Modelagem Geométrica de Sólidos

CIV 2802 – Sistemas Gráficos para Engenharia

Luiz Fernando Martha (PUC-Rio)
André Maués Brabo Pereira (UFF)

Conteúdo

- Motivação
- Modelagem de Sólidos
- Modelagem em Engenharia
- Modelagem Geométrica
- Modelagem Paramétrica

Introdução

Desenho

Abordagem Tradicional - Primeira Geração de CAD

(*Computer Aided Design*)

Desenho

Abordagem Tradicional - Primeira Geração de CAD

(*Computer Aided Design*)

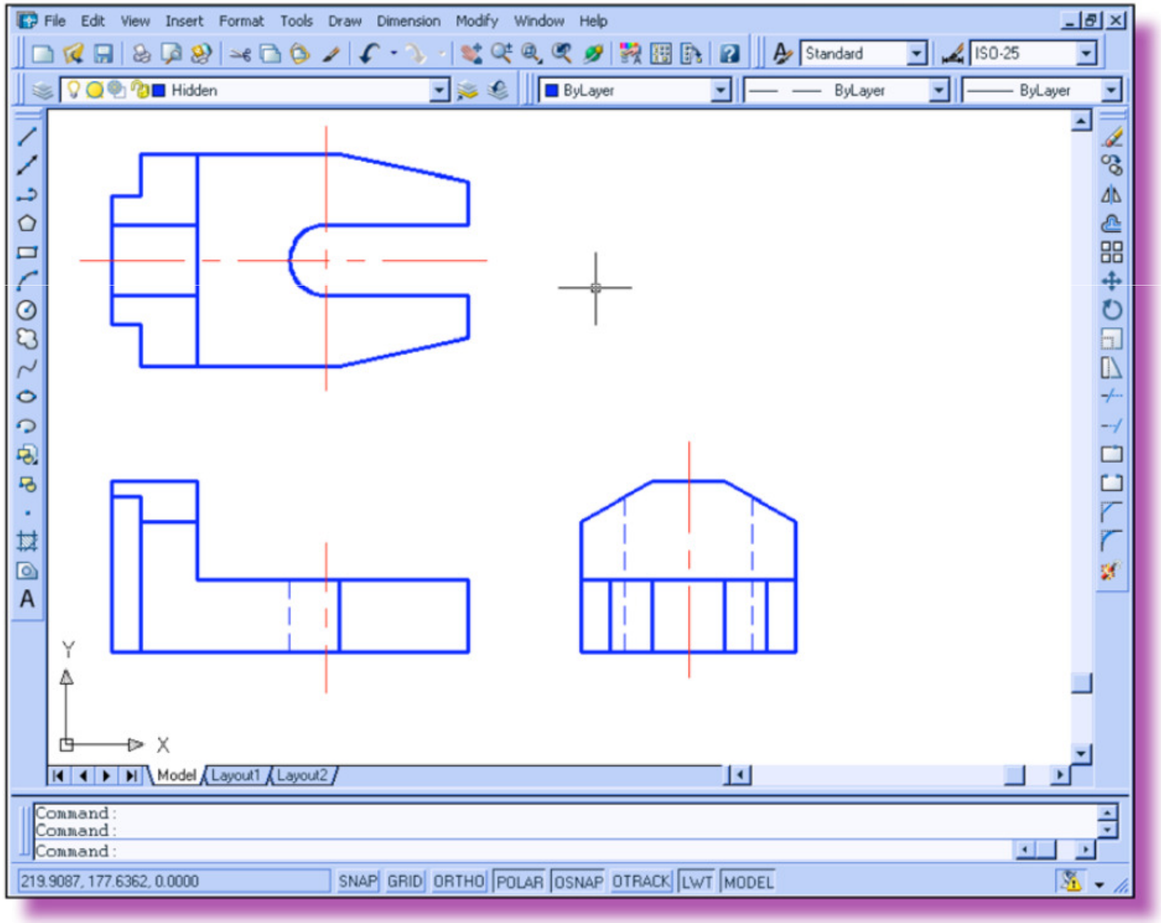
As primeiras gerações de CAD são apenas em 2D, basicamente substituindo lápis e papel.

Desenho

Abordagem Tradicional - Primeira Geração de CAD

(*Computer Aided Design*)

[SHIH2006]



As primeiras gerações de CAD são apenas em 2D, basicamente substituindo lápis e papel.

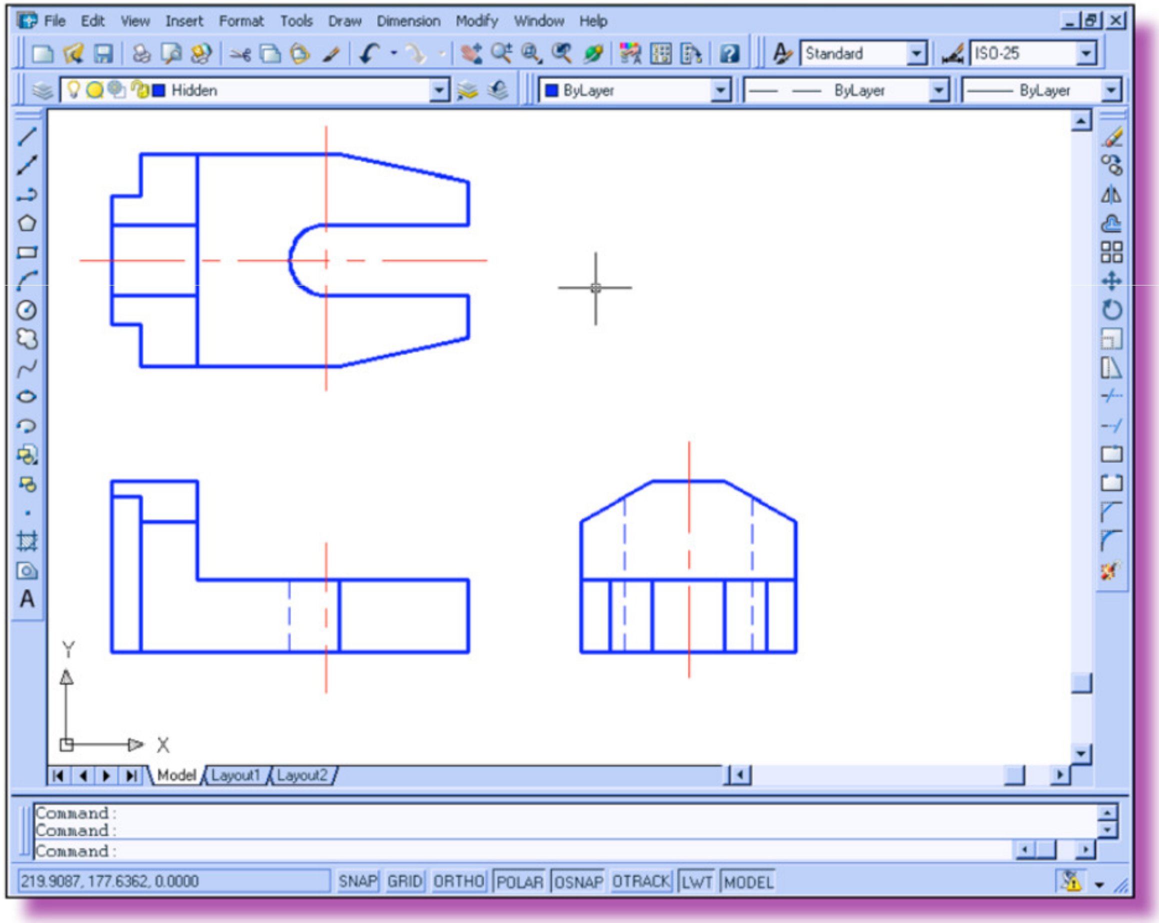
O tão popular AutoCAD, distribuído pela primeira vez em 1981, ganhou popularidade e é um dos principais sistemas CAD.

Desenho

Abordagem Tradicional - Primeira Geração de CAD

(*Computer Aided Design*)

[SHIH2006]



As primeiras gerações de CAD são apenas em 2D, basicamente substituindo lápis e papel.

O tão popular AutoCAD, distribuído pela primeira vez em 1981, ganhou popularidade e é um dos principais sistemas CAD.

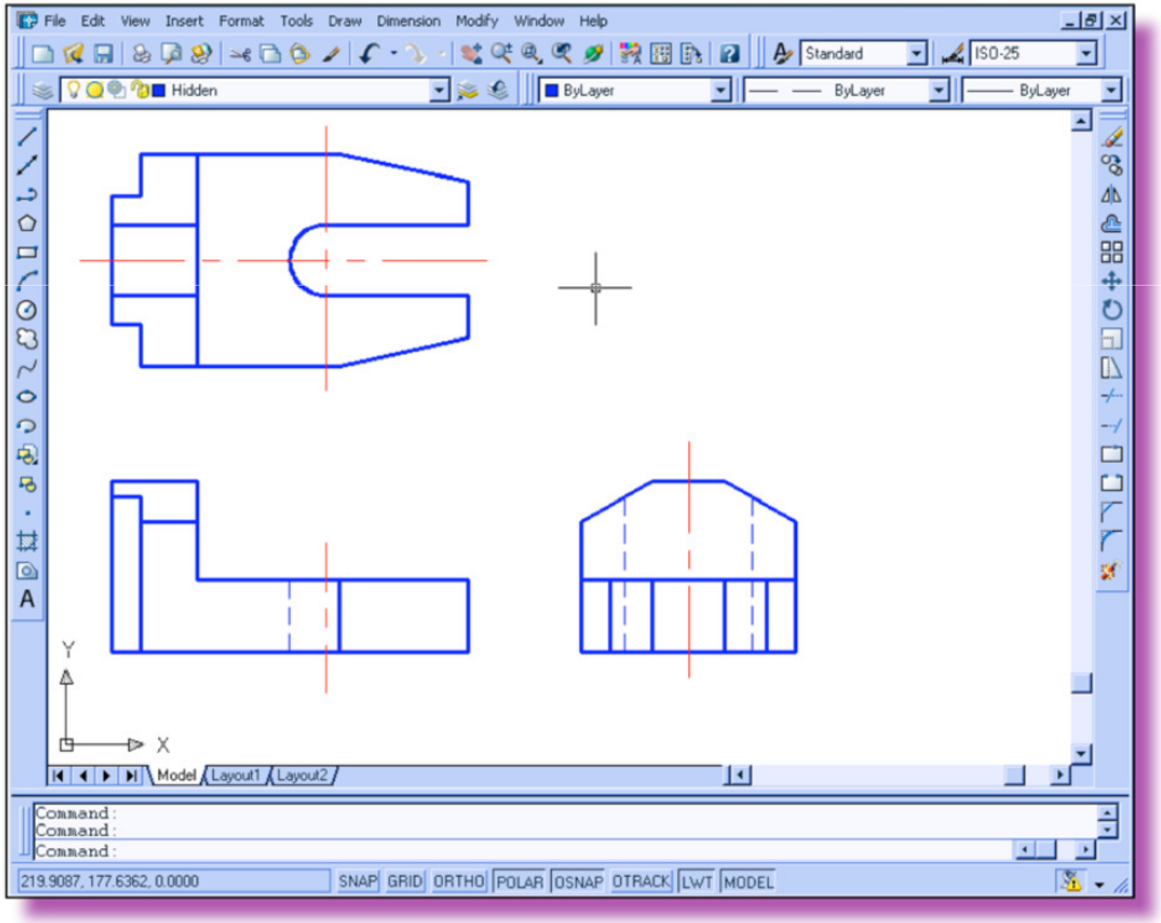
Ainda hoje, muitas empresas utilizam 2D CAD para criar projetos.

Desenho

Abordagem Tradicional - Primeira Geração de CAD

(*Computer Aided Design*)

[SHIH2006]



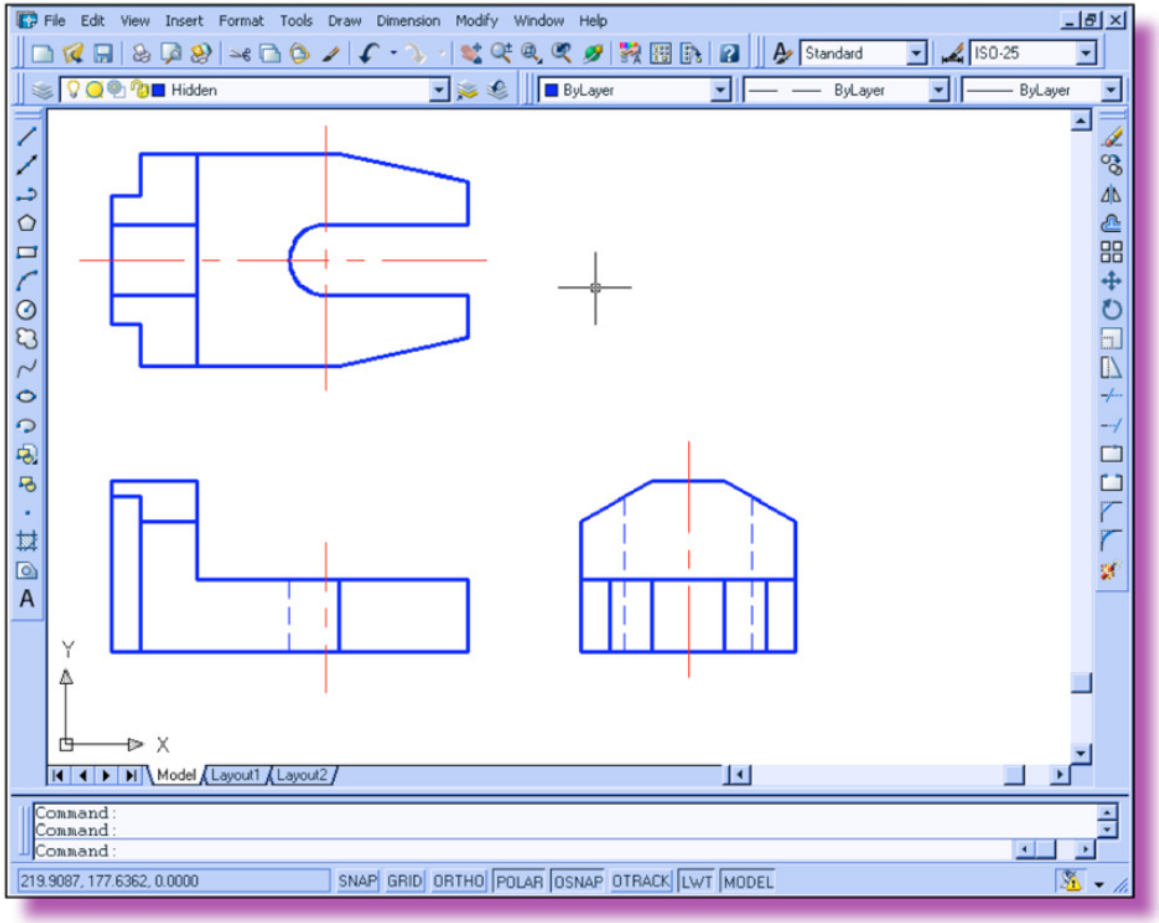
Esse tipo de abordagem requer o conhecimento das dimensões reais de projeto, sendo portanto pouco flexível.

Desenho

Abordagem Tradicional - Primeira Geração de CAD

(*Computer Aided Design*)

[SHIH2006]



Esse tipo de abordagem requer o conhecimento das dimensões reais de projeto, sendo portanto pouco flexível.

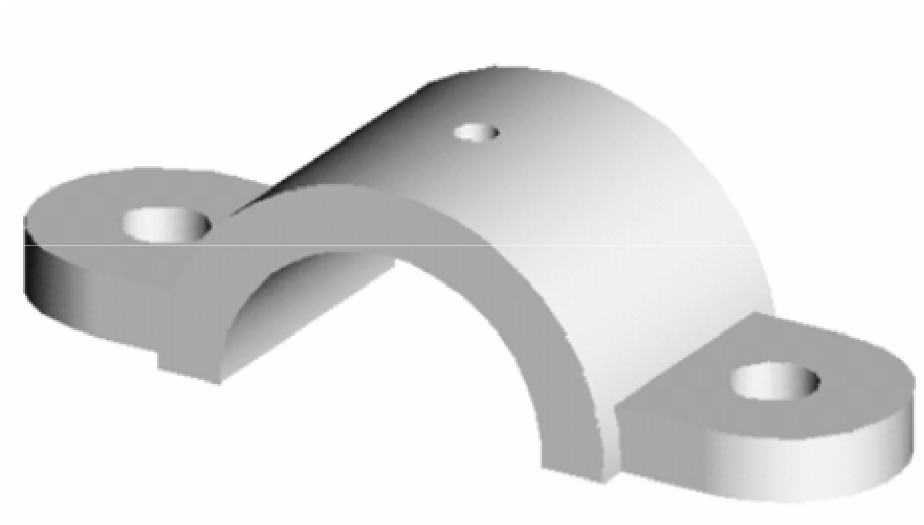
Note na Figura que :

(1) A criação dessas vistas requer o conhecimento das dimensões.

(2) Cada uma das três vistas é criada e editada independentemente das outras.

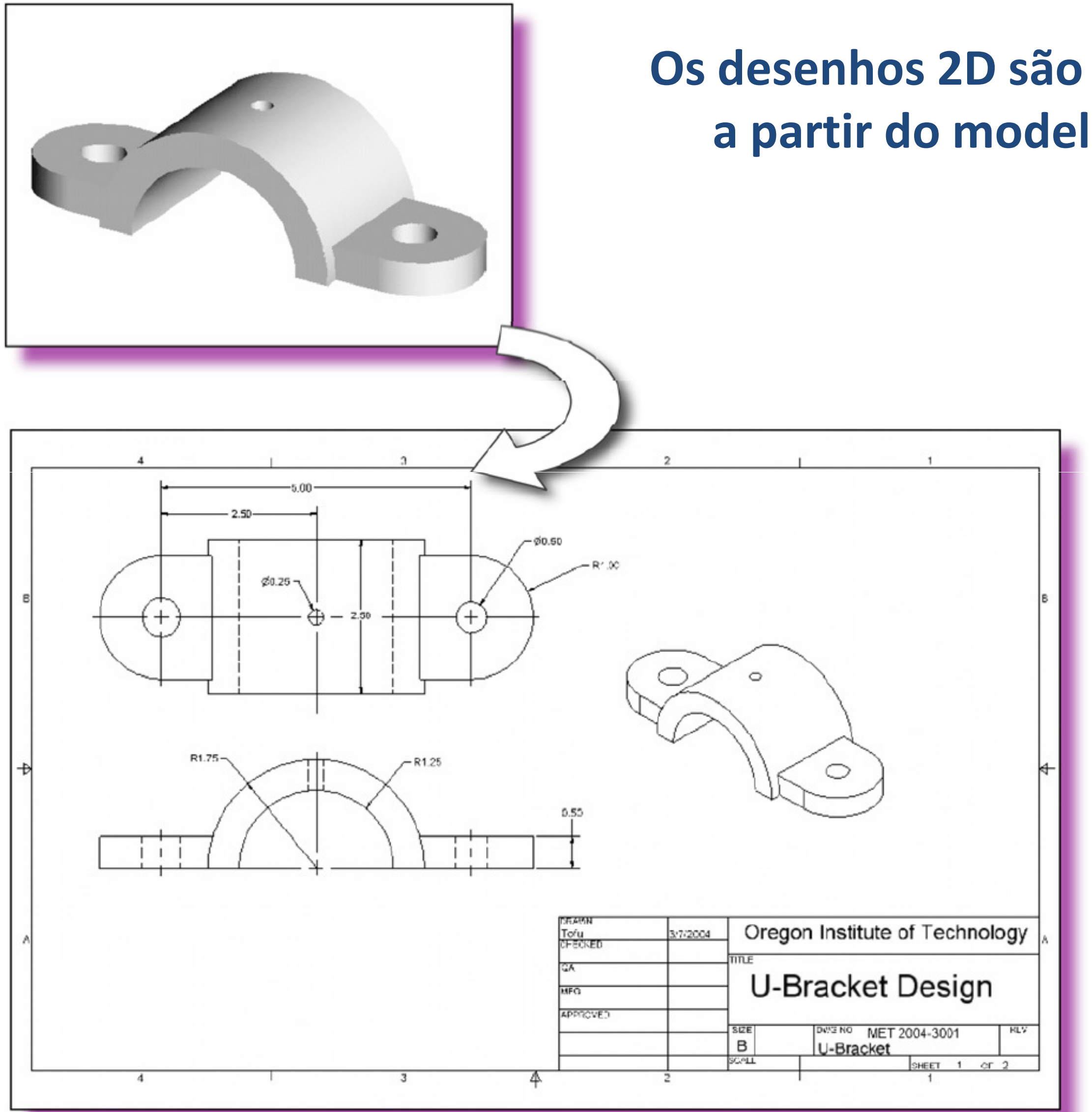
Modelagem de Sólidos

Modelagem de Sólidos

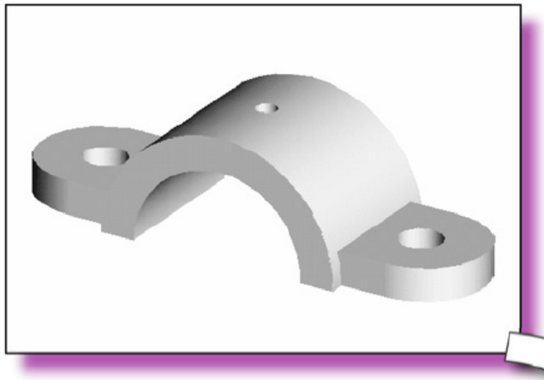


Modelagem de Sólidos

Os desenhos 2D são gerados
a partir do modelo 3D.

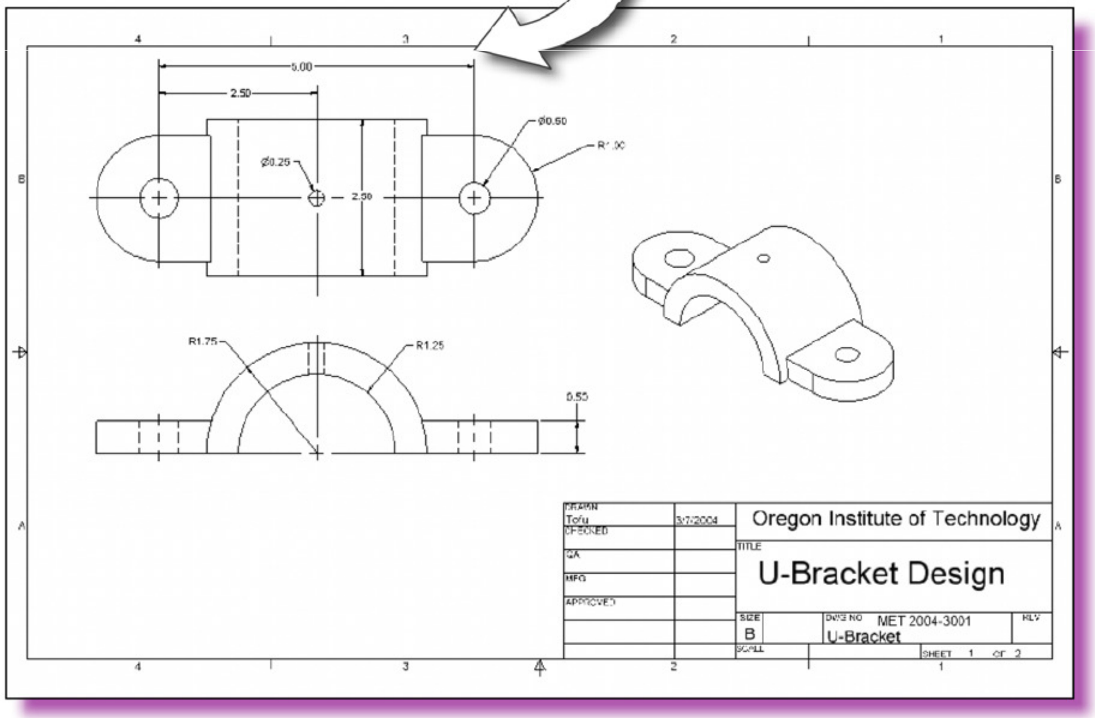


Modelagem de Sólidos



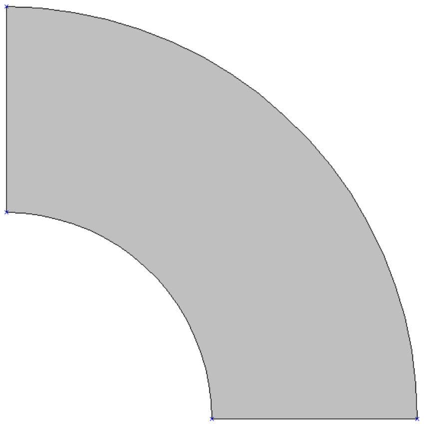
Os desenhos 2D são gerados a partir do modelo 3D.

Modificações são atualizadas automaticamente.

