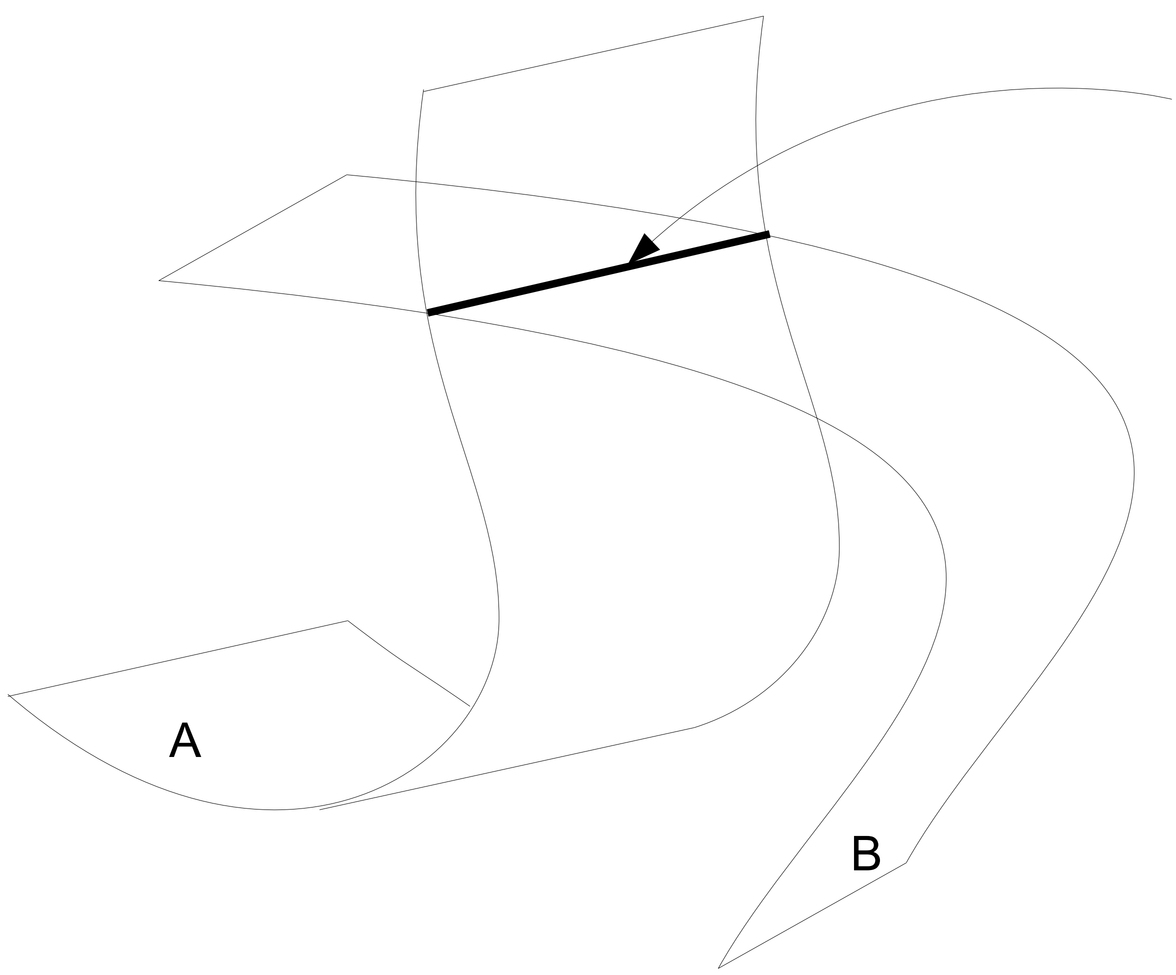
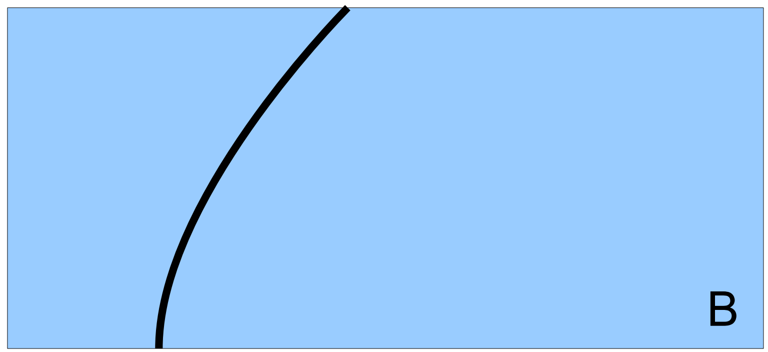
Solid models and B-REP

- Classical modelling problem : the intersection



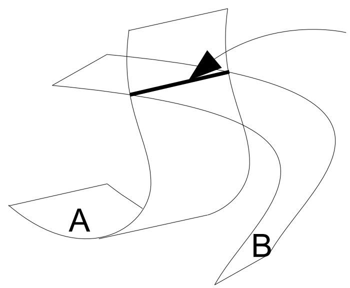
3 independent representations of the intersection :
- a 3D NURBS curve (giving points in the global XYZ coordinate system)
- a 2D NURBS curve
  in the parametric space of surface A (giving 2D points in the coordinate system of the parametric space of surface A)
- Idem for surface B



Parametric space of surface B

Solid modelling

- Theoretically, these three representations are equivalent ...



3 independent representations of the intersection :
- a 3D NURBS curve
- a 2D NURBS curve (parametric space of surface A)
- a 2D NURBS curve (parametric space of surface B)

- In practice, there are numerical approximations
  - NURBS are finite approximation spaces; therefore approximation/interpolation errors do occur.
  - The use of floating point numbers with a finite binary representation of the mantissa lead to numerical errors
- There is no robust way to ensure, in **a geometrical sense**, that a curve located on surface A is the same as the corresponding curve on surface B, *i.e.* that both surfaces are neighbours, and share the same edge.

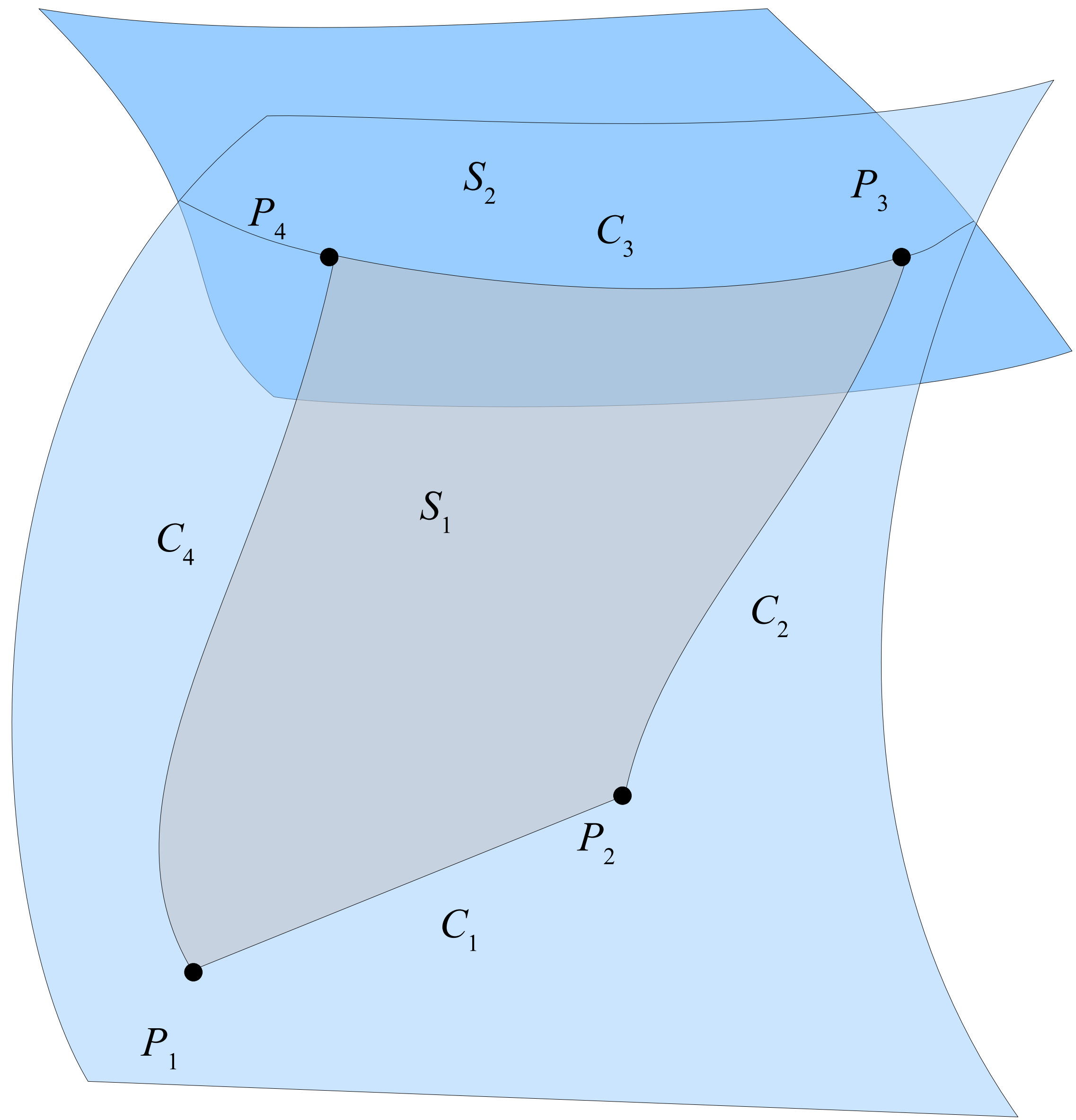
Solid modelling

- Definition of a **topology** : non geometric relations between entities.
- This allows to unify the calculations (of points, normals, etc...) on entities shared (or bounding) other entities (*eg.* an edge shared by surfaces).
- It also allows the explicit definition of volumes – from the surfaces that bound the volume.
- It may also solve the problem of orientation of surfaces

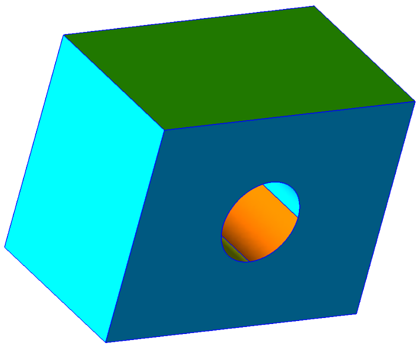
Solid modelling

- Topology

B-REP model



- B-Rep model
  - « Boundary representation »
  - Model based on the representation of surfaces
  - Model of exchange (STEP format) and definition
  - The “natural” set of operators is richer than for CSG
    - Extrusion, chamfer etc ...
  - Does not carry the history of construction of the model (whereas CSG usually does)



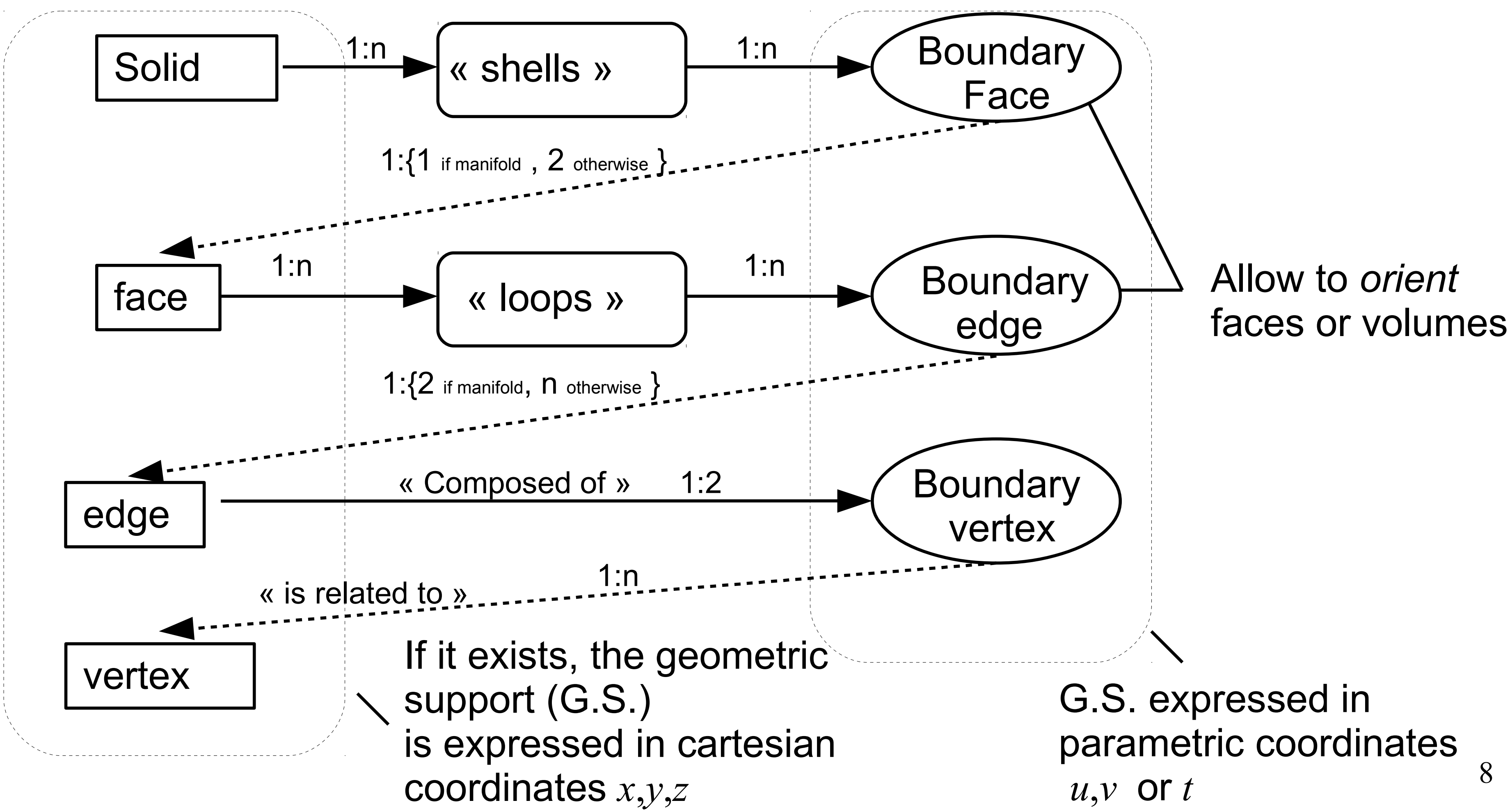
- B-Rep model
  - Consists of two types of information :
    - Geometric

      Geometric information is used for defining the spatial position, the curvatures, etc...

