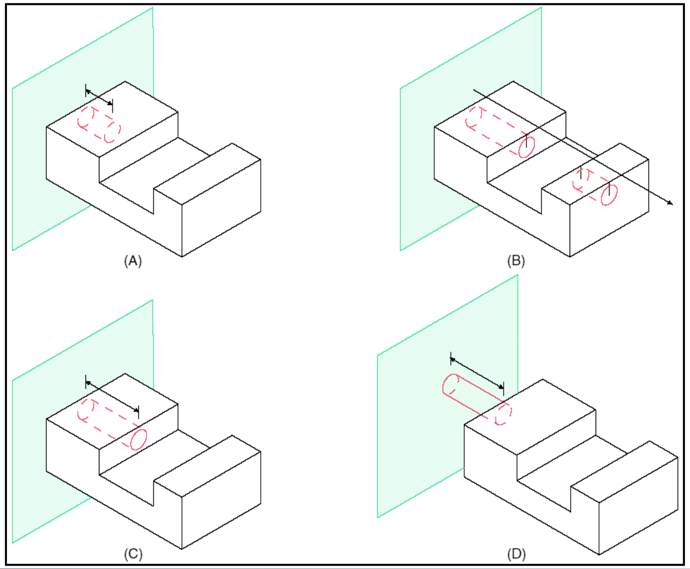
C) The overlap constraint to an existing feature

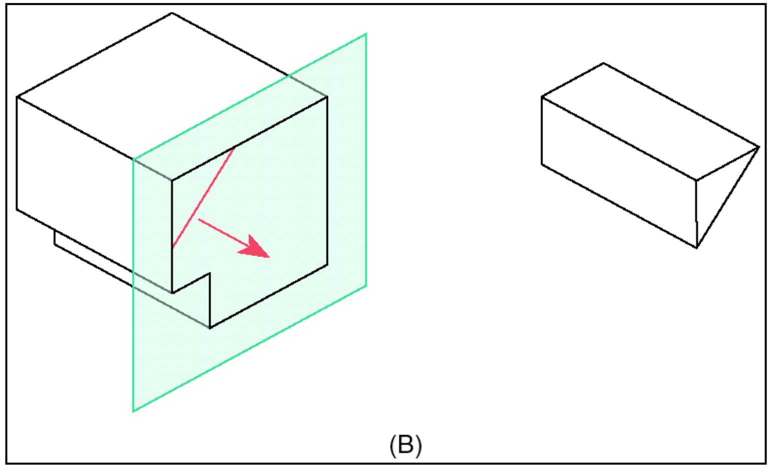
Sketch

The profile can be constrained to existing geometry and can therefore also be non-closed





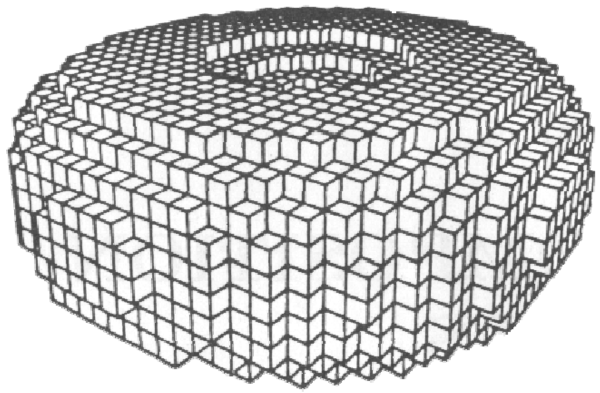




Spatial enumeration

Voxel Models

- A set of 3D adjacent cells **“approximating by filling”** the space occupied by the solid to be modelled
- Cells are usually cubic, not overlapping, of uniform size (voxel)
- Modeling is done by:
  - Defining a regular grid oriented along the 3 reference axis
  - Setting a TRUE value to all cells containing some material and FALSE to all the others
  - Indexing each cell by coordinates of a predefined vertex.

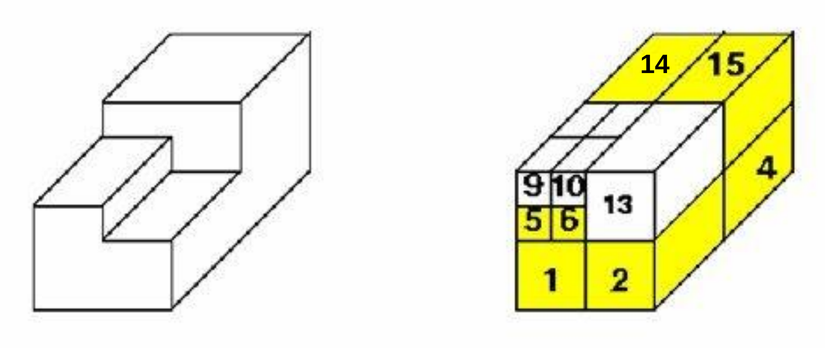


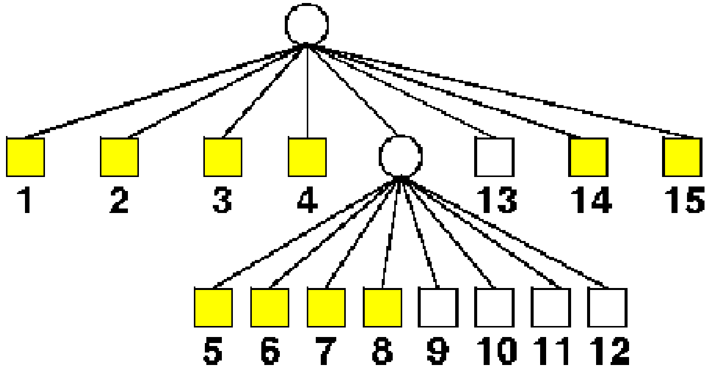
Spatial enumeration

OCTREE Models

Based on recursive subdivision in 8 cubic regions
• Optimize memory space
• Improve accuracy