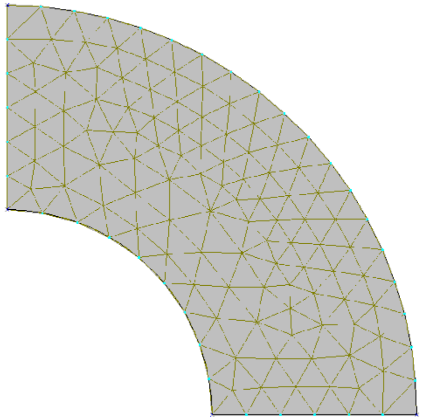


Modelagem em Engenharia

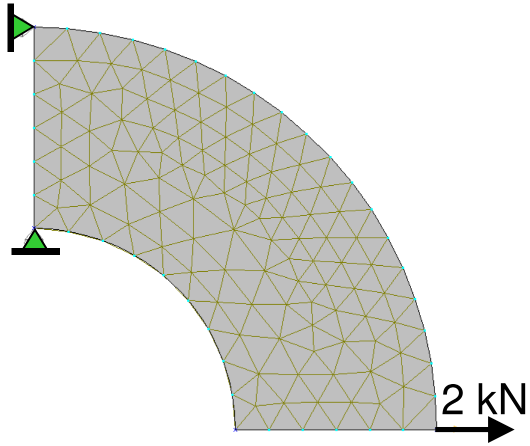
Traditional FE Simulation Process



1. Build geometric model

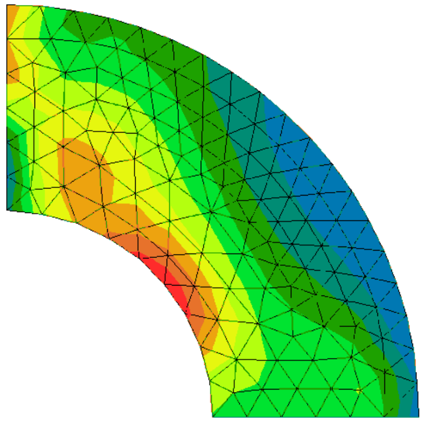


2. Mesh



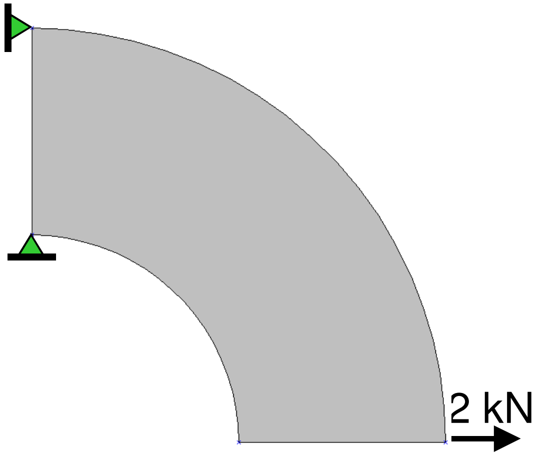
3. Apply boundary conditions

4. Computational analysis

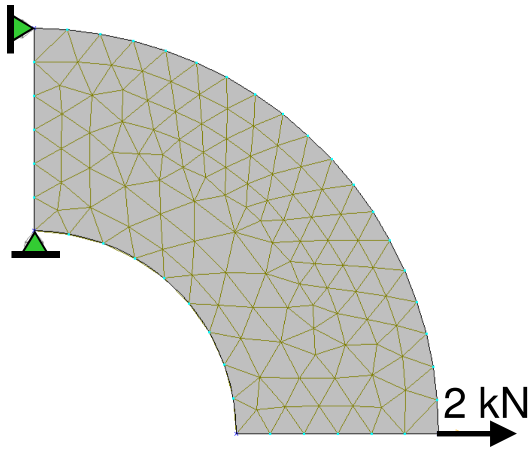


5. Result visualization

Geometry-based Simulation Process

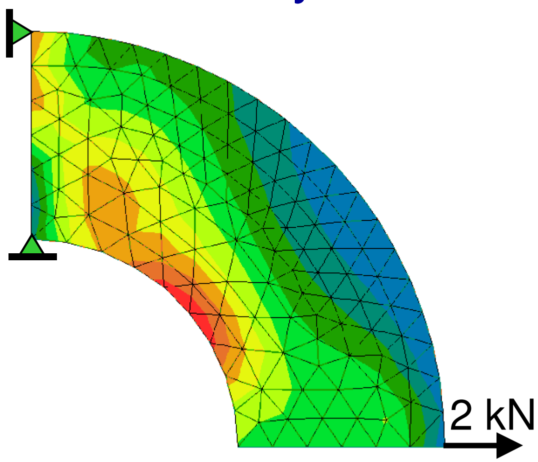


1. Geometric modelling, apply attributes and boundary conditions



2. FE mesh generation, apply boundary conditions

3. Computational analysis

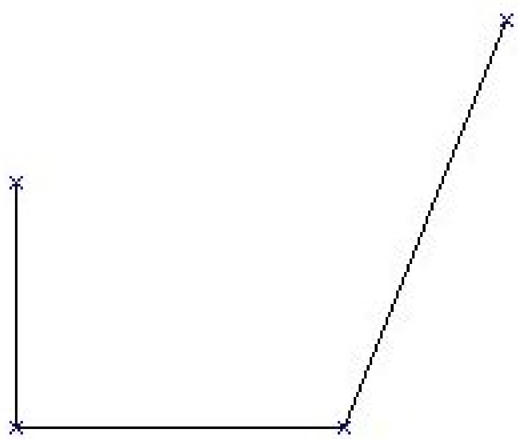


4. Result visualization

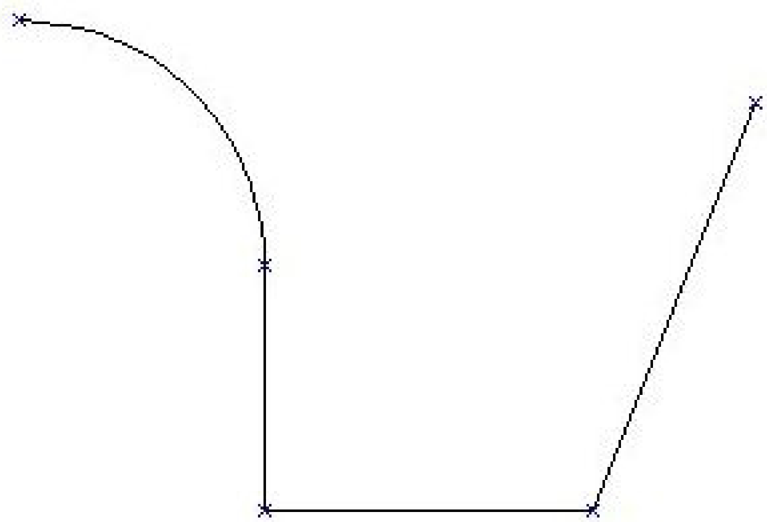
Construction of a Simple FE Model

Construction of a Simple FE Model

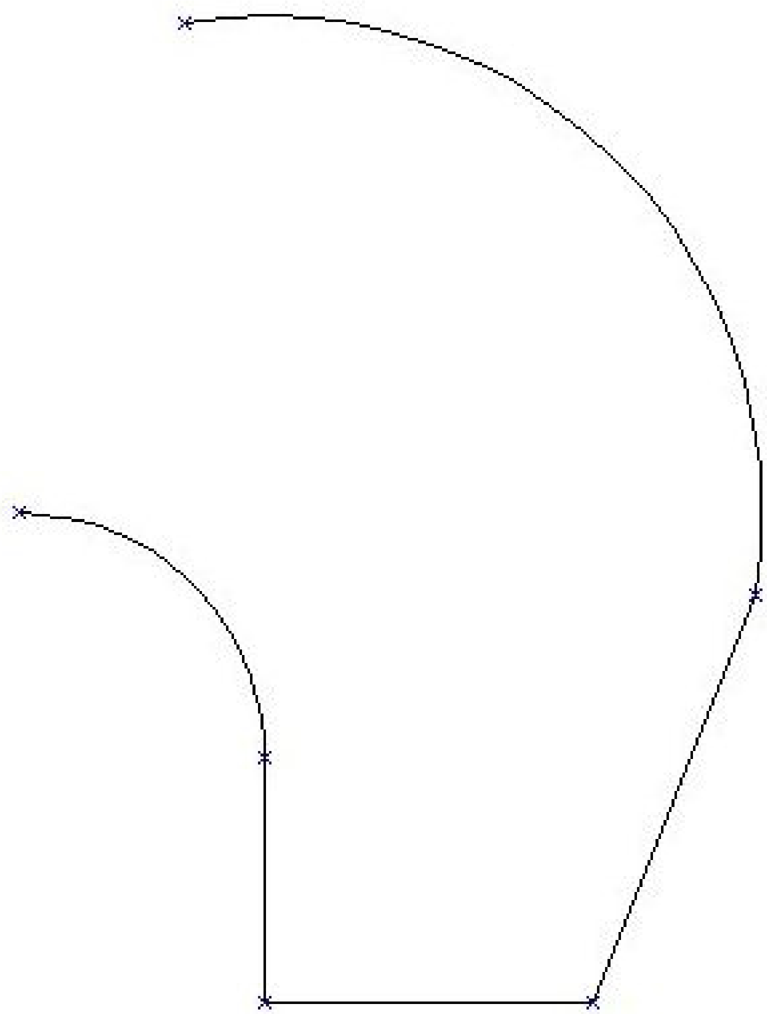
Construction of a Simple FE Model



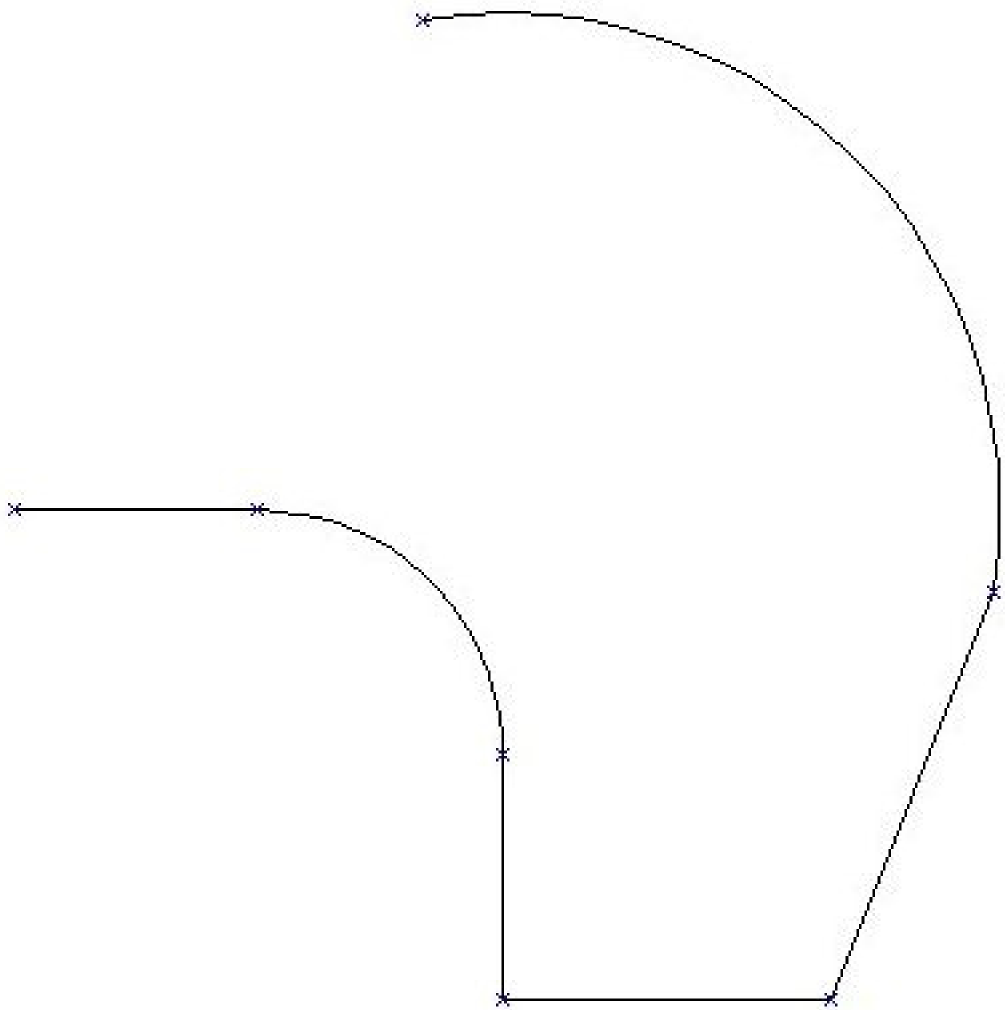
Construction of a Simple FE Model



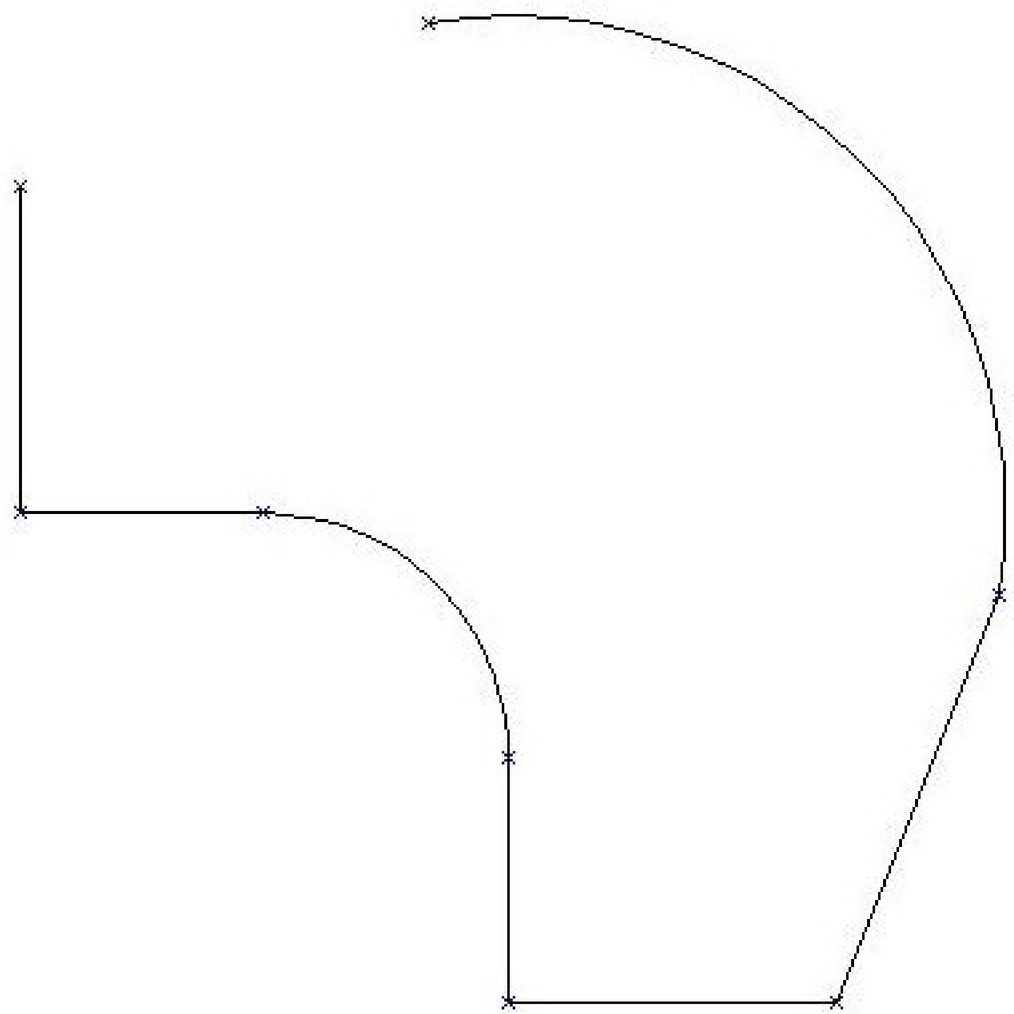
Construction of a Simple FE Model



Construction of a Simple FE Model

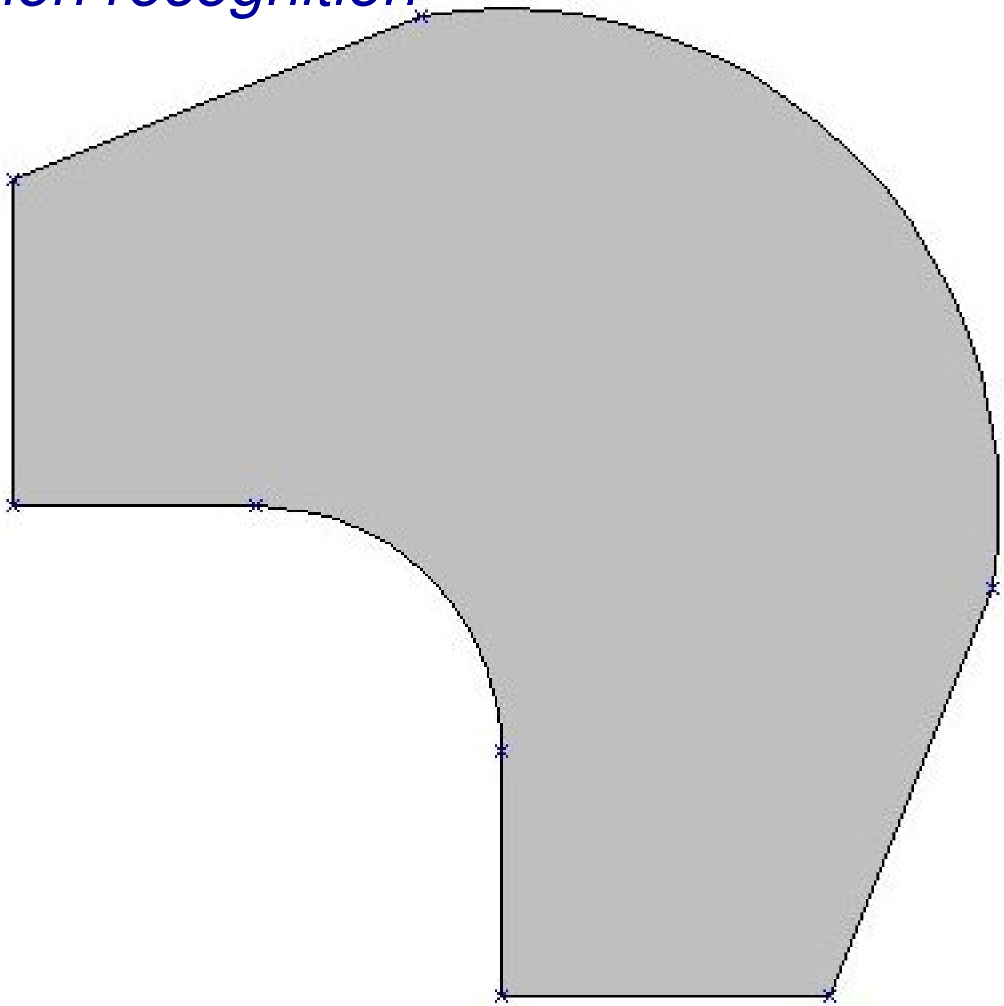


Construction of a Simple FE Model

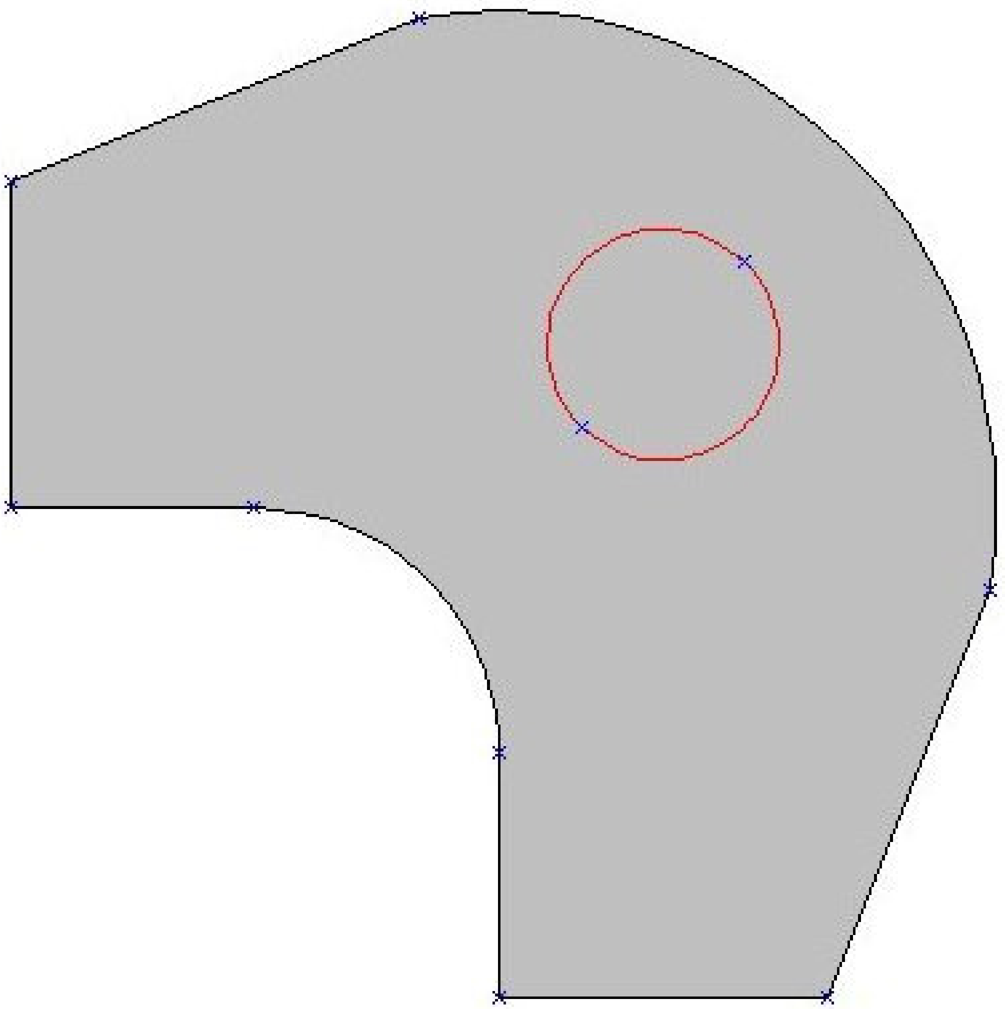


Construction of a Simple FE Model

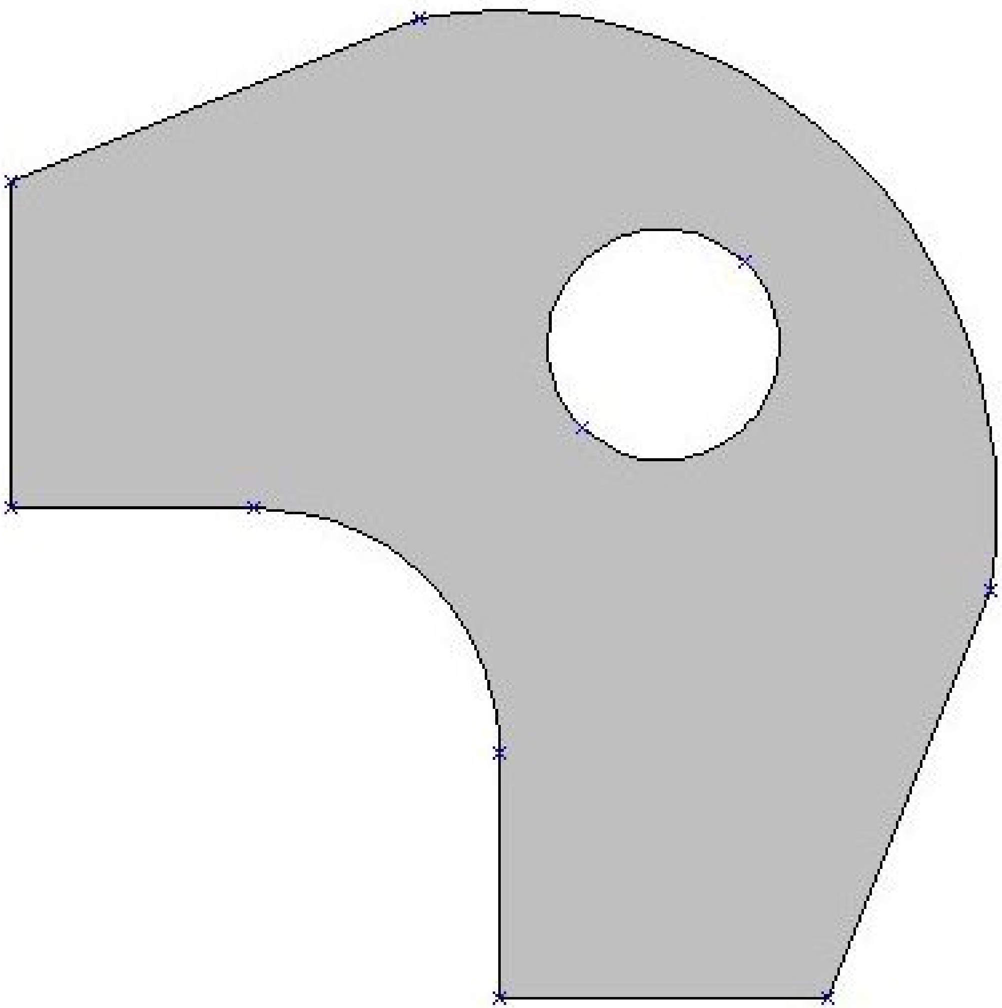
Automatic region recognition



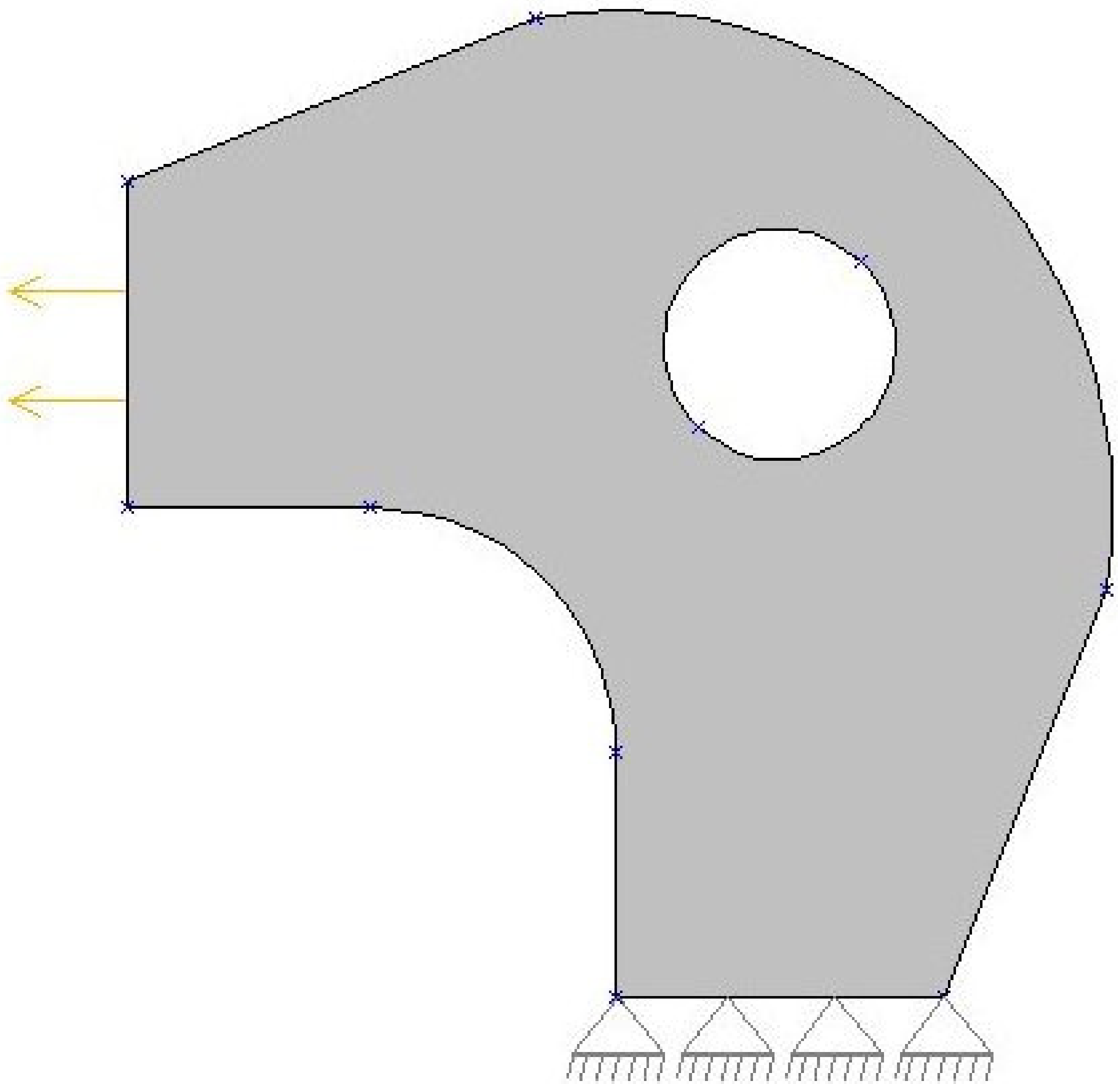
Creating a hole



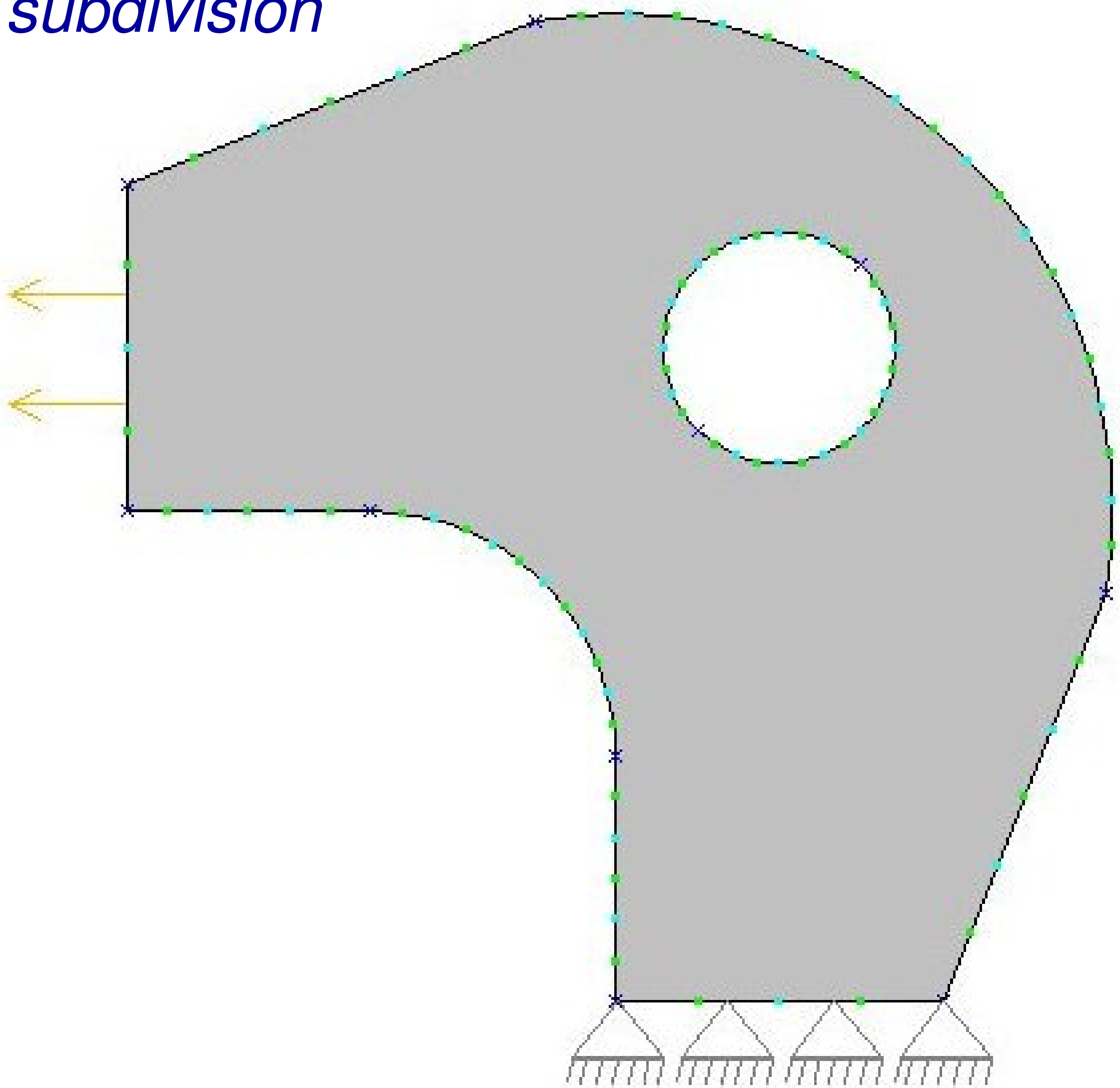
Assigning hole attribute



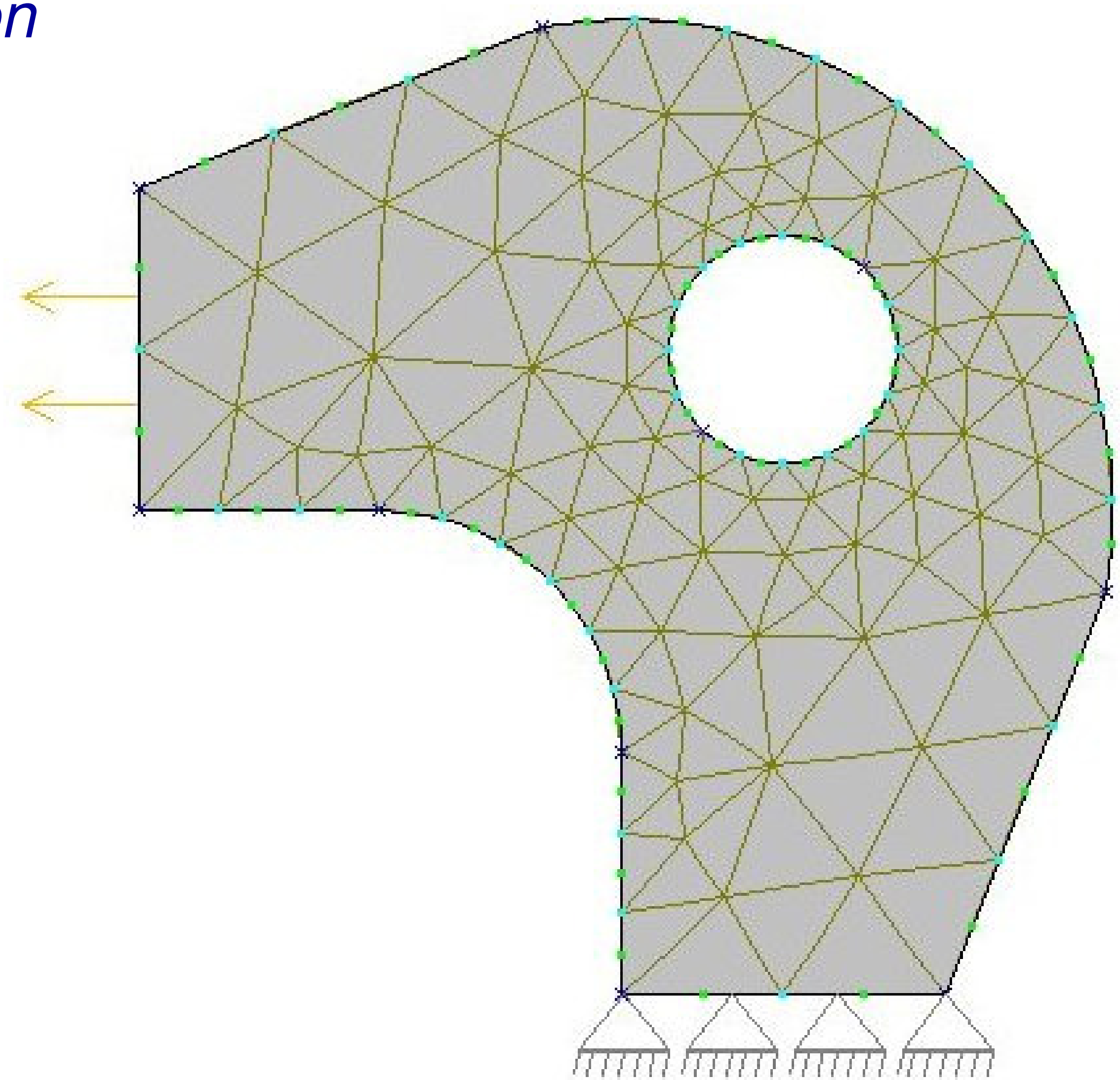
Applying attributes to geometry



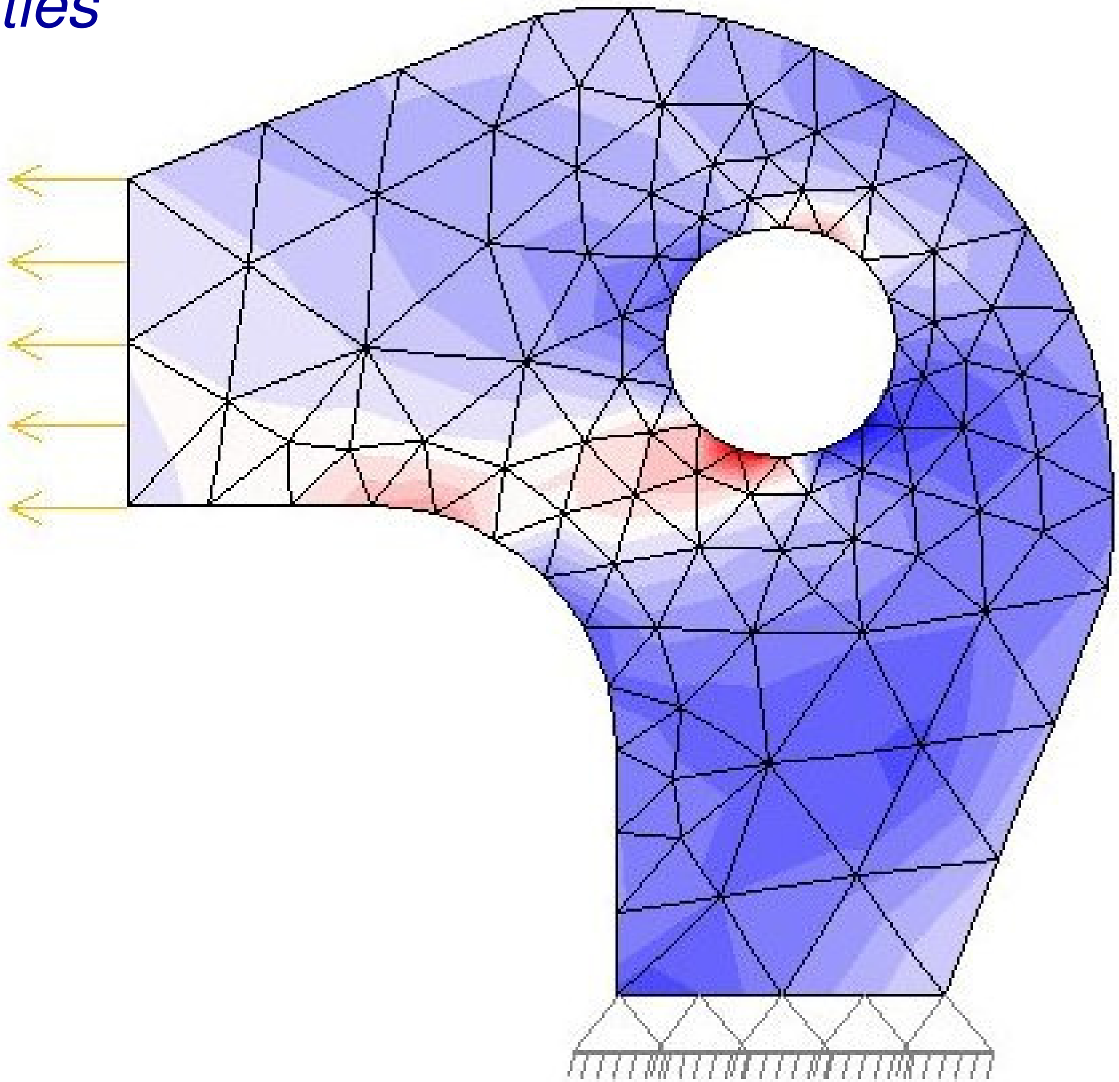
Defining meshing refinement parameters: boundary subdivision



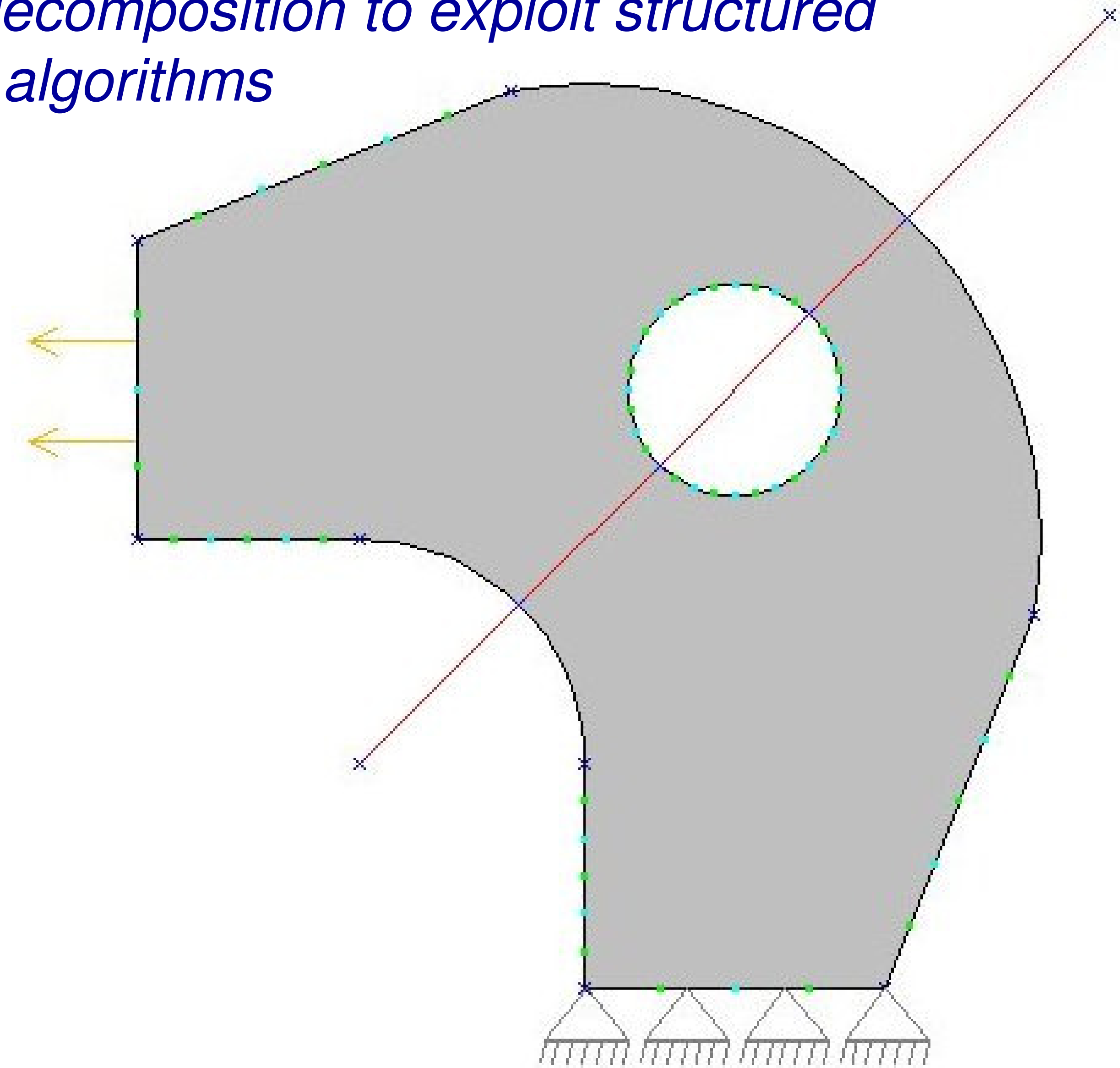
Automatic unstructured mesh generation



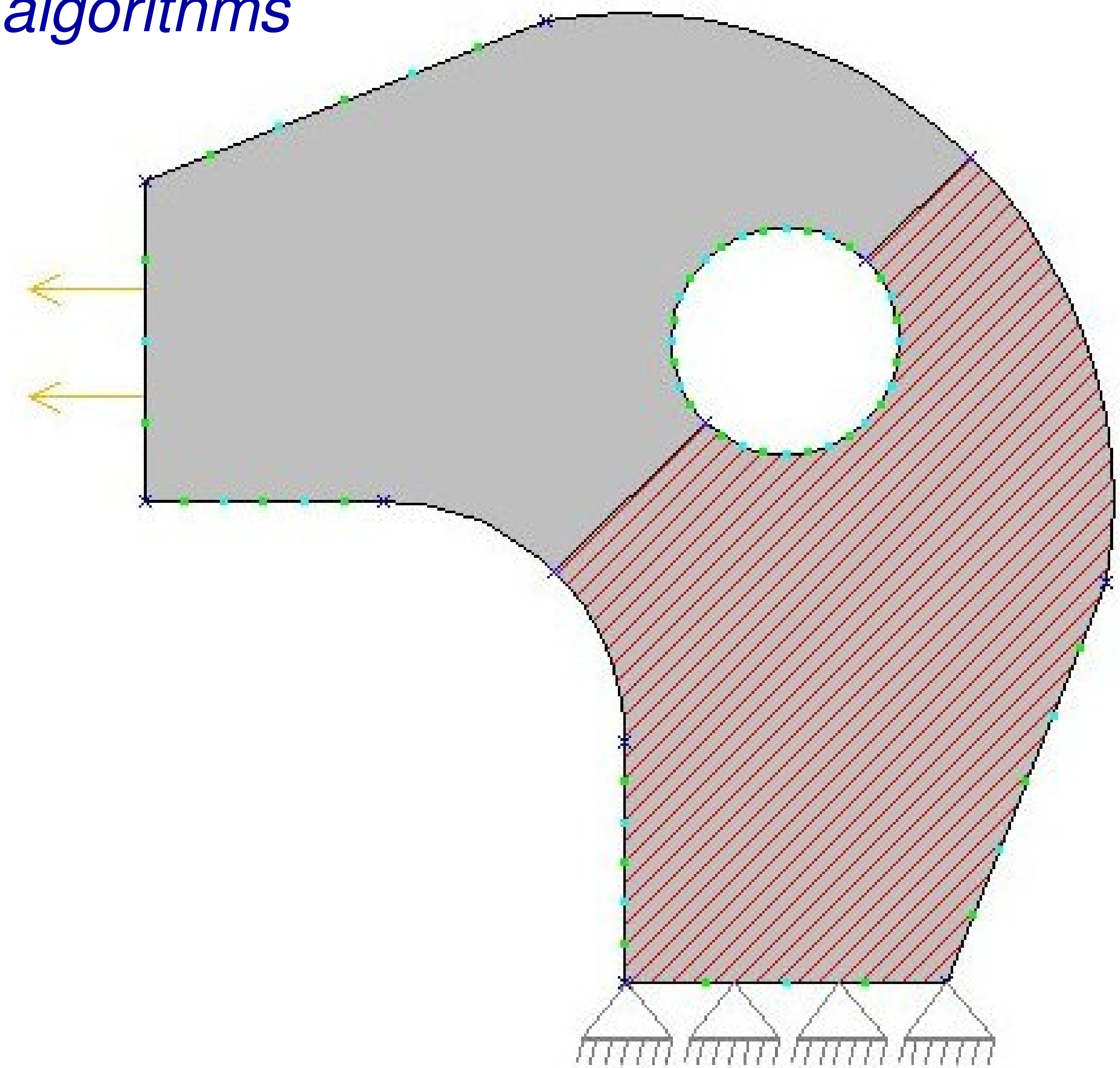
Attributes automatically assigned to mesh entities



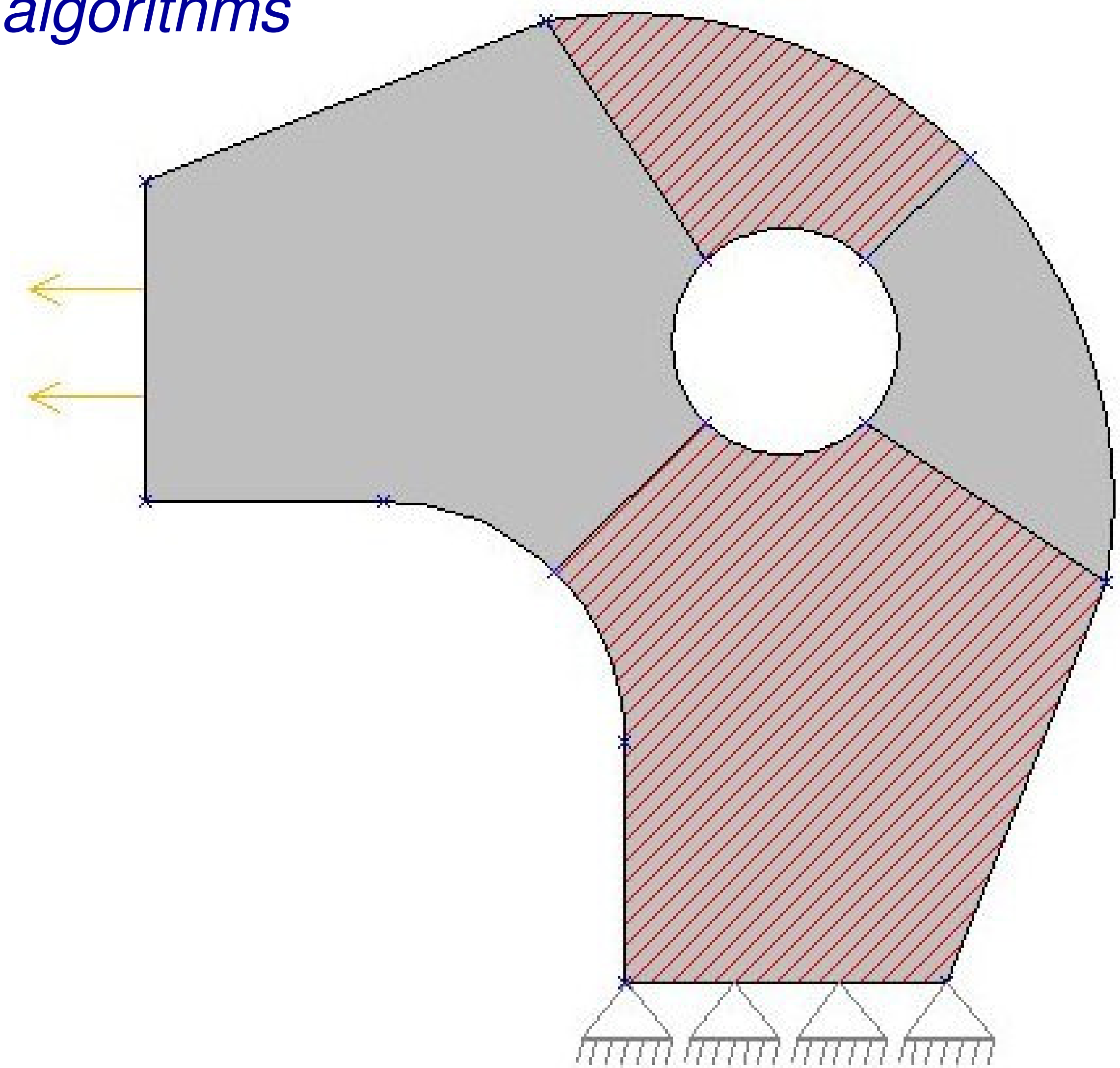
Region decomposition to exploit structured meshing algorithms