      That's what we have seen until now – NURBS curves and surfaces !
    - Topological

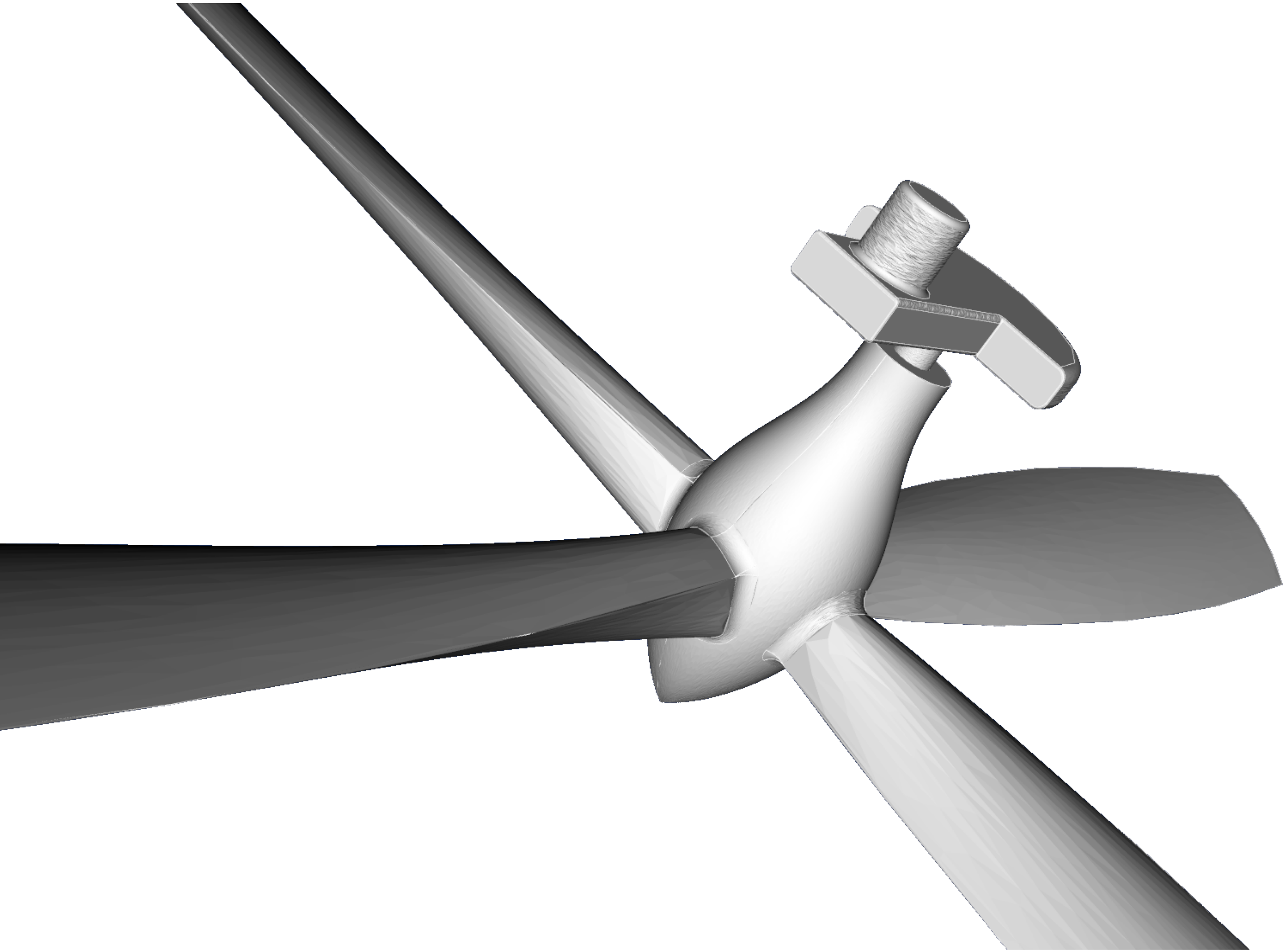
      This allows to make links between geometrical entities.
  - Two types of entities
    - Geometric entities: (volume), surface, curve, point
    - Topological entities : solid, face, edge, vertex
  - A topological entity “lies on” a geometric entity, which is its geometrical support (when existing)

- B-Rep model
  - Complete hierarchical model



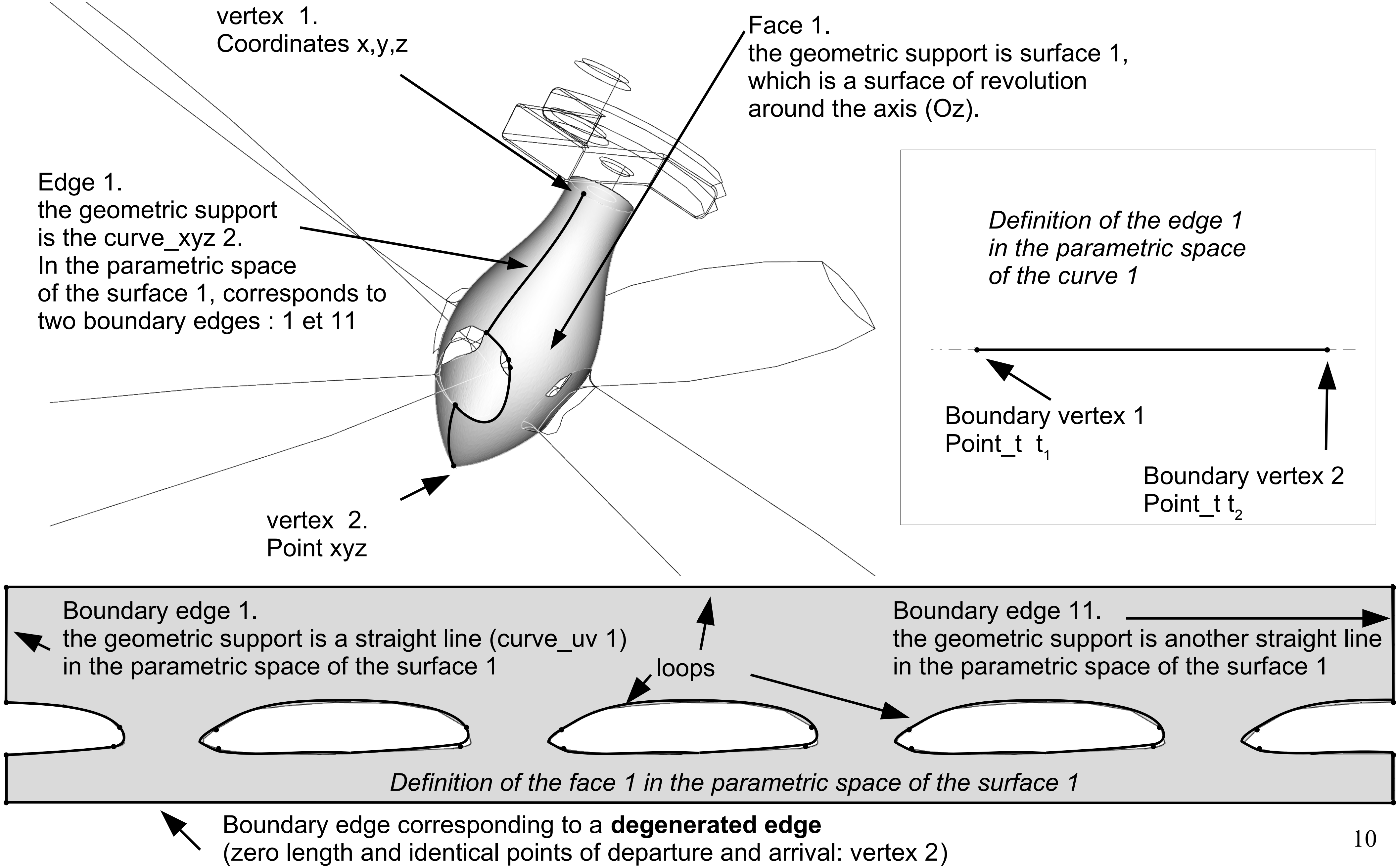
Computer Aided Design

Solid modelling



Computer Aided Design

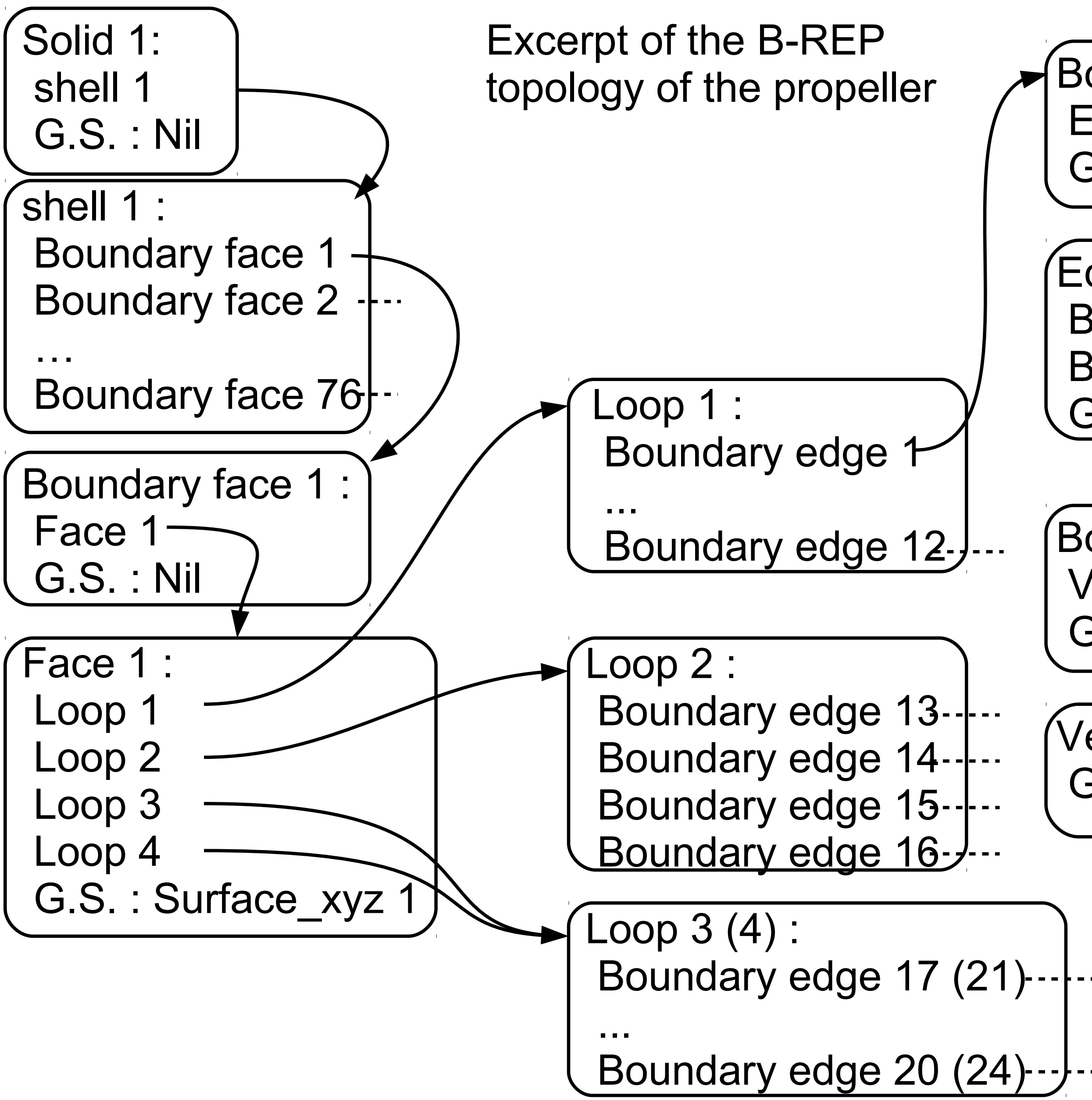
Solid modelling



Computer Aided Design

Solid modelling

Excerpt of the B-REP topology of the propeller



Computer Aided Design

Solid modelling

Links between the B-REP topology and the actual geometry of the propeller

Boundary edge 1
Edge 1
G.S. : Curve_uv 1

Curve_uv 1 :
Straight line
application (t) → (u,v)