The model is a tree : each node (father) has 8 child
The space is recursively subdivided : at each step 8 child are generated
To any child a code is associated defining his status:
• Fully included into the solid
• Completely outside the solid
• Partially including the solid





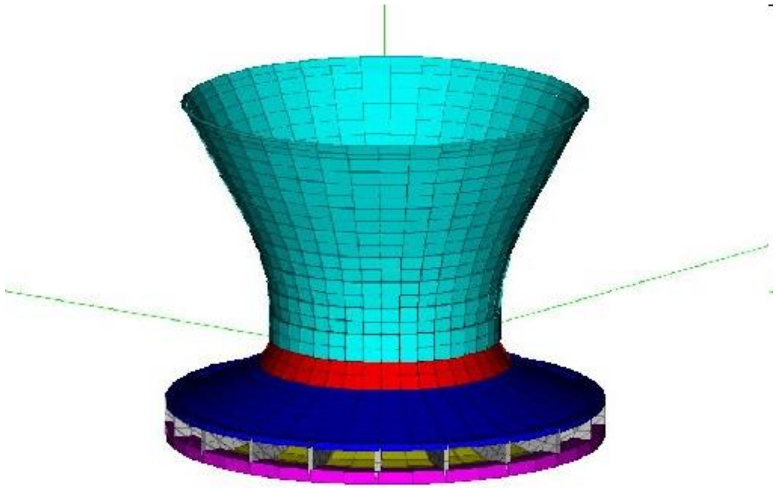
Spatial enumeration

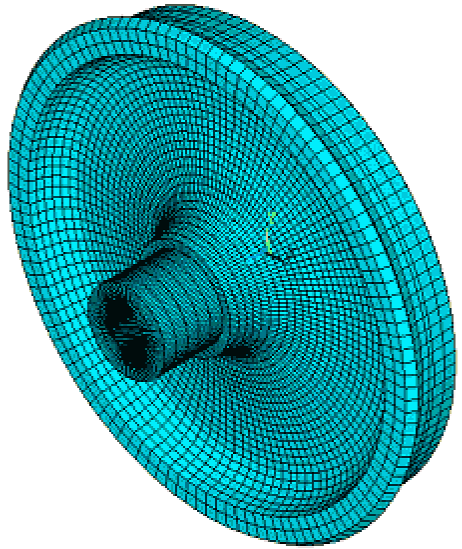
Cellular Models

They decompose space by means of blocks of irregular form (tetrahedra, hexahedron) connected to each other.

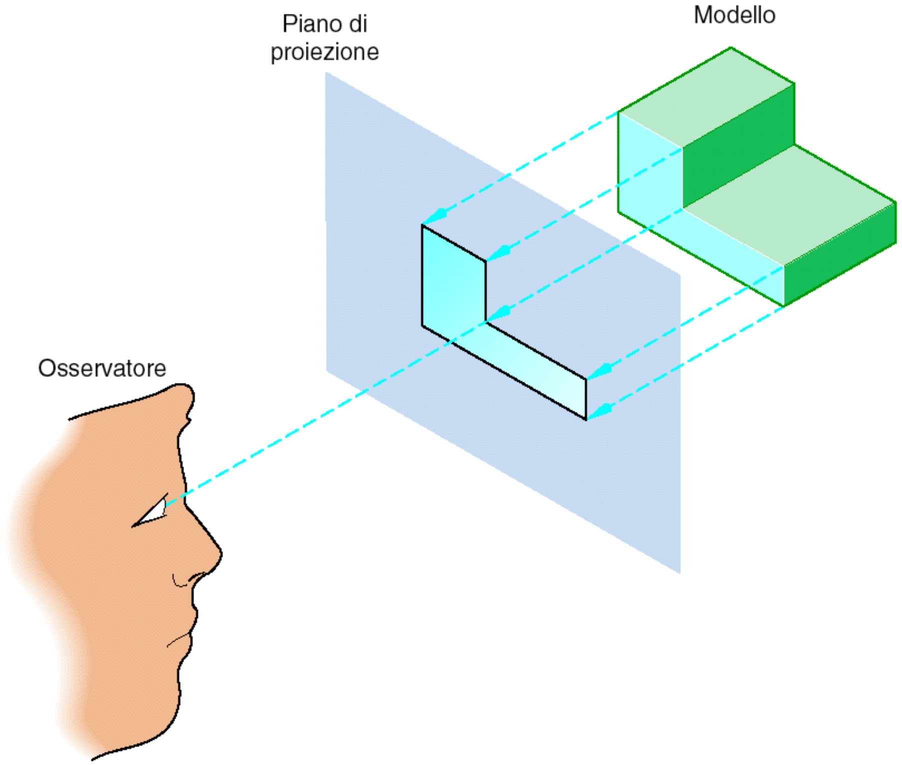
Example: finite element models.