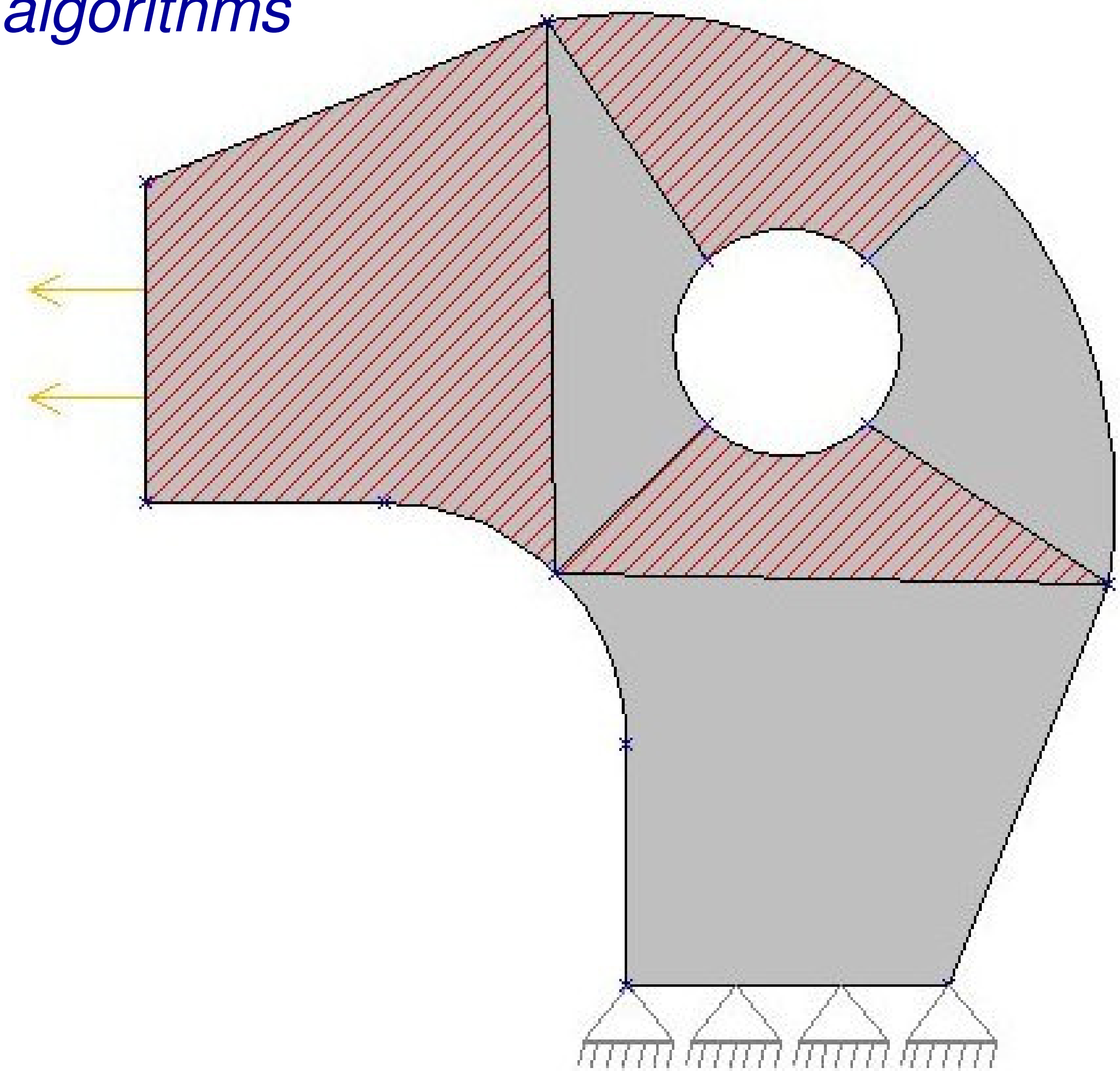
Region decomposition to exploit structured meshing algorithms



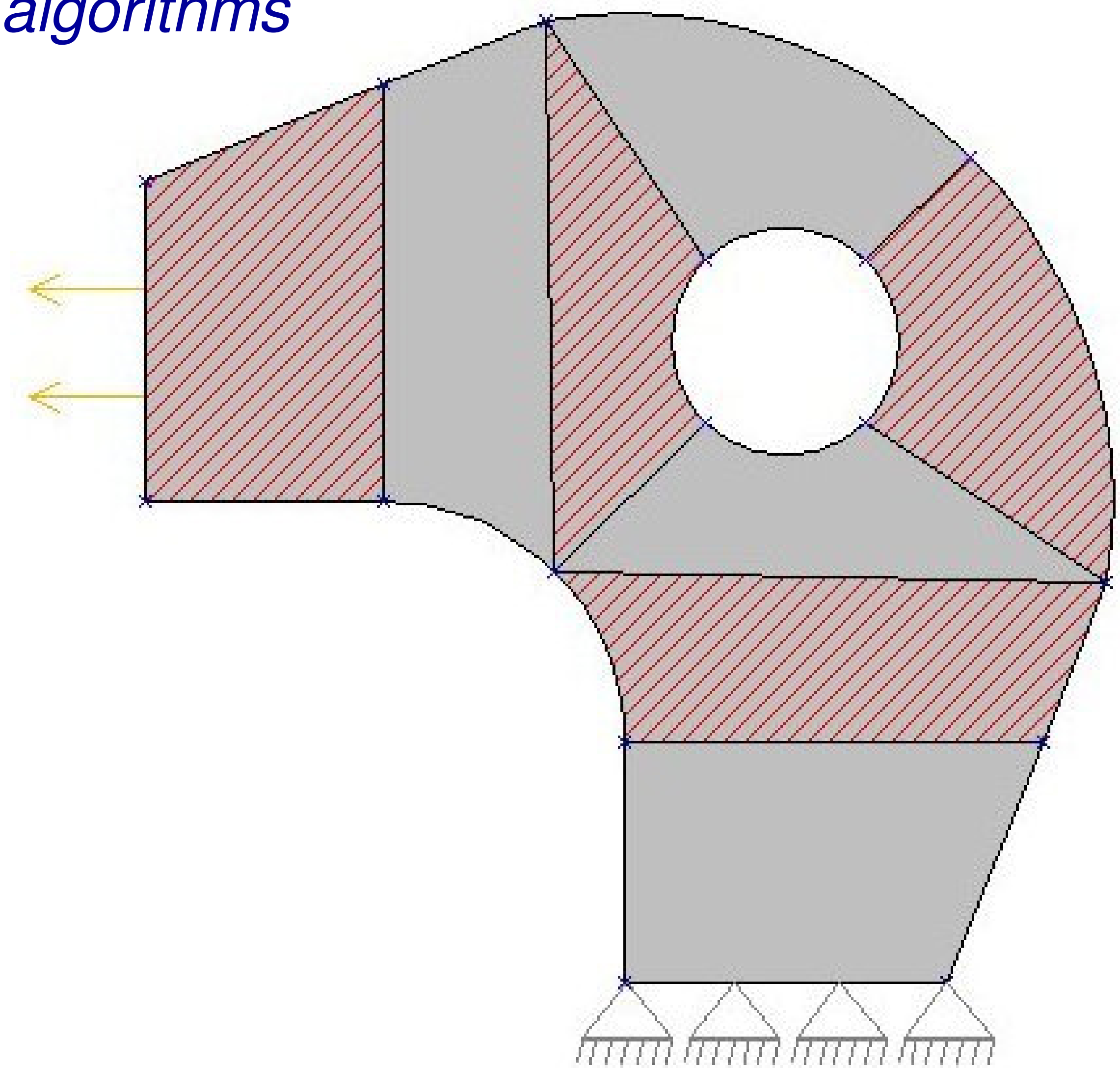
Region decomposition to exploit structured meshing algorithms



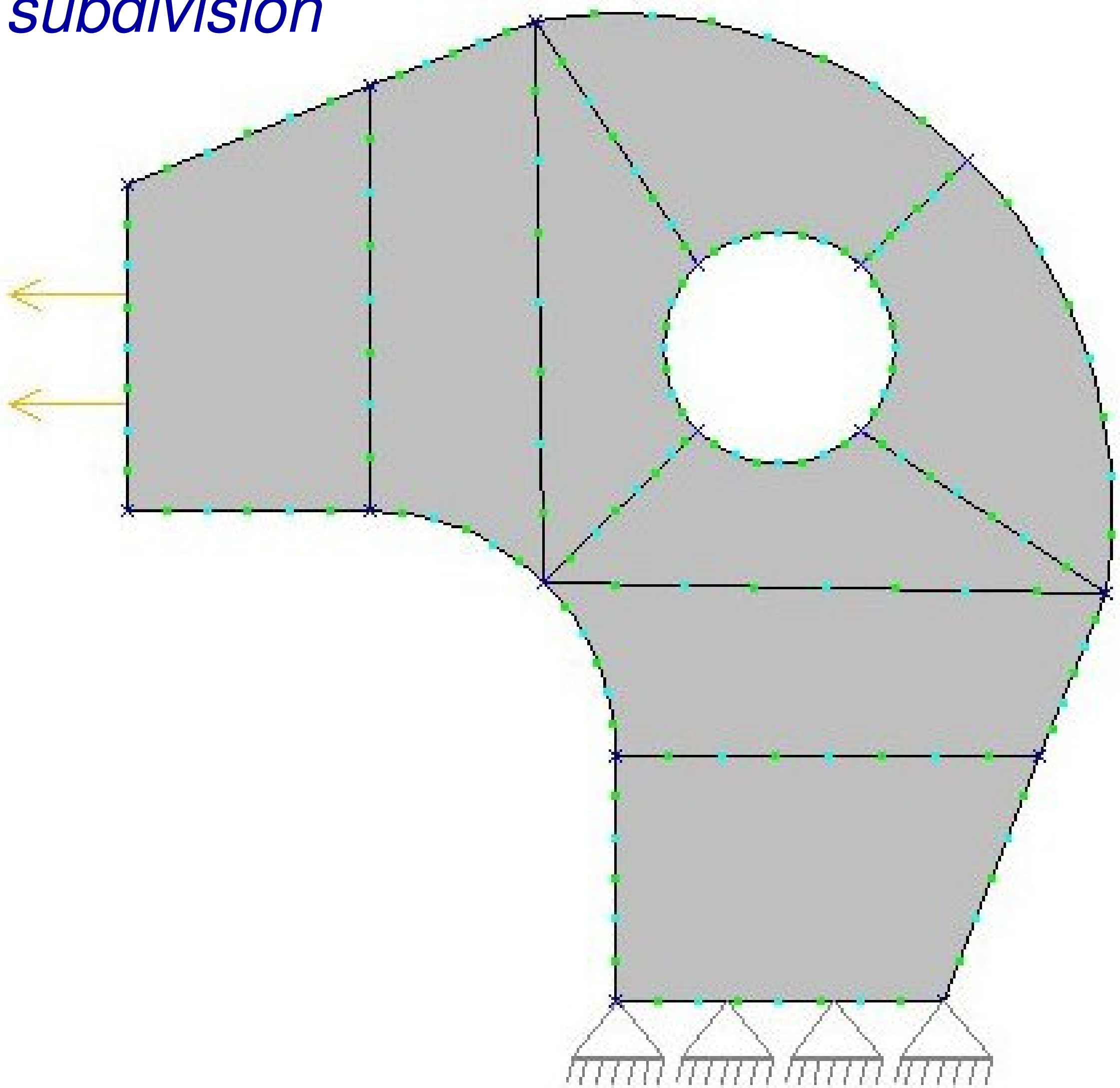
Region decomposition to exploit structured meshing algorithms



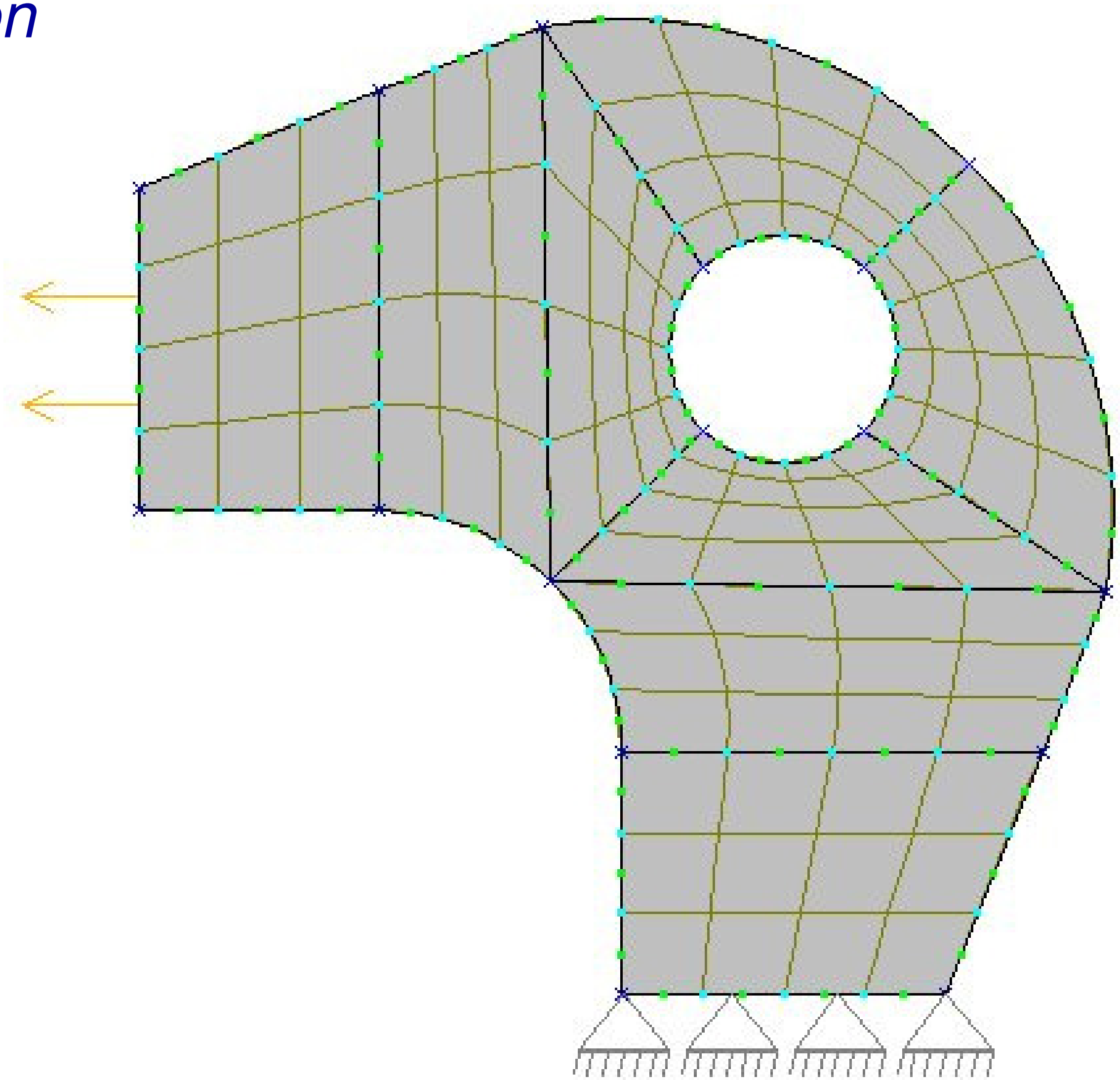
Region decomposition to exploit structured meshing algorithms



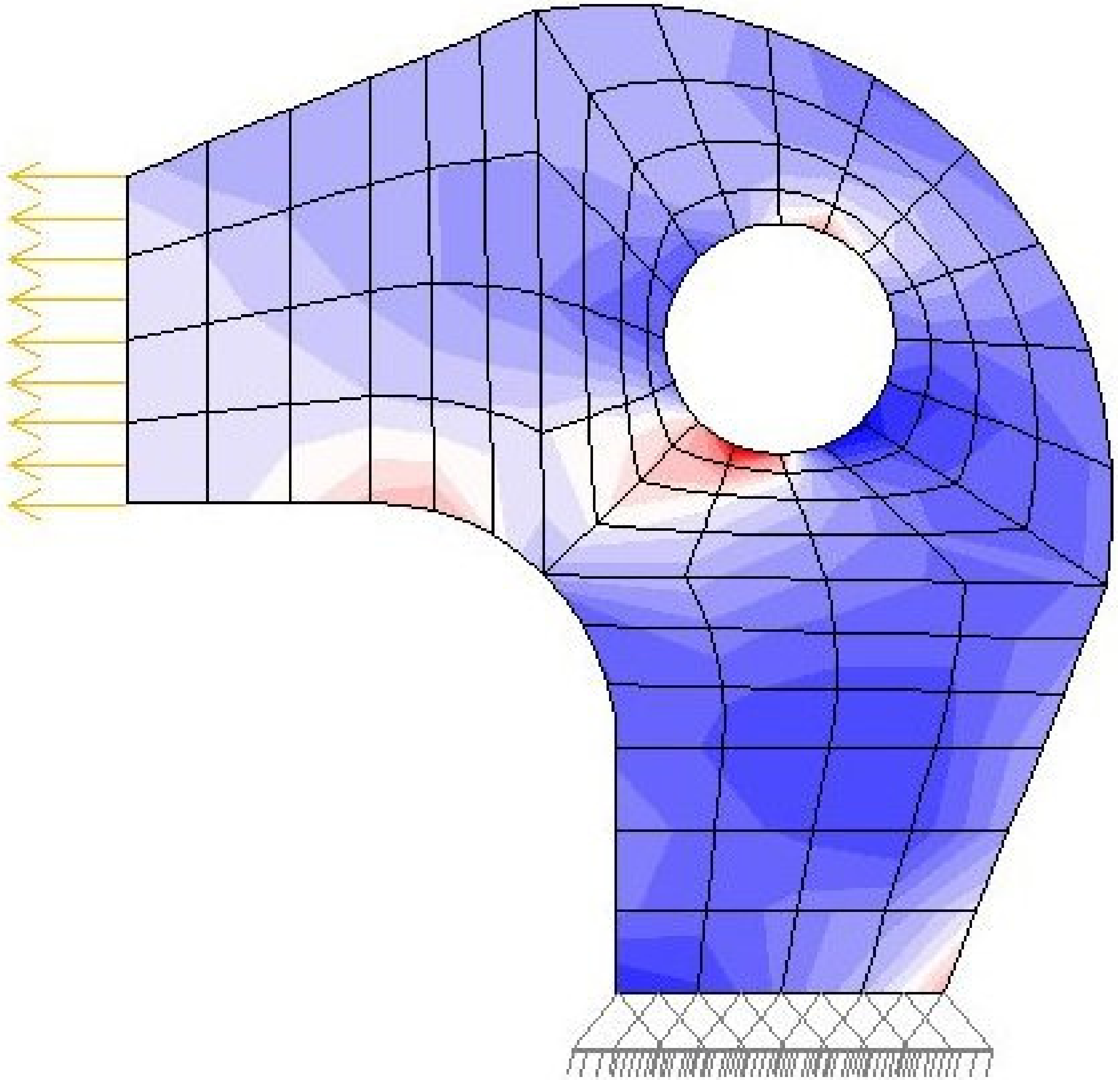
Defining meshing refinement parameters: boundary subdivision



Automatic unstructured mesh generation

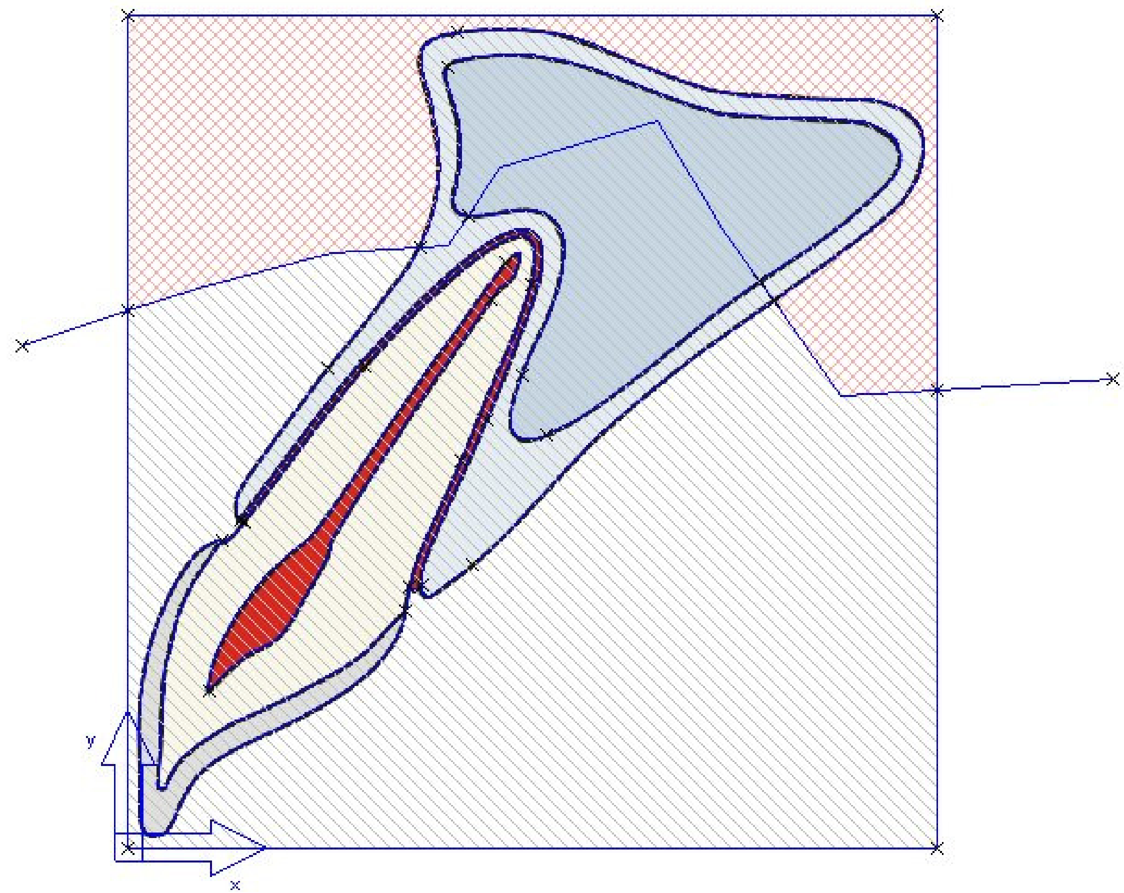


What is the technology behind this?
What issues we have to address?



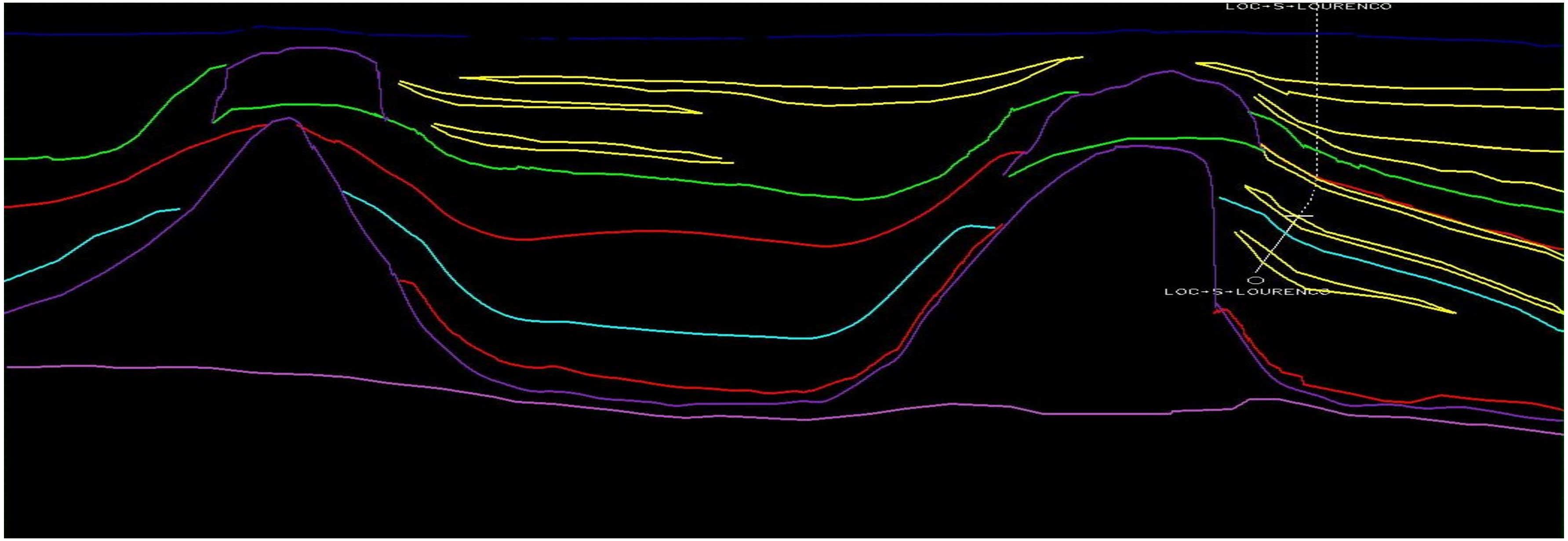
Generic Space Subdivion: Many Applications

An environment in which curves and surfaces are inserted randomly.
Automatic region recognition and full adjacency information.



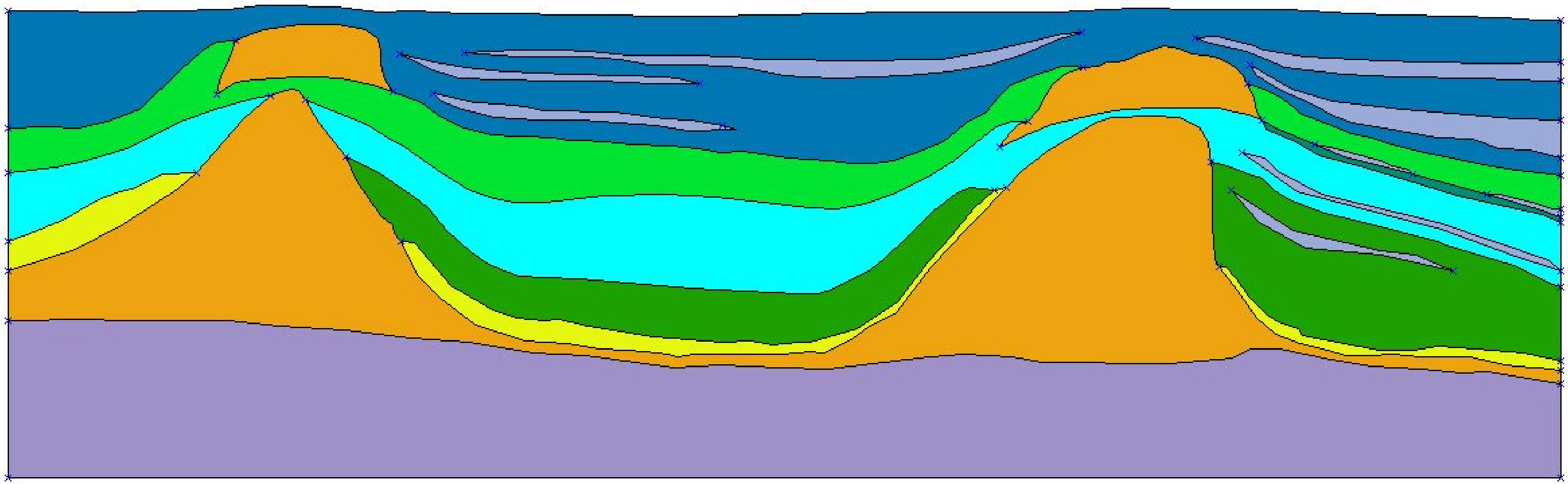
2D Subsurface Simulation Modeling

Sidon-Tiro



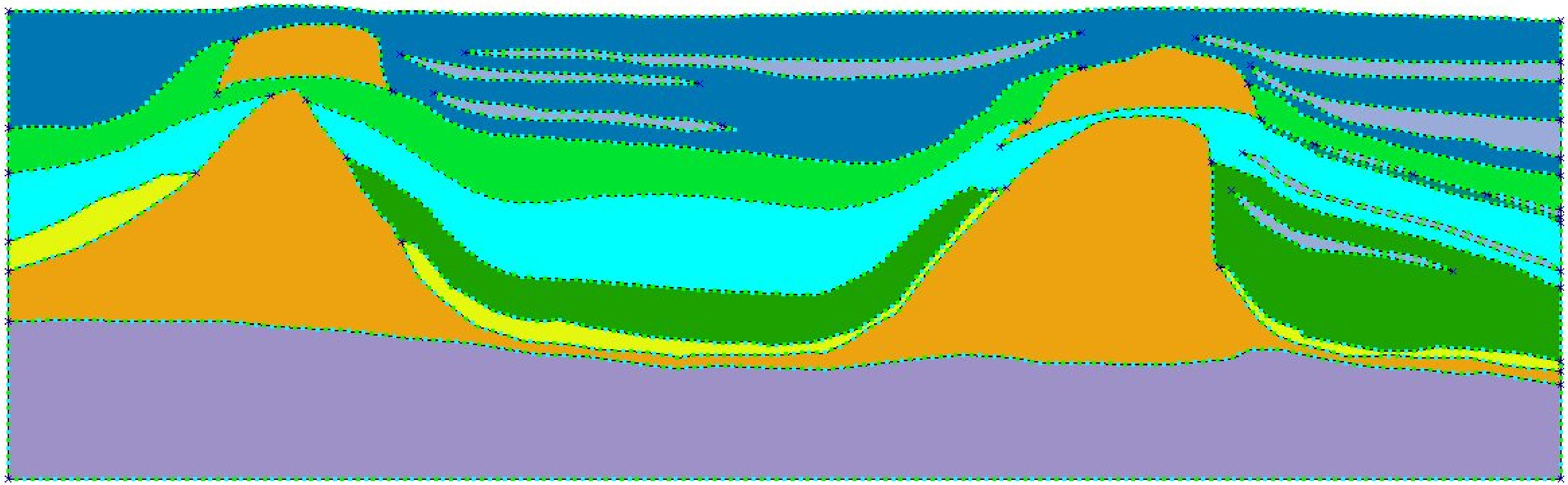
2D Subsurface Simulation Modeling

Curve digitalization



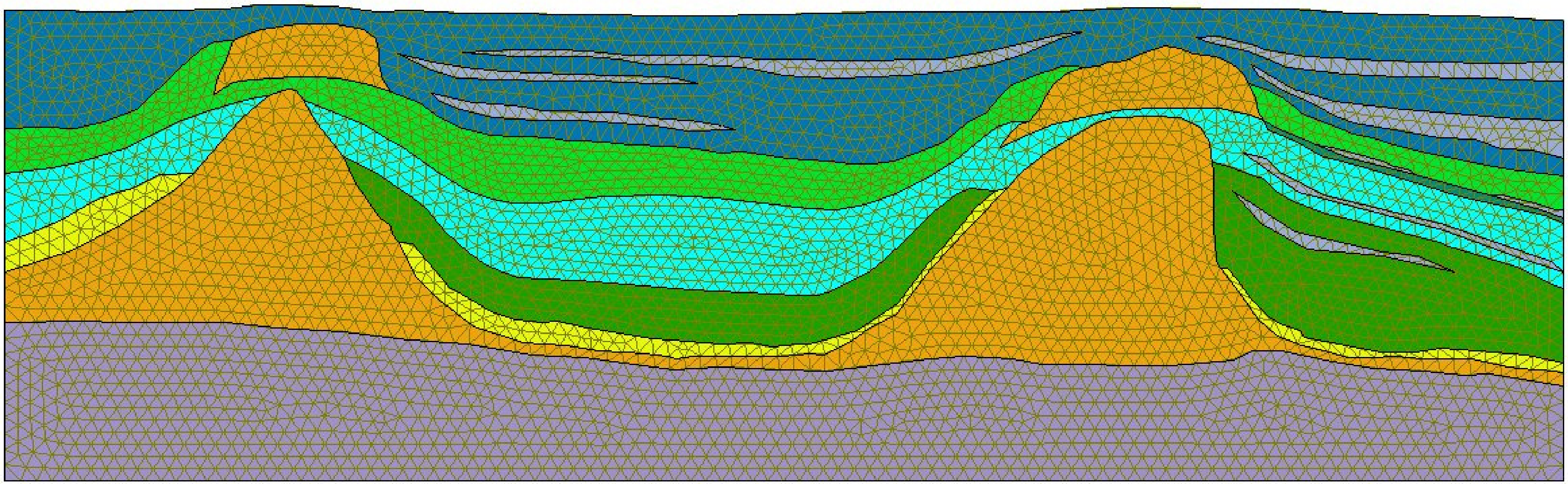
2D Subsurface Simulation Modeling

Curve subdivision



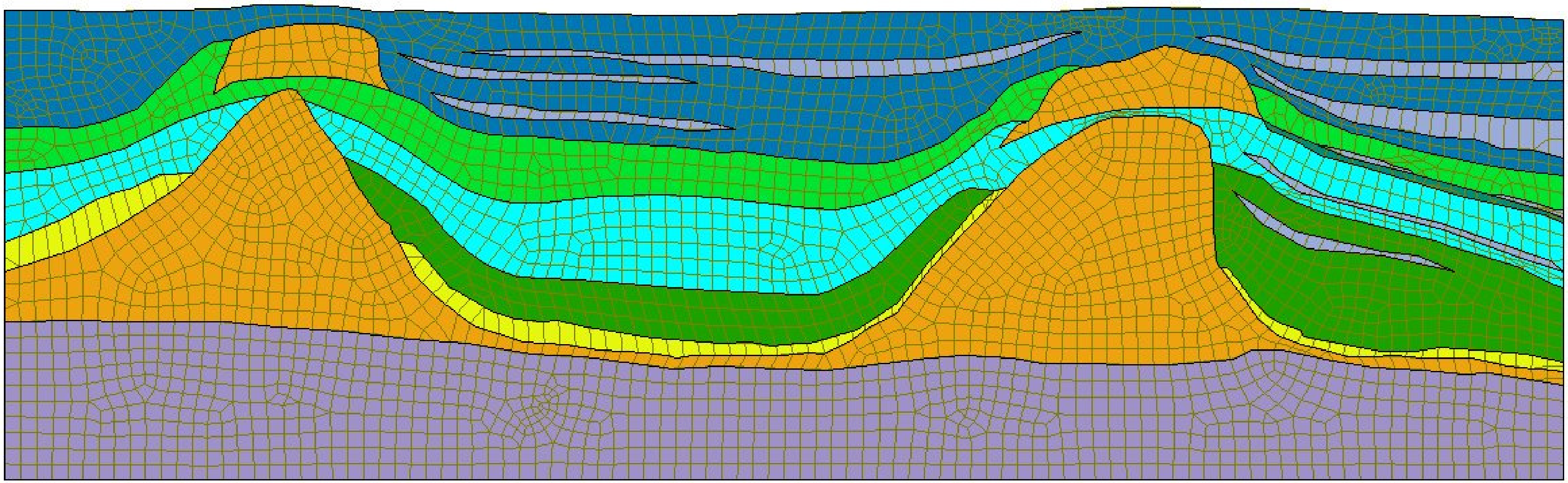
2D Subsurface Simulation Modeling

Mesh generation: triangular elements



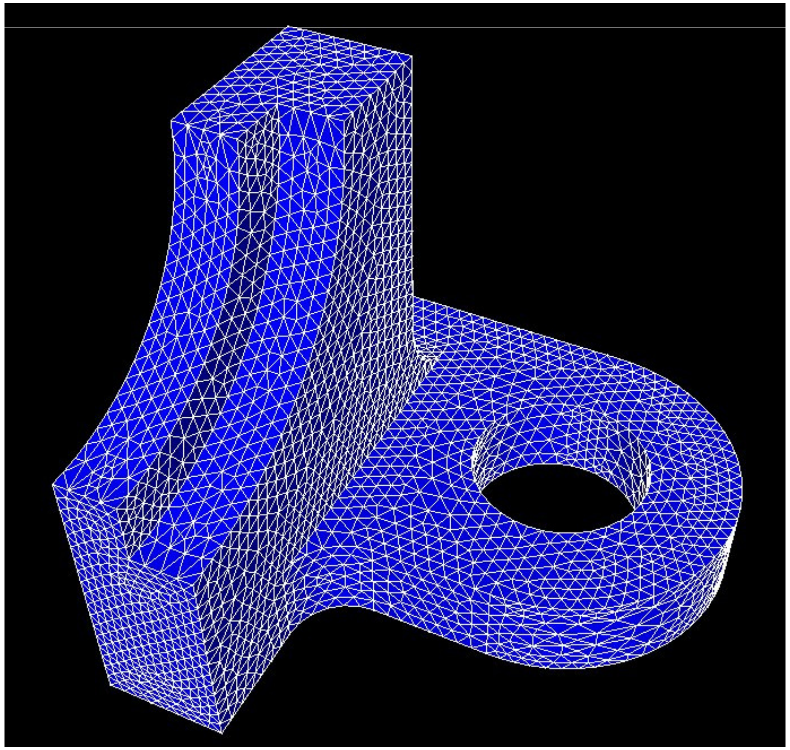
2D Subsurface Simulation Modeling

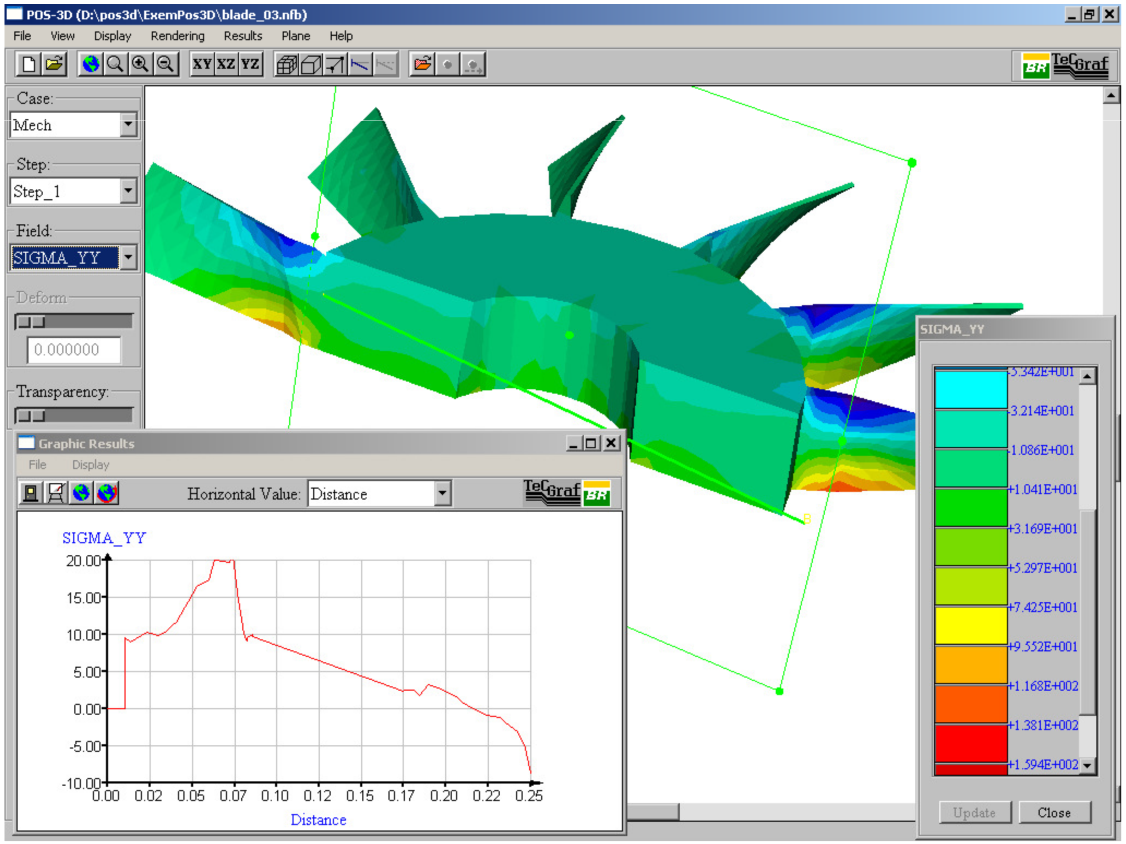
Mesh generation: quadrilateral elements



Requirements for underlying data representation

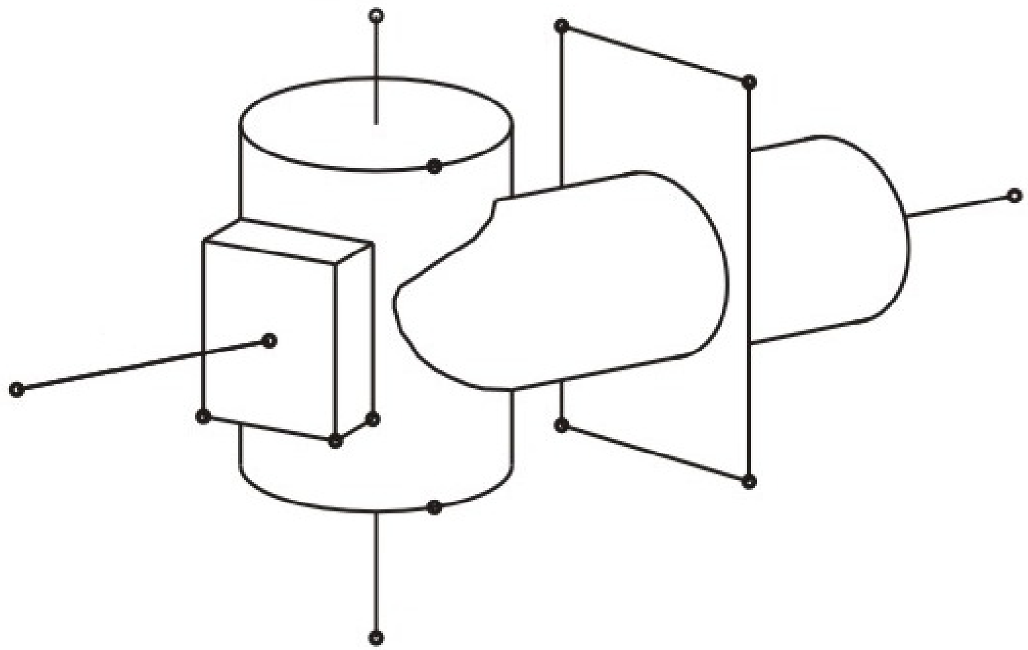
– ***The data structures must provide a natural navigation across all phases of a simulation: pre-processing (model creation), numerical analysis, and post-processing (model results visualization).***