Edge 1
Boundary vertex 1
Boundary vertex 2
G.S. : Curve_xyz 2

Curve_xyz 2 :
NURBS Curve
application (t) → (x,y,z)

Boundary vertex 1
vertex 1
G.S. : Point_t 1

Point_t 1
t = t1

Vertex 1
G.S. : Point_xyz 2

Point_xyz 2
x=x1, y=y1, z=z1

Face 1 :
Loop 1
Loop 2
Loop 3
Loop 4
G.S. : Surface_xyz 1

Surface_xyz 1 :
Surface of revolution
NURBS Surface
application (u,v) → (x,y,z)

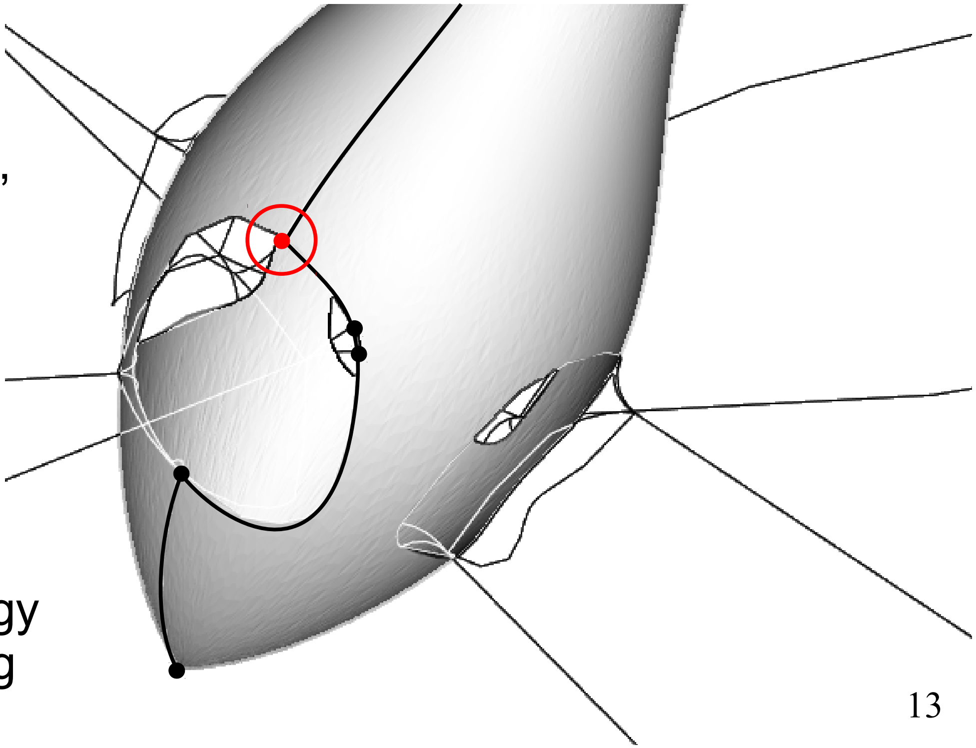
How to obtain the (x,y,z) coordinates of the encircled point ?

1 – Use the 3D vertex directly (Point_xyz xxx)

2 – Use the boundary vertices for every 3D edge (there are 3 such edges) (Point_t t1,t2,t3)
Then use those (t) to get (x,y,z) by the 3D edges

3 – Use the boundary vertices of the 2D boundary edges in the face (there are 2 faces, so 4 of them), (Point_t t'1,t'2,t'3,t'4). Then use those (t) to obtain coordinates (u,v) in the parametric space of the face, thanks to 2D curves, finally, use those (u,v) to obtain (x,y,z) thanks to the geometry of the face.

So there exists 8 different ways. Nothing indicates that the 8 set of 3D coordinates are exactly equal (there are numerical approximations). Only topology allows us to say that those 8 points are all referring to the same point … at least conceptually.



- B-Rep model
  - Euler characteristic for polyhedra

$$\chi(S) = v - e + f$$

  - Euler – Poincaré formula

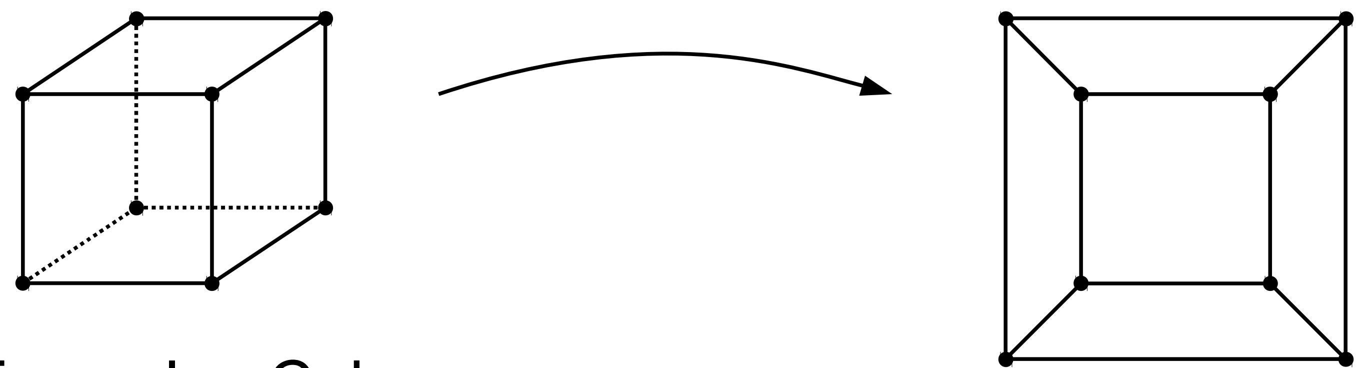
$$\chi(S) = v - e + f - r = 2(s - h)$$

with

v	=	number	of vertices
f	=	"	of faces
e	=	"	of edges
s	=	"	of solids (independent volumes)
h	=	"	of holes – going through (topol. gender)
r	=	"	of internal loops (ring)

Euler's formula

- Euler characteristic



Example : Cube

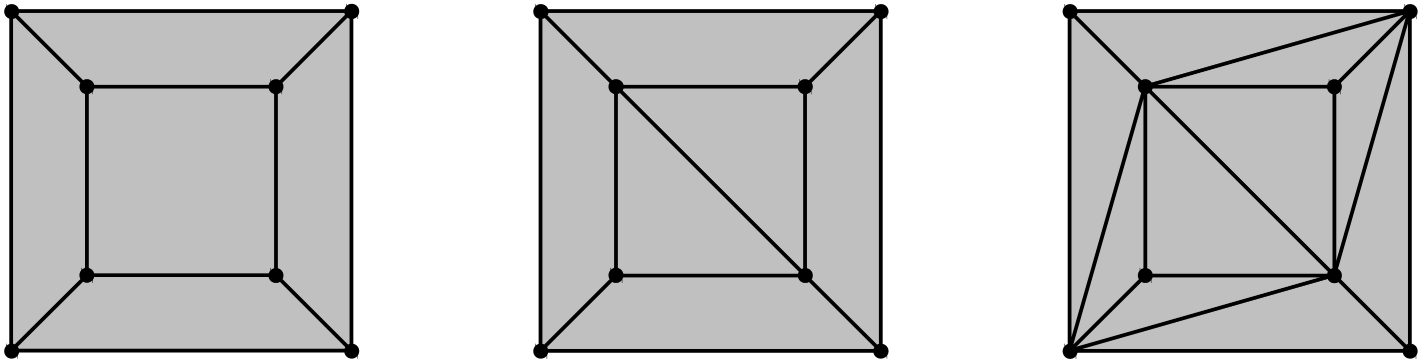
$v - e + f = \kappa$

Opened and flattened Cube

$v - e + f = \kappa - 1$

Step 0 : we take a face off the polyhedron and flatten it to obtain a plane graph

Euler's formula



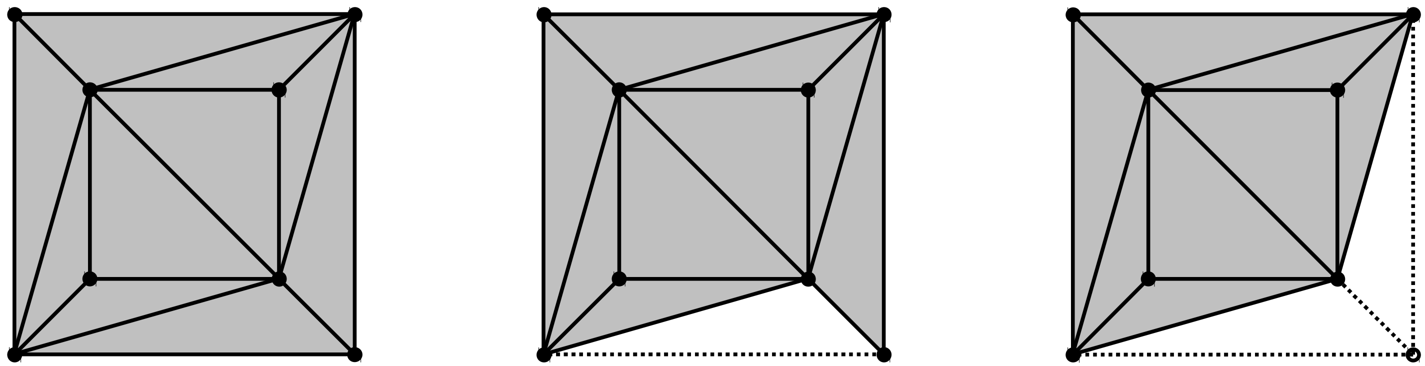
$v - e + f = \kappa - 1$

$+1e\,,\ +1f$
$v - e + f = \kappa - 1$

$+5e\,,\ +5f$
$v - e + f = \kappa - 1$

Step 1 : Repeat the following operation :
For each non triangular face, add one edge linking non related vertices.
Each tome, the number of edges and faces is increased by 1.
This is repeated until no non triangular faces remain.

Euler's formula



$v - e + f = \kappa - 1$

$-1e,\ -1f$

$v - e + f = \kappa - 1$

$-2e,\ -1f,\ -1v$

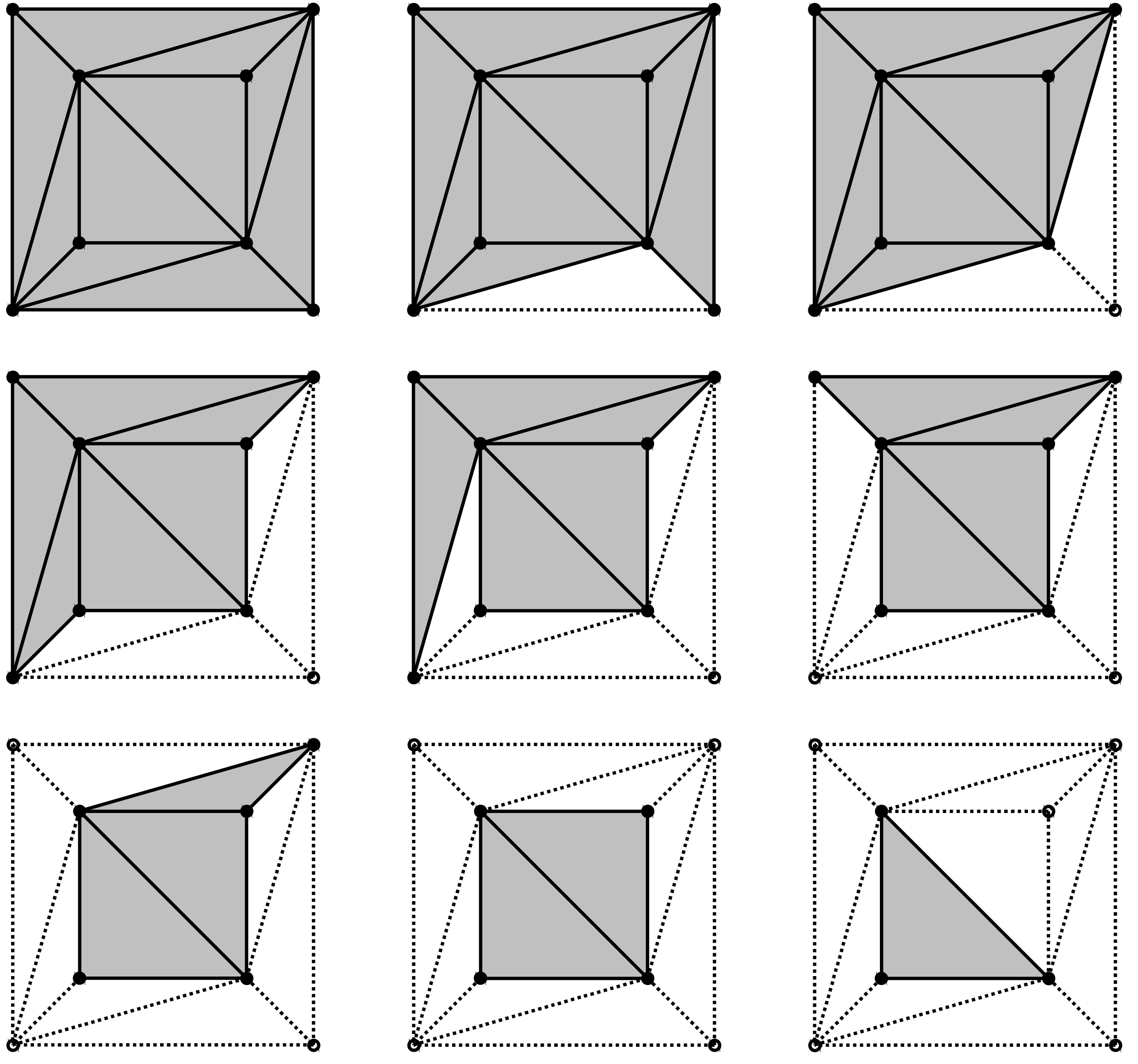
$v - e + f = \kappa - 1$

Step 2 : One alternates betwee these two operations

- Preferentially, delete triangles that have 2 boundary edges. Every time; *e* decreases by 2 and *f* and *v* by 1.
- Then, delete triangles with only one boundary edege. Each tile, *e* and *f* decrease by 1.