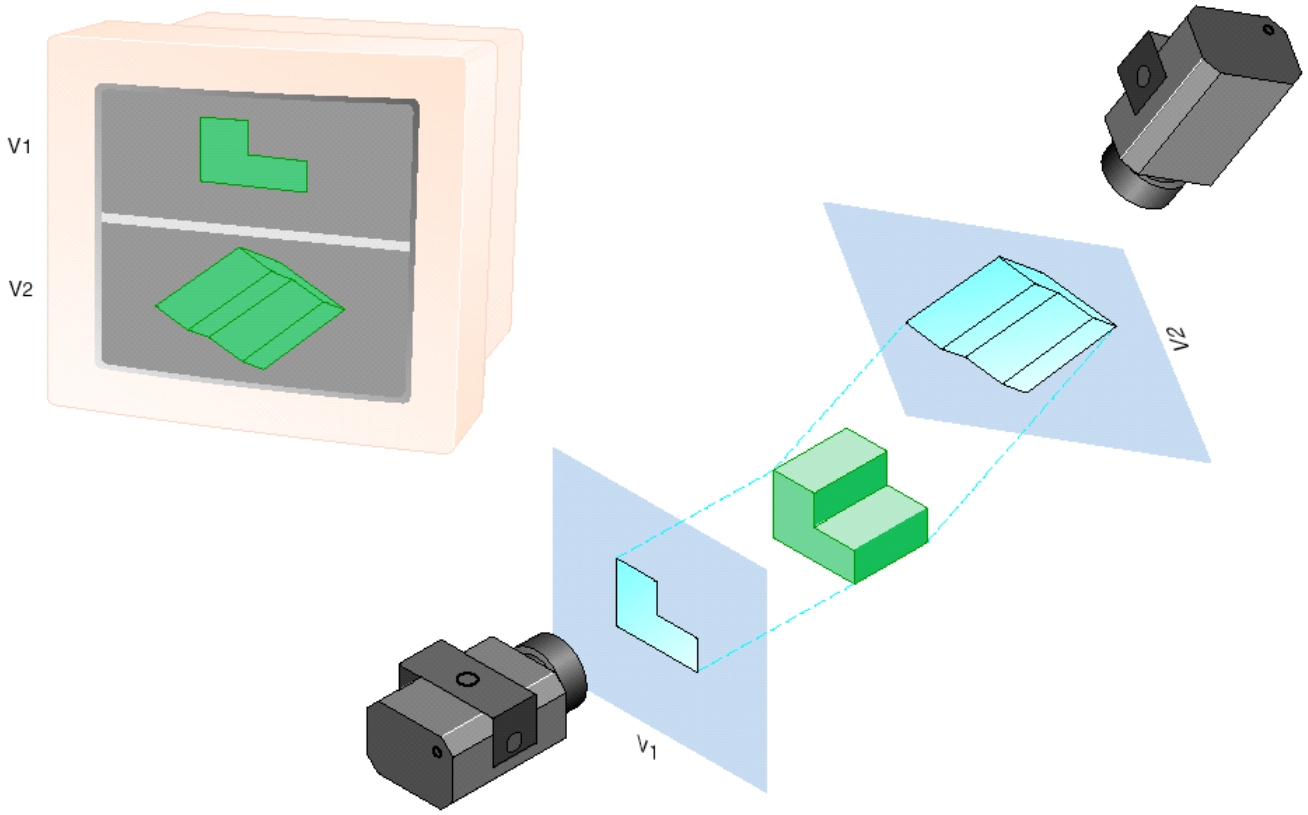
The cells are connected to each other at the vertices