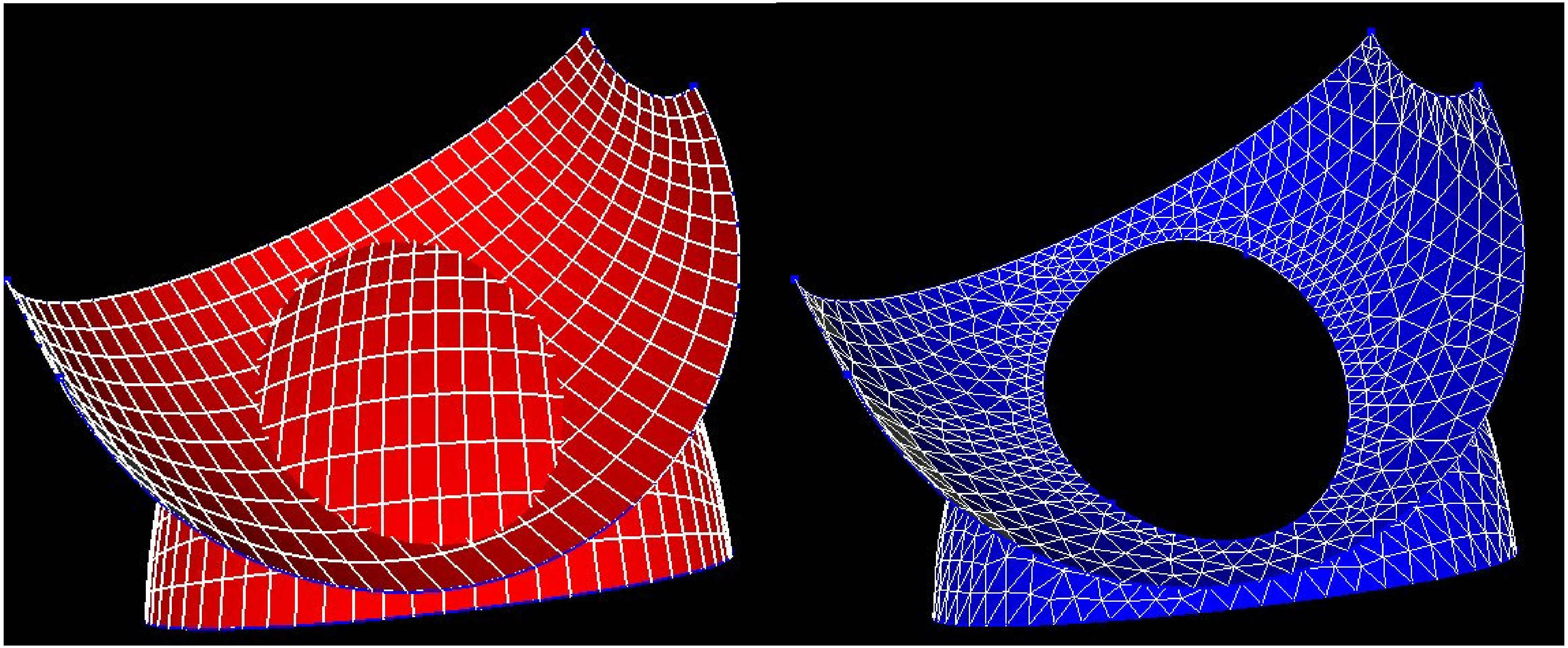
Requirements for underlying data representation (cont.)

–***The data structures must take into account that the simulation may induce, at least temporarily during model creation, geometric objects (curves and surfaces) that are inconsistent with the target final model. This requires a non-manifold topology representation capability.***



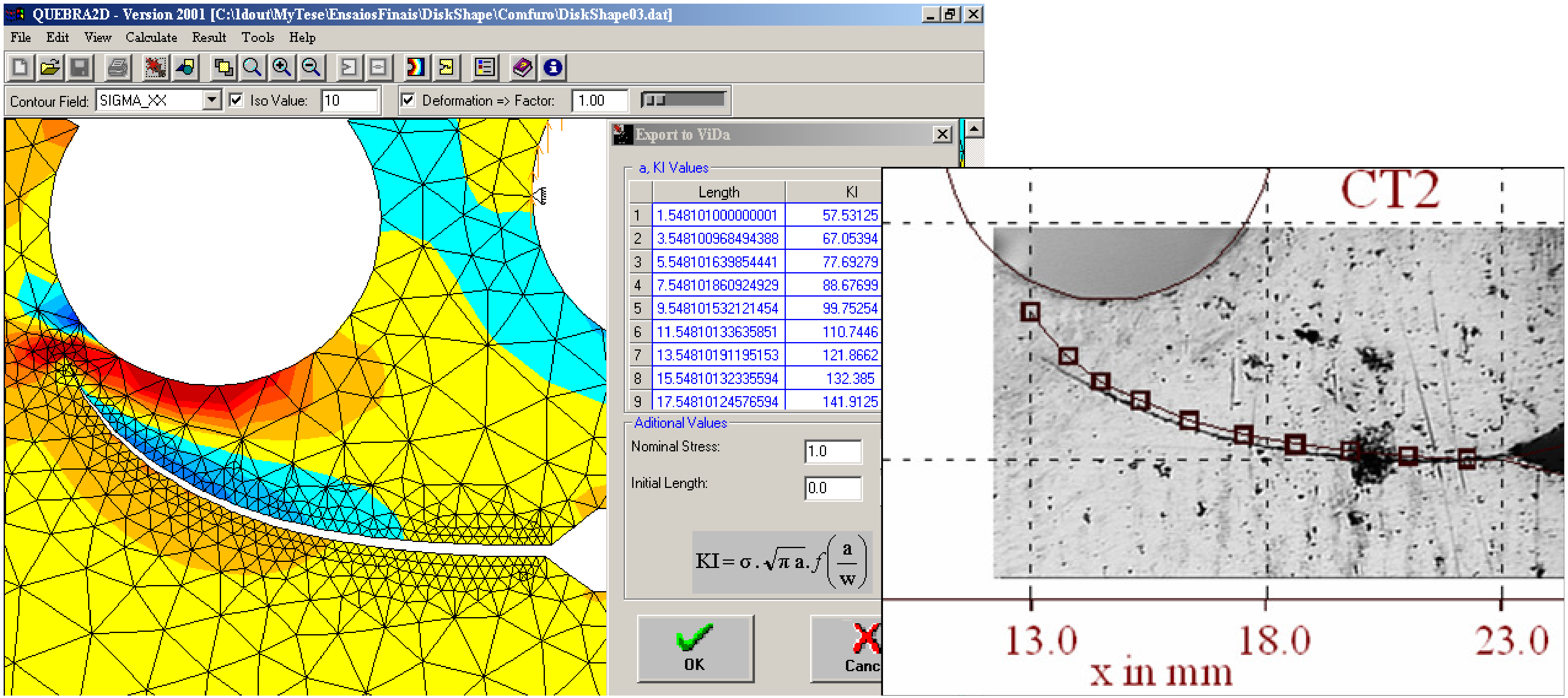
Requirements for underlying data representation (cont.)

–*The data structure should aid in key aspects of geometric modeling, such as surface intersection and automatic region recognition, as well as in surface and solid finite element mesh generation in arbitrary domains.*



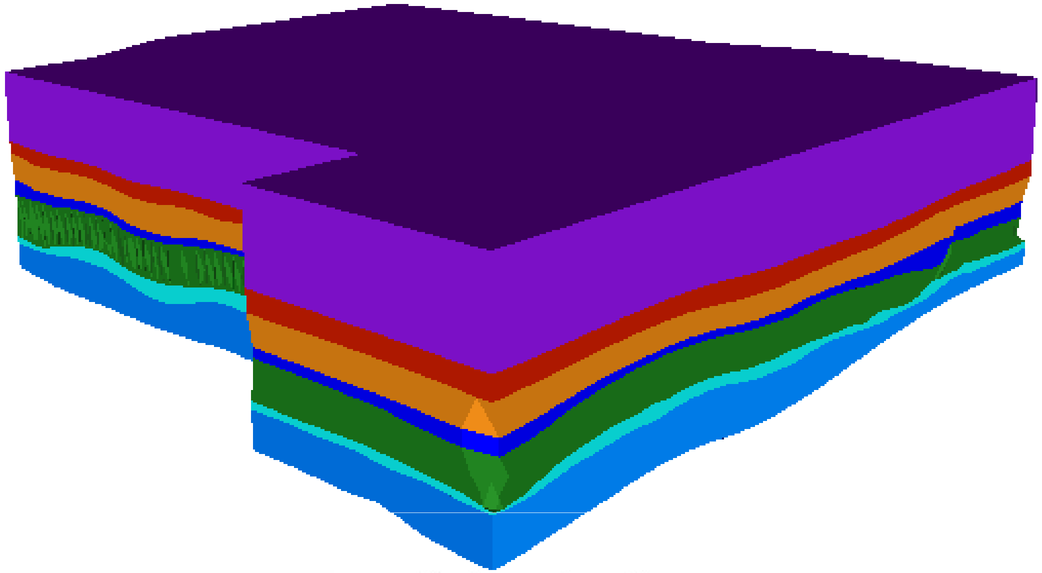
Requirements for underlying data representation (cont.)

–*The data structure must provide for efficient geometric operators, including automatic intersection detection and processing.*
This is necessary in simulations with evolving topology and geometry.

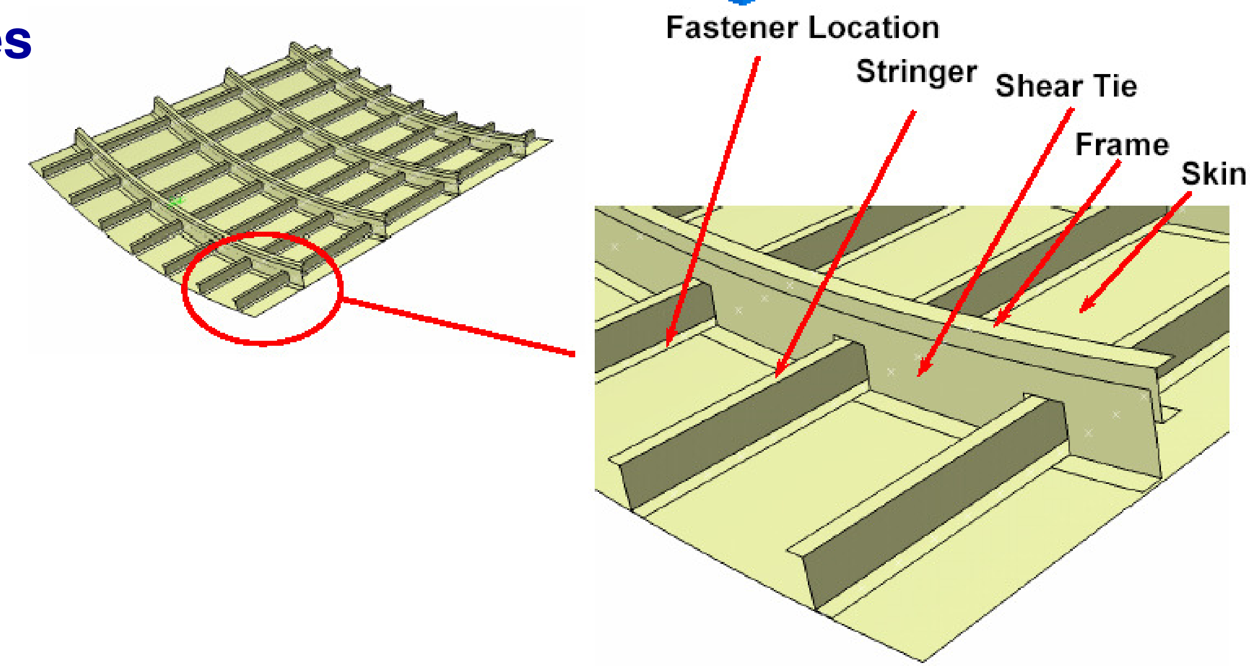


The need for non-manifold modeling

Multi-region modeling

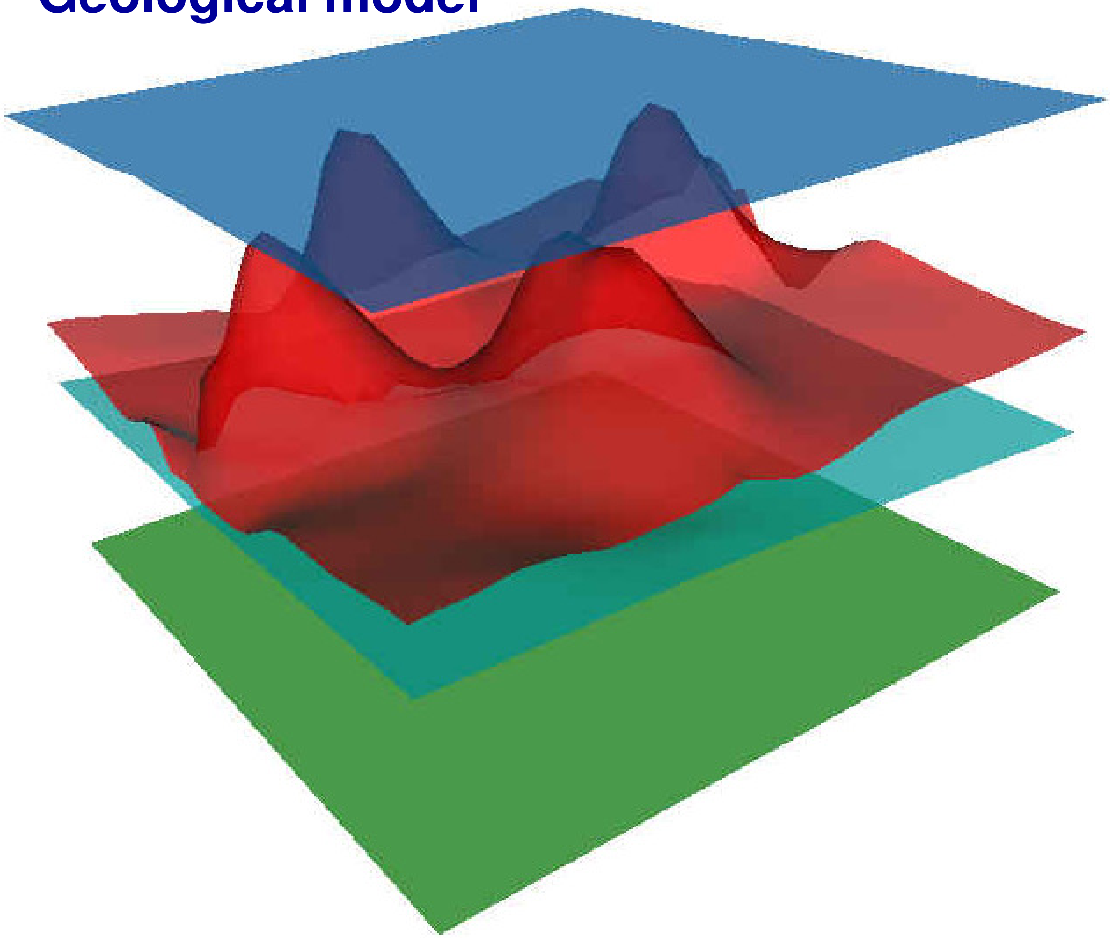


Degenerated structures

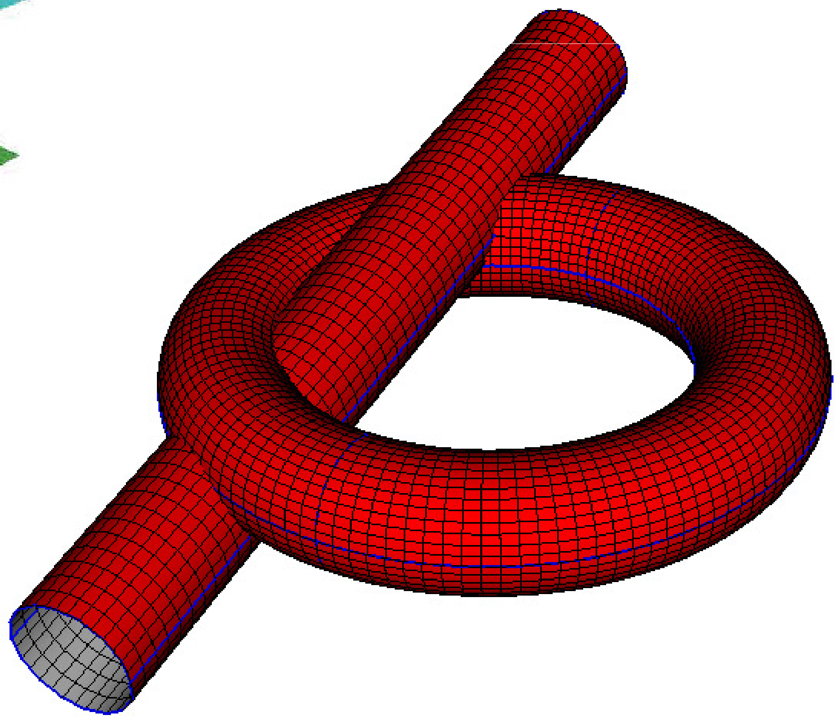


Natural modeling: surface patches as primitives

Geological model

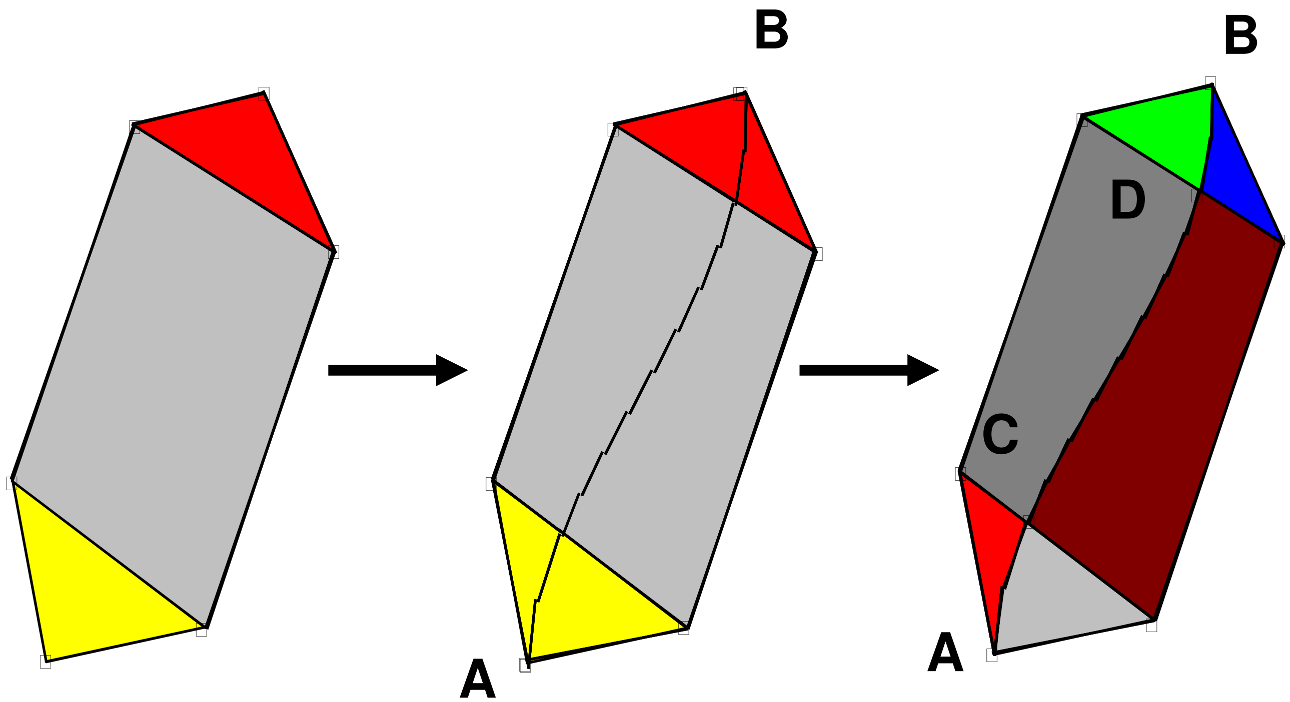


Manufactured model



Ideal environment: complete space subdivision

Space subdivision in 2D: high level operations



User action
+ basic function

System
response

Modelagem em Engenharia

Computer Graphics

Research areas:

- Geometric Modeling

needs due to visual inspections

- Visualization

Representation forms:

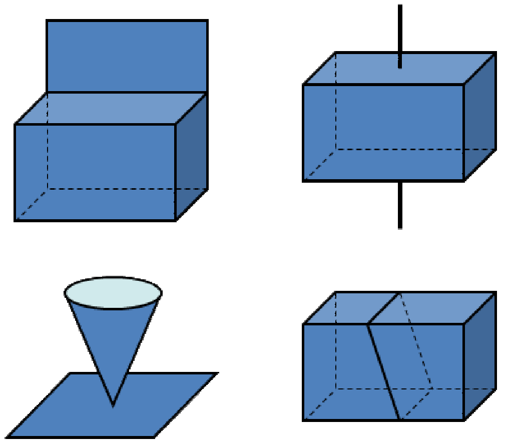
- Wireframe Modeling Technique
- Surface Modeling Technique
- Solid Modeling Technique

are traditionally constrained to work only with two-manifold solids.

Non-manifold Modeling

embods all of the capabilities of the three modeling forms in a unified representation.

allows topological conditions such as



has superior flexibility

but at a cost of a larger size and more complex data structure

Modelagem em Engenharia

Computer Graphics

Research areas:

- Geometric Modeling

  needs due to visual inspections

- Visualization

Representation forms:

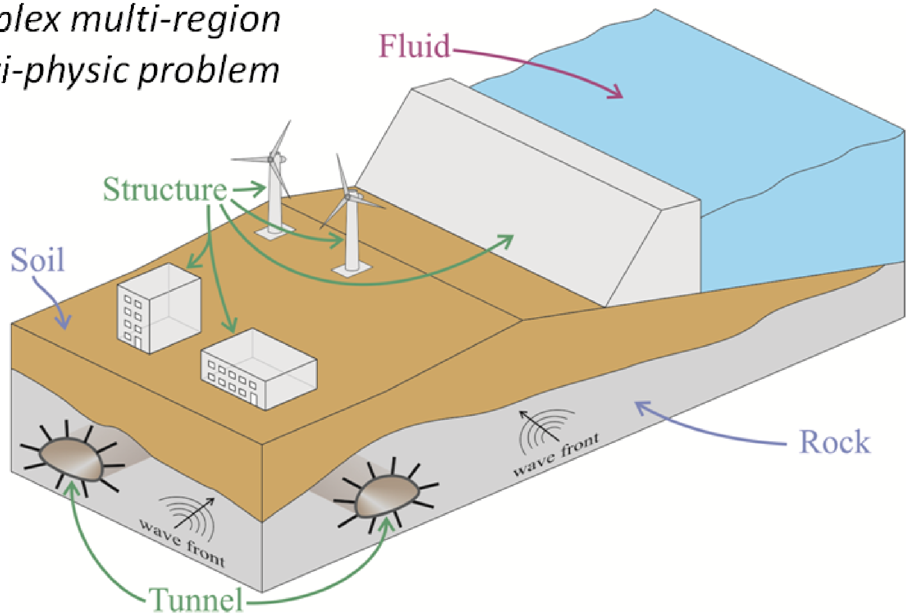
- Wireframe Modeling Technique
- Surface Modeling Technique
- Solid Modeling Technique

are traditionally constrained to work only with two-manifold solids.

Non-manifold Modeling

embods all of the capabilities of the three modeling forms in a unified representation.

complex multi-region multi-physic problem



Modelagem Geométrica

Modelagem Geométrica

- Criação, manipulação, manutenção e análise das representações das formas geométricas de objetos bi e tridimensionais.

- Aplicação em diversas áreas, como na produção de filmes, design de peças mecânicas industriais, visualização científica e reprodução de objetos para análise em engenharia.

Modelagem Geométrica

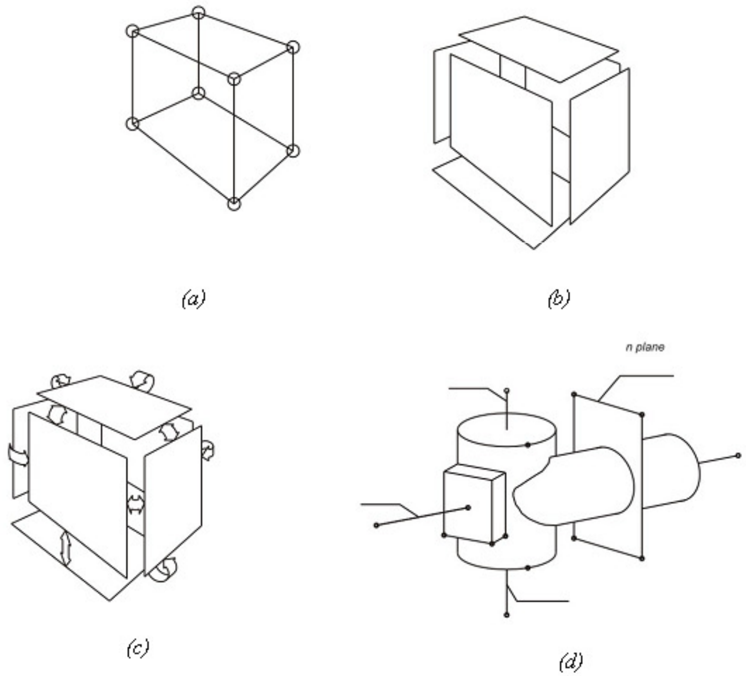
- **Evolução Histórica:**

a) Modelagem por arames

a) Modelagem por superfícies

b) Modelagem de sólidos

a) Modelagem *non-manifold*

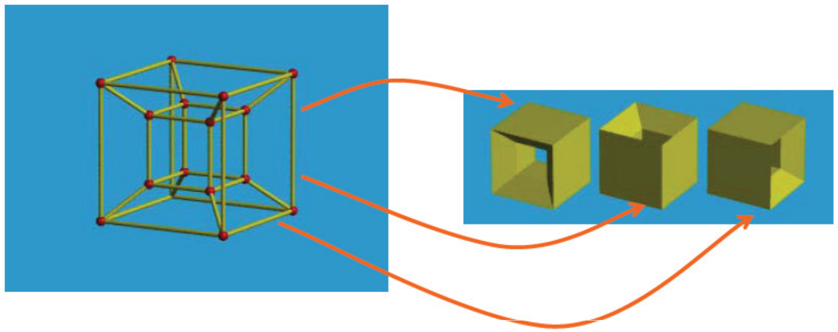


Modelagem Geométrica

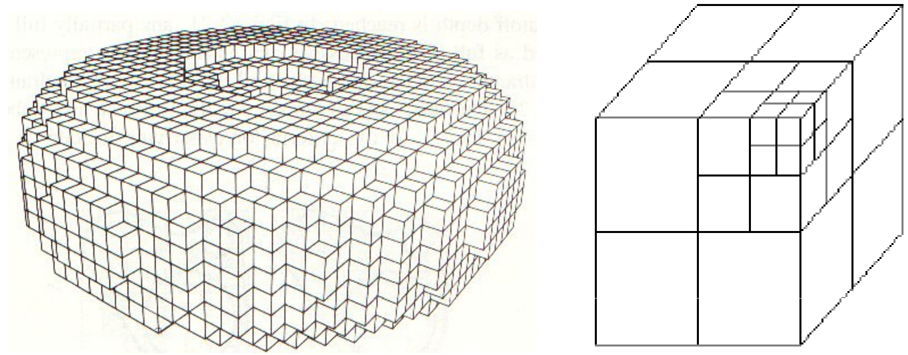
- **Formas de representação de sólidos**
  - ➢Modelos de decomposição
  - ➢Modelos B-Rep
  - ➢Modelos construtivos (CSG)
  - ➢Modelos híbridos

Modelagem de Sólidos

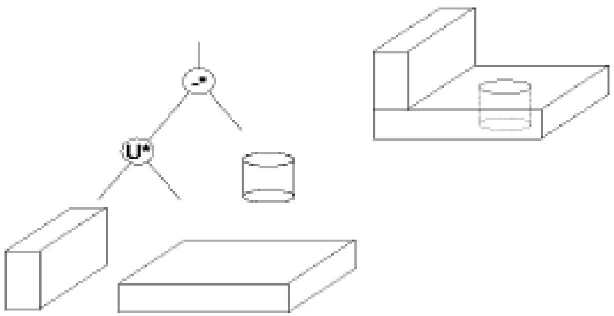
Wire Frame



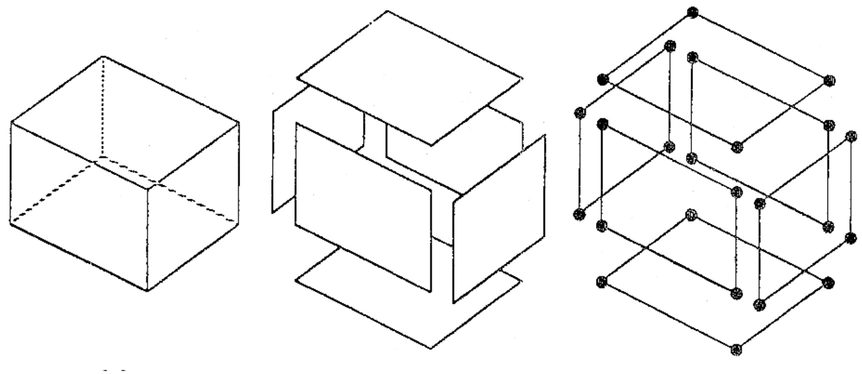
Cell Decomposition / Space Enumeration



Constructive Solid Geometry (CSG)

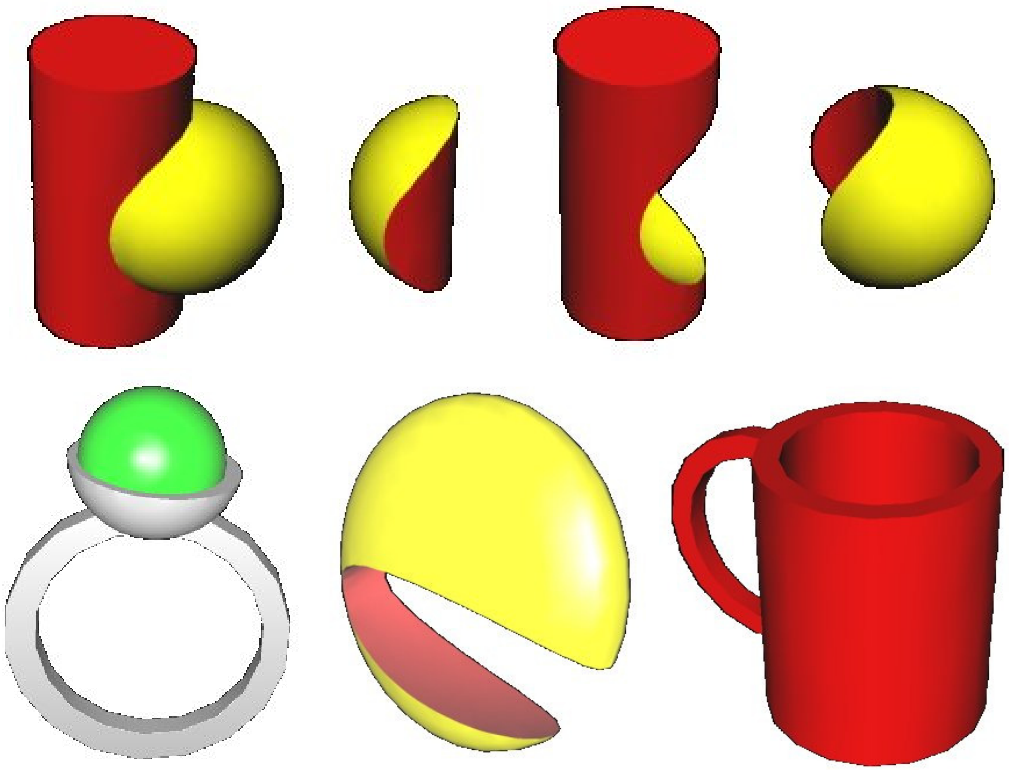


Boundary Representation (B-Rep)



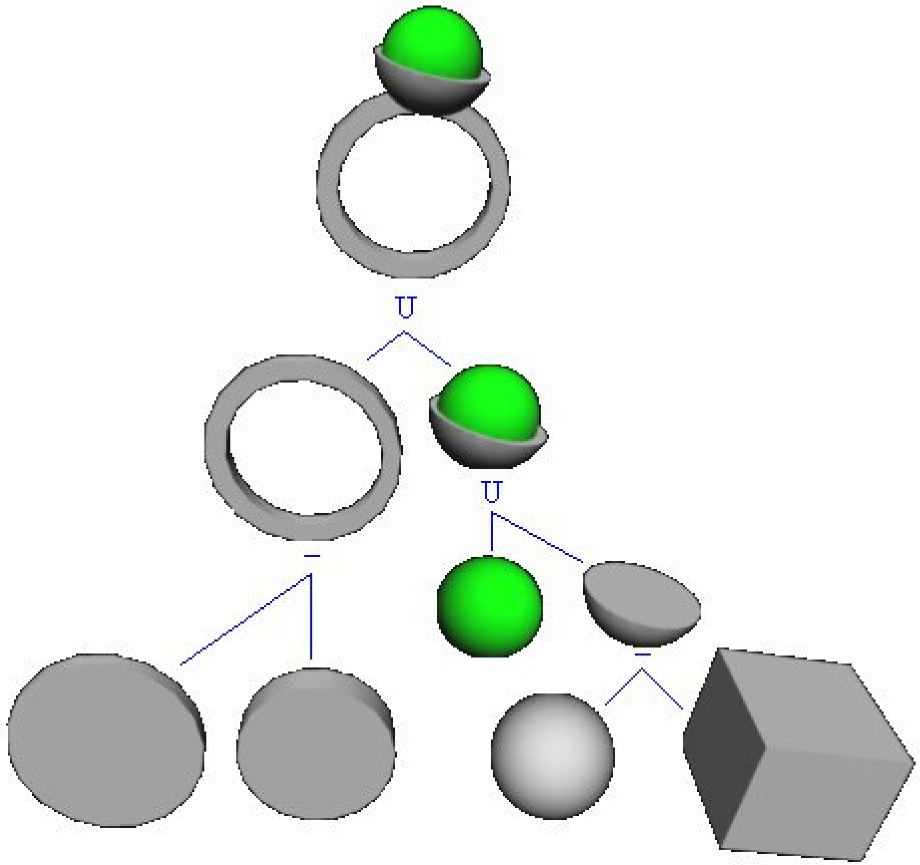
Modelagem Geométrica

- A Geometria Construtiva de Sólidos (CSG) utiliza as operações booleanas e de movimentos rígidos em primitivas simples para construir objetos sólidos mais complexos.