This until only one triangle remain.

Euler's formula



$$v - e + f = \kappa - 1$$

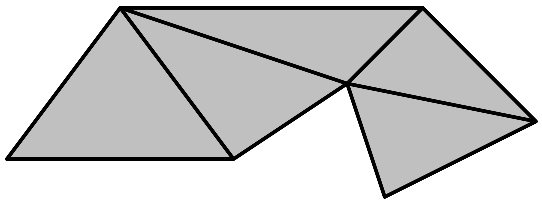
Euler's formula

- Every polygon can be decomposed into triangles
  Therefore, by applying the three operations described in the previous slides, we can transform the planar graph into a triangle without changing Euler's characteristic. The triangle satisfies obviously

  $$v - e + f = \kappa - 1$$

  with $\kappa\text{-}1 = 1$
  Therefore, the planar graph verifies the formula.
- So the initial polyhedron satisfies :

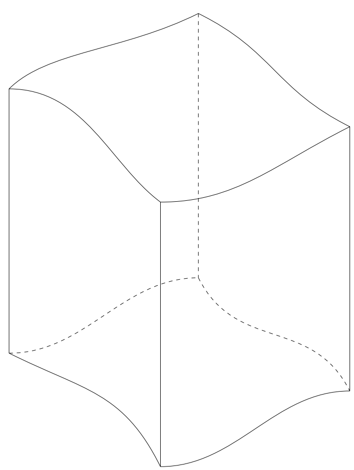
$$v - e + f = \kappa = 2$$



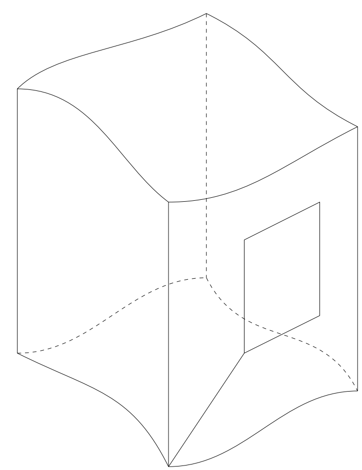
Computer Aided Design

Solid modelling

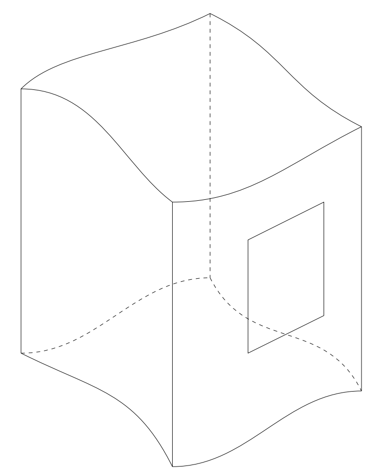
- Necessity to take “rings” into account - inside faces



polyhedron



polyhedron



Polyhedron with rings

$$\chi(S)=2$$

$$\chi(S)=v-e+f$$

$$\chi(S)=8-12+6=2$$

$$\chi(S)=12-17+7=2$$

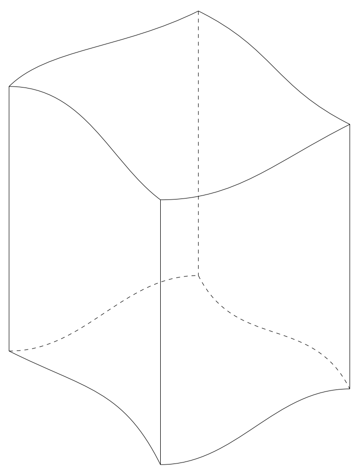
OK

$$\chi(S)=12-16+7\neq 2$$

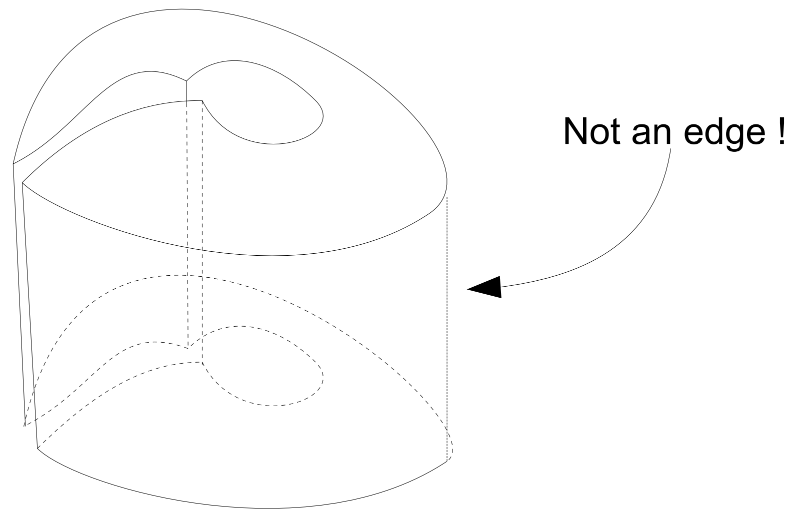
Not OK !
Contribution of the ring

$$\chi(S)=v-e+f-r=2$$

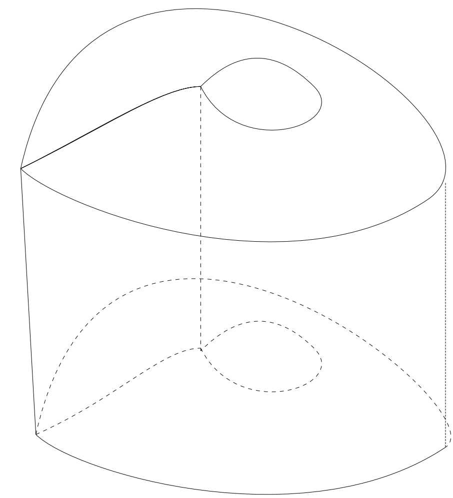
- Necessity to take “holes” into account



polyhedron



“Warped”polyhedron



Polyhedron with one hole, 4 edges less, 2 faces less, 4 vertices less

$$\chi(S)=2$$

$$\chi(S)=v-e+f$$

$$\chi(S)=8-12+6=2$$

$$\chi(S)=8-12+6=2$$

OK

$$\chi(S)=4-8+4\neq 2$$

Not OK !
Contribution of the hole

$$\chi(S)=v-e+f-r+2h=2$$

Solid modelling

- Every B-rep model is identifiable (topologically) to a « point » in a 6-dimensional vector space.
  - Vector space of coordinates v, e, f, s, h, r.
- Any topologically valid model shall verify the Euler-Poincaré relation
  - This relation defines an « hyperplane » (of dimension 5) in a 6-dimensional space
  - The equation of this hyperplane is :

$$v - e + f - 2s + 2h - r = 0$$

Solid modelling

$$v - e + f - 2s + 2h - r = 0$$

- We can update a valid solid and modify the 6 numbers characterising a model with a transformation that yields a valid solid for which :

$$v + \Delta v - e - \Delta e + f + \Delta f$$
$$- 2s - 2\Delta s + 2h + 2\Delta h - r - \Delta r = 0$$

$$\Rightarrow \Delta v - \Delta e + \Delta f - 2\Delta s + 2\Delta h - \Delta r = 0$$

  - In this way, add a vertex ($\Delta v = 1$) must be accompanied, one way or another, by addition of an edge ($\Delta e = 1$) OR of the withdrawal of a face ($\Delta f = -1$), etc...
  - Elementary operations satisfying the Euler-Poincaré relation are called **Euler operators**.
  - They allow staying on the « hyperplane » of validity while changing the topological configuration

- Euler operators
  - The use of Euler operators guarantees the *topological* validity of the result
  - Here we don't check the *geometric* validity (self-intersections etc...)
  - We identify them under the form : ***MaKb*** where ***M*** = Make ***K*** = Kill and ***a*** and ***b*** are a sequence of entities : vertex, edge, face, solid, hole or ring.
    - In total, there are 99 Euler operators aiming to modify the number of entities by **at most** one unit.
      - These are divided in 49 + 49 inverses, plus the identity operator.
    - Among those 49 operators , we can chose 5 linearly independent operators (the hyperplane has 5 dimensions)
    - Those 5 independent operators form a base for the hyperplane of topologically admissible models

- Example of a set of Euler operators

inverses

MEV, **M**ake an **E**dge and a **V**ertex
MEF, **M**ake an **E**dge and a **F**ace
MEKR, **M**ake an **E**dge and **K**ill a **R**ing
MVFS, **M**ake a **V**extex, a **F**ace and a **S**hell
KFMRH, **K**ill a **F**ace **M**ake a **R**ing and a **H**ole

KEV, **K**ill an **E**dge and a **V**ertex
KEF, **K**ill an **E**dge and a **F**ace
KEMR, **K**ill an **E**dge and **M**ake a **R**ing
KVFS, **K**ill a **V**ertex, a **F**ace and a **S**hell
MFKRH, **M**ake a **F**ace, **K**ill a **R**ing and a **H**ole

- Proof by Mäntylä (1984) that those operators allow to build every valid solid (since they are independent)
- Those operators form a base of the space of valid configurations (the « hyperplane »)

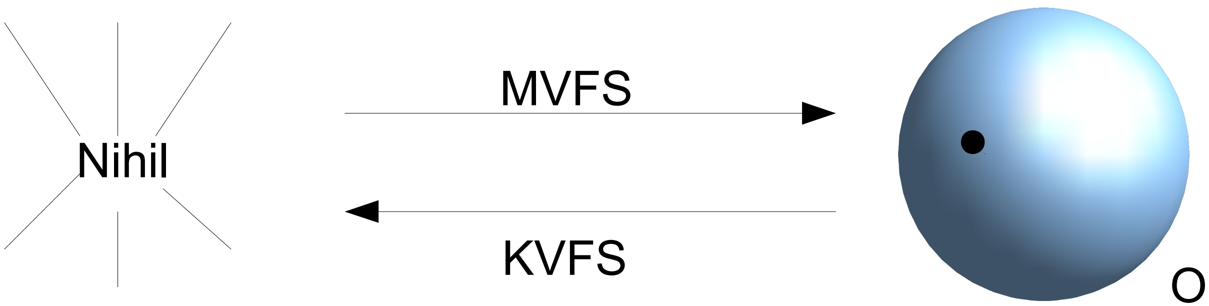
There are three types of operators in this set :

- Skeleton operators **MVFS** and **KVFS**
  - Allow to build/destroy elementary volumes
- Local operators **MEV**, **KEV**, **MEF**, **KEF**, **KEMR**, **MEKR**
  - Allow to modify connectivities for existing volumes
  - Don't modify fundamental topological characteristics of the surfaces - nb of handles/ holes (topological gender) and number of independent volumes
- Global operators **KFMRH** and **MFKRH**
  - Allow to add / remove “handles” (change the topological gender)

- Only the skeleton and global operators do change the topological gender.

- Euler operators
  - « Skeleton » operators

    MVFS; **KVFS**
    - Allow to « build » an « elementary » volume from void (which is an admissible topological structure) – or destroy it.