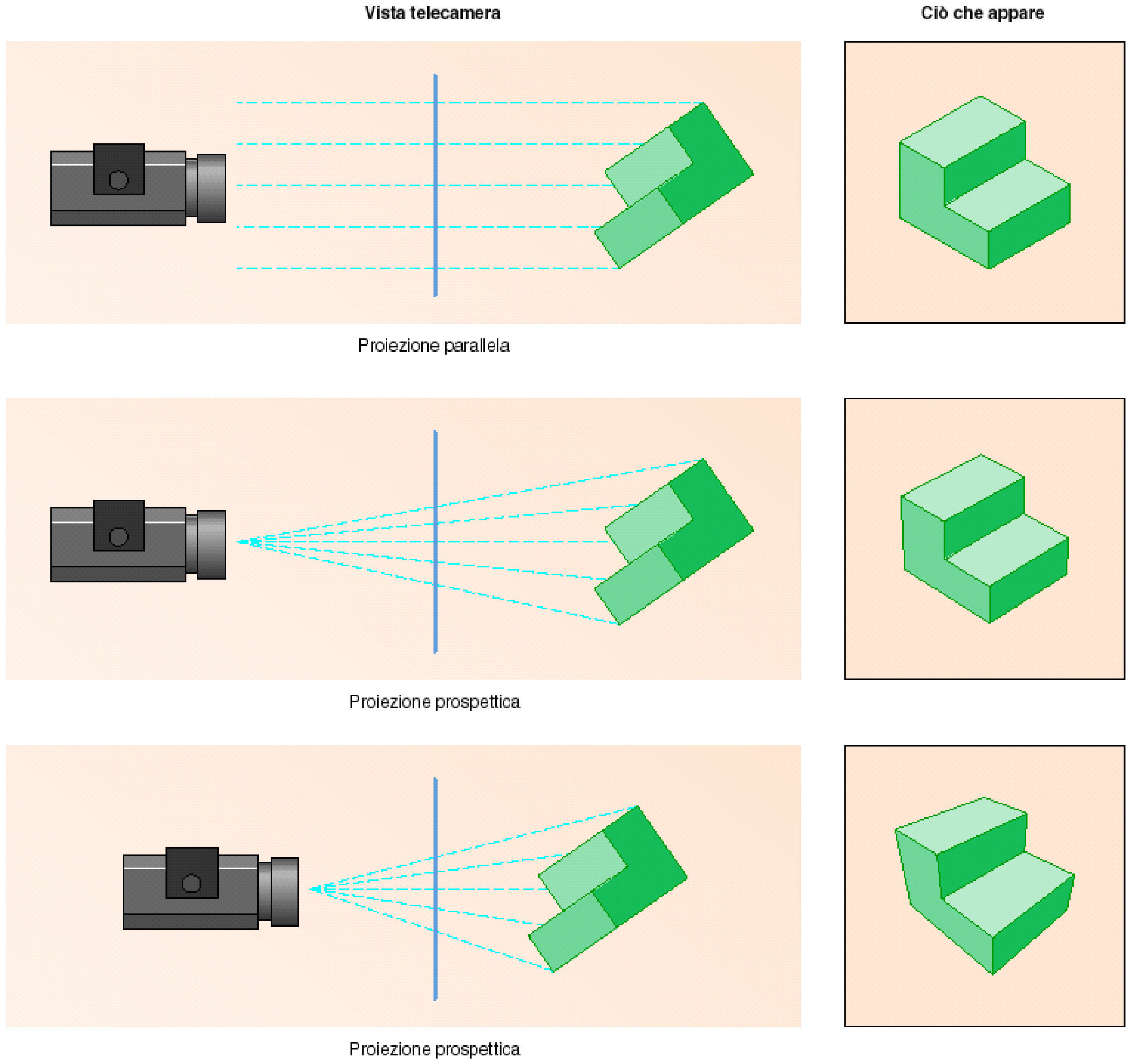


Display of the model on the screen

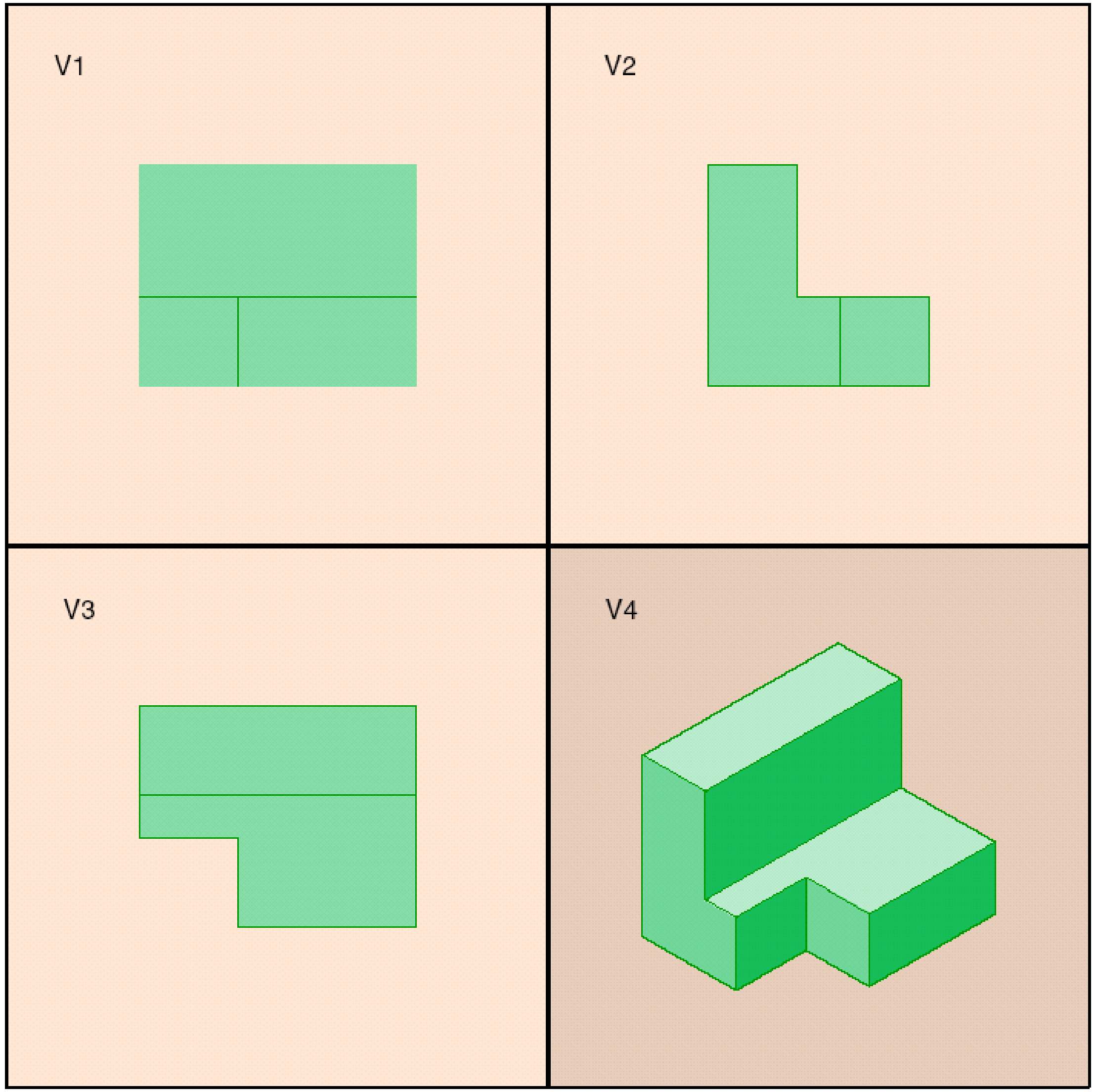




Display of the model on the screen



Display of the model on the screen



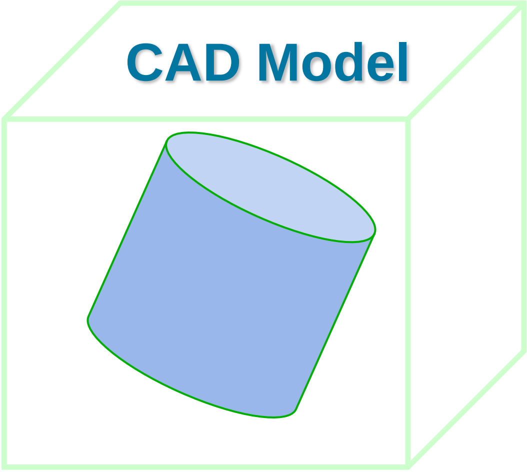
Parametric vs Variational approaches

Parametric approach

- $N_{unknown} = N_{equations}$
- Unique solution
- equations are solved sequentially