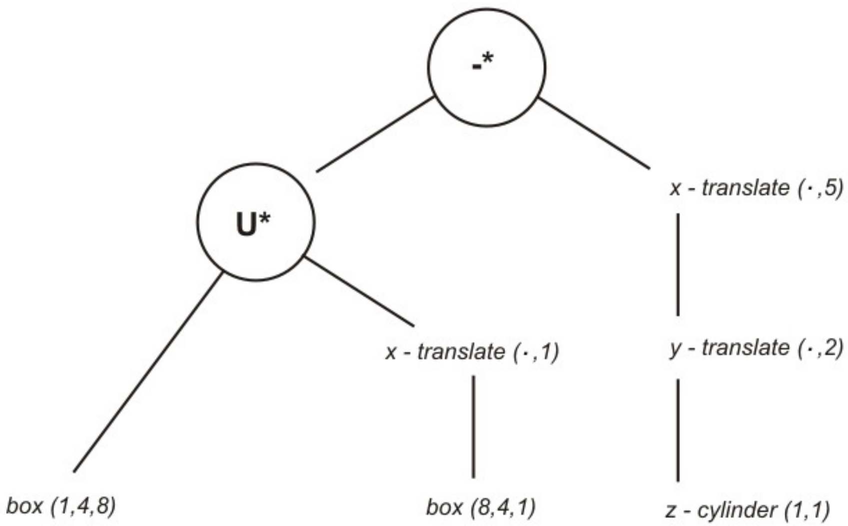
Modelagem Geométrica

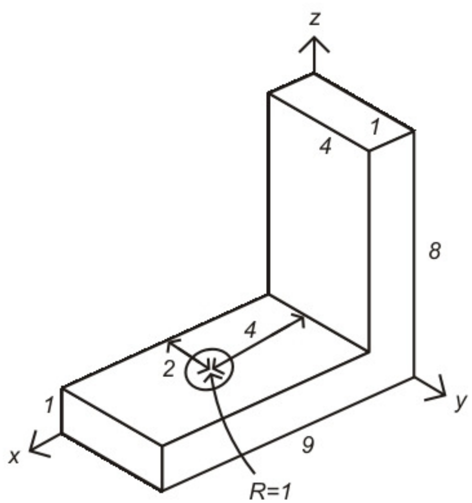
- **Árvore CSG**



Modelagem Geométrica

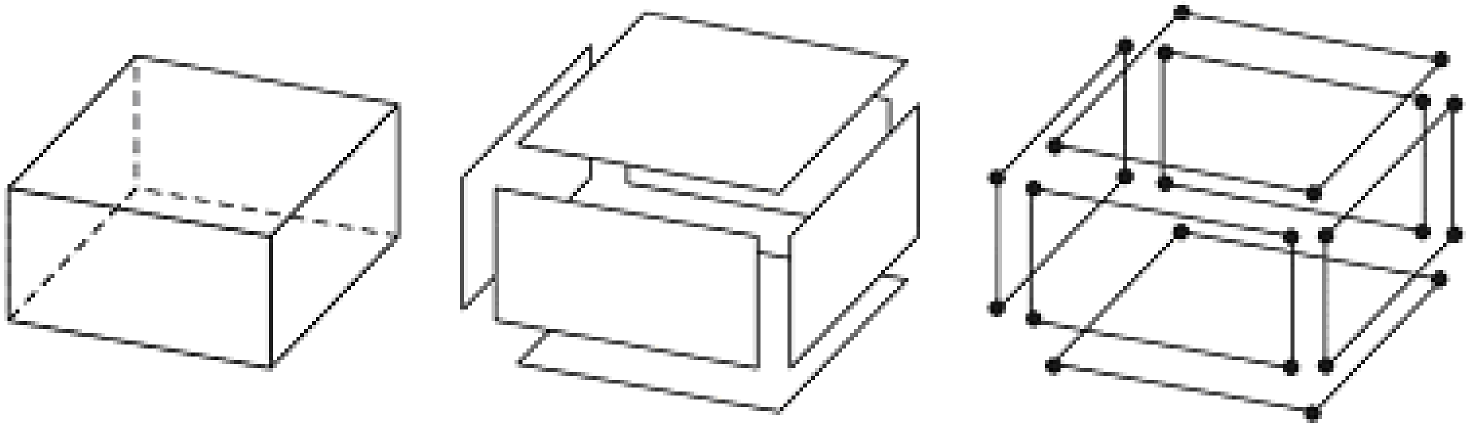
- **Árvore CSG**





Modelagem Geométrica

- Modelos B-Rep utilizam explicitamente as relações de adjacência entre os elementos topológicos (vértices, arestas e faces) para definir a fronteira topológica dos objetos.

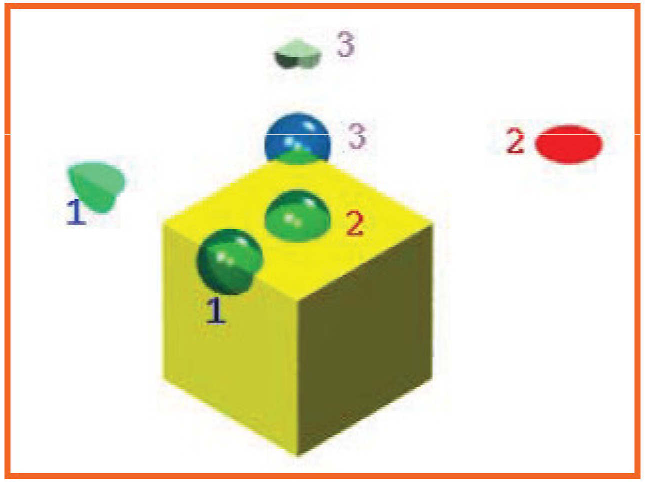


Modelagem Geométrica

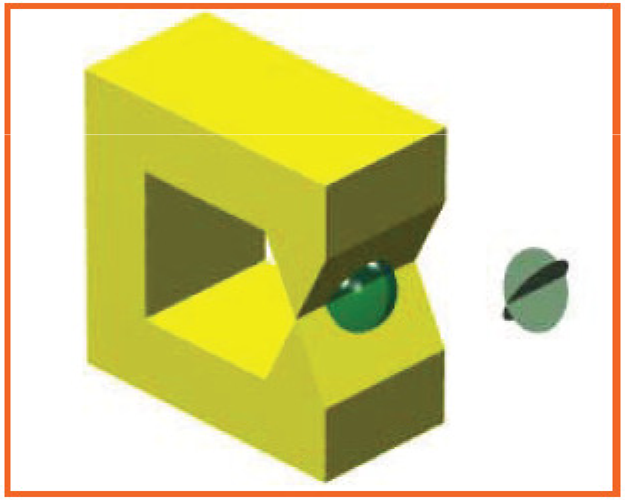
- **Modelagem *non-manifold***
  - Agrega todas as capacidades dos três tipos de modelagem anteriores.
  - Elimina as restrições ao domínio dos modelos analisados.
  - Permite a representação de estruturas internas ou pendentes de dimensão inferior.

Modelagem Geométrica

Manifold



Non-manifold



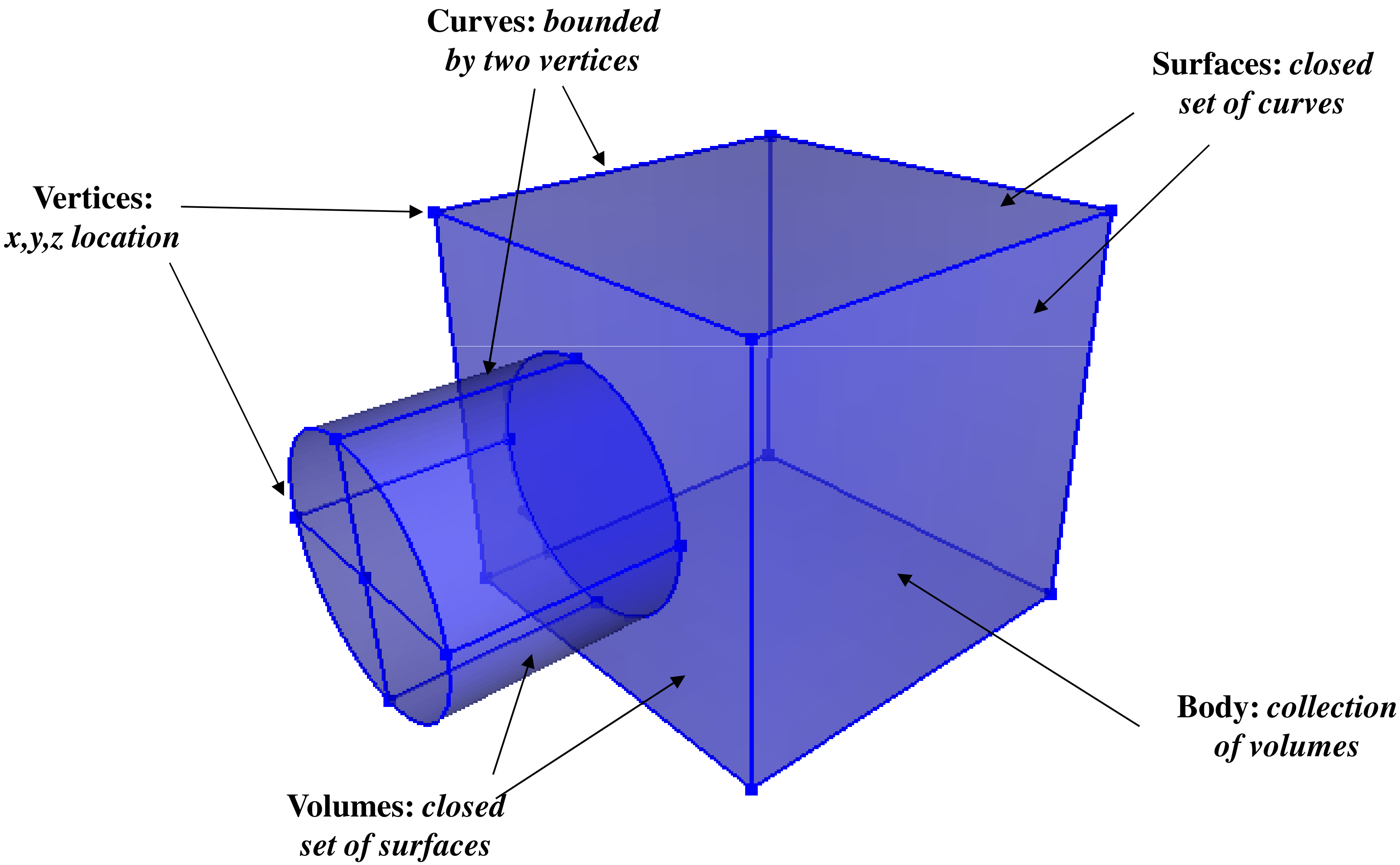
Modelagem Geométrica

- **Topologia e Geometria**

  - **Geometria –** conjunto de informações completas e essenciais para definir a forma e a localização espacial dos objetos.

  - **Topologia –** subconjunto de informações obtidas a partir da geometria do objeto. Invariante após a aplicação de transformações geométricas no objeto.

Entidades Geométricas e Topológicas



Modelagem Geométrica

- **Uso da topologia como base de um sistema de modelagem:**

1) Estabilidade do sistema

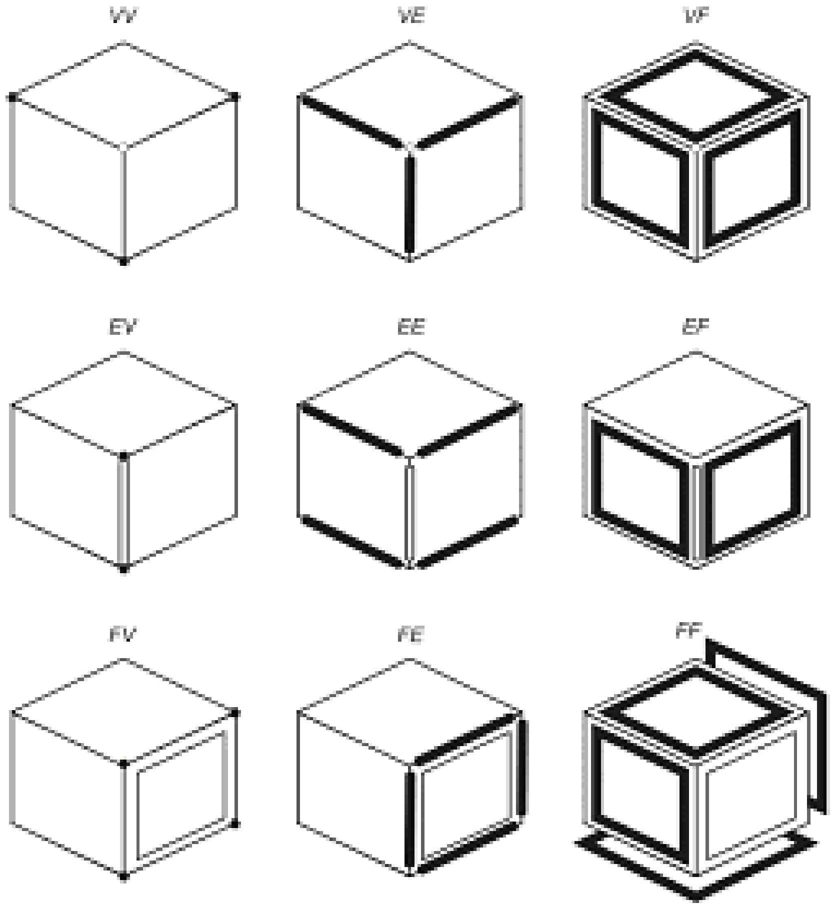
2) Evitação de erros numéricos

3) Separação das informações geométricas e topológicas

Modelagem Geométrica

- **Relações de adjacência**
  - Conectividade entre os elementos topológicos
  - Extraídas das informações geométricas do modelo
  - Utilização como base da estrutura de modelagem, garantindo a implementação de algoritmos mais simples e eficientes
  - Determinação de um conjunto mínimo suficiente de relações de adjacência

Modelagem Geométrica



Relações de adjacência entre vértices, arestas e faces

Modelagem Geométrica

- **Estruturas de dados topológicas**

  - Sistematização e organização das informações topológicas de um modelo a partir do armazenamento de um conjunto suficiente de relações de adjacência.

  - Principais elementos topológicos: vértices, arestas e faces.

  - Elementos topológicos adicionais: *loops*, cascas, regiões, uso de vértices, semi-arestas, uso de arestas, uso de *loops*, uso de faces.

Modelagem Geométrica

- **Estruturas de dados topológicas**

  - Exemplos de estruturas de dados consagradas em modelagem *manifold*:
    - *Winged-edge*
    - *Half-edge*

  - Estrutura de dados consagrada em modelagem *non-manifold*:
    - *Radial Edge*

Estrutura de Dados Topológica
Winged-Edge

Winged-Edge (Baumgart, 1972)

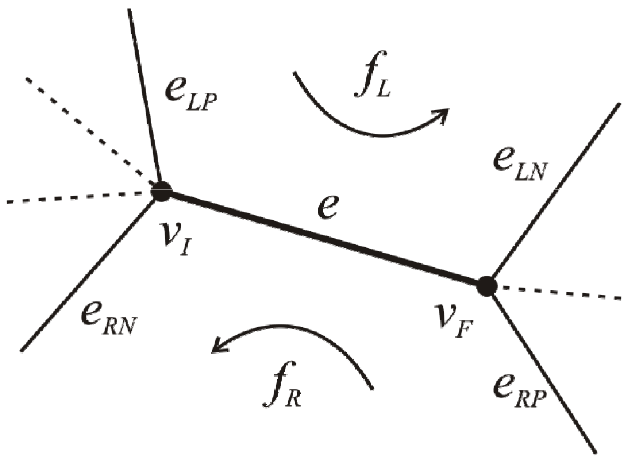


Tabela de vértices

v	x	y	z	e_I

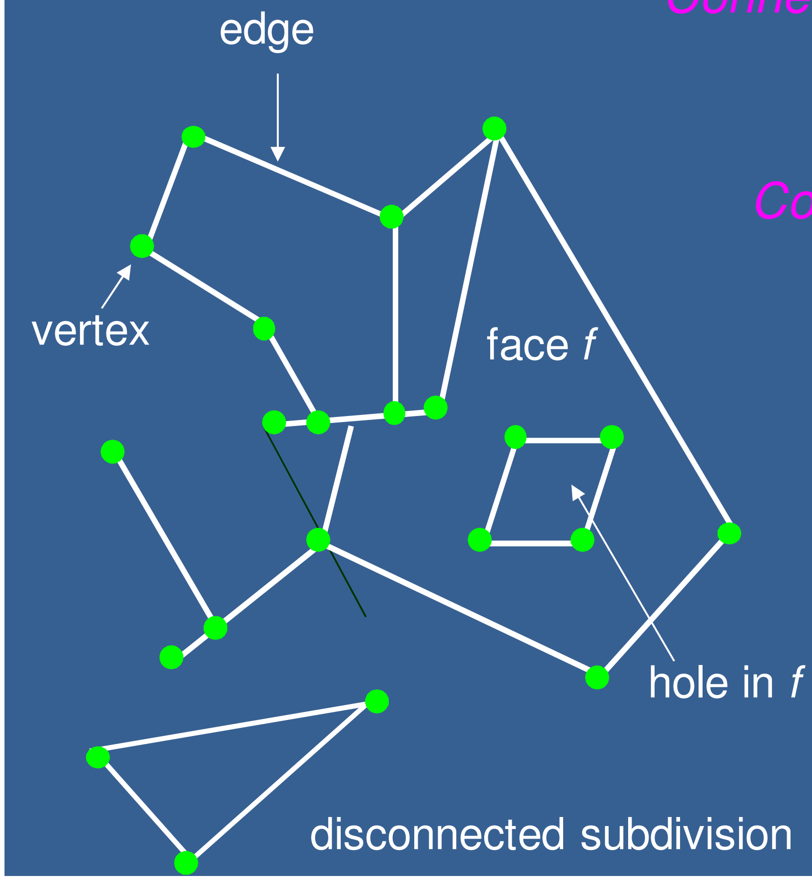
Tabela de faces

f	e_I

Tabela de arestas

e	v_I	v_F	f_L	f_R	e_{LP}	e_{LN}	e_{RP}	e_{RN}

Topological Data Structure - Planar Subdivision



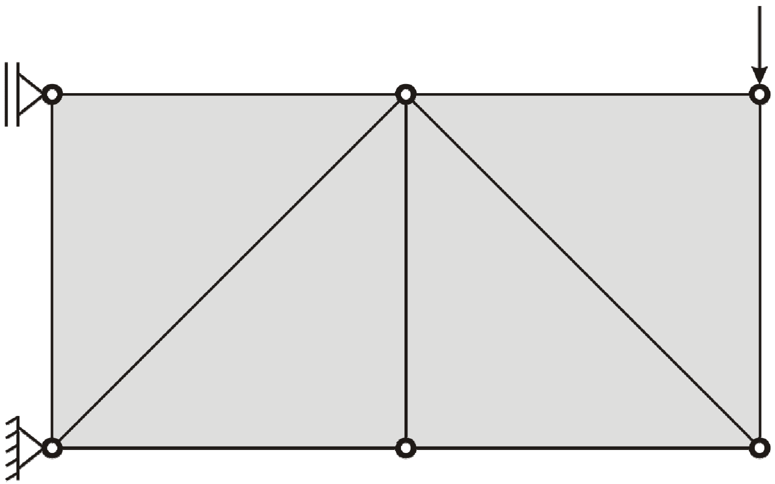
Induced by planar embedding of a graph.

Connected if the underlying graph is.

Complexity = #vertices + #edges + #faces

Typical operations:

- Walk around a face.
- Access one face from an adjacent one via a common edge.
- Visit all the edges adjacent to a vertex.



geometria

ponto

curva

superfície

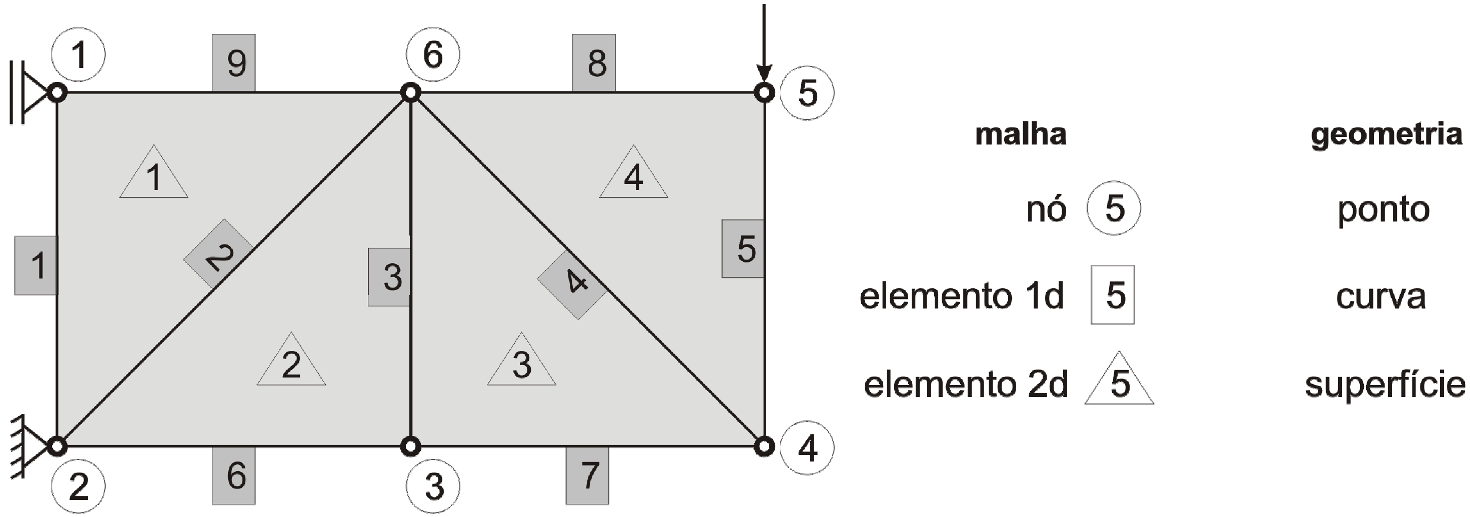


Tabela de nós

N	x	y	z
1	0	1	0
2	0	0	0
3	1	0	0
4	2	0	0
5	2	1	0
6	1	1	0

Tabela de conectividades

E	N_1	N_2	N_3
1	1	2	6
2	2	3	6
3	6	3	4
4	4	5	6

E	N_1	N_2
1	1	2
2	2	6
3	6	3
4	6	4
5	4	5
6	2	3
7	3	4
8	5	6
9	6	1

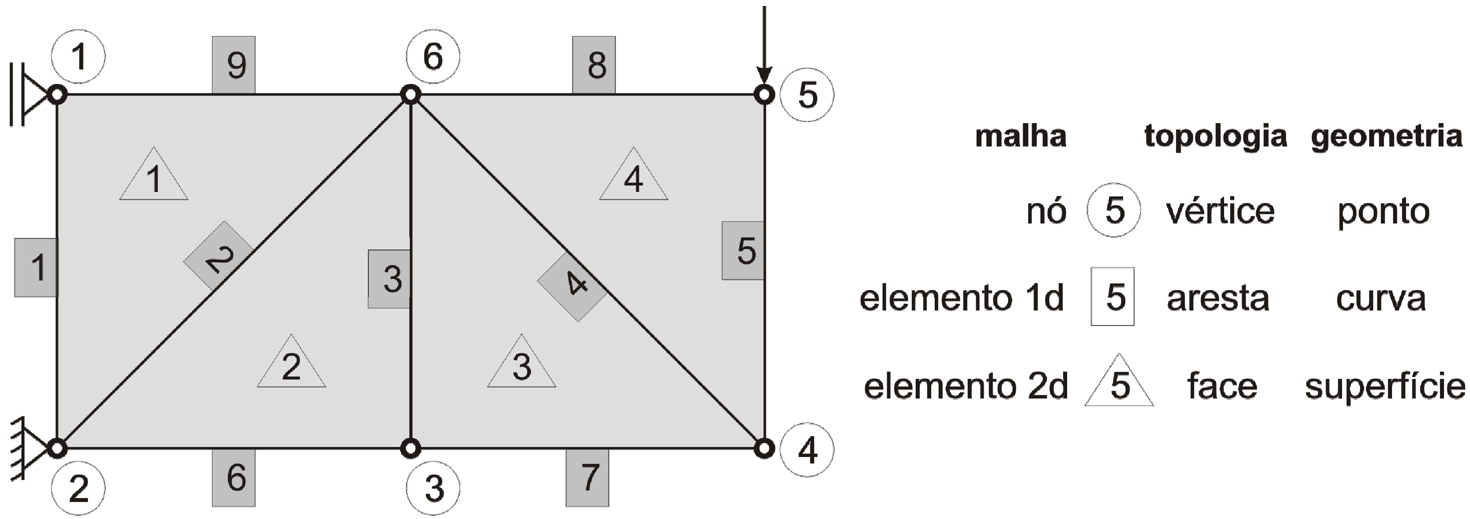


Tabela de nós

N	x	y	z
1	0	1	0
2	0	0	0
3	1	0	0
4	2	0	0
5	2	1	0
6	1	1	0

Tabela de conectividades

E	N_1	N_2	N_3
1	1	2	6
2	2	3	6
3	6	3	4
4	4	5	6

E	N_1	N_2
1	1	2
2	2	6
3	6	3
4	6	4
5	4	5
6	2	3
7	3	4
8	5	6
9	6	1

Tabela de vértices

v	x	y	z	e_I
1	0	1	0	1
2	0	0	0	6
3	1	0	0	7
4	2	0	0	4
5	2	1	0	8
6	1	1	0	2

Tabela de faces

f	e_I
0	1
1	2
2	6
3	4
4	4

Tabela de arestas

e	v_I	v_F	f_L	f_R	e_{LP}	e_{LN}	e_{RP}	e_{RN}
1	1	2	1	0	9	2	6	9
2	2	6	1	2	1	9	3	6
3	6	3	3	2	4	7	6	2
4	6	4	4	3	8	5	7	3
5	4	5	4	0	4	8	8	7
6	2	3	2	0	2	3	7	1
7	3	4	3	0	3	4	5	6
8	5	6	4	0	5	4	9	5
9	6	1	1	0	2	1	1	8

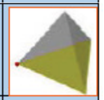
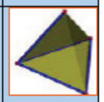
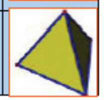
Euler Operators

Operator Name	Meaning	V	E	F	L	S	G
MEV	Make an edge and a vertex	+1	+1				
MFE	Make a face and an edge		+1	+1	+1		
MSFV	Make a shell, a face and a vertex	+1		+1	+1	+1	
MSG	Make a shell and a hole					+1	+1
MEKL	Make an edge and kill a loop		+1		-1		

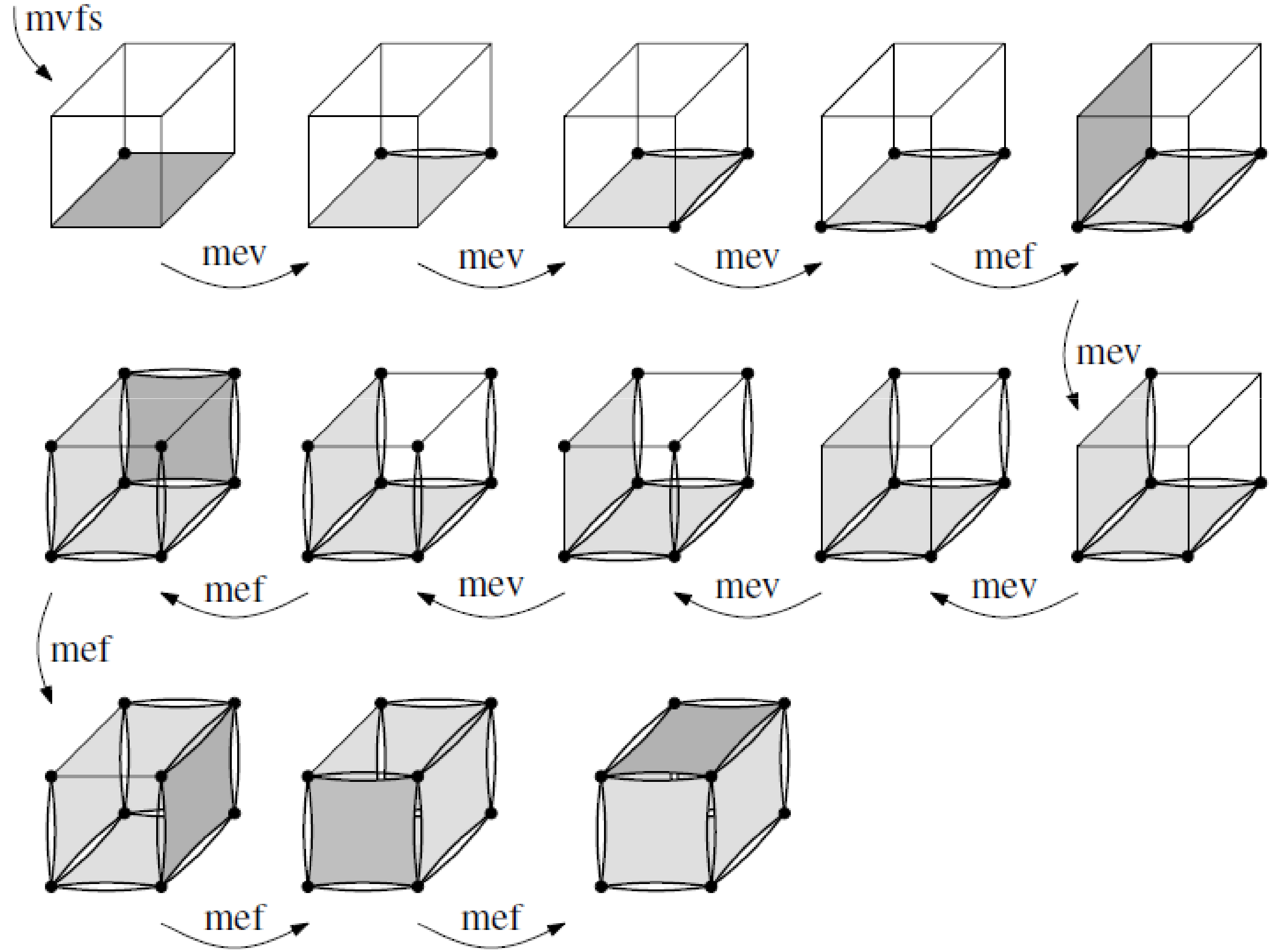
$$V - E + F = 2$$

Euler Operators

Operator Name	Meaning	V	E	F	L	S	G
MEV	Make an edge and a vertex	+1	+1				
MFE	Make a face and an edge		+1	+1	+1		
MSFV	Make a shell, a face and a vertex	+1		+1	+1	+1	
MSG	Make a shell and a hole					+1	+1
MEKL	Make an edge and kill a loop		+1		-1		

Operator Name	*Meaning*	*V*	*E*	*F*	*L*	*S*	*G*	**Result**
MSFV	**Make a shell, a face and a vertex**	**+1**		**+1**	**+1**	**+1**		
MEV	**Make an edge and a vertex**	**+1**	**+1**					
MEV	**Make an edge and a vertex**	**+1**	**+1**					
MEV	**Make an edge and a vertex**	**+1**	**+1**					
MFE	**Make a face and an edge**		**+1**	**+1**		**+1**		
MFE	**Make a face and an edge**		**+1**	**+1**		**+1**		
MFE	**Make a face and an edge**		**+1**	**+1**		**+1**		

Using Euler Operators to Construct a Solid



Estrutura de Dados Topológica
Half-Edge

Half-Edge (Mäntylä, 1988)