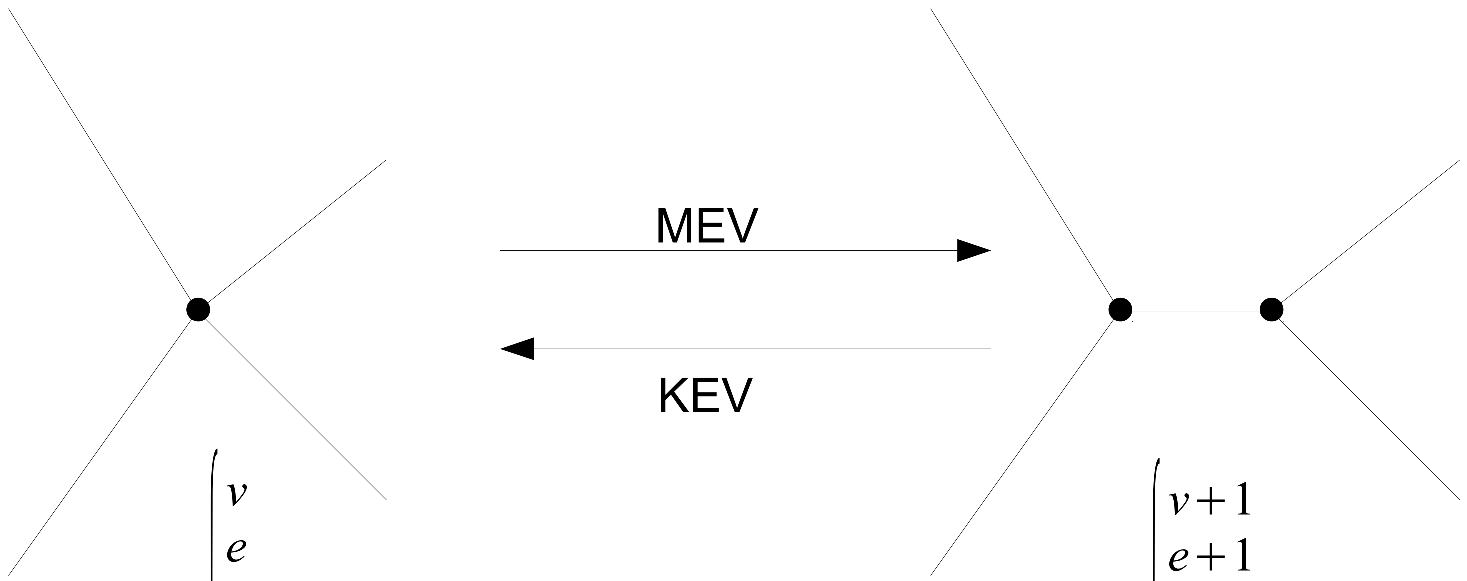
$$\begin{cases} v=0 \\ e=0 \\ f=0 \\ h=0 \\ r=0 \\ s=0 \end{cases}$$

$$\begin{cases} v=1 \\ e=0 \\ f=1 \\ h=0 \\ r=0 \\ s=1 \end{cases}$$

Only one face
- its boundary is reduced
to a single vertex

- Euler operators
  - Local operators

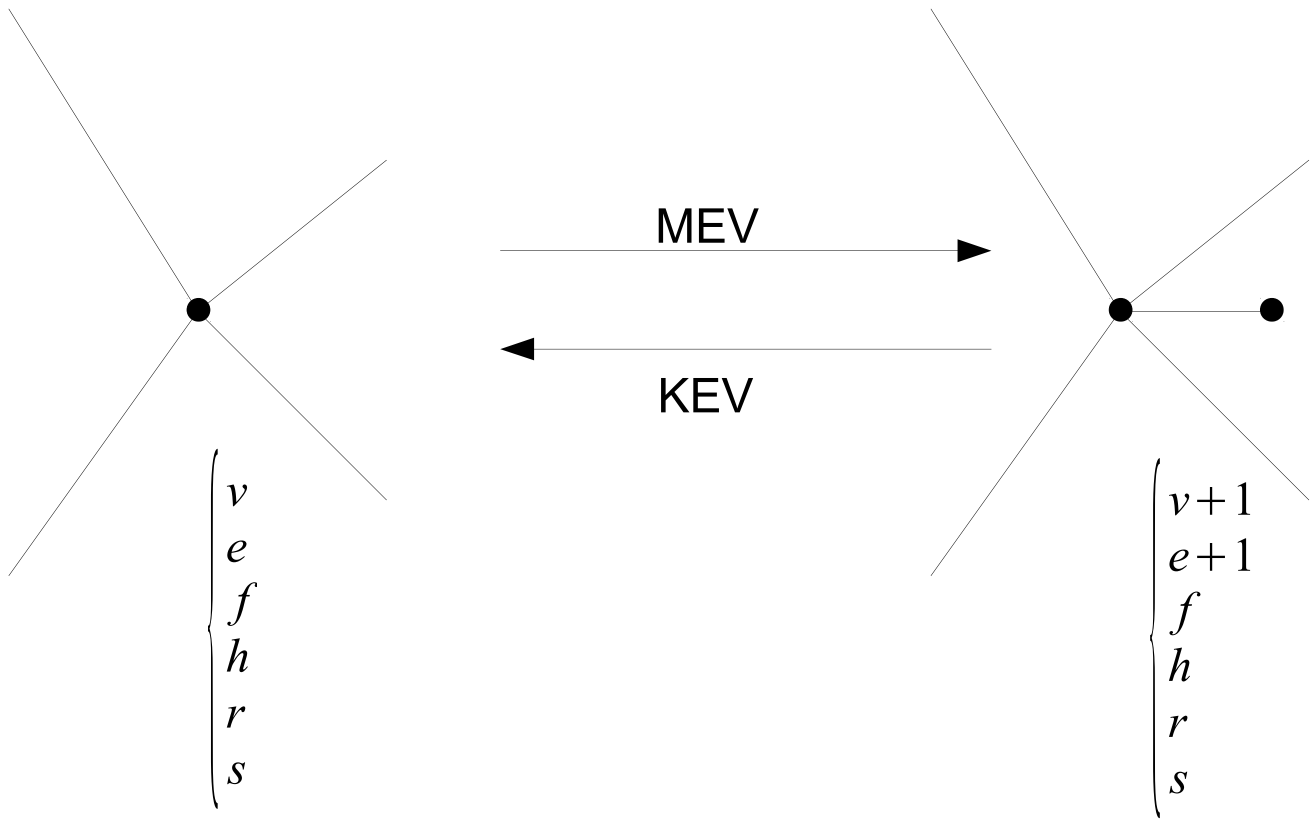
    MEV, **KEV** (case 1)



$$\begin{cases} v \\ e \\ f \\ h \\ r \\ s \end{cases} \qquad \begin{cases} v+1 \\ e+1 \\ f \\ h \\ r \\ s \end{cases}$$

- Euler operators
  - Local operators

    MEV, **KEV** (case 2)



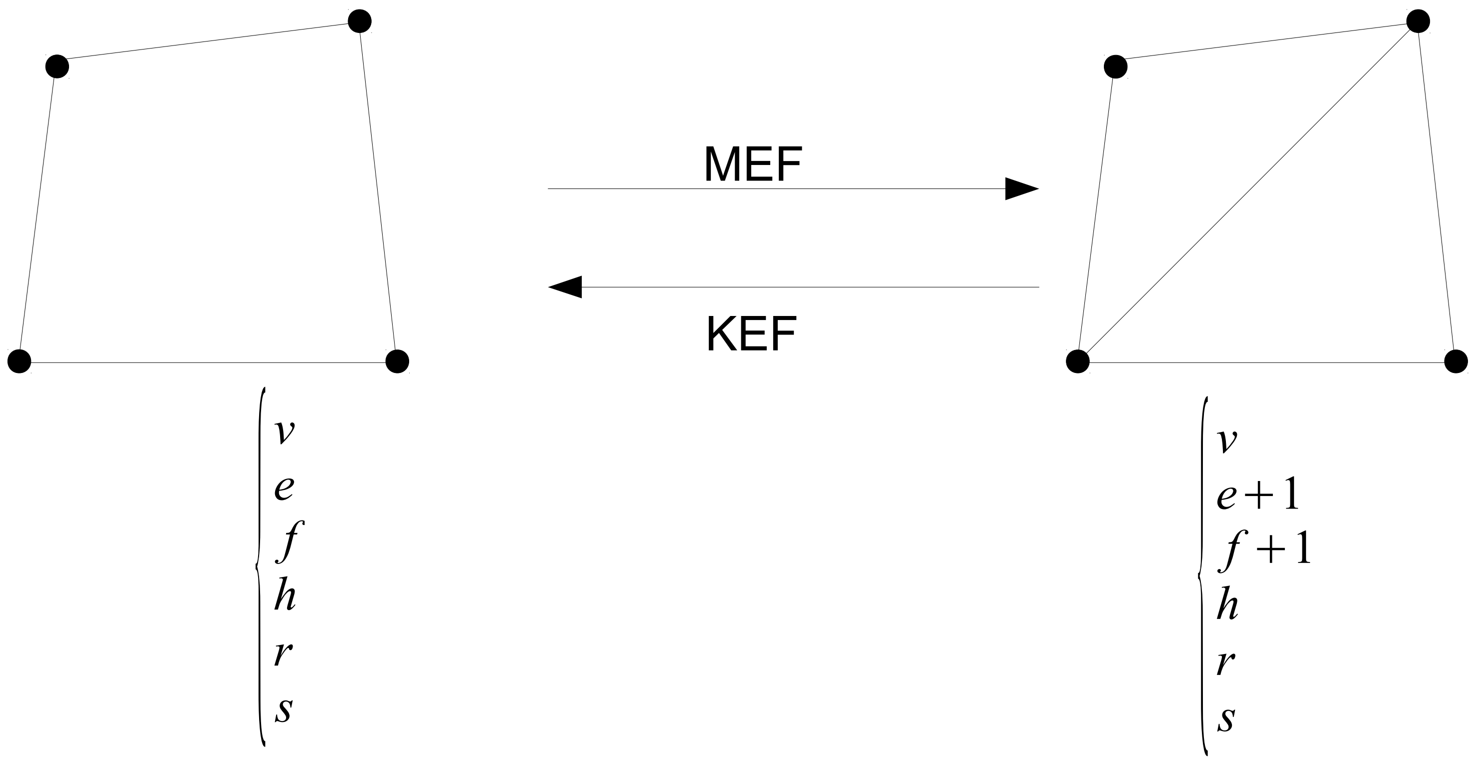
- Euler operators
  - Local operators

    MEV, **KEV** (case 3)

$$\begin{cases} v=1 \\ e=0 \\ f=1 \\ h=0 \\ r=0 \\ s=1 \end{cases} \qquad \begin{cases} v=2 \\ e=1 \\ f=1 \\ h=0 \\ r=0 \\ s=1 \end{cases}$$

- Euler operators
  - Local operators

    MEF, **KEF** (case 1)



- Euler operators
  - Local operators

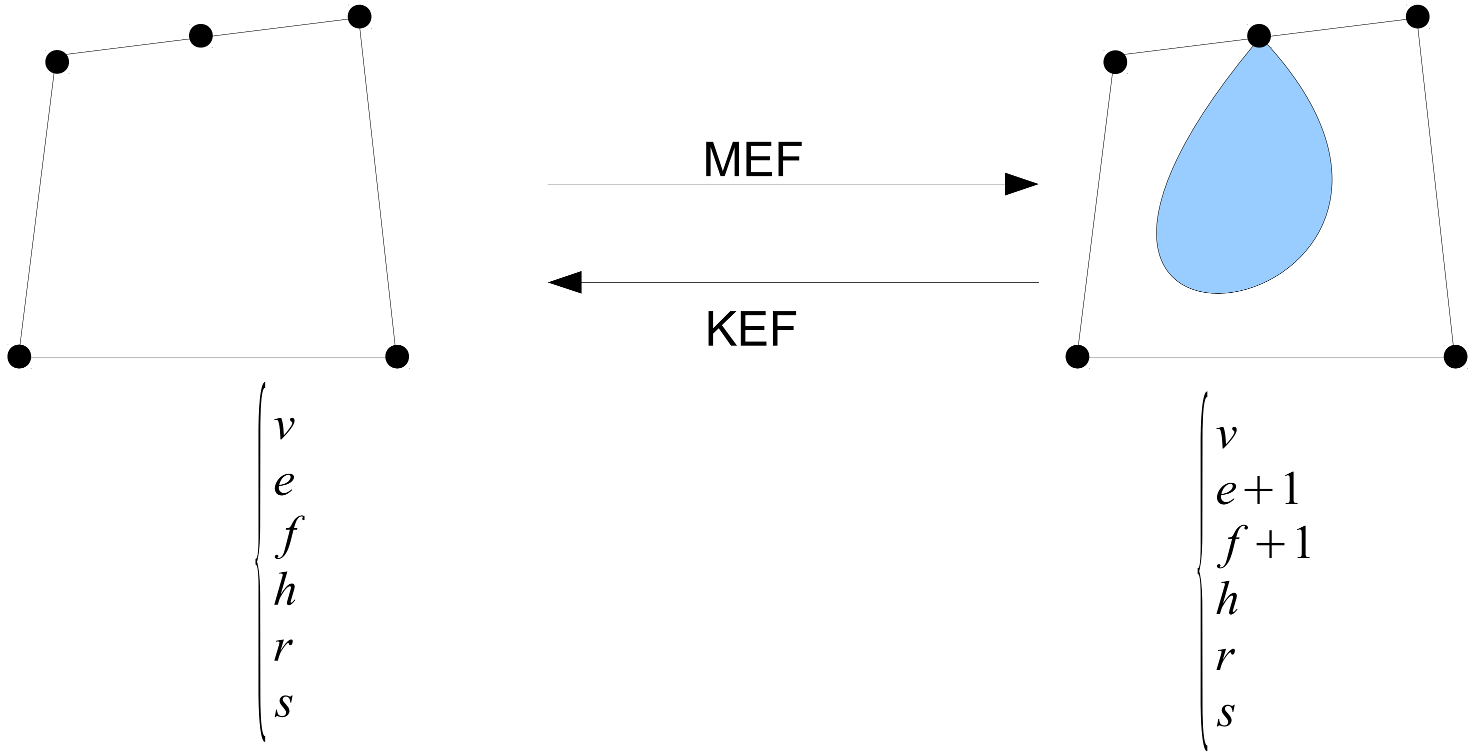
    MEF, **KEF** (case 2)