Dimensional and geometric constraints

Sketch definition

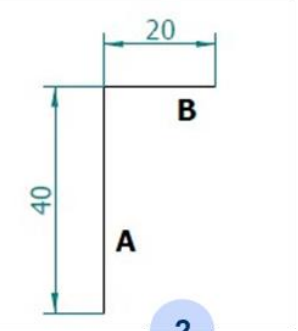


Variational approach

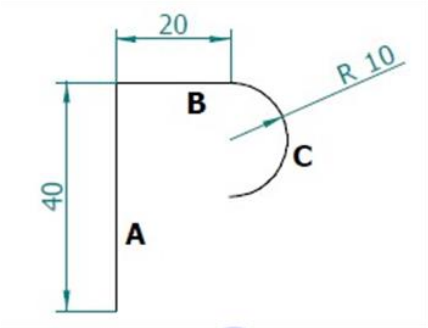
- $N_{unknown} > N_{equations}$
- Infinite solutions
- equations are solved simulateously

Parametric vs Variational approaches

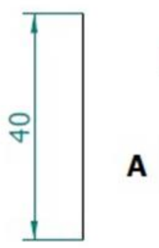
Parametric approach



2

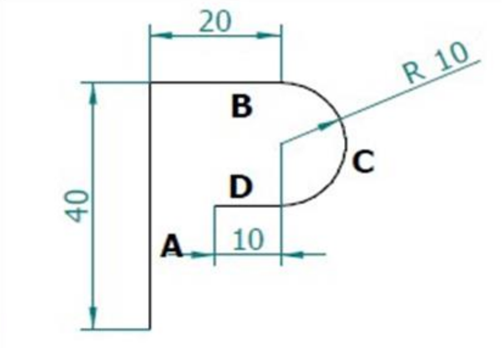


3



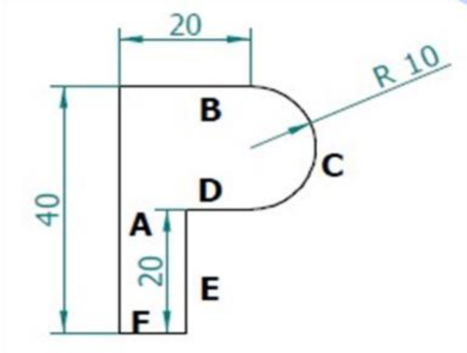
1

Sequential construction of geometric entities, parameters and relations

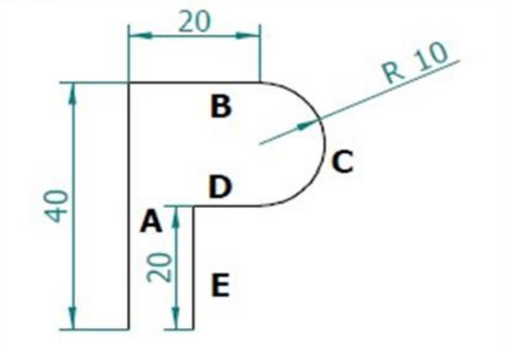


4

6

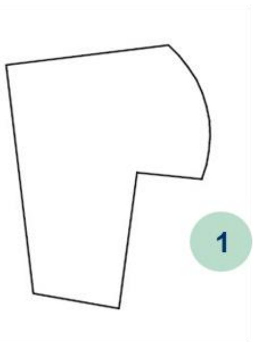


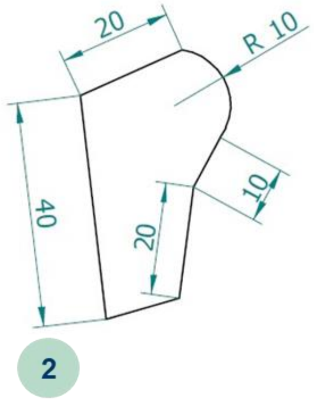
5

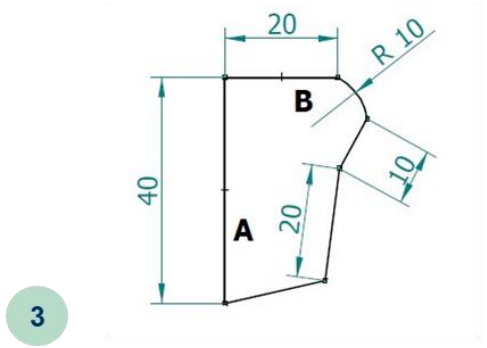