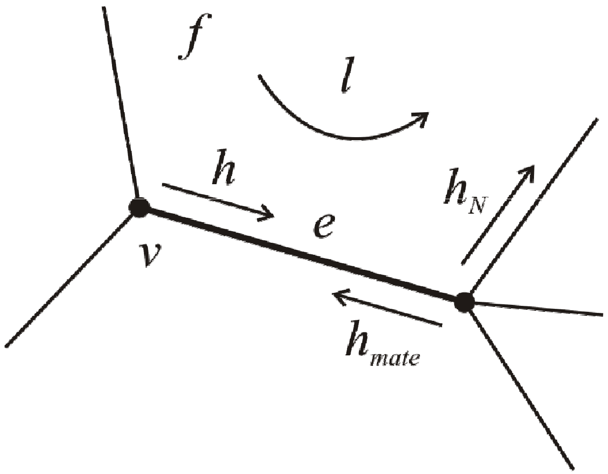


Tabela de vértices

v	x	y	z	h

Tabela de semi-arestas

h	e	v	l	h_N

Tabela de arestas

e	h_1	h_2

Tabela de laços

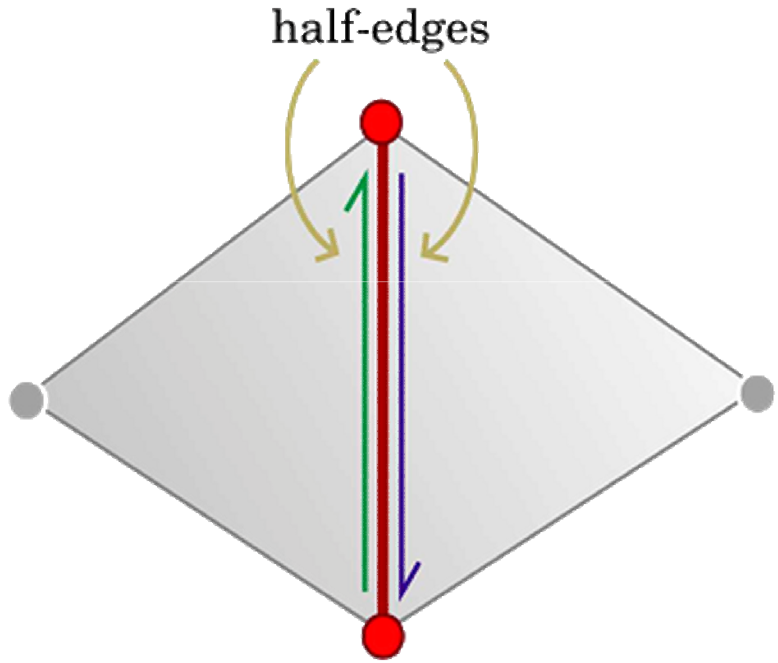
l	h	f	l_N

Tabela de faces

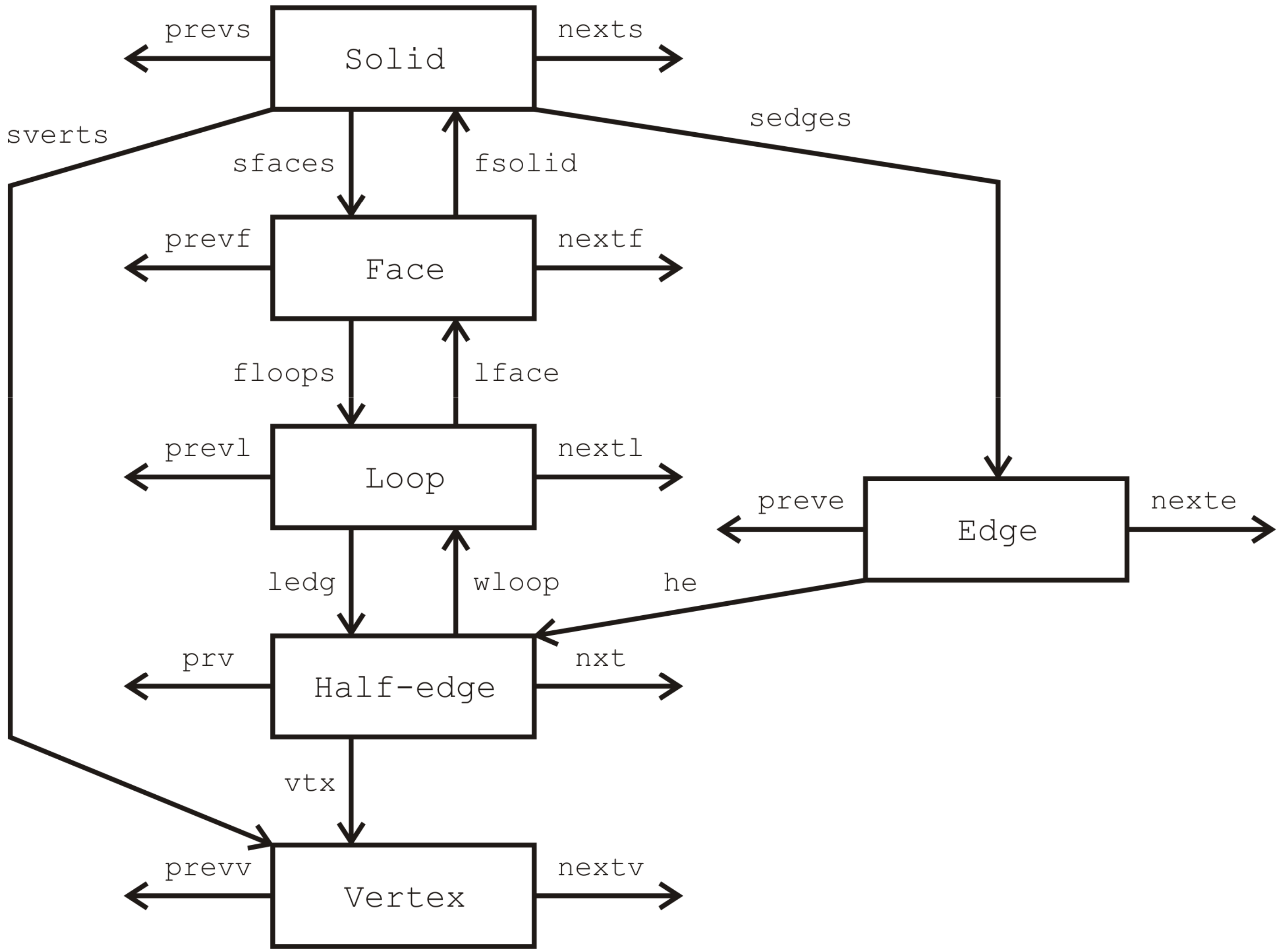
f	l_{out}	l_{in}

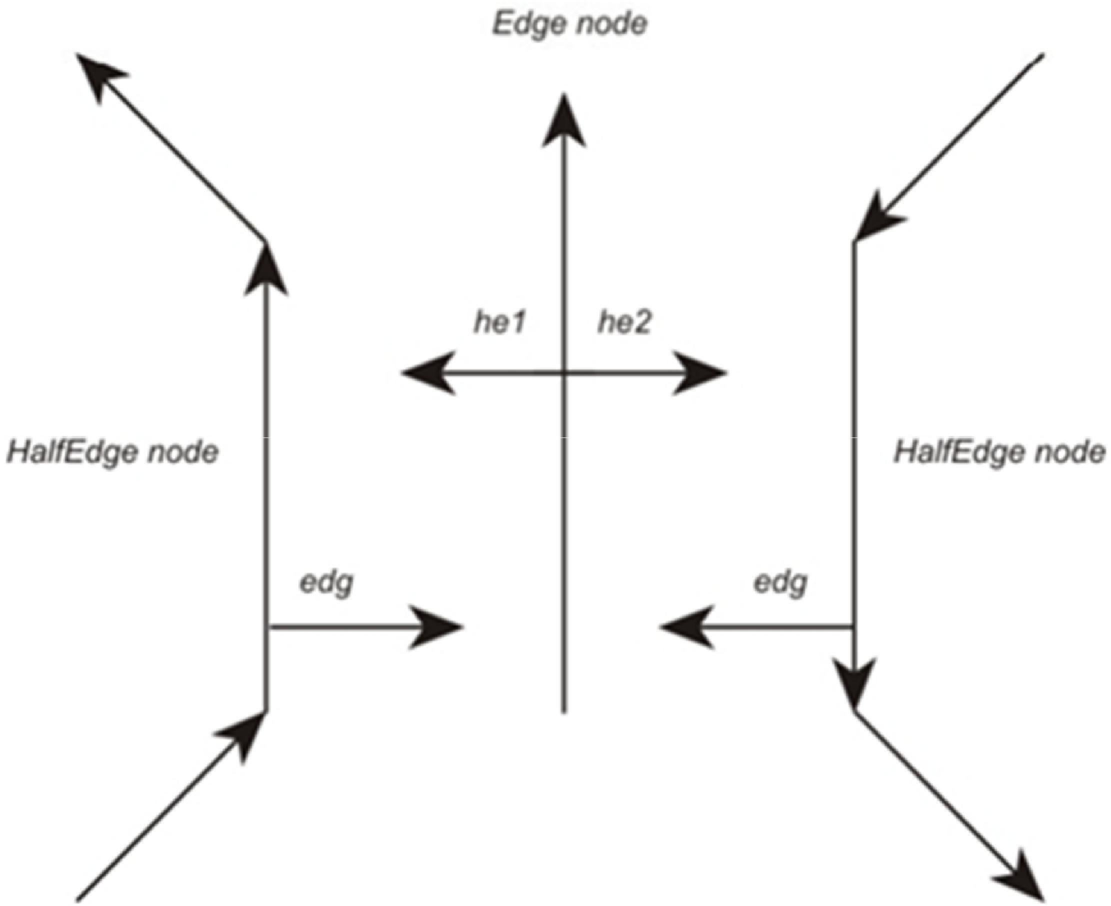
Hierarchy of Topological Levels

- Solid
- Face
- L*oop*
- *Half-Edge*
- Vertex
- Edge*

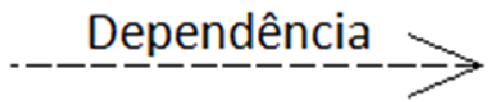


Half-Edge Data Structure Entities



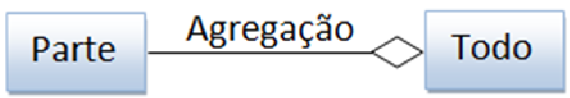

Edge node
he1
he2
HalfEdge node
HalfEdge node
edg
edg

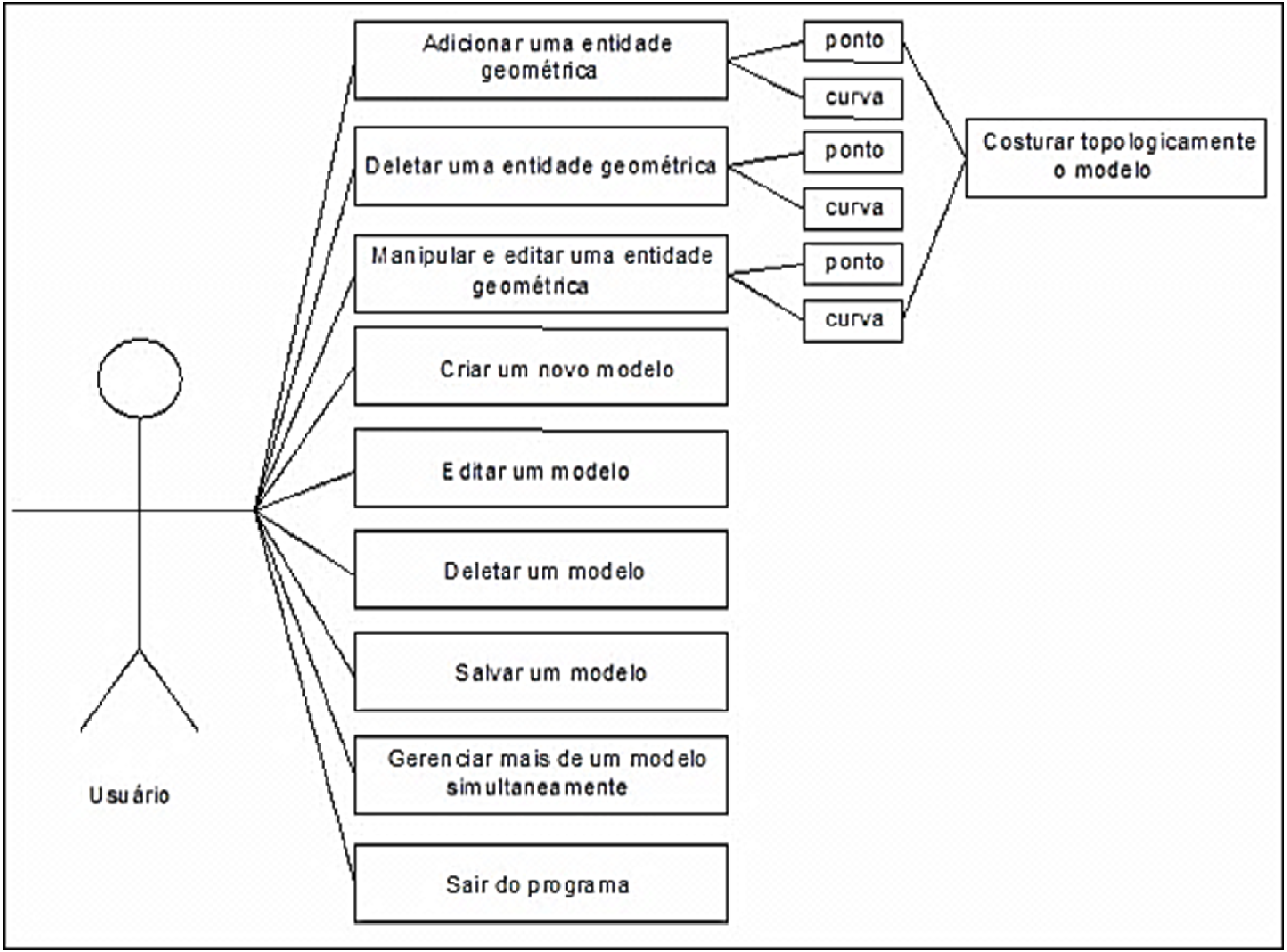
- Elements
  - Objects
  - Classes
- UML Diagrams
  - Use Cases
  - Robustez*
  - Sequence
  - Classes
- Relationships
  - Dependency



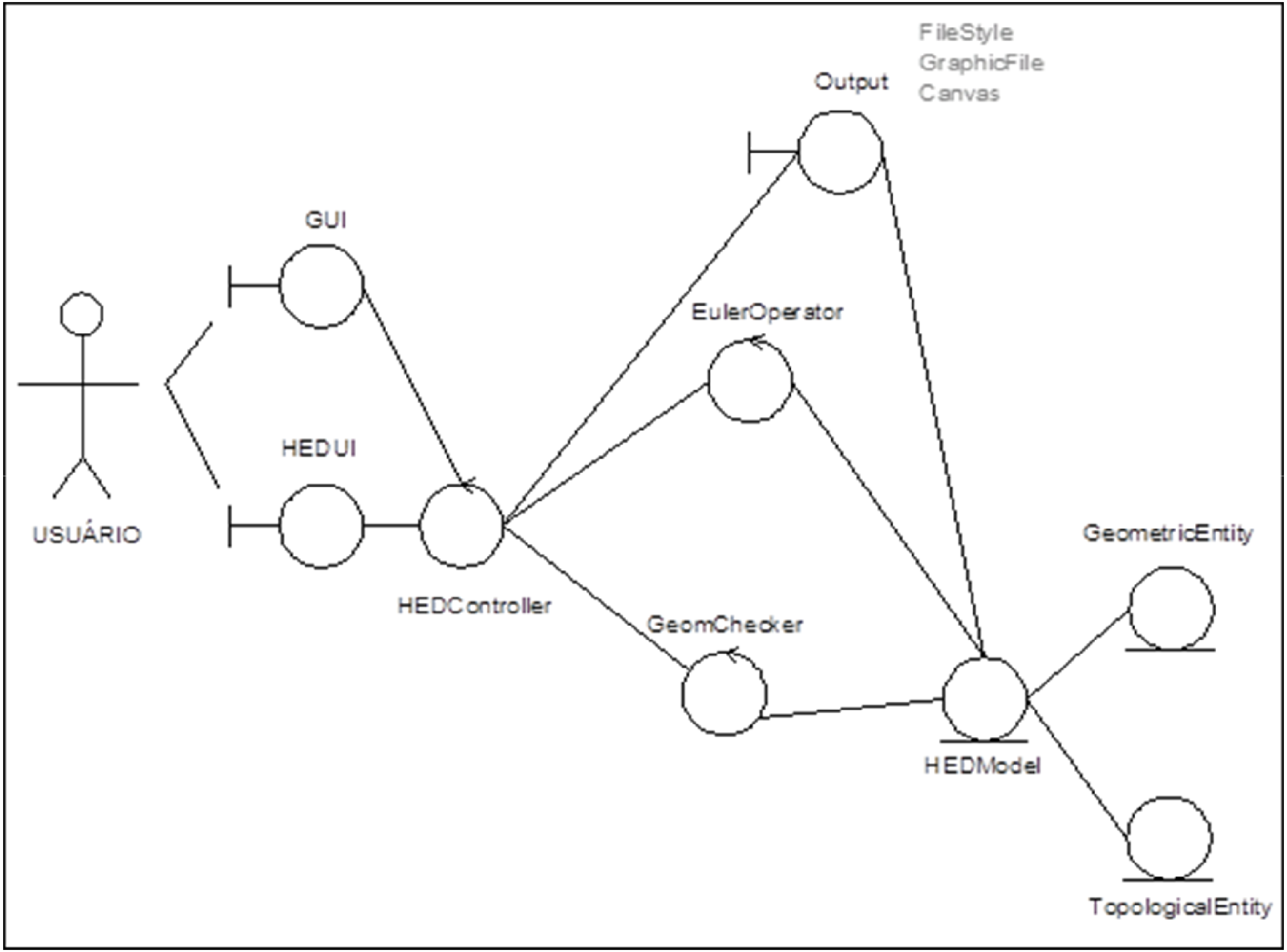
  - Association

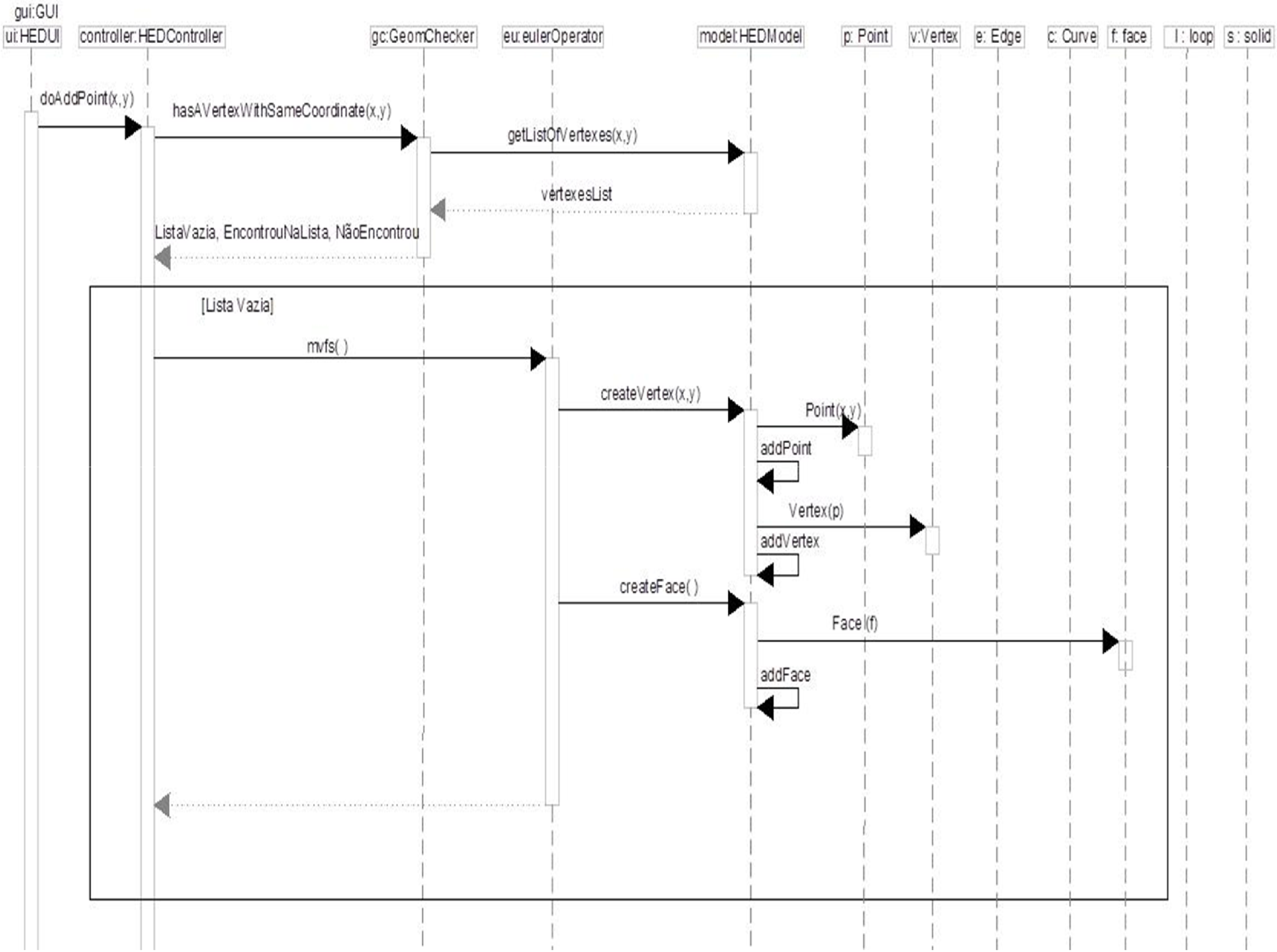
  - Aggregation

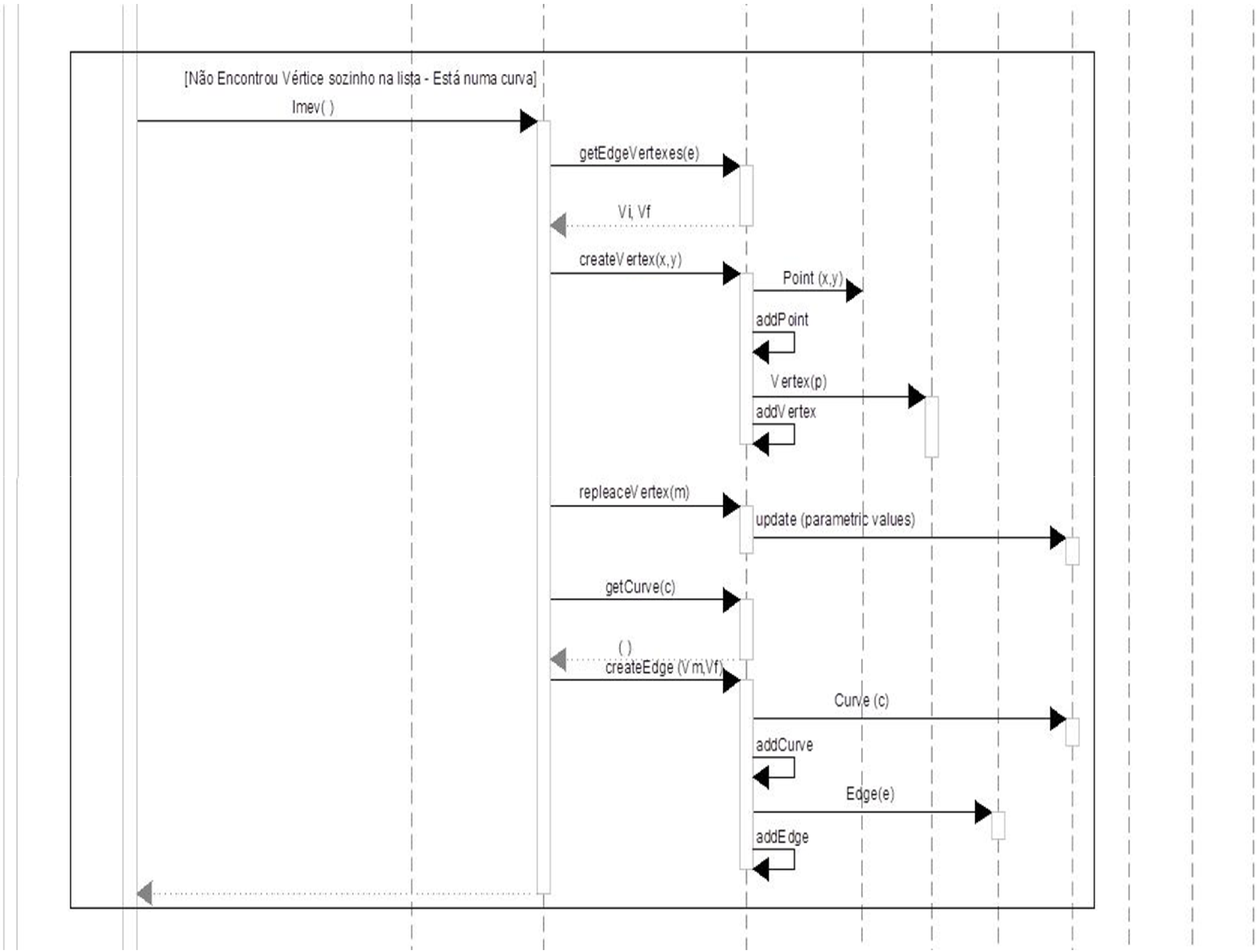


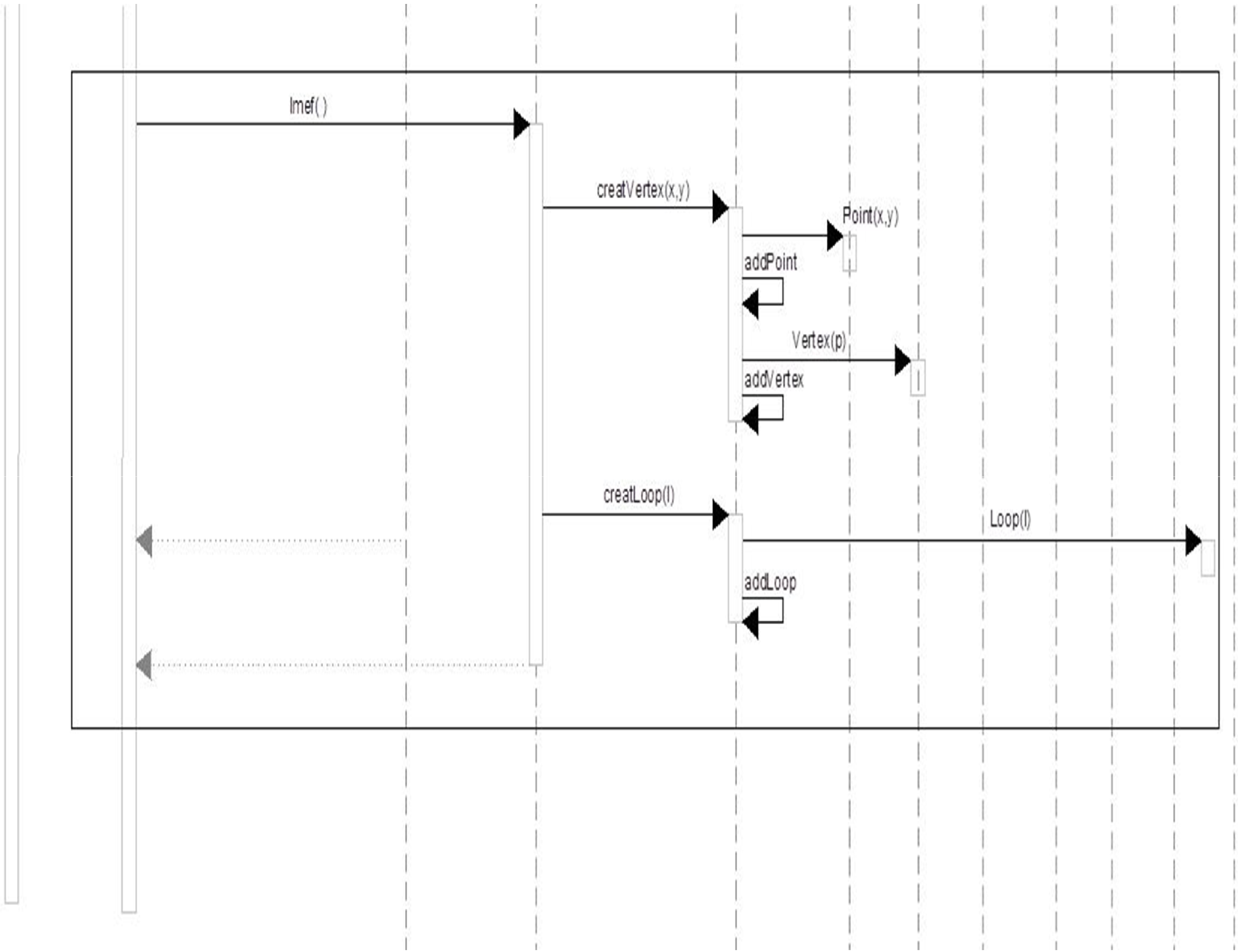


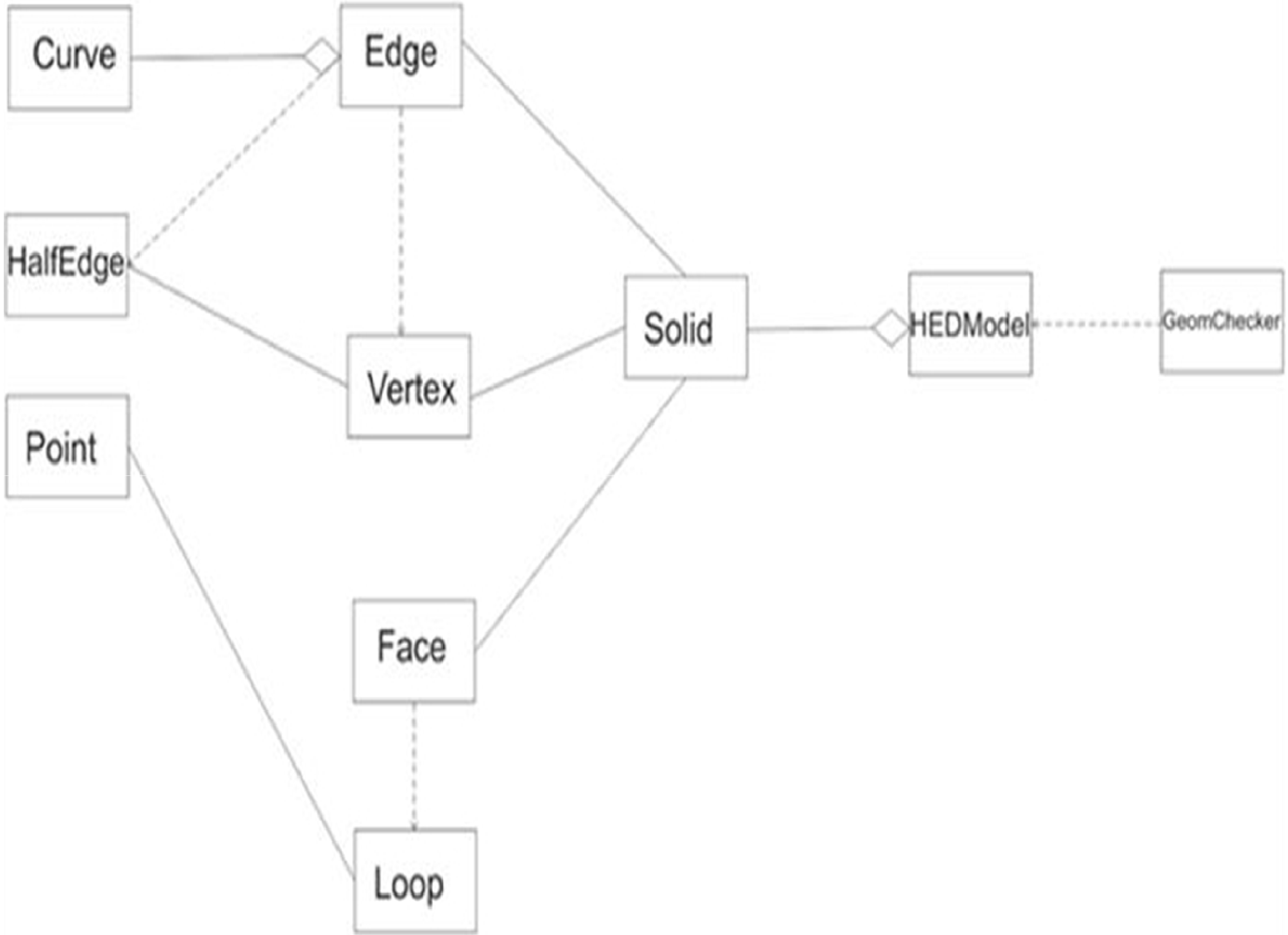
Adicionar uma entidade geométrica
ponto
curva
Deletar uma entidade geométrica
ponto
curva
Costurar topologicamente o modelo
Manipular e editar uma entidade geométrica
ponto
curva
Criar um novo modelo
Editar um modelo
Deletar um modelo
Salvar um modelo
Gerenciar mais de um modelo simultaneamente
Sair do programa
Usuário


FileStyle
GraphicFile
Canvas
Output
GUI
EulerOperator
HEDUI
USUÁRIO
HEDController
GeomChecker
GeometricEntity
HEDModel
TopologicalEntity

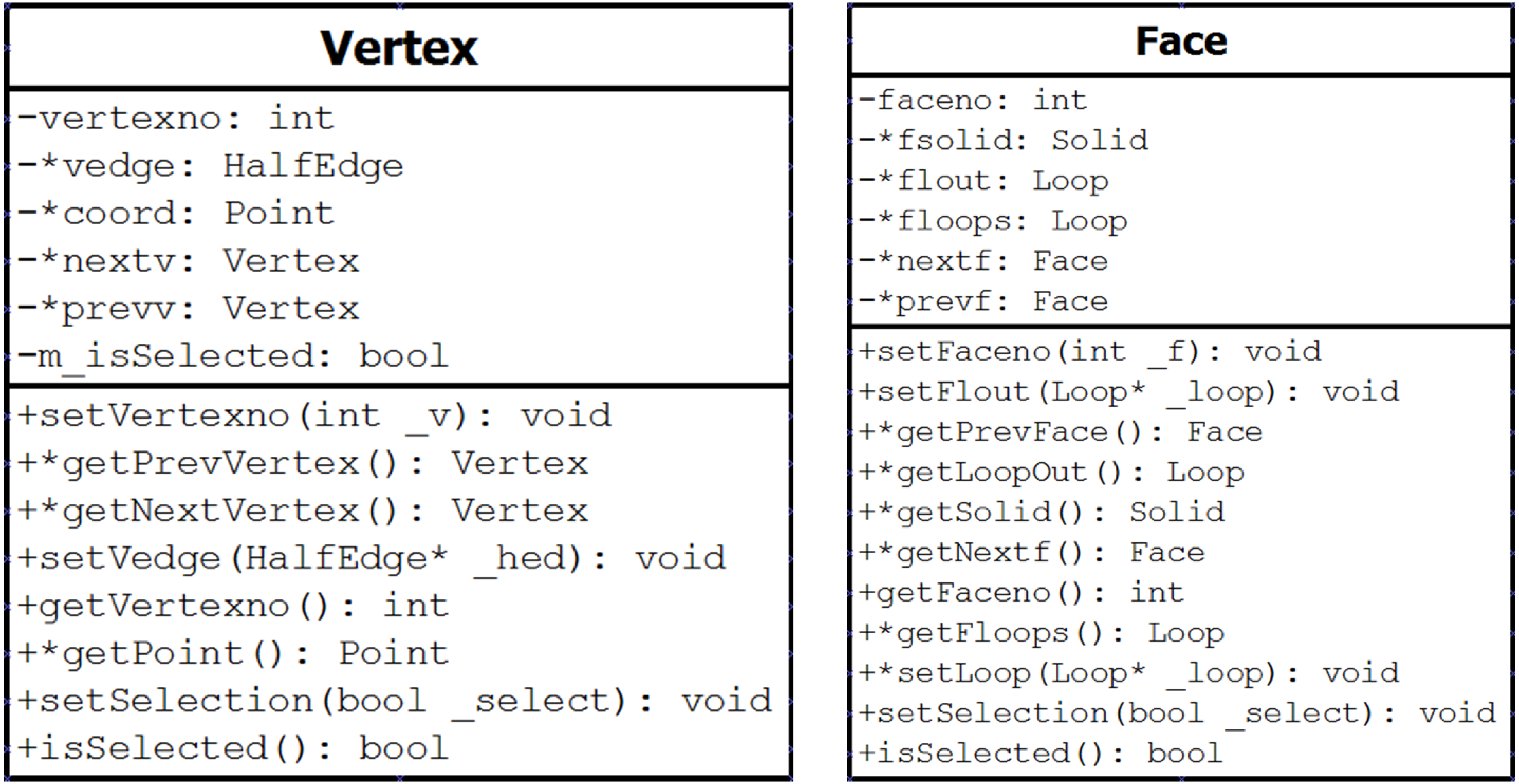

gui:GUI
ui:HEDUI
controller:HEDController
gc:GeomChecker
eu:eulerOperator
model:HEDModel
p: Point
v:Vertex
e: Edge
c: Curve
f: face
l : loop
s : solid
doAddPoint(x,y)
hasAVertexWithSameCoordinate(x,y)
getListOfVertexes(x,y)
vertexesList
ListaVazia, EncontrouNaLista, NãoEncontrou
[Lista Vazia]
mvfs( )
createVertex(x,y)
Point(x,y)
addPoint
Vertex(p)
addVertex
createFace( )
Face(f)
addFace

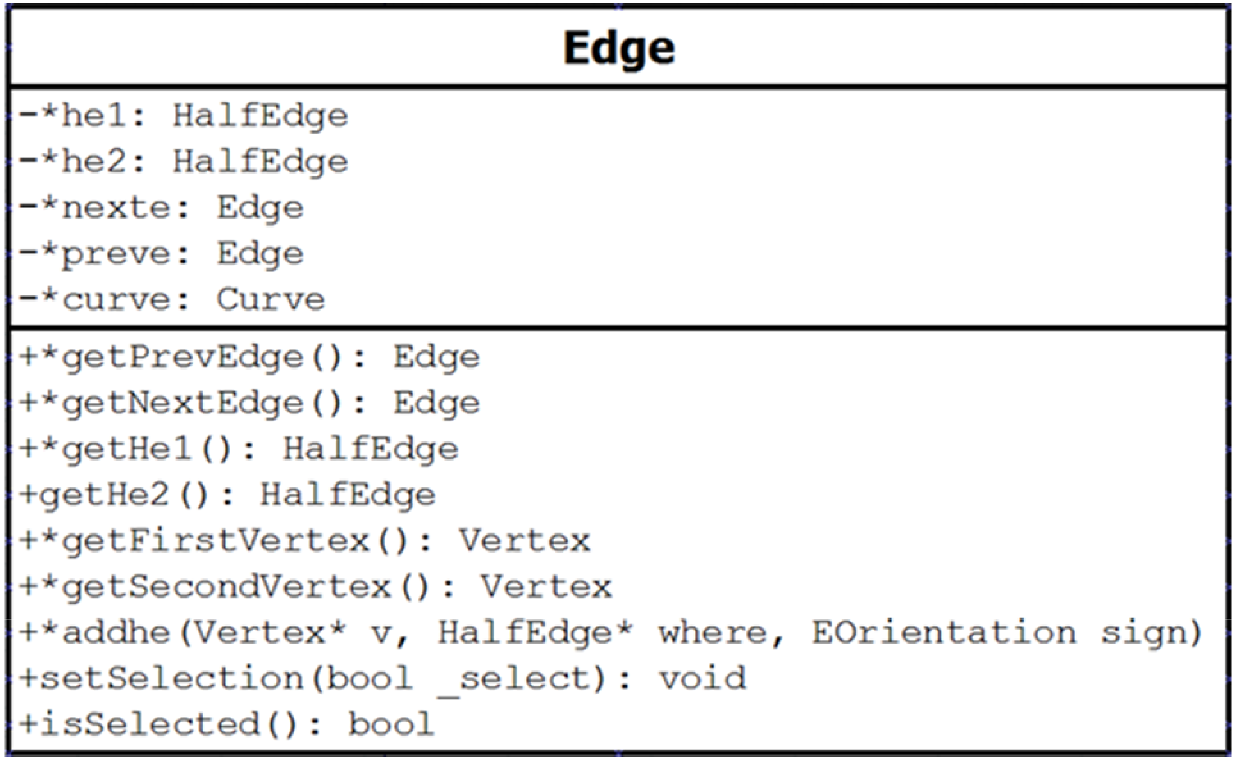

[Não Encontrou Vértice sozinho na lista - Está numa curva]
lmev( )
getEdgeVertexes(e)
Vi, Vf
createVertex(x,y)
Point (x,y)
addPoint
Vertex(p)
addVertex
repleaceVertex(m)
update (parametric values)
getCurve(c)
( )
createEdge (Vm,Vf)
Curve (c)
addCurve
Edge(e)
addEdge

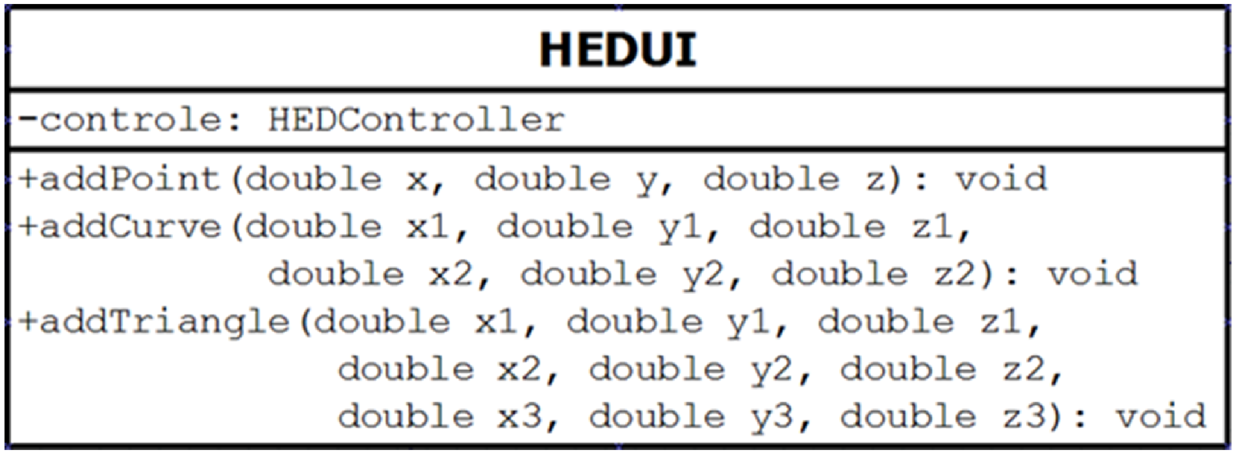

lmef( )
creatVertex(x,y)
Point(x,y)
addPoint
Vertex(p)
addVertex
creatLoop(l)
Loop(l)
addLoop



Curve
Edge
HalfEdge
Solid
HEDModel
GeomChecker
Vertex
Point
Face
Loop


Vertex
-vertexno: int
-*vedge: HalfEdge
-*coord: Point
-*nextv: Vertex
-*prevv: Vertex
-m_isSelected: bool
+setVertexno(int _v): void
+*getPrevVertex(): Vertex
+*getNextVertex(): Vertex
+setVedge(HalfEdge* _hed): void
+getVertexno(): int
+*getPoint(): Point
+setSelection(bool _select): void
+isSelected(): bool
Face
-faceno: int
-*fsolid: Solid
-*flout: Loop
-*floops: Loop
-*nextf: Face
-*prevf: Face
+setFaceno(int _f): void
+setFlout(Loop* _loop): void
+*getPrevFace(): Face
+*getLoopOut(): Loop
+*getSolid(): Solid
+*getNextf(): Face
+getFaceno(): int
+*getFloops(): Loop
+*setLoop(Loop* _loop): void
+setSelection(bool _select): void
+isSelected(): bool


Edge
-*he1: HalfEdge
-*he2: HalfEdge
-*nexte: Edge
-*preve: Edge
-*curve: Curve
+*getPrevEdge(): Edge
+*getNextEdge(): Edge
+*getHe1(): HalfEdge
+getHe2(): HalfEdge
+*getFirstVertex(): Vertex
+*getSecondVertex(): Vertex
+*addhe(Vertex* v, HalfEdge* where, EOrientation sign)
+setSelection(bool _select): void
+isSelected(): bool


HEDUI
-controle: HEDController
+addPoint(double x, double y, double z): void
+addCurve(double x1, double y1, double z1,
double x2, double y2, double z2): void
+addTriangle(double x1, double y1, double z1,
double x2, double y2, double z2,
double x3, double y3, double z3): void

EulerOperator

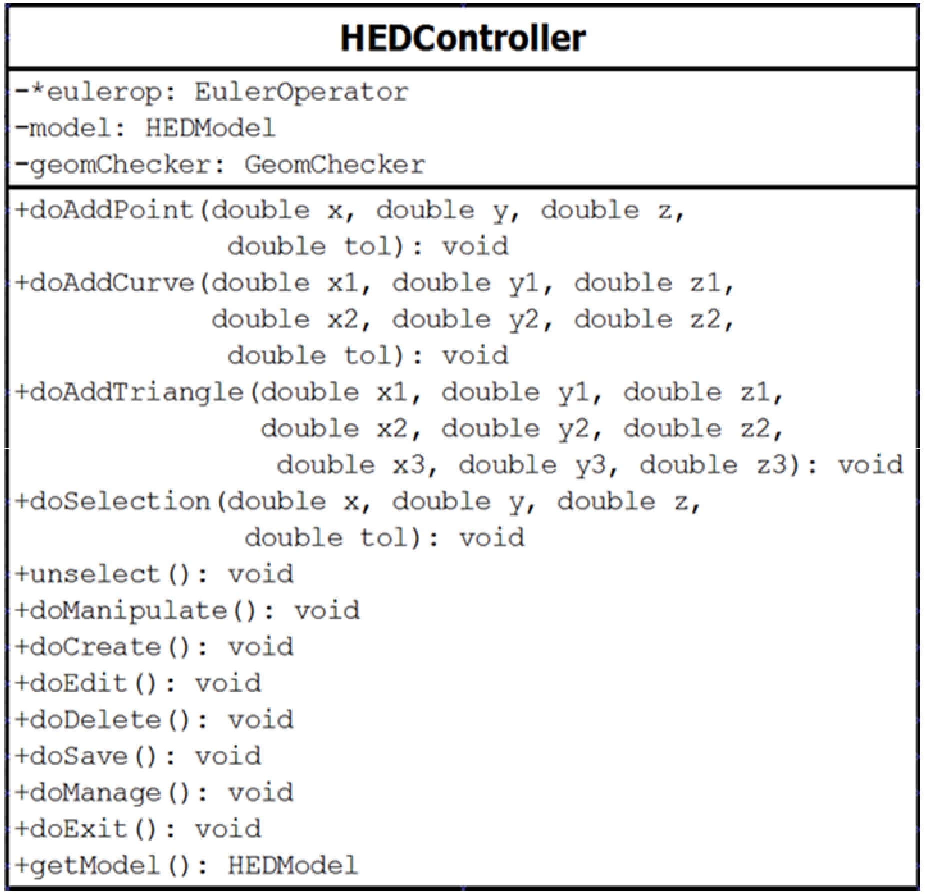
```
-lmev (HalfEdge* he1, HalfEdge* he2, int v, double x, double y, double z): void
-*lmef (HalfEdge* he1, HalfEdge* he2, int f): Face
-lkmer (HalfEdge* he1, HalfEdge* he2): void
```