$$\begin{cases} v=1 \\ e=0 \\ f=1 \\ h=0 \\ r=0 \\ s=1 \end{cases}$$

$$\begin{cases} v=1 \\ e=1 \\ f=2 \\ h=0 \\ r=0 \\ s=1 \end{cases}$$

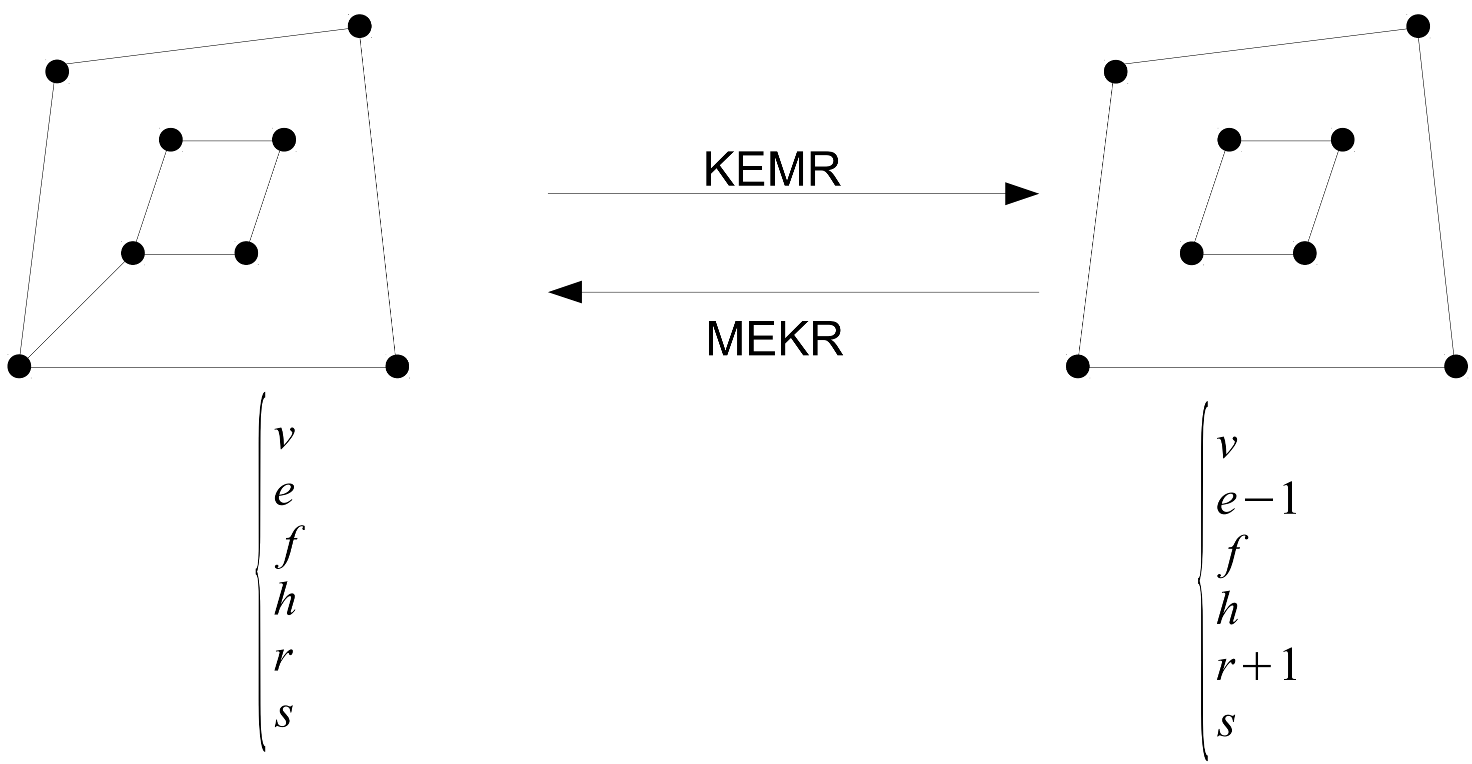
- Euler operators
  - Local operators

    MEF, **KEF** (case 3)



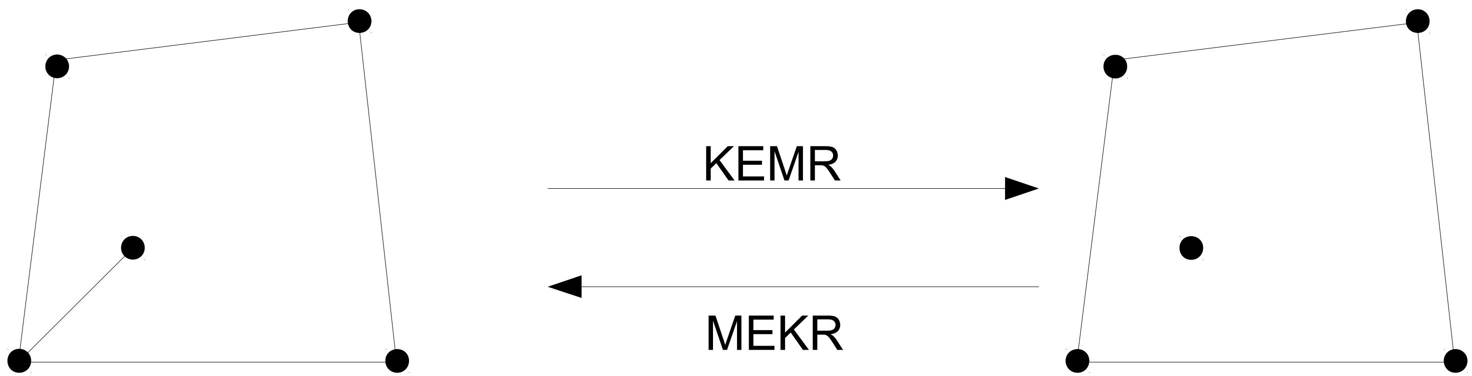
- Euler operators
  - Local operators

    KEMR, **MEKR** (case 1)



- Euler operators
  - Local operators

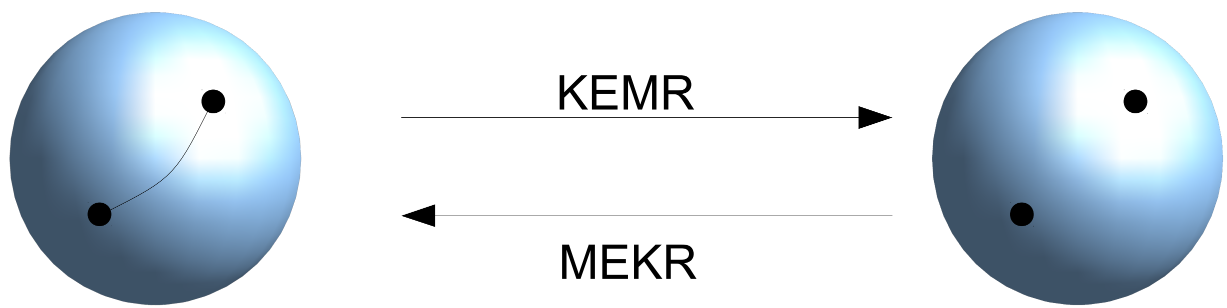
    KEMR, **MEKR** (case 2 : the loop for the internal ring is reduced to a single vertex)



$$\begin{cases} v \\ e \\ f \\ h \\ r \\ s \end{cases} \qquad \begin{cases} v \\ e-1 \\ f \\ h \\ r+1 \\ s \end{cases}$$

- Euler operators
  - Local operators

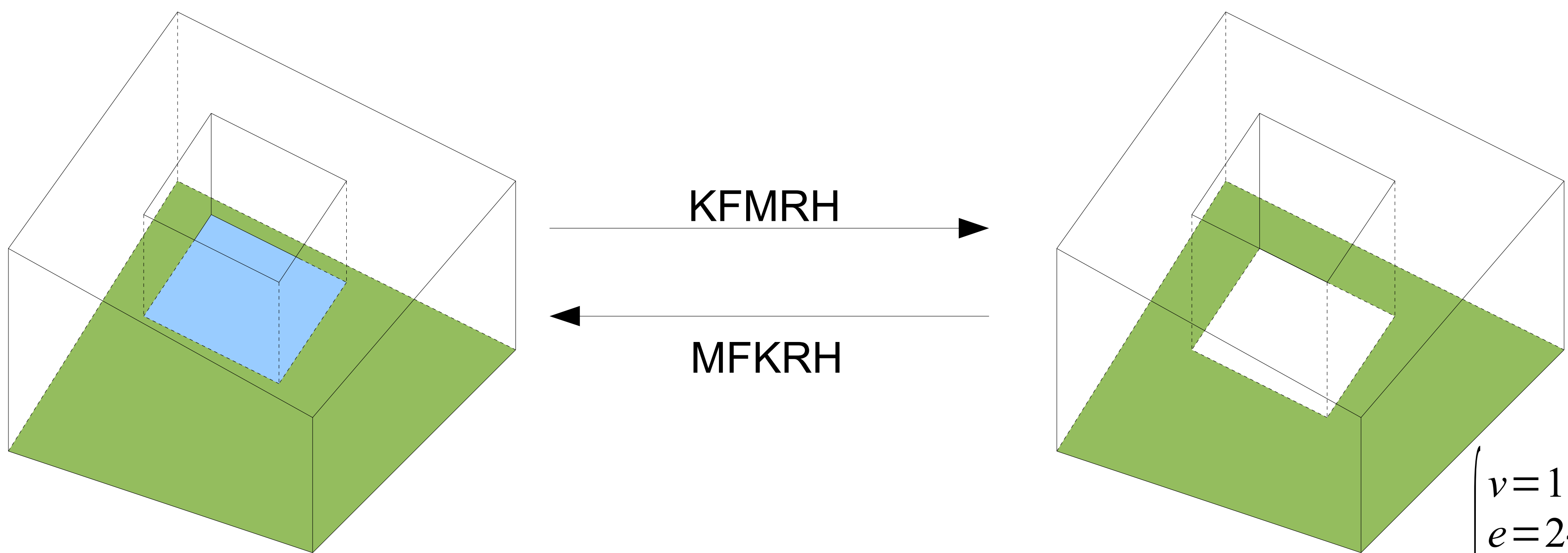
    KEMR, **MEKR** (case 3 : both loops are reduced to one vertex – one is the ring; the other is the external loop of the face)



$$\begin{cases} v=2 \\ e=1 \\ f=1 \\ h=0 \\ r=0 \\ s=1 \end{cases} \qquad \begin{cases} v=2 \\ e=0 \\ f=1 \\ h=0 \\ r=1 \\ s=1 \end{cases}$$

- Euler operators
    - Global operators

      KFMRH, **MFKRH** (case 1 : allow the creation / destruction of holes in a solid)