Parametric vs Variational approaches

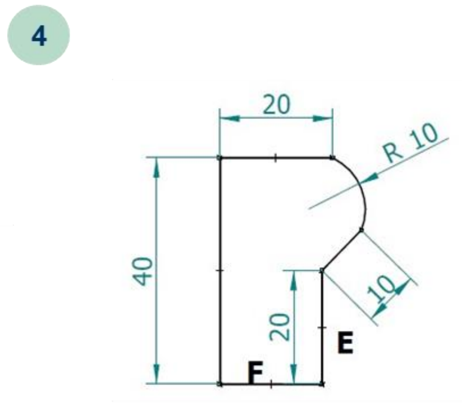
Variational approach

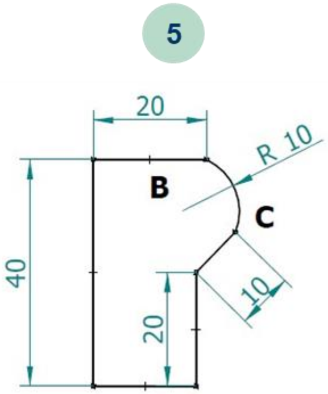


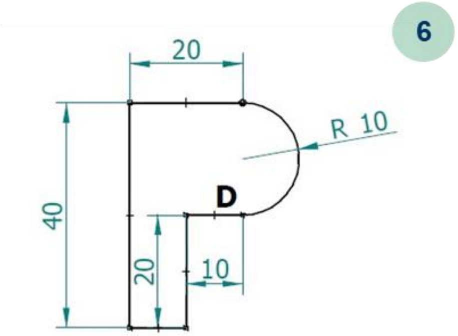




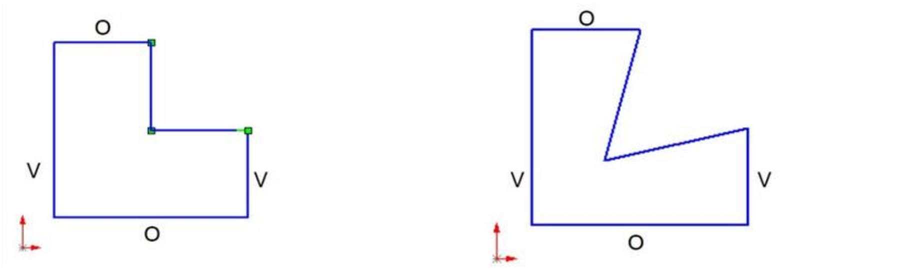
Initial construction of a free form and addition of geometric and dimensional constraints



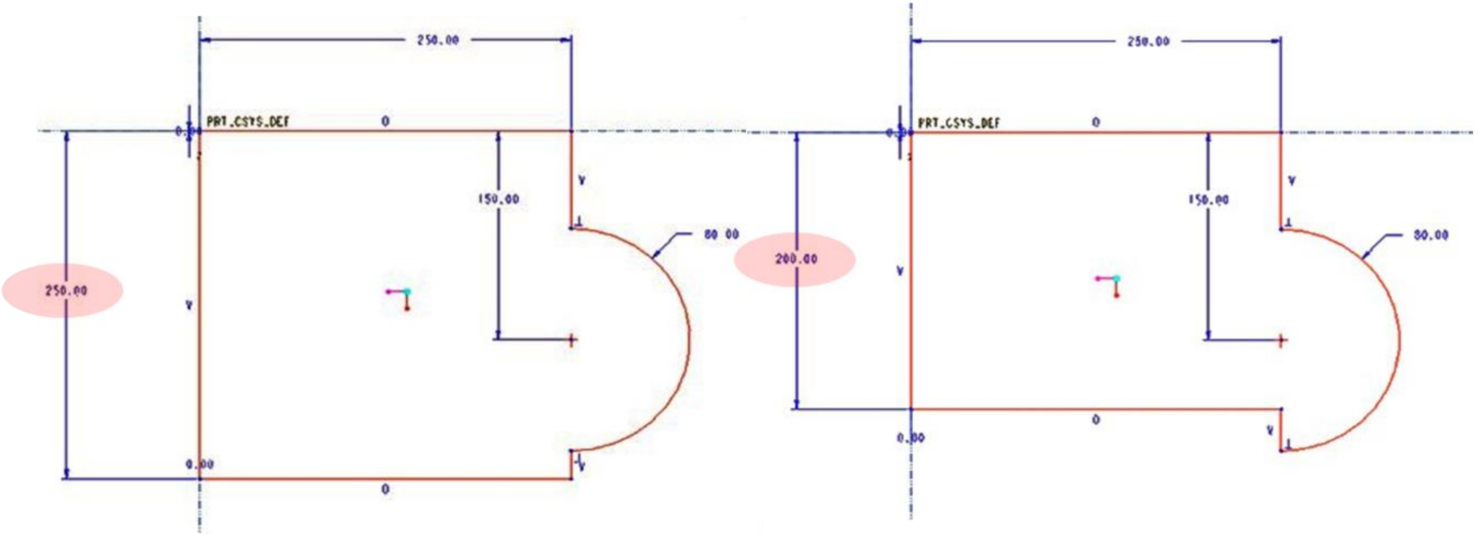




Parametric vs Variational approaches



Under constrained sketch → Drawbacks in updating parameters

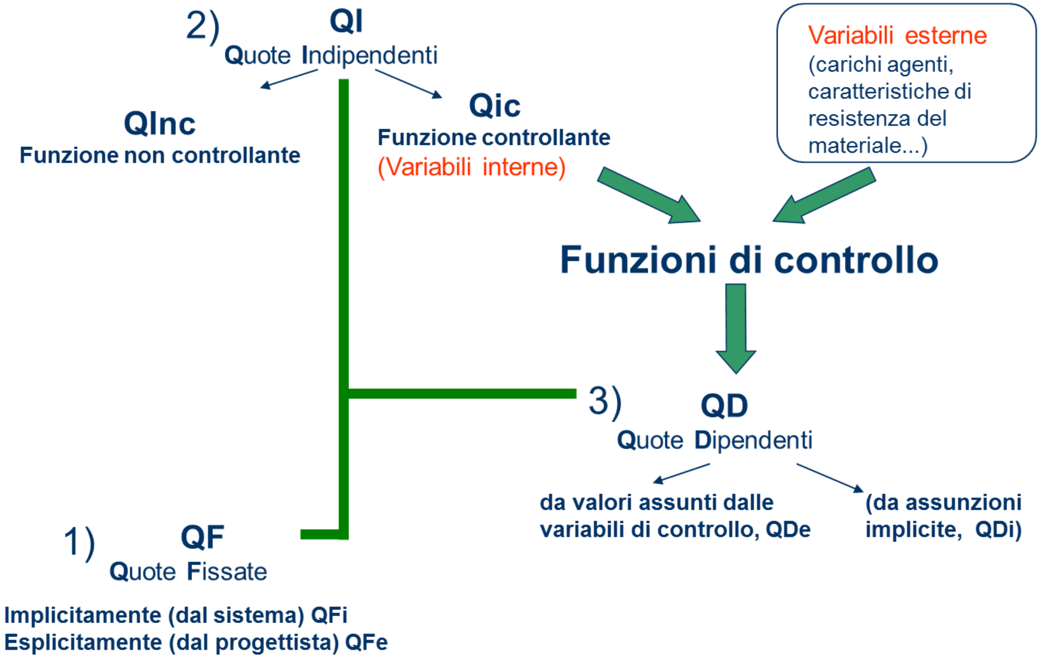


Even a fully-constrained sketch may not retain the designer's intent during modification → *It is necessary to define a parametric-variational control system*