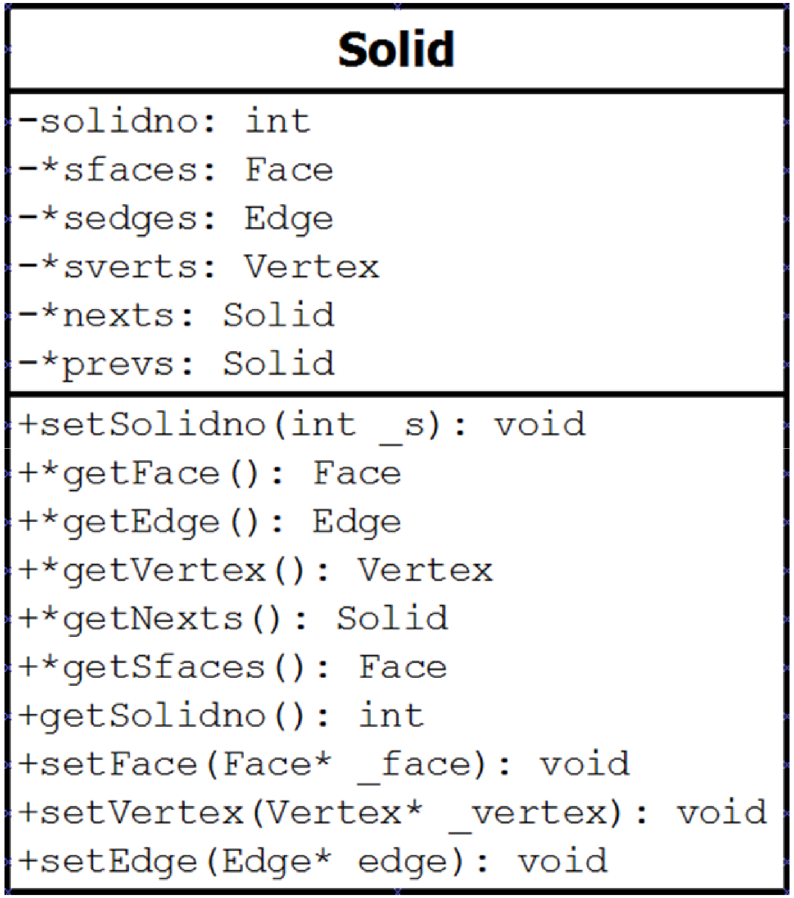
```
+mvfs(HEDModel& _model, int s, int f, int v,
       double x, double y, double z): void
+mev(HEDModel& _model, int s, int f1, int f2,
      int v1, int v2, int v3, int v4, double x,
      double y, double z): void
+mef(HEDModel& _model, int s, int f1, int v1,
      int v2, int v3, int v4, int f2): void
```

GeomChecker

```
-computerLineLineIntersection(double x1, double y1, double x2, double y2,
                              double x3, double y3, double x4, double y4): int
```

```
+hasAnyVertex(HEDModel& _model): bool
+hasVertexWithGivenCoords(HEDModel& _model,
                          int& id, double x,
                          double, y, double z,
                          double _tol): bool
+liesOnCurve(HEDModel& _model, double x,
             double y, double z): bool
+intersectsVertex(HEDModel& _model, double* curvePts,
                  int&n, double* intersPts): bool
+intersectsCurves(HEDModel& _model, double* curvePts,
                  int&n, double* intersPts): bool
```


HEDController
-*eulerop: EulerOperator
-model: HEDModel
-geomChecker: GeomChecker
+doAddPoint(double x, double y, double z,
double tol): void
+doAddCurve(double x1, double y1, double z1,
double x2, double y2, double z2,
double tol): void
+doAddTriangle(double x1, double y1, double z1,
double x2, double y2, double z2,
double x3, double y3, double z3): void
+doSelection(double x, double y, double z,
double tol): void
+unselect(): void
+doManipulate(): void
+doCreate(): void
+doEdit(): void
+doDelete(): void
+doSave(): void
+doManage(): void
+doExit(): void
+getModel(): HEDModel


Solid
-solidno: int
-*sfaces: Face
-*sedges: Edge
-*sverts: Vertex
-*nexts: Solid
-*prevs: Solid
+setSolidno(int _s): void
+*getFace(): Face
+*getEdge(): Edge
+*getVertex(): Vertex
+*getNexts(): Solid
+*getSfaces(): Face
+getSolidno(): int
+setFace(Face* _face): void
+setVertex(Vertex* _vertex): void
+setEdge(Edge* edge): void

HEDModel

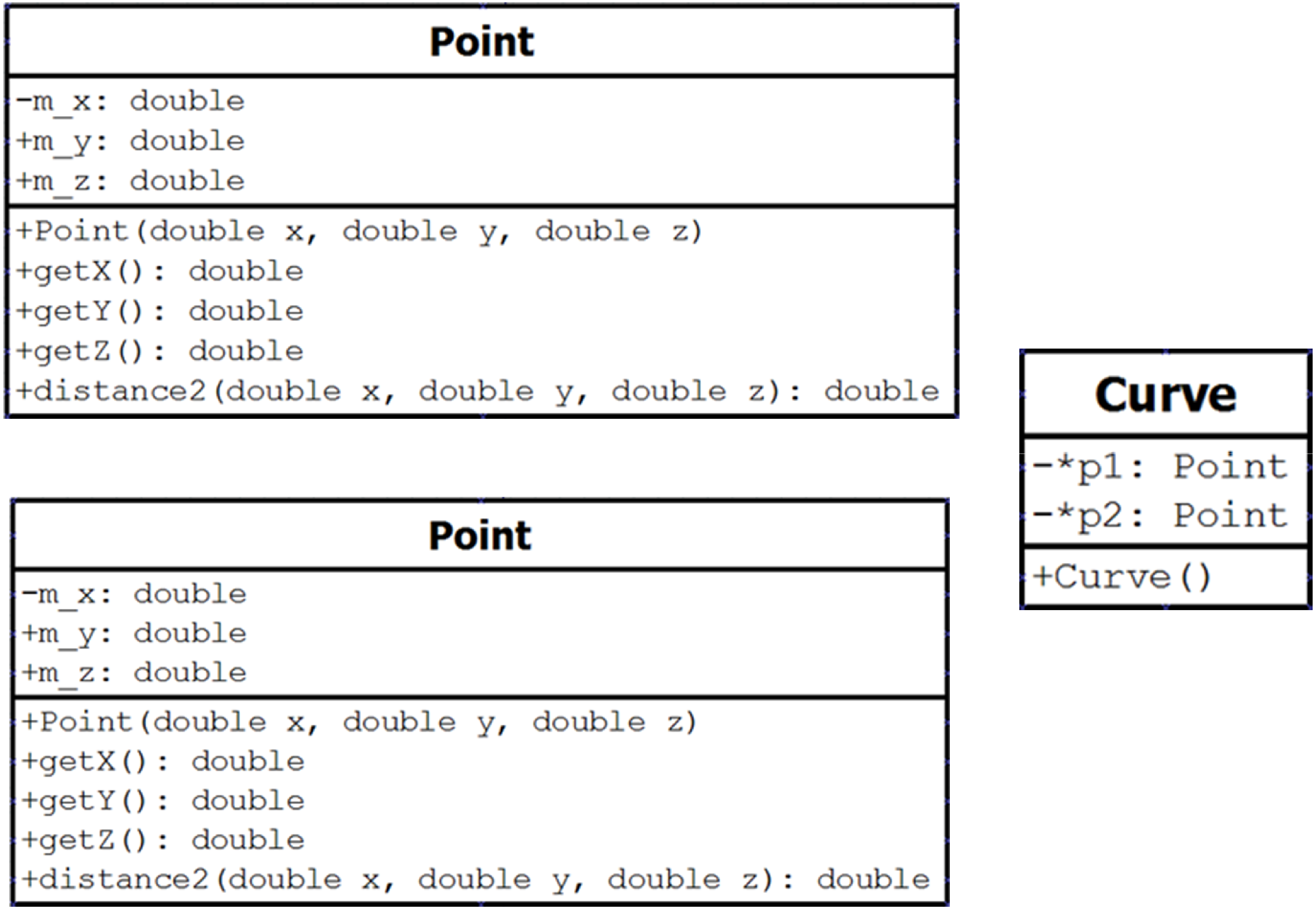
```
-*firsts: Solid
```

```
+*getFirsts(): Solid
+setFirsts(Solid* _firsts): void
+*getsolid(int sn): Solid
+*fface(Solid* s, int fn): Face
+*fhe(Face* f, int vn1, int vn2): HalfEdge
+*getVertex(double _x, double _y, double _z,
            double _tol): Vertex
+*getEdge(double _x, double _y, double _z,
          double _tol): Edge
+*getFace(double _x, double _y, double _z,
          double _tol): Face
+getNumberOfVertexes(): int
+getNumberOfFaces(): int
+getEdgeVertexes(): int
+replaceVertex(): void
+getCurve(): void
```

Loop

```
-*ledg: HalfEdge
-*lface: Face
-*nextl: Loop
-*prevl: Loop
```

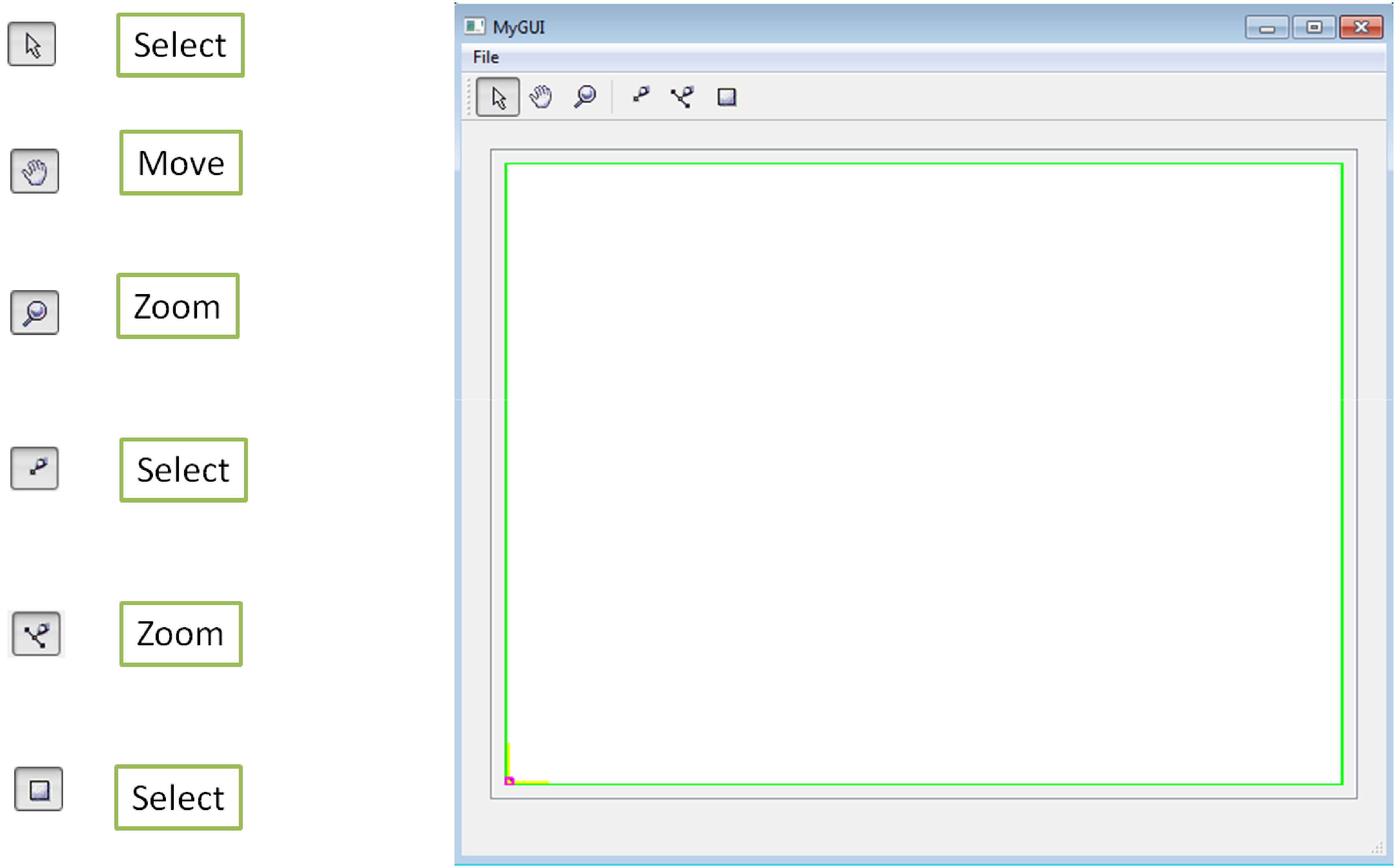
```
+setLedg(HalfEdge* _newhe): void
+*getPrevLoop(): Loop
+*getFace(): Face
+*getNextl(): Loop
+*getLedg(): HalfEdge
```



Point
-m_x: double
+m_y: double
+m_z: double
+Point(double x, double y, double z)
+getX(): double
+getY(): double
+getZ(): double
+distance2(double x, double y, double z): double
Curve
-*p1: Point
-*p2: Point
+Curve()
Point
-m_x: double
+m_y: double
+m_z: double
+Point(double x, double y, double z)
+getX(): double
+getY(): double
+getZ(): double
+distance2(double x, double y, double z): double

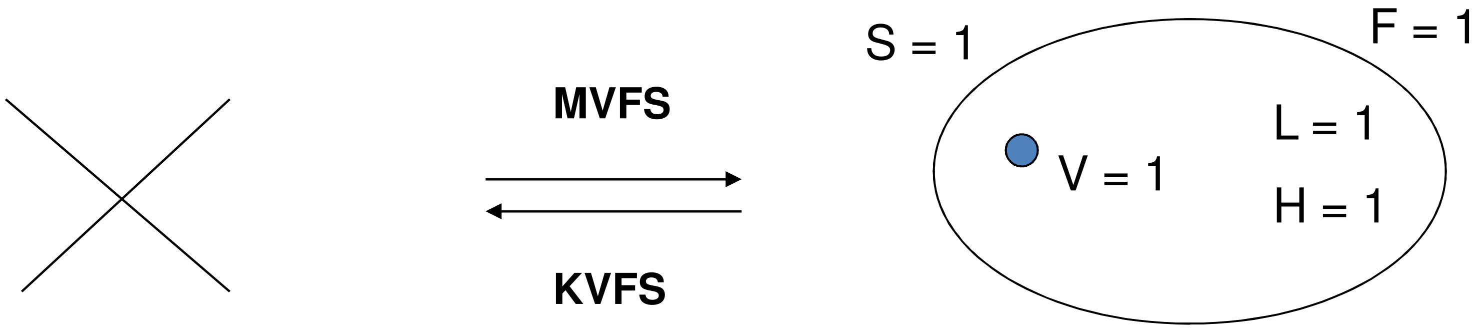
- **Operador de Euler - LMEV**

```
void EulerOperator::lmev(HalfEdge* he1, HalfEdge* he2, int v, double x, y, z)
{
	Loop* loop = he1->getLoop();
	Face* face = loop->getFace();
	Solid* solid = face->getSolid();
	Edge* newedge = new Edge(solid);
	Point* point = new Point(x,y,z);
	Vertex* newvertex = new Vertex(point,solid);
	newvertex->setVertexno(v);
	HalfEdge* he = he1;
	while(he != he2)
	{
		he->setVtx(newvertex);
		he = he->mate()->getNxt();
	}
	newedge->addhe(he2->getVtx(), he1, MINUS);
	newedge->addhe(newvertex, he2, PLUS);
	newvertex->setVedge(he2->getPrv());
	he2->getVtx()->setVedge(he2);
}
```


Select
Move
Zoom
Select
Zoom
Select
MyGUI
File

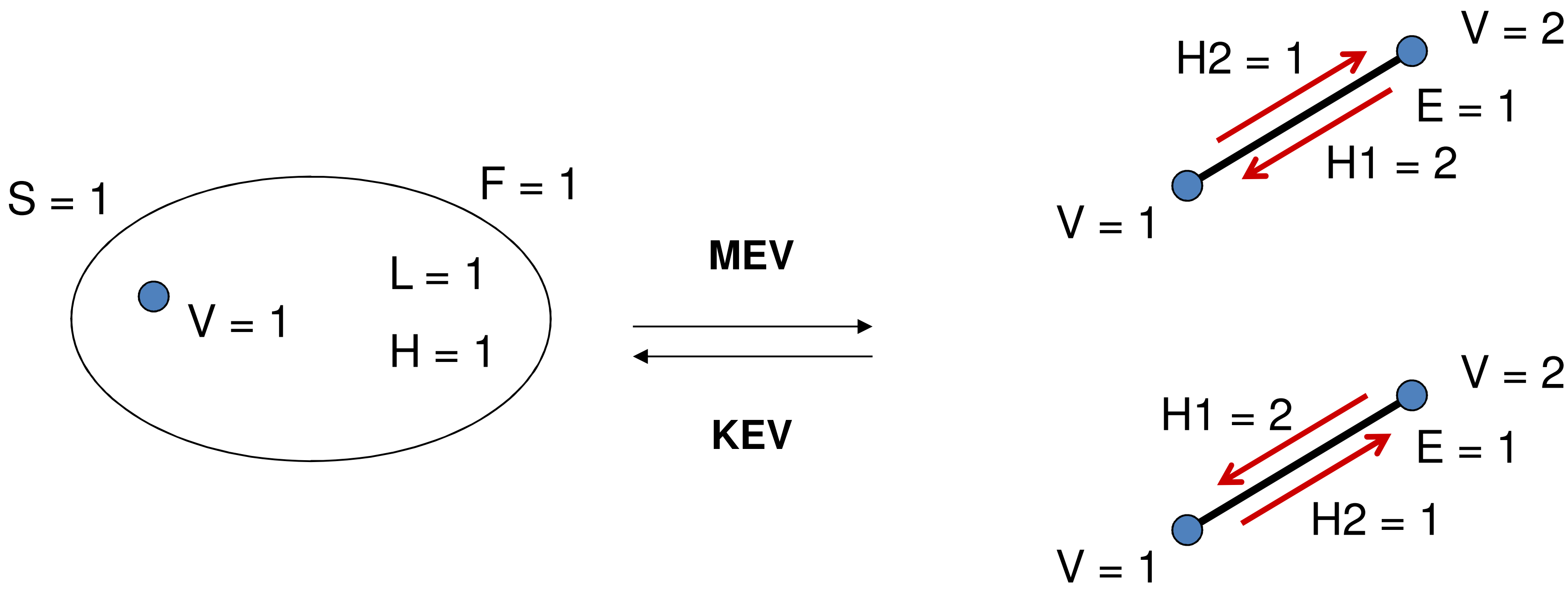
MVFS

V = 1 (H = 1) N = 0 P = 0
H = 1 (V = 1, E = 0, L = 1) N = 0 P = 0
E = 0 (H1 = 0, H2 = 0) N = 0 P = 0
L = 1 (H = 1, F = 1) N = 0 P = 0
F = 1 (S = 1, LOUT = 0 / LOOPS = 1) N = 0 P = 0
S = 1 (V = 1, F = 1, E = 0) N = 0 P = 0



MEV

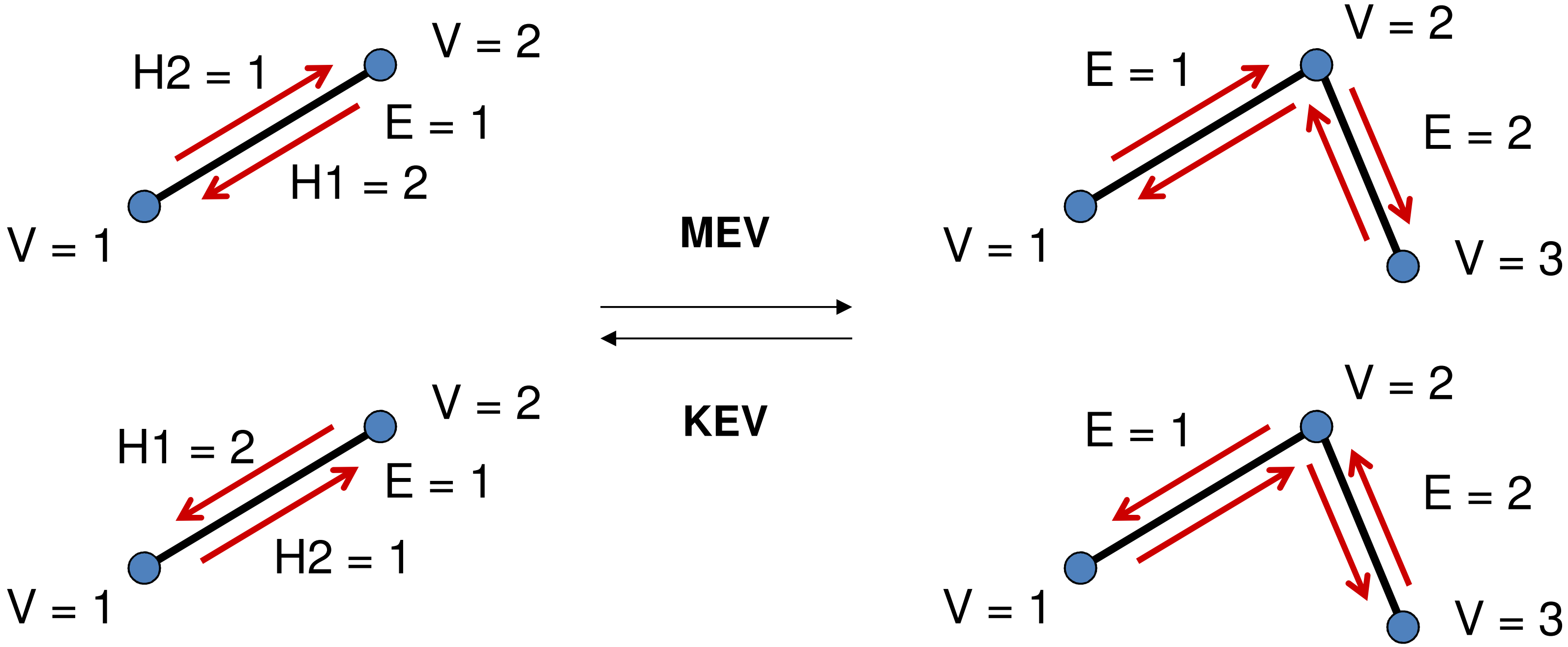
V = 1 (H = 1) N = 0 P = 2
V = 2 (H = 2) N = 1 P = 0
H = 1 (V = 1, E = 1, L = 1) N = 2 P = 2
H = 2 (V = 2, E = 1, L = 1) N = 1 P = 1
E = 1 (H1 = 2, H2 = 1) N = 0 P = 0
L = 1 (H = 2, F = 1) N = 0 P = 0
F = 1 (S = 1, LOUT = 0 / LOOPS = 1) N = 0 P = 0
S = 1 (V = 1, F = 1, E = 1) N = 0 P = 0



MEV

For a single strip there is no definition of the sequence (ccw nor ucw)

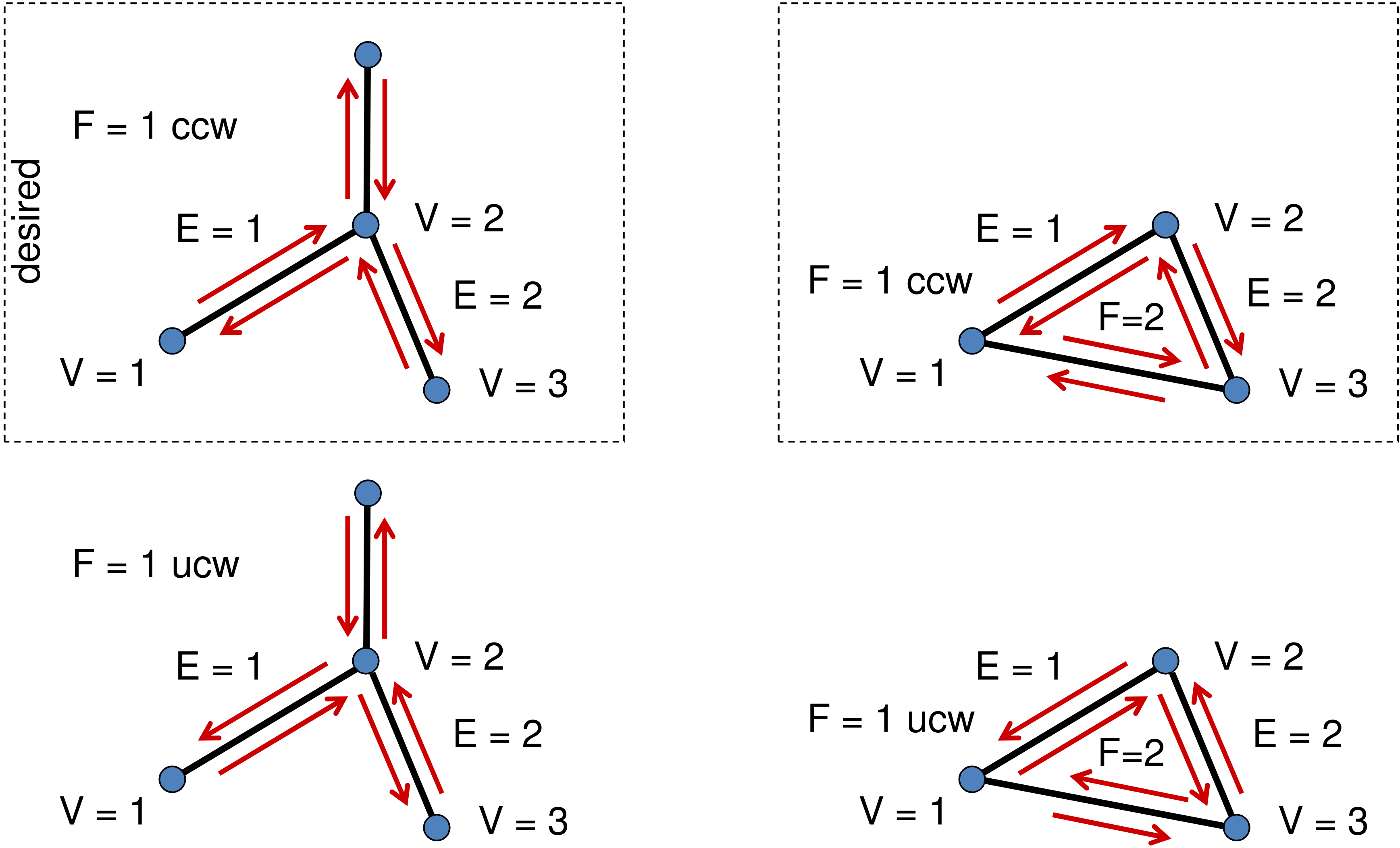
V = 1 (H = 1) N = 0 P = 2
V = 2 (H = 3) N = 1 P = 3
V = 3 (H = 4) N = 2 P = 0
H = 1 (V = 1, E = 1, L = 1) N = 3 P = 2
H = 2 (V = 2, E = 1, L = 1) N = 1 P = 4
H = 3 (V = 2, E = 2, L = 1) N = 4 P = 1
H = 4 (V = 3, E = 2, L = 1) N = 2 P = 3
E = 1 (H1 = 2, H2 = 1) N = 0 P = 2
E = 2 (H1 = 3, H2 = 4) N = 1 P = 0
L = 1 (H = 2, F = 1) N = 0 P = 0
F = 1 (S = 1, LOUT = 0 / LOOPS = 1) N = 0 P = 0
S = 1 (V = 1, F = 1, E = 1) N = 0 P = 0



Defines the sequence if occurs two situations:

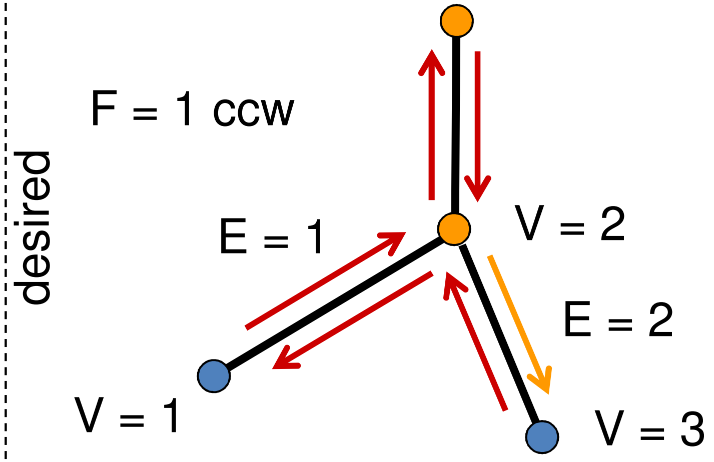
a MEV between two edges

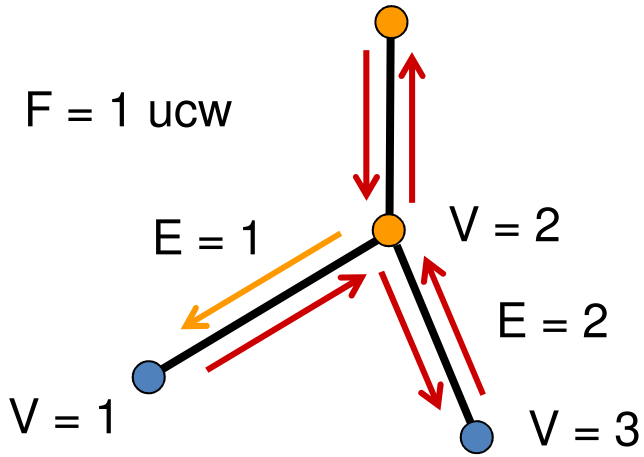
the first MEF



Which are the parameter to define each situation?

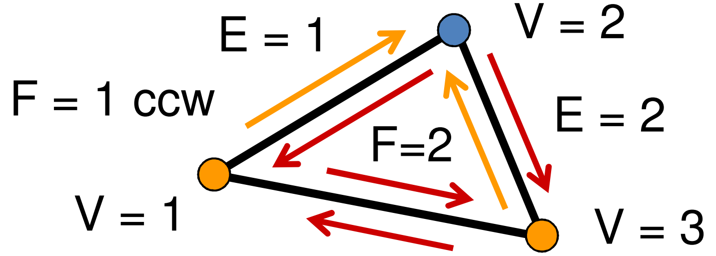
MEV

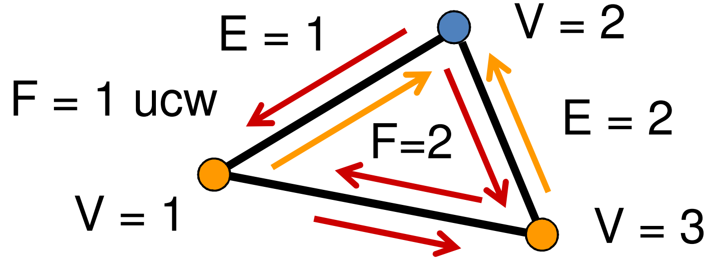




MEF

In this case, the half-edge of edge 2 (if it is the first parameter of MEF) receives the new face/loop. It is decided if the new loop area is positive! TIP: Always keep the first face with negative area (as the outside face).



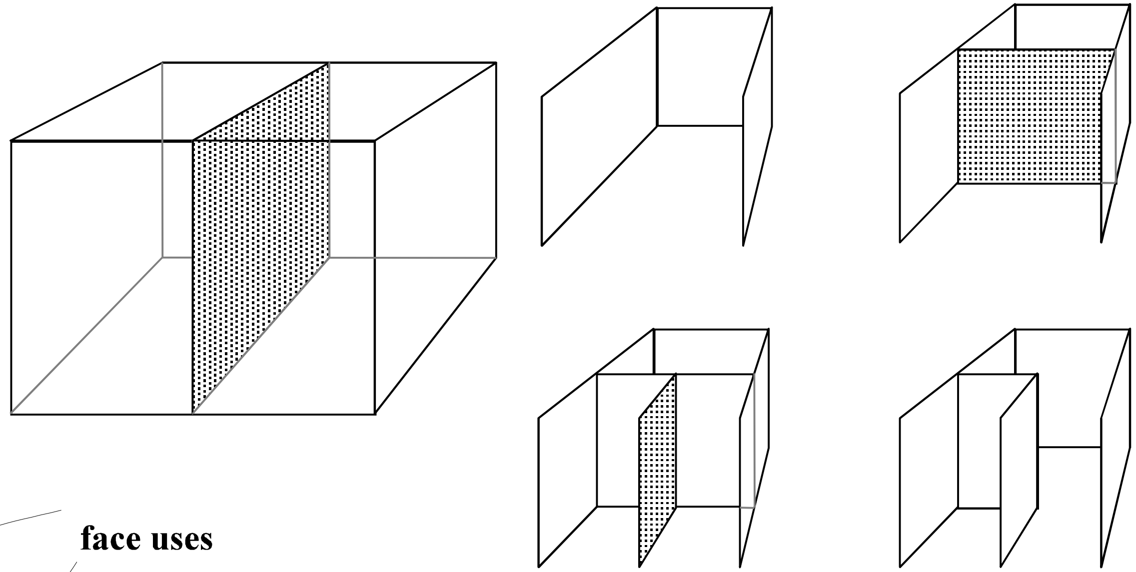


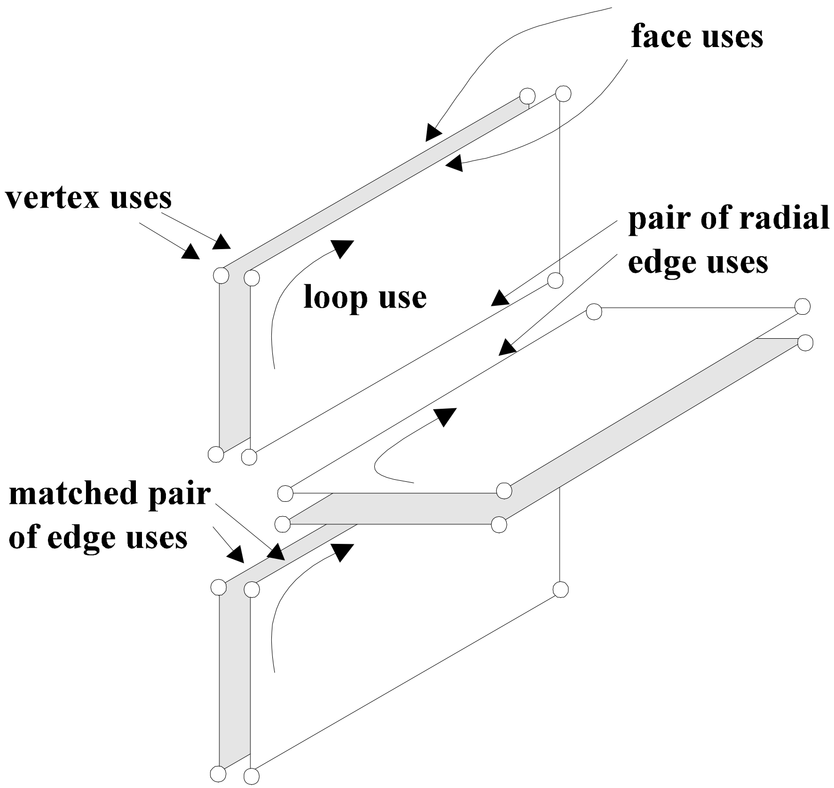
Modelagem Geométrica
Non-manifold

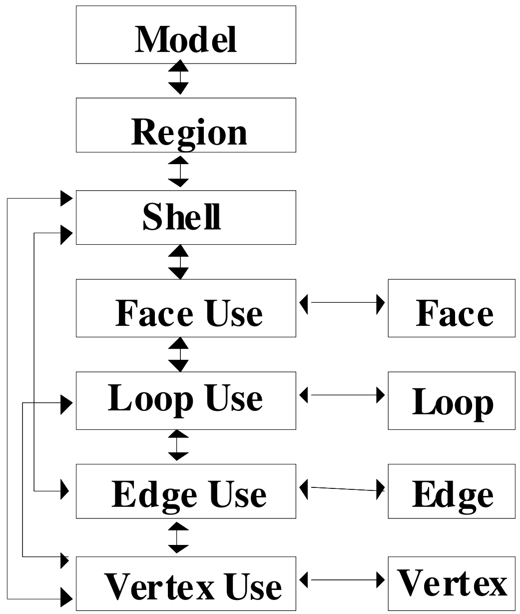
Modelagem Geométrica

- **Topologia em representações *non-manifold***
  - Áreas de aplicação da modelagem geométrica que usufruem das vantagens adicionais da representação *non-manifold*
    - **Modelagem** – transição entre modelos, detecção de regiões, armazenamento de informações geométricas arbitrárias
    - **Análise** – implementação de ferramentas de criação e análise simultâneas do modelo
    - **Representação de objetos heterogêneos** – regiões com volumes comuns, faces coincidentes, estruturas internas, sólidos constituídos de materiais diferentes

Radial-Edge
(Weiler 1986)







Modelagem Paramétrica

Modelagem Paramétrica

MCAD (*Mechanical Computer Aided Design*)

Tecnologia relativamente nova.

Modelagem Paramétrica

MCAD (*Mechanical Computer Aided Design*)

Tecnologia relativamente nova.

Seu desenvolvimento vem ocorrendo desde +40 anos em paralelo ao desenvolvimento da tecnologia de hardware.

Foi primeiramente apresentada no final de 1980, e recentemente se tornou o novo paradigma da projetos CAD mecânicos.