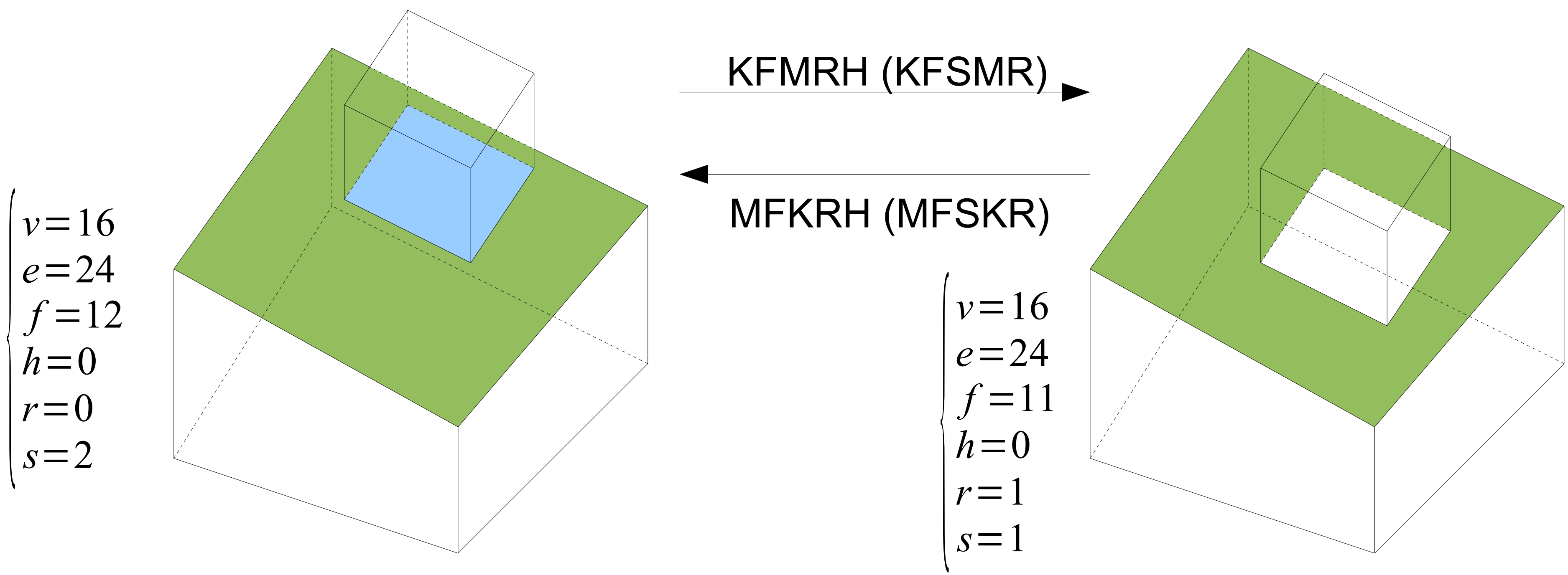


$$\begin{cases} v=16 \\ e=24 \\ f=11 \\ h=0 \\ r=1 \\ s=1 \end{cases}$$

$$\begin{cases} v=16 \\ e=24 \\ f=10 \\ h=1 \\ r=2 \\ s=1 \end{cases}$$

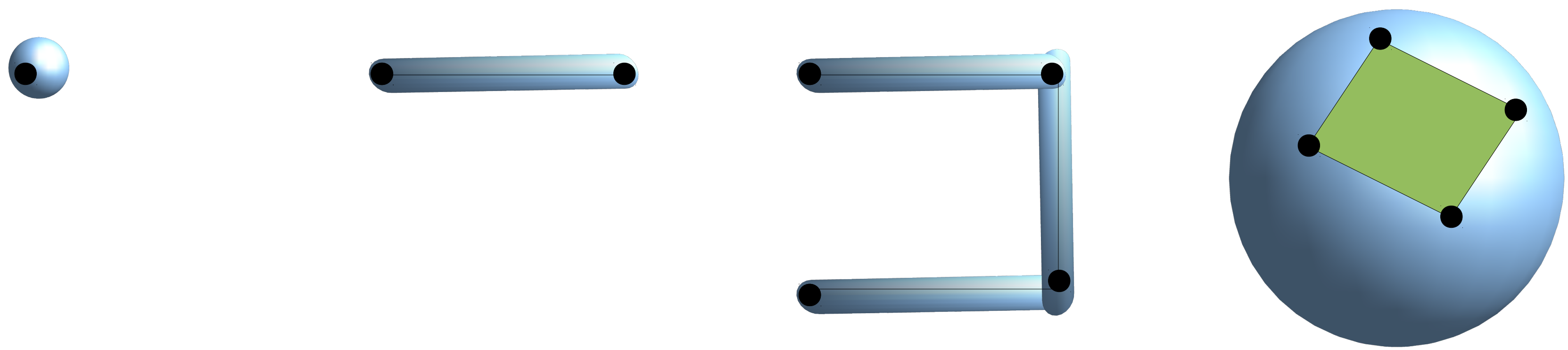
- Euler operators
  - Global operators

    KFMRH, **MFKRH** (case 2) : join two independent solids : here more judiciously called **K**ill **F**ace, **S**olid and **M**ake **R**ing (**KFSMR**)
  - Interpretation of global operators is sometimes confusing



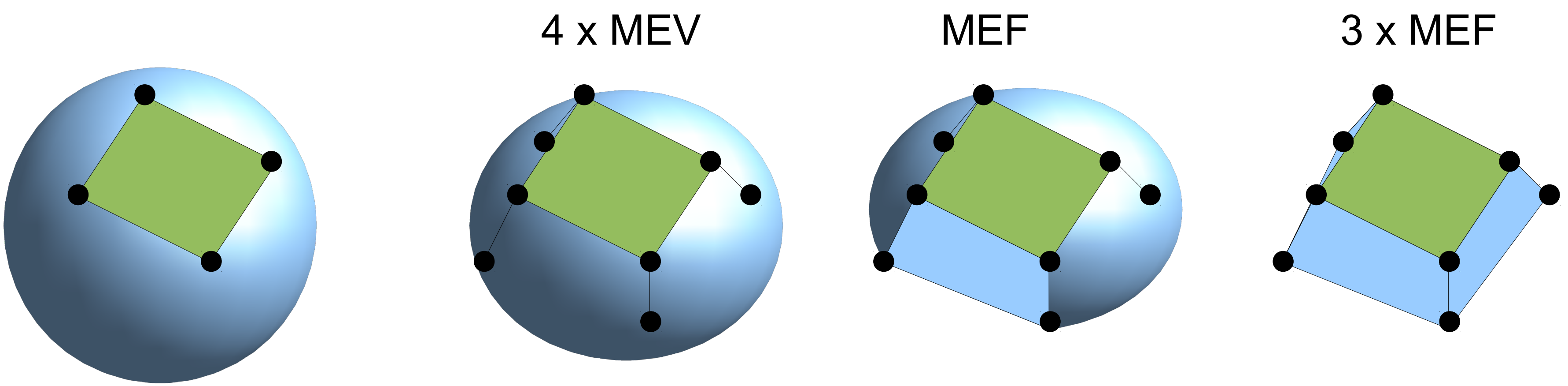
- Example of use of Euler operators

MVFS MEV 2 x MEV MEF



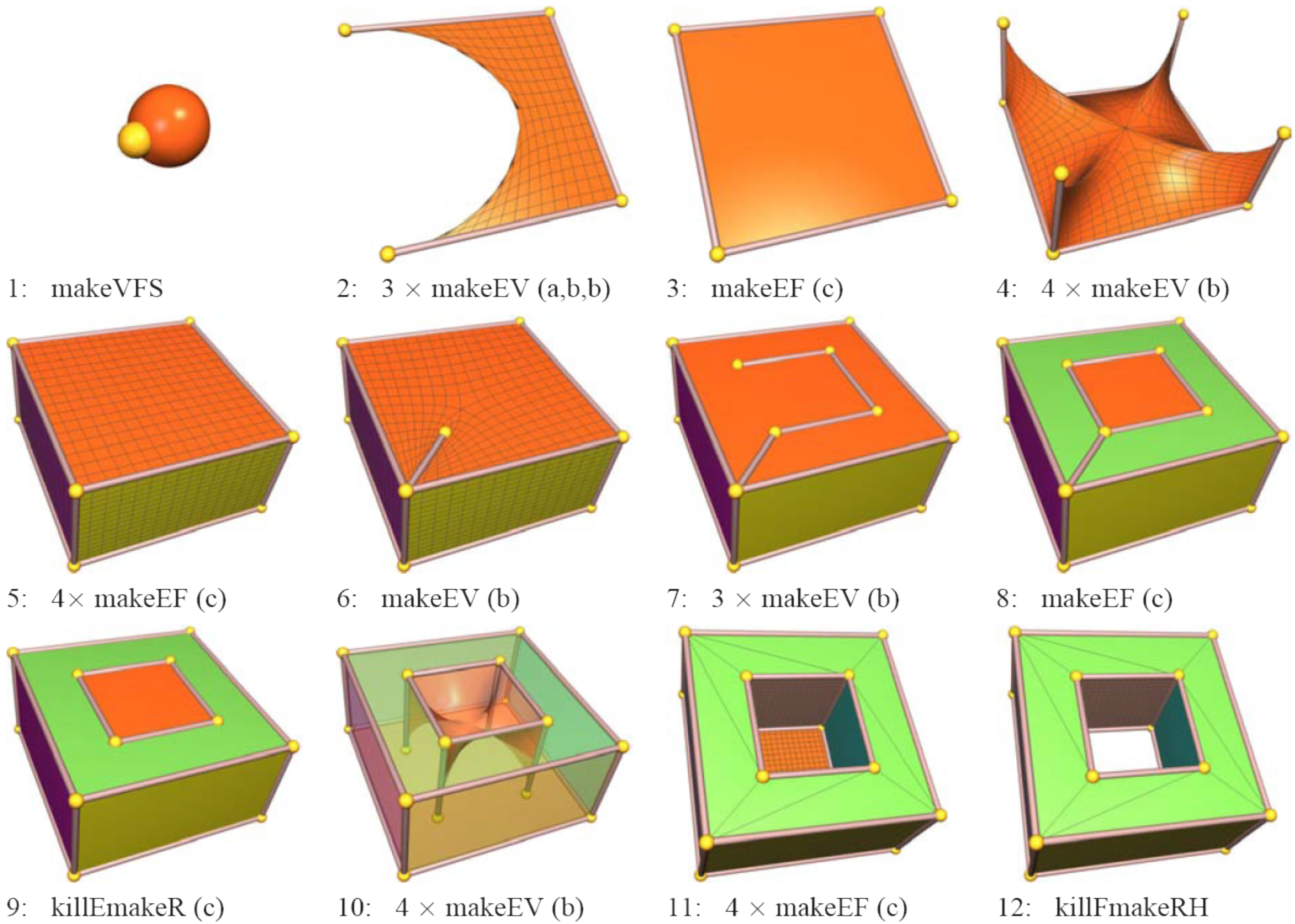
$$\begin{cases} v=1 \\ e=0 \\ f=1 \\ h=0 \\ r=0 \\ s=1 \end{cases} \qquad \begin{cases} v=2 \\ e=1 \\ f=1 \\ h=0 \\ r=0 \\ s=1 \end{cases} \qquad \begin{cases} v=4 \\ e=3 \\ f=1 \\ h=0 \\ r=0 \\ s=1 \end{cases} \qquad \begin{cases} v=4 \\ e=4 \\ f=2 \\ h=0 \\ r=0 \\ s=1 \end{cases}$$

- Example of use of Euler operators



$$\begin{cases} v=4 \\ e=4 \\ f=2 \\ h=0 \\ r=0 \\ s=1 \end{cases} \qquad \begin{cases} v=8 \\ e=8 \\ f=2 \\ h=0 \\ r=0 \\ s=1 \end{cases} \qquad \begin{cases} v=8 \\ e=9 \\ f=3 \\ h=0 \\ r=0 \\ s=1 \end{cases} \qquad \begin{cases} v=8 \\ e=12 \\ f=6 \\ h=0 \\ r=0 \\ s=1 \end{cases}$$

- Example of use of Euler operators



S. Havemann

- Those operators have a vectorial form in the basis of elementary entities

```
  v  e  f  h  r  s
( 1, 1, 0, 0, 0, 0) – MEV, Make an Edge and a Vertex
( 0, 1, 1, 0, 0, 0) – MEF, Make a Face and an Edge
( 0,-1, 0, 0, 1, 0) – KEMR, Kill an Edge Make a Ring
( 1, 0, 1, 0, 0, 1) – MVFS, Make a Vertex, a Face and a Solid
( 0, 0,-1, 1, 1, 0) – KFMRH,  Kill a Face, Make a Ring and a Hole
```

  - In order to have a complete basis of the configuration space, a vector orthogonal to the hyperplane of acceptable configurations must be added

$$v-e+f-2s+2h-r=0$$

  - The coefficients of the equation of hyperplane are precisely the coordinates of the orthogonal vector...

```
  v  e  f  h  r  s
( 1,-1, 1, 2,-1,-2) – Euler-Poincaré
```

- Any transformation can thus be expressed easily using matrix operations
  - $\mathbf{A}$ is a basis of the topological configurations space
  - The columns of $\mathbf{A}$ are the variation of the number of entities for each operator, and the E-P relation.

$$\mathbf{A}=\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & -1 & 0 & 0 & -1 \\ 0 & 1 & 0 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 & 0 & -2 \end{pmatrix}$$

Columns corresponding to each of the Euler operators

Column corresponding to Euler-Poincaré's relation

$$q=\mathbf{A}\cdot p$$

Vector representing the number of times that each operator is applied

Vector representing the number (or the variation of the number) of elementary entities

- $\mathbf{A}$ is composed of linearly independent vectors, thus one can get the inverse...

$$q = \mathbf{A} \cdot p$$
$$\mathbf{A}^{-1} \cdot q = \mathbf{A}^{-1} \cdot \mathbf{A} \cdot p$$
$$p = \mathbf{A}^{-1} \cdot q$$

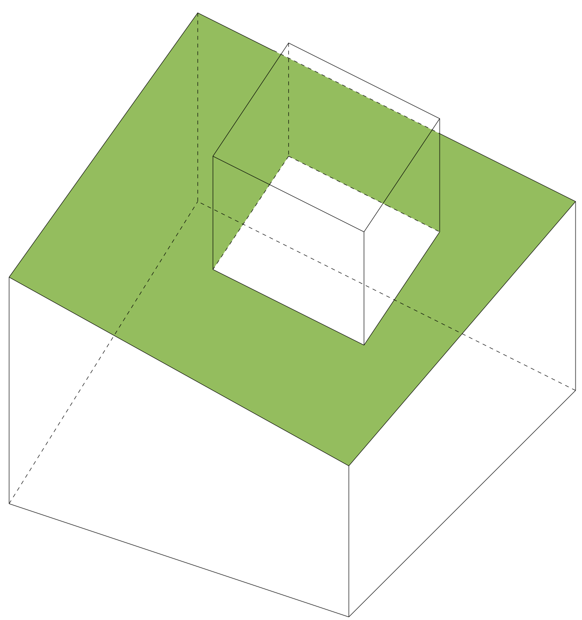
$$\mathbf{A}^{-1} = \frac{1}{12} \begin{pmatrix} 9 & 3 & -3 & -6 & 3 & -6 \\ -5 & 5 & 7 & 2 & 5 & -2 \\ 3 & -3 & 3 & -6 & 9 & -6 \\ 2 & -2 & 2 & 4 & -2 & 8 \\ -2 & 2 & -2 & 8 & 2 & 4 \\ 1 & -1 & 1 & 2 & -1 & -2 \end{pmatrix}$$

Vector representing the number of times each operator is applied
They are the **Euler Coordinates**

Vector representing the number of elementary entities

A Vector that is orthogonal to the « hyperplane »...