PARAMETRIC-VARIATIONAL CONTROL SYSTEM

The **QF** are quotas "fixed" by morphological-geometric design choice; they are the dimensions that constrain the modeled object to maintain a defined geometric shape. Among them we can distinguish those fixed

- "implicitly" **QFi**, (assumptions made by the modeler during the sketching phase) - for example angles of 90 ° between elements assumed perpendicular to each other;
- "explicitly" **QFe**, those fixed directly by the designer (for example the dimensions of the internal angles of an equilateral triangle).



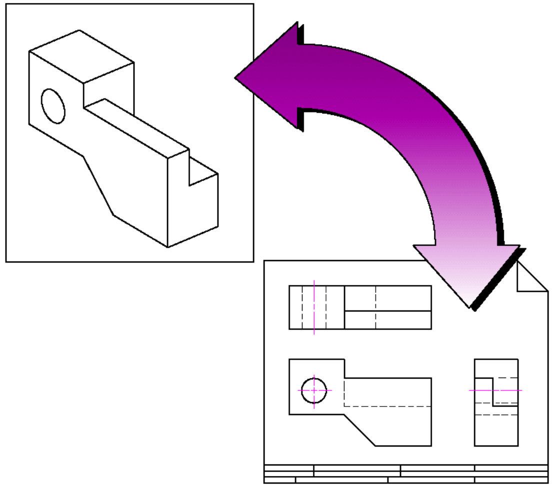
The **QD** Quotas are "controlled". Among them we can distinguish those:

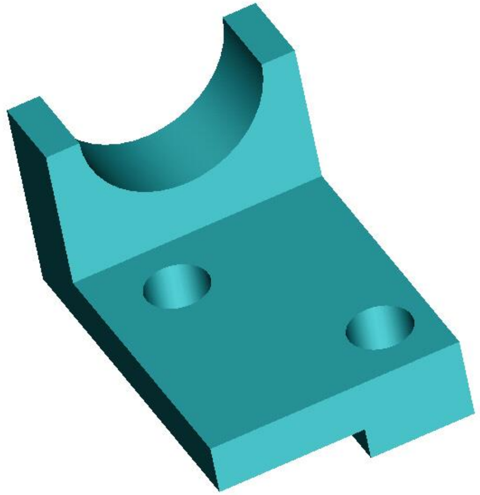
- **QDi** "implicitly assumed"; (assumptions made by the modeler in the sketch phase) - for example, dimensions of elements assumed by the modeler of the same length compared to an element dimensioned by the designer;
- **QDe** "controlled" by values assumed by control factors;

The **QI** are Quotas remained "independent". Among them we can distinguish those:

- **QIc** "parent companies" other quotas; they are quotas whose value determines that of "controlled quotas" **QD**. Control can be exercised through control functions in which the controlling quotas are assumed to control variables of an "internal" type.
- **QInc** "non-controlling" other quotas; they are "isolated" dimensions in the sense that any modification of the value does not involve changes in any of the other dimensions of the model.