Modelagem Paramétrica

MCAD (*Mechanical Computer Aided Design*)

Tecnologia relativamente nova.

Seu desenvolvimento vem ocorrendo desde +40 anos
em paralelo ao desenvolvimento da tecnologia de hardware.

Foi primeiramente apresentada no final de 1980, e recentemente
se tornou o novo paradigma da projetos CAD mecânicos.

Tem elevado as tecnologias de CAD ao nível de ser
uma ferramenta de projetos muito poderosa.

Ela automatiza o projeto e os procedimentos de revisão
pelo uso de *parametric features.*

Modelagem Paramétrica

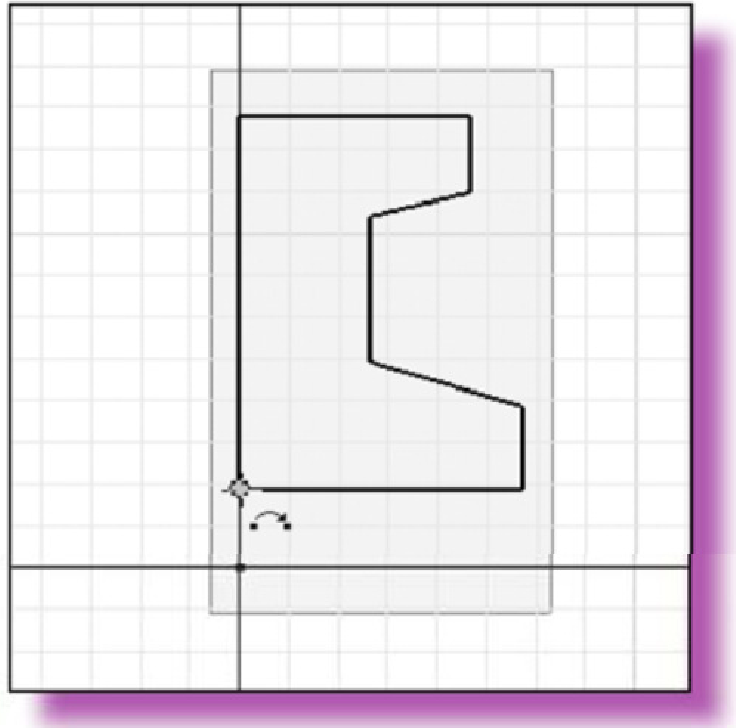
A palavra *paramétrico* significa que as definições da geometria do projeto, tal como dimensões, podem ser mudadas em qualquer momento no processo de projeto

Modelagem paramétrica recebe esse nome por causa do projeto de parâmetros ou variáveis que são modificados durante o processo de simulação do projeto.

Vocabulário e Formalização:
- *Features*
- *Part* (Parte)
- *Constrains* (Restrições)
- *Assembly* (Montagem)
- *Sketch* (Esboço)

Modelagem Paramétrica

Sketcher

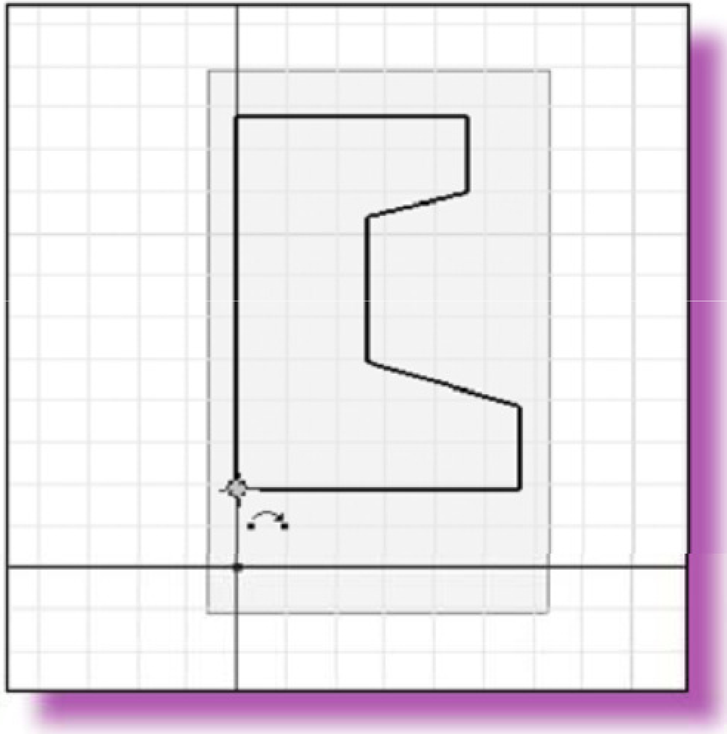


Modelagem Paramétrica

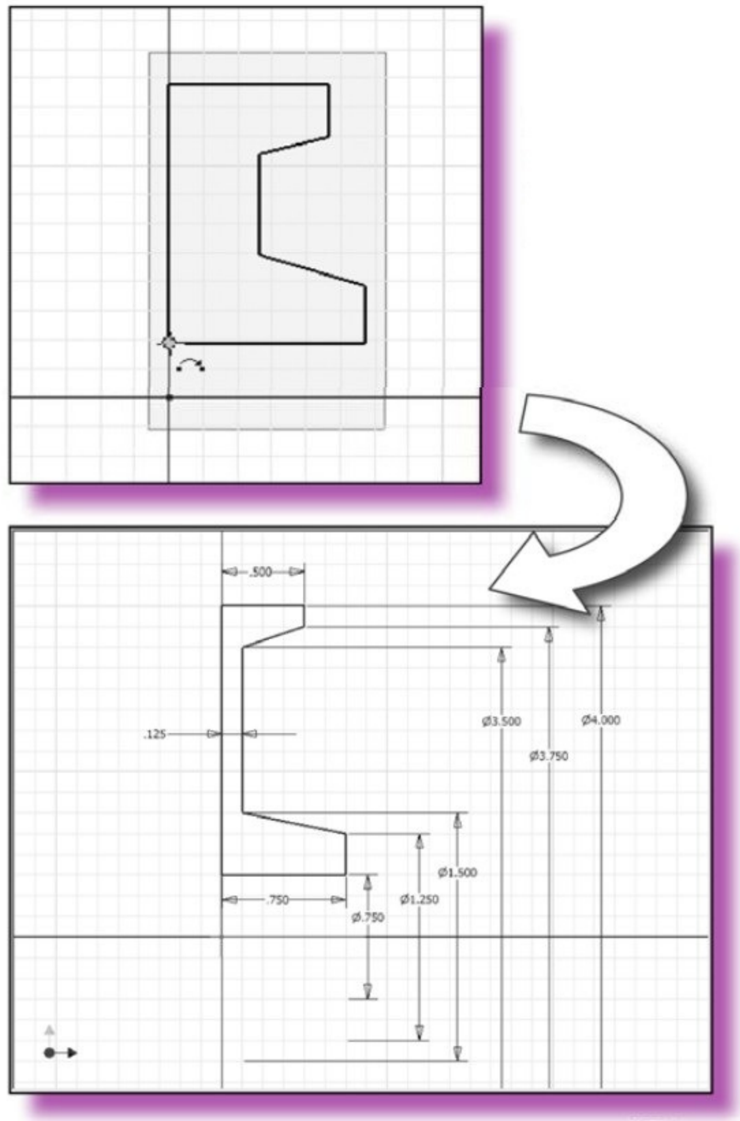
Ferramentas Geométricas

- **Point**: Draws a point
- **Arc**: Draws an arc segment from center, radius, start angle and end angle
- **Circle**: Draws a circle from center and radius
- **2-point Line**: Draws a line segment from 2 points
- **Polyline (multiple-point line)**: Draws a line made of multiple line segments
- **Rectangle**: Draws a rectangle from 2 opposite points
- **Fillet**: Makes a fillet between two lines joined at one point. Select both lines or click on the corner point, then activate the tool.
- **Trimming**: Trims a line, circle or arc with respect to the clicked point.
- **External Geometry**: Creates an edge linked to external geometry.
- **Construction Mode**: Toggles an element to/from construction mode. A construction object will not be used in a 3D geometry operation.

Sketcher



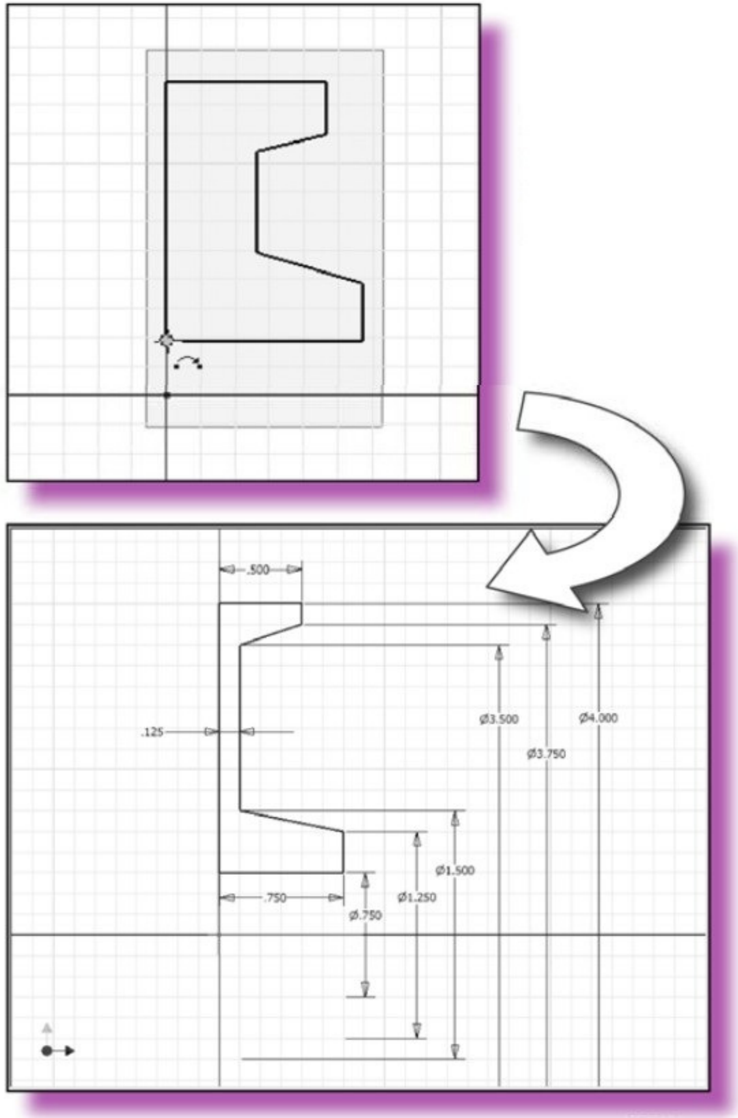
Modelagem Paramétrica



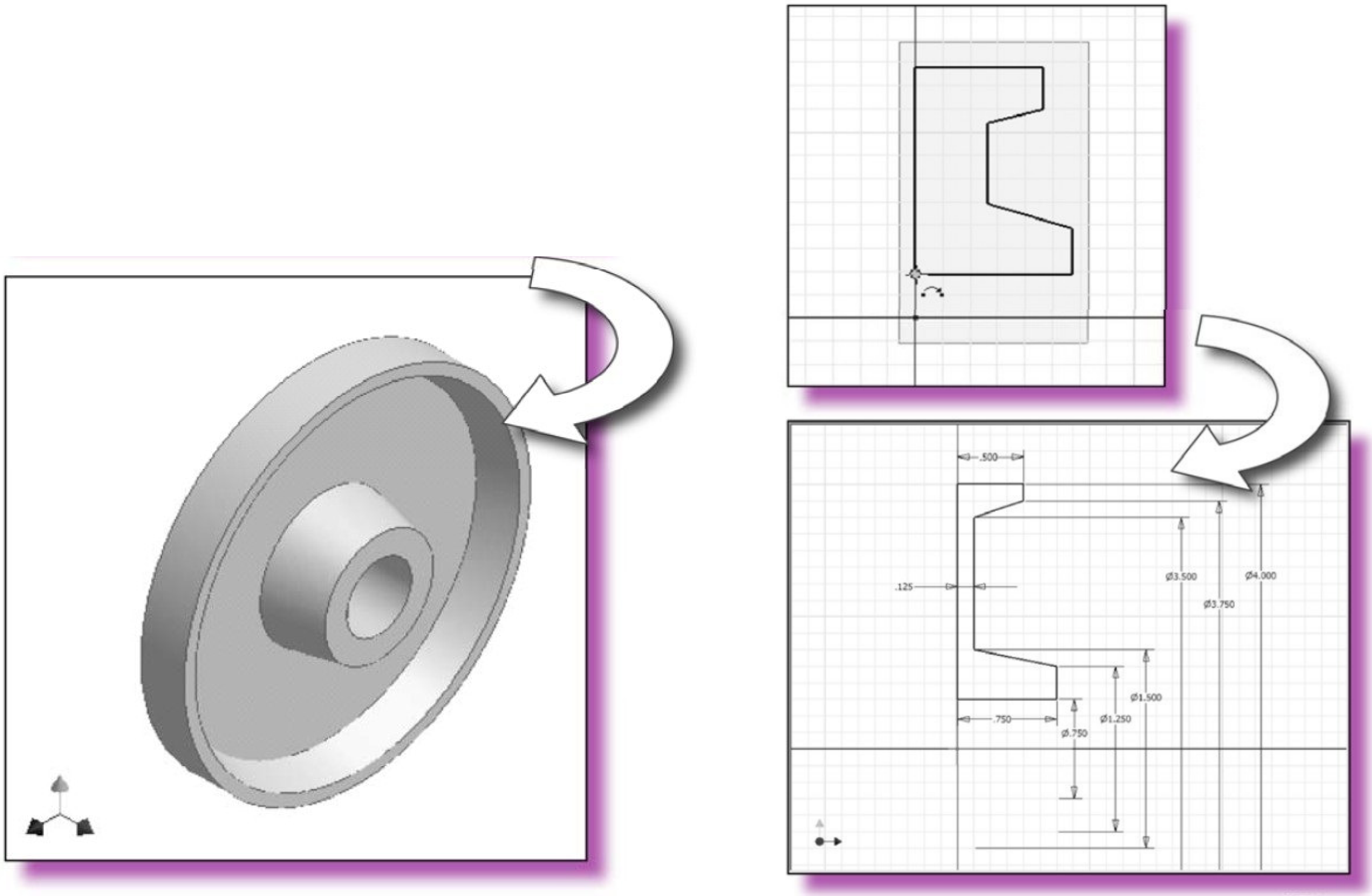
Modelagem Paramétrica

Aplicação das Restrições

- **Lock**: Creates a lock constraint on the selected item by setting vertical and horizontal dimensions relative to the origin (dimensions can be edited afterwards).
- **Coincident**: Creates a coincident (point-on-point) constraint between two selected points.
- **Point On Object**: Creates a point-on-object constraint on selected items.
- **Horizontal Distance**: Fixes the horizontal distance between 2 points or line ends. If only one item is selected, the distance is set to the origin.
- **Vertical Distance**: Fixes the vertical distance between 2 points or line ends. If only one item is selected, the distance is set to the origin.
- **Vertical**: Creates a vertical constraint to the selected lines or polylines elements. More than one object can be selected.
- **Horizontal**: Creates a horizontal constraint to the selected lines or polylines elements. More than one object can be selected.
- **Length**: Creates a length constraint on a selected line.
- **Radius**: Creates a radius constraint on a selected arc or circle.
- **Parallel**: Creates a parallel constraint between two selected lines.
- **Perpendicular**: Creates a perpendicular constraint between two selected lines.
- **InternalAngle**: Creates an internal angle constraint between two selected lines.
- **Tangent**: Creates a tangent constraint between two selected entities, or a colinear constraint between two line segments.
- **Equal Length**: Creates an equality constraint between two selected entities. If used on circle or arcs, the radius will be set equal.
- **Symmetric**: Creates a symmetric constraint between 2 points with respect to a line.

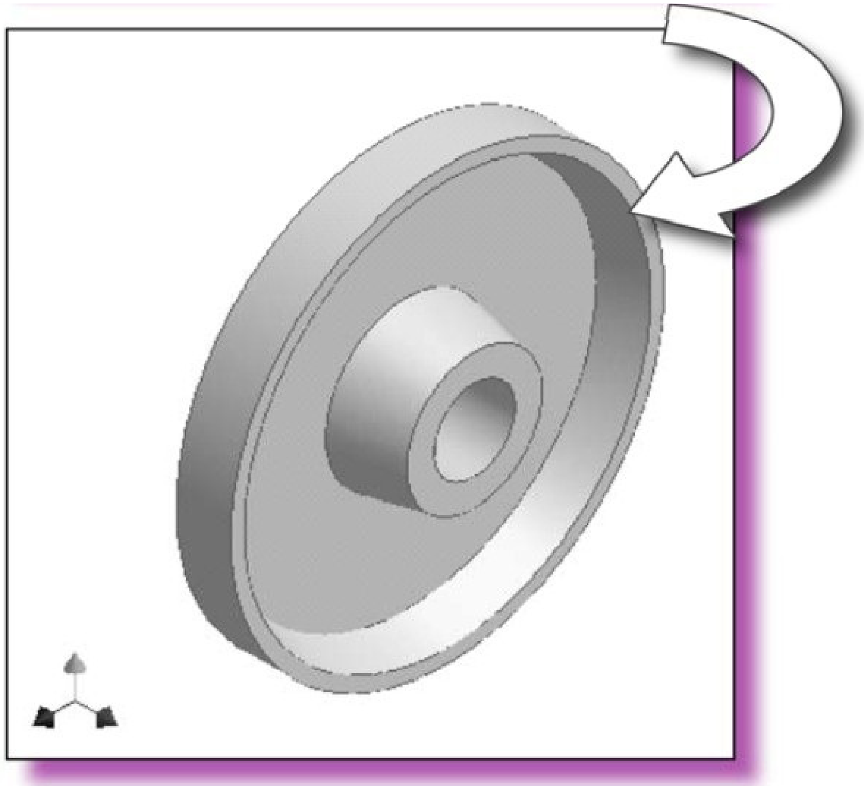


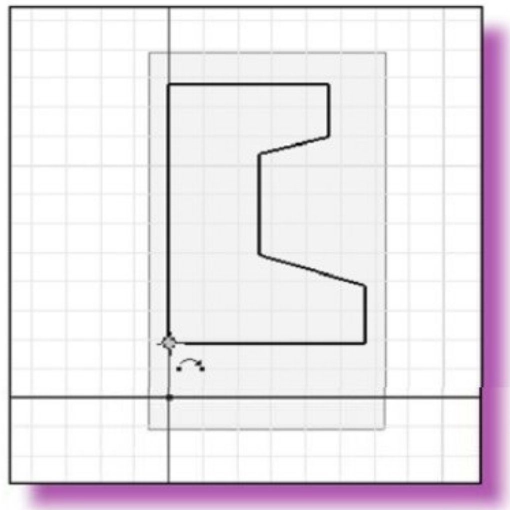
Modelagem Paramétrica

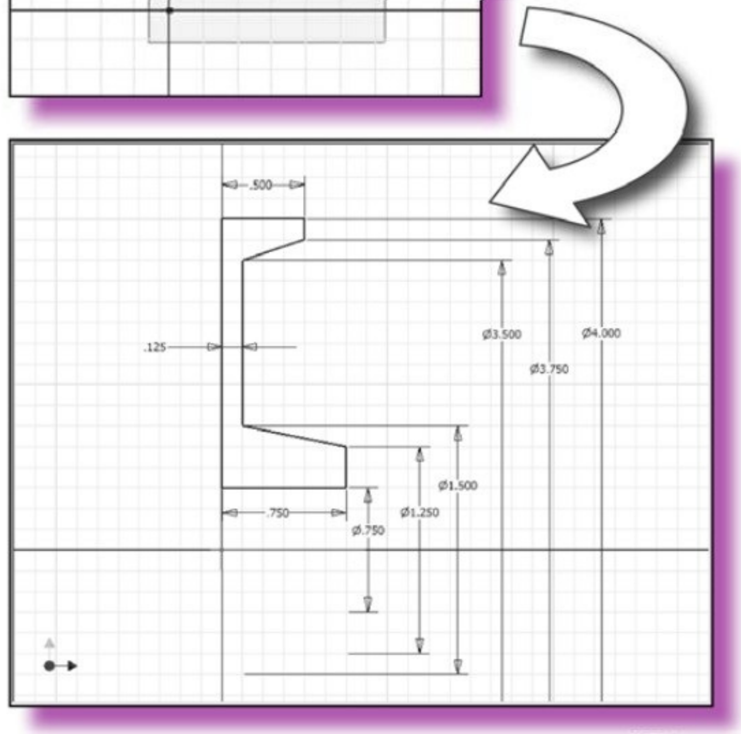


Modelagem Paramétrica

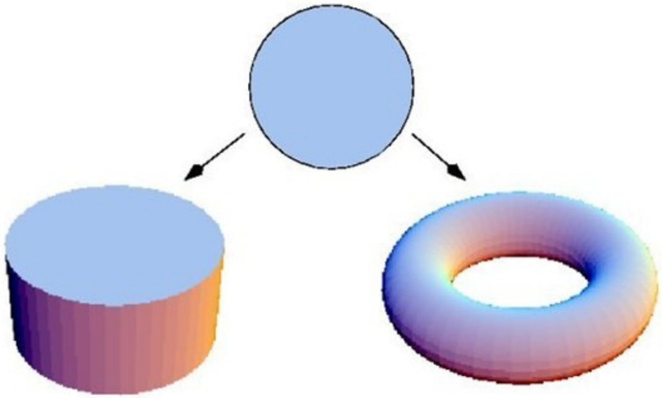
Features:
- ***Extrude*** (Extrusão)
- ***Revolute*** (Revolução)
- ***Sweep*** (Varredura)
- ***Loft***

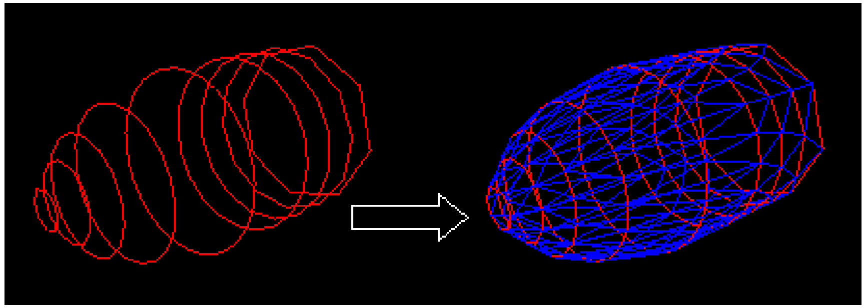


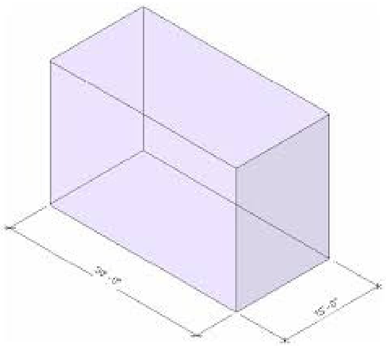


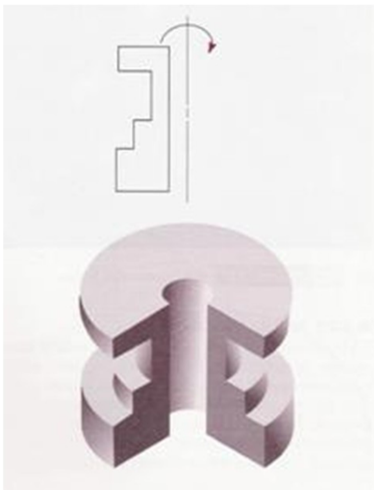


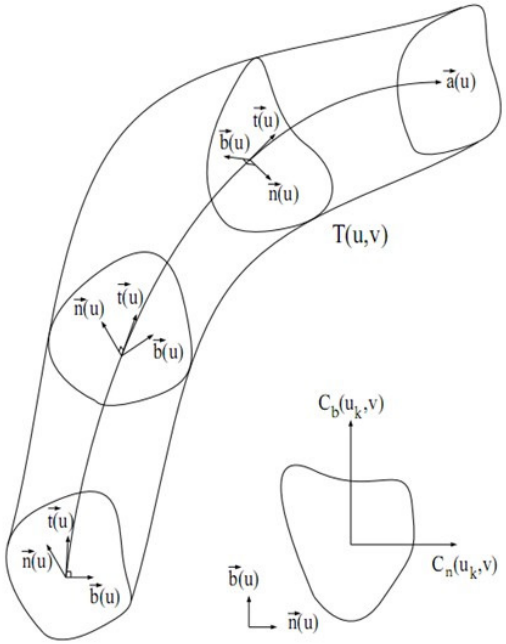
Modelagem Paramétrica



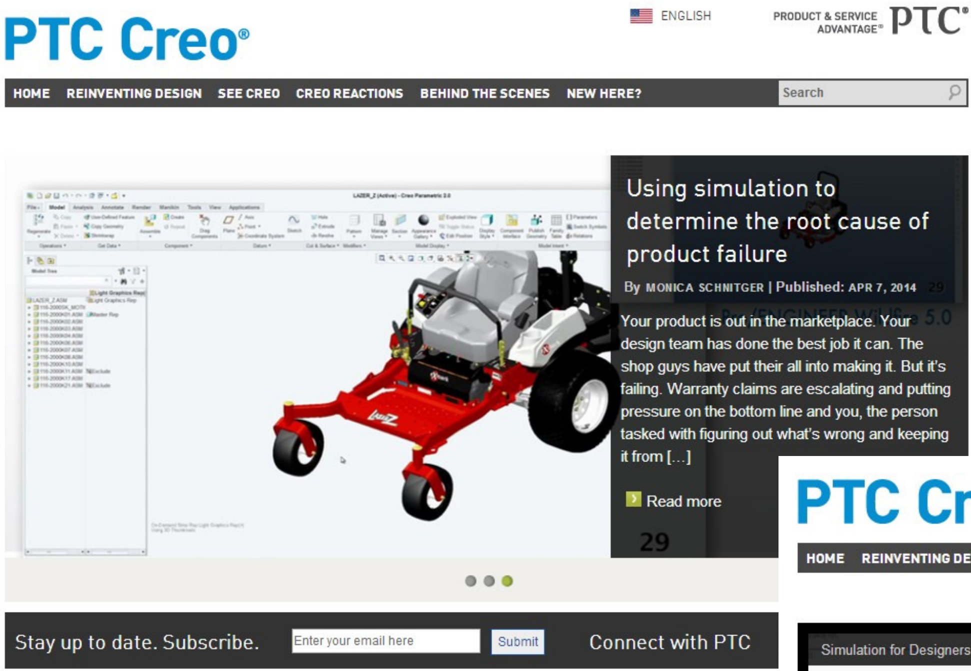








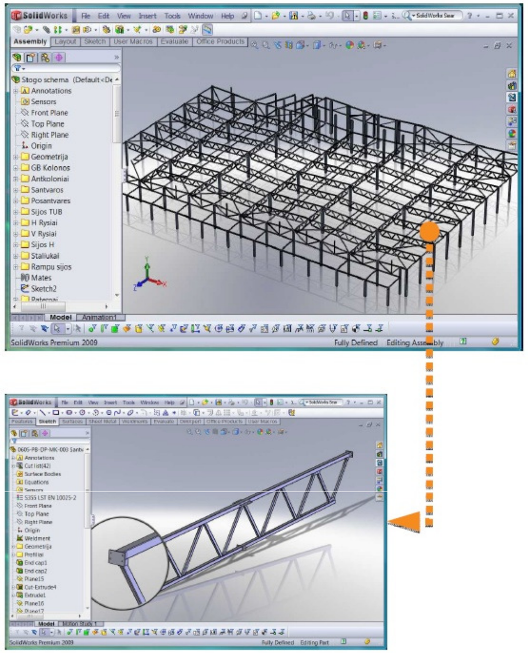
Modelagem Paramétrica



Modelagem Paramétrica

[POPOV2009]

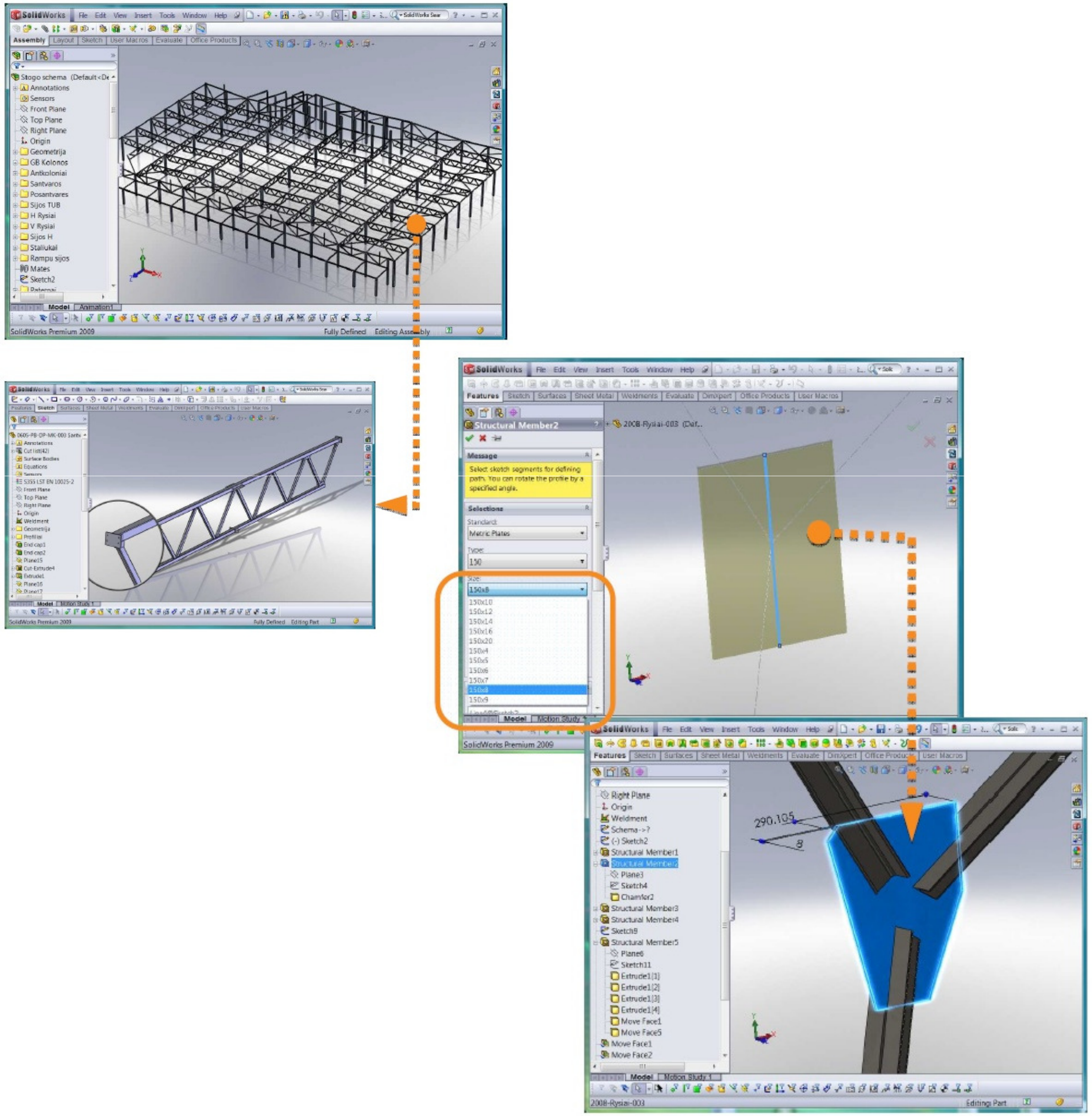
Modelagem Paramétrica

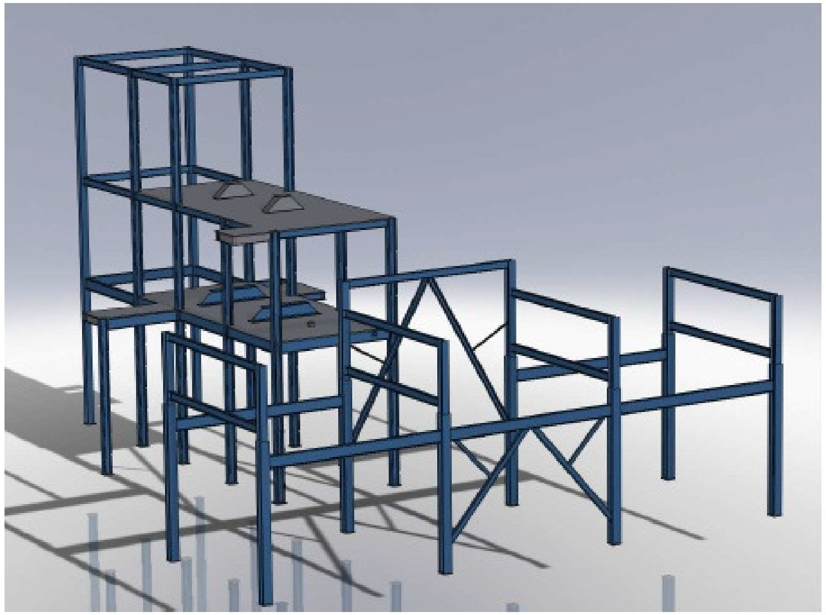


[POPOV2009]

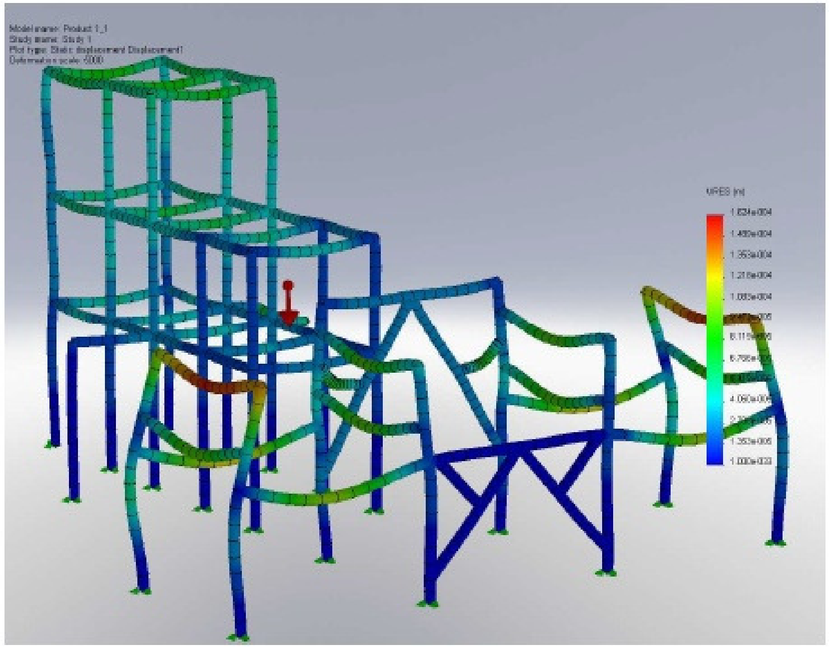
Modelagem Paramétrica

[POPOV2009]

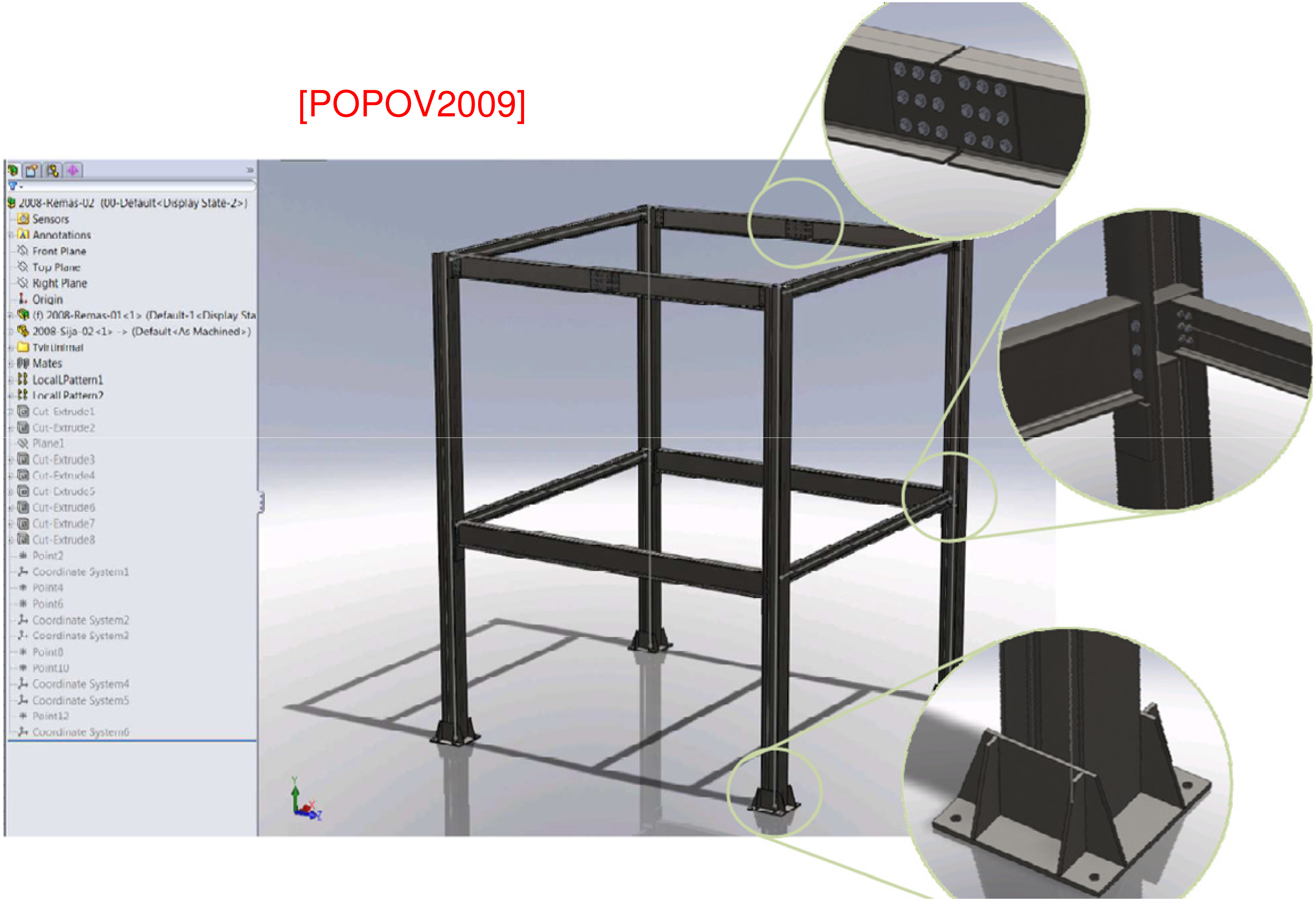