- Determination of elementary operations ...



$$q=\begin{cases} v=16 \\ e=24 \\ f=11 \\ h=0 \\ r=1 \\ s=1 \end{cases}=(16,24,11,0,1,1)$$

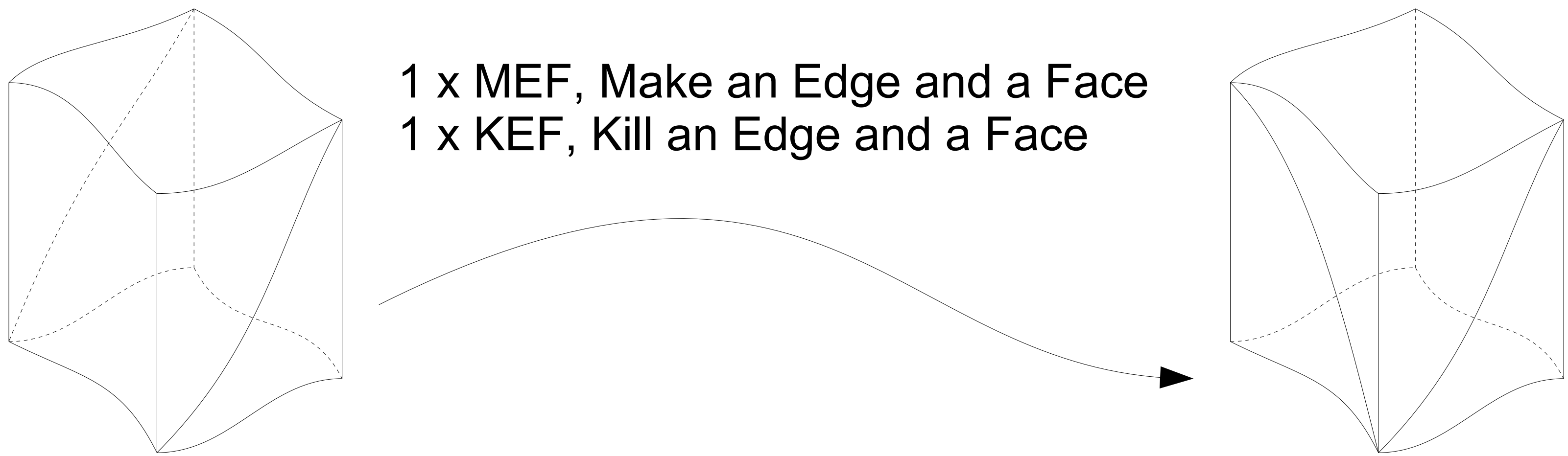
$$p=\mathbf{A}^{-1}\cdot q$$

$$p=(15,10,1,1,0,0)^{\mathrm{T}}$$

15 x MEV, Make an Edge and a Vertex
10 x MEF, Make a Face and an Edge
1 x MVFS, Make a Vertex, a Face and a Solid
1 x KEMR, Kill an Edge Make a Ring
0 x KFMRH, Kill a Face, Make a Ring and a Hole

The vector q respects the Euler-Poincaré relation

- Are Euler coordinates sufficient to define the topology of a solid ? → No.



7 x MEV, Make an Edge and a Vertex
7 x MEF, Make an Edge and a Face
1 x MVFS, Make a Vertex, a Face and a Solid
0 x KEMR, Kill an Edge Make a Ring
0 x KFMRH, Kill a Face, Make a Ring and a Hole

7 x MEV
7 x MEF
1 x MVFS
0 x KEMR
0 x KFMRH

Identical Euler coordinates

- Each Euler operator takes a certain number of parameters, in principle the entities to destroy/ or replace, and data necessary to creation of new entities.
- These depend on the structure of data used to represent the B-Rep object
- Application of an Euler operator is not always possible, the entities involved must exist and respect some conditions

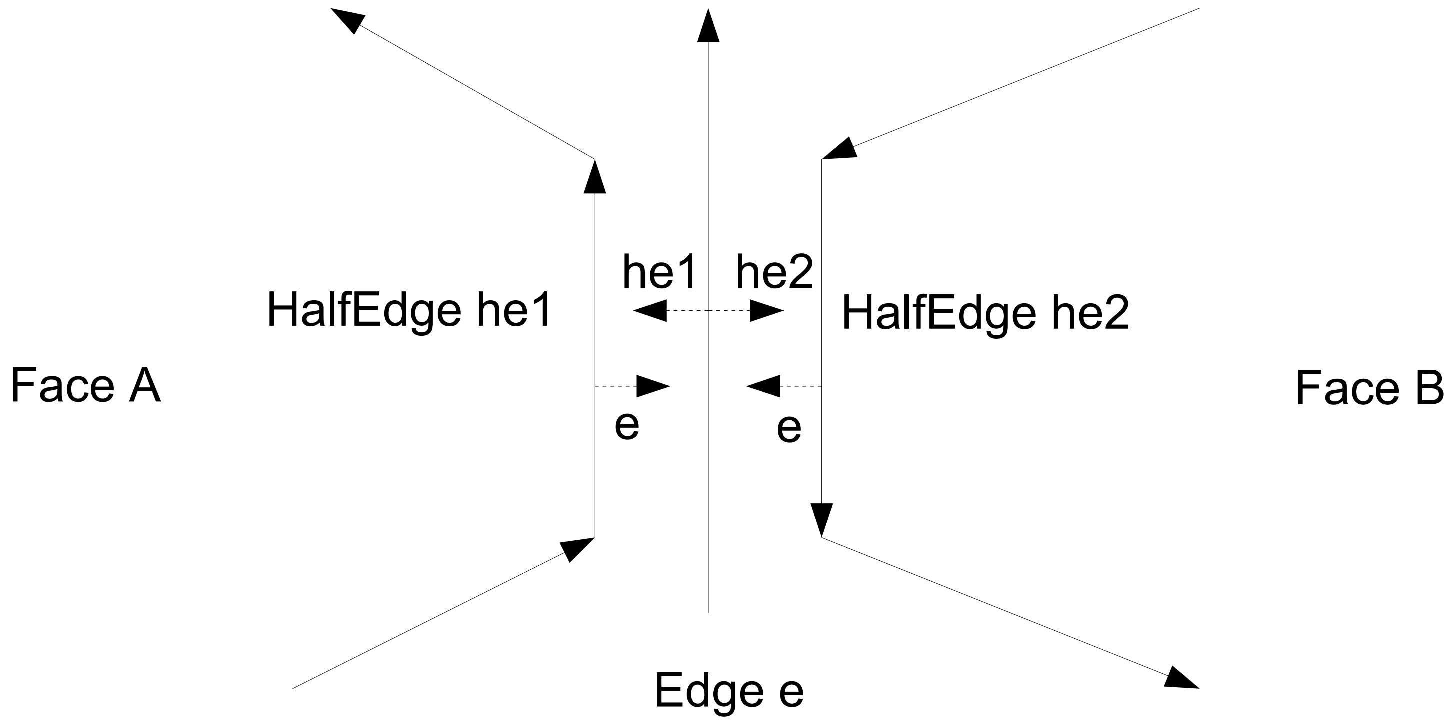
  KEF for example may only be applied on an edge separating two distinct faces... if not, one does not remove any face from the model !

- Conditions of application of Euler operators

MEV, Make an Edge and a Vertex	Empty space
KEV, Kill an Edge and a Vertex	The edge has two distinct vertices
MEF, Make an Edge and an Face	Vertices belonging to the same boundary loop of one face
KEF, Kill an Edge and a Face	Distinct faces located on both sides of the edge
MEKR, Make an Edge and Kill a Ring	Vertices belong to distinct boundary loops of the same face
KEMR, Kill an Edge and Make a Ring	Same face located on both sides of the edge, which is not part of a ring
MVFS, Make a Vextex, a Face and a Shell	Empty space
KVFS, Kill a Vertex, a Face and a Shell	The shell (solid) has no edges and has only one vertex (elementary volume)
KFMRH, Kill Face Make a Ring and a Hole	The face cannot hold any ring
MFKRH, Make a Face, Kill a Ring and a Hole	May be only applied to a ring

- Some examples of the application of Euler operators (not shown here)
  - Extrusion of a face
  - Junction of two solids
  - Cutting out a solid by a plane
  - Boolean operations between solids

- The most used data structure in a manifold B-Rep representation : Half-Edge data structure



- Basic entities
  - *shell* contains :
    - Solid number
    - Reference to *face* , *edge*, *vertex* of solid
  - *face* contains :
    - Face number
    - Ref. to an external *loop*
    - Ref. to a list of internal *loop*
    - Ref. to *shell*
    - Ref. to surface – nurbs or other – (the geometric support)

- *loop* contains :
  - Ref. to a list of *halfedge*
  - Ref. to *face*
- *edge* contains :
  - Ref. to *halfedge* of straight line
  - Ref. To the left *halfedge*
  - Ref. to a curve – nurbs or other – (the geometric support)
- *halfedge* contains :
  - Ref. to the parent *edge*
  - Ref. to the starting *vertex*
  - Ref. to the holding *loop*

- *vertex* contains :
  - Vertex number
  - Reference to one of the halfedges
  - Coordinates (the geometric support)

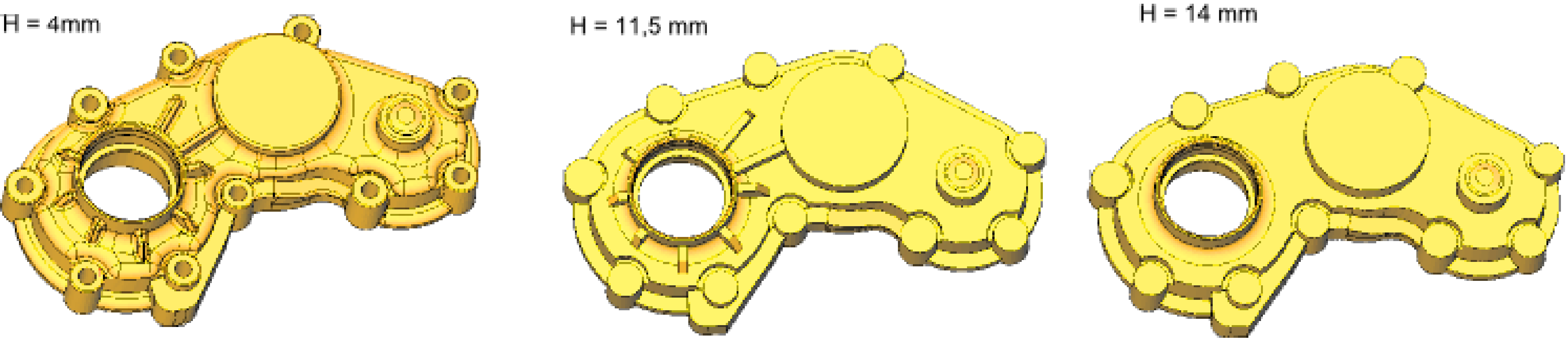
- A simplified implementation (without geometry other than vertices coordinates) in C++ of a B-rep modeller based on these ideas is available:

http://www.cs.utah.edu/~xchen/euler-doc/

Computer Aided Design

Solid modelling

- B-Rep model
  - Possibility of automatic topological operations
    - Here, elimination of small features in order to generate a mesh for numerical simulation in mechanical engineering.



- Bibliographic note

M. Mäntylä, An Introduction to Solid Modelling, Computer Science Press, 1988

I. Stroud, Solid Modelling and CAD Systems : How to Survive a CAD System, Springer, 2011(available on-line from the university campus)