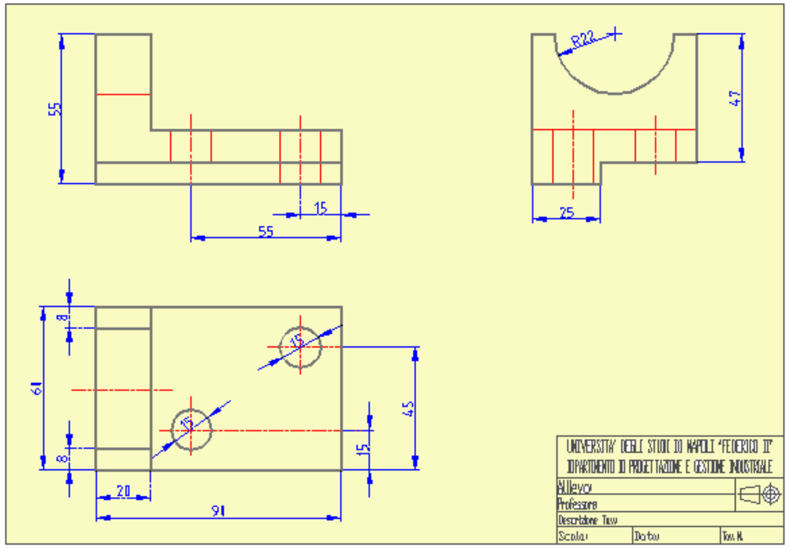
Concept of ASSOCIATIVITY





Automatic drawing

associativity



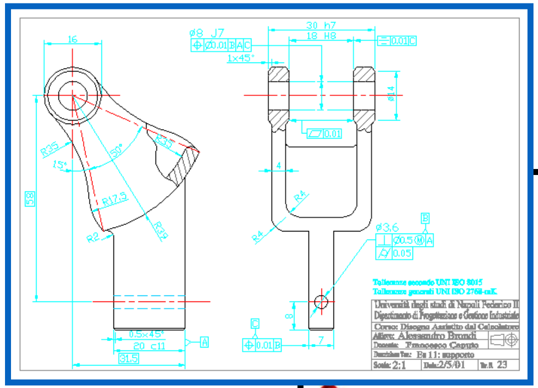
3D model

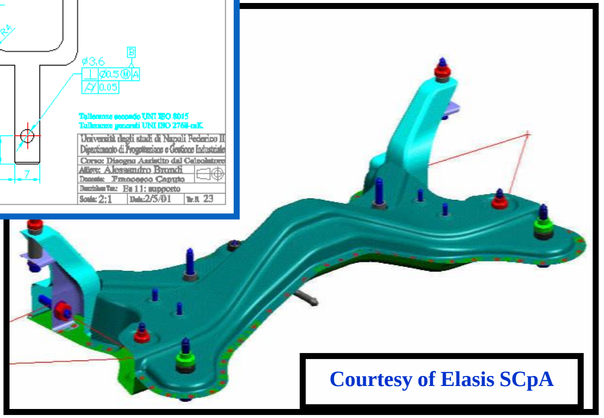
Technical drawing

Modern 3D CAD systems

Computer Aided Drafting

Computer Aided Design

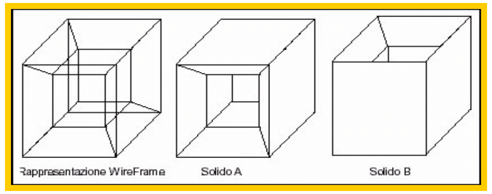


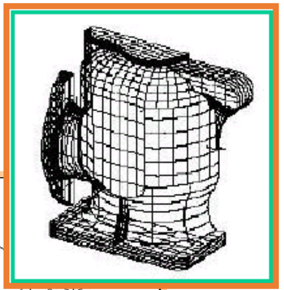


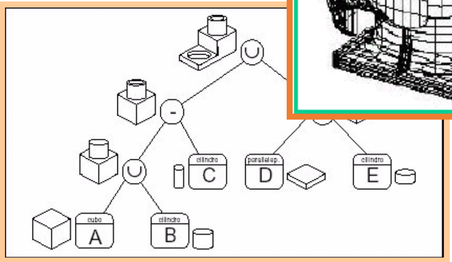
Courtesy of Elasis SCpA

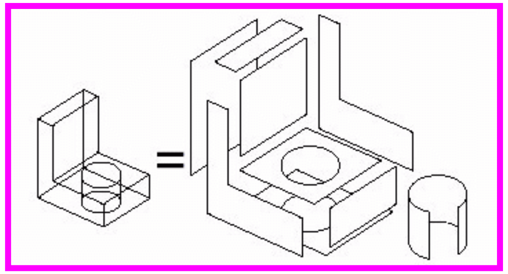
Modeling techniques

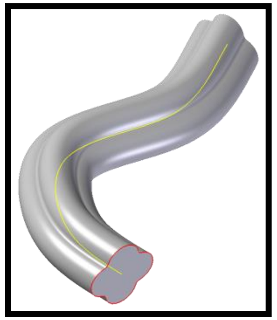
- *Wireframe*
- *Surfaces*
- *Solids*
  - *CSG*
  - *B-Rep*
  - *Sweep*
  - *Hybrid*

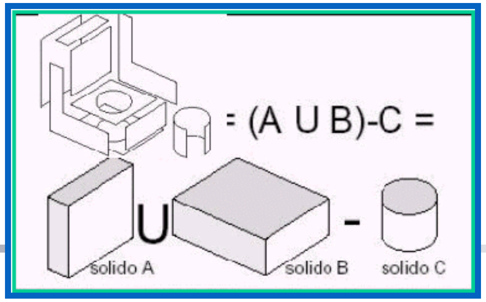








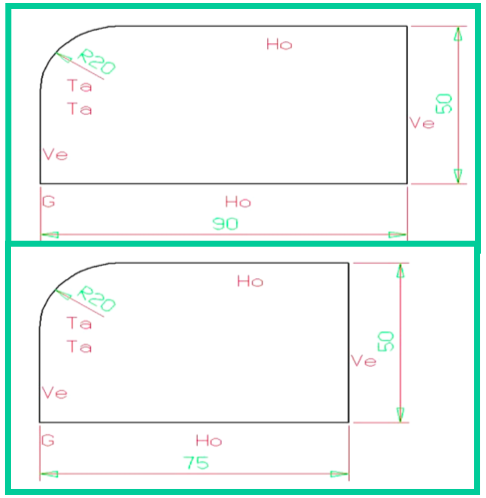


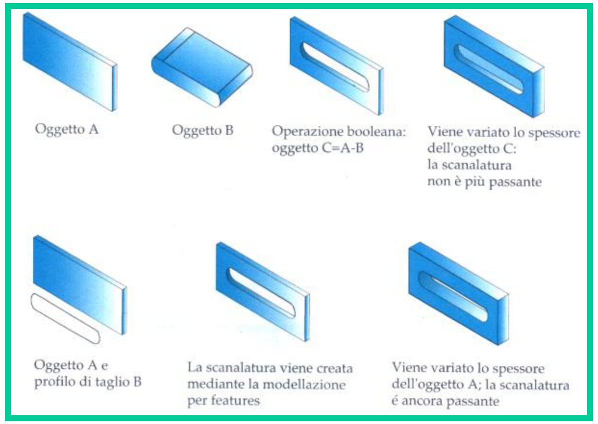


Modern 3D CAD systems

Characteristics

- *Esplicit*
- *Variable-driven*
  - *Parametric*
  - *Variational*
  - *Param.\Variaz.*
- *Feature-based*





Kernel

- *Licenced*
  - *ACIS*
  - *Parasolid*
  - *Open Cascade*
  - *SMlib*
- *Proprietary*

CAD application	Kernel		
	ACIS	Parasolid	Proprietary
AutoCAD	•		
CADKEY	•		
CATIA			•
I-DEAS			•
IronCAD	•		
IX Design	•		
Mechanical Desktop	•		
MicroStation		•	
Pro/ENGINEER			•
SolidEdge		•	
SolidWorks		•	
ThinkDesign			•
Unigraphics		•	
VX CAD/CAM		•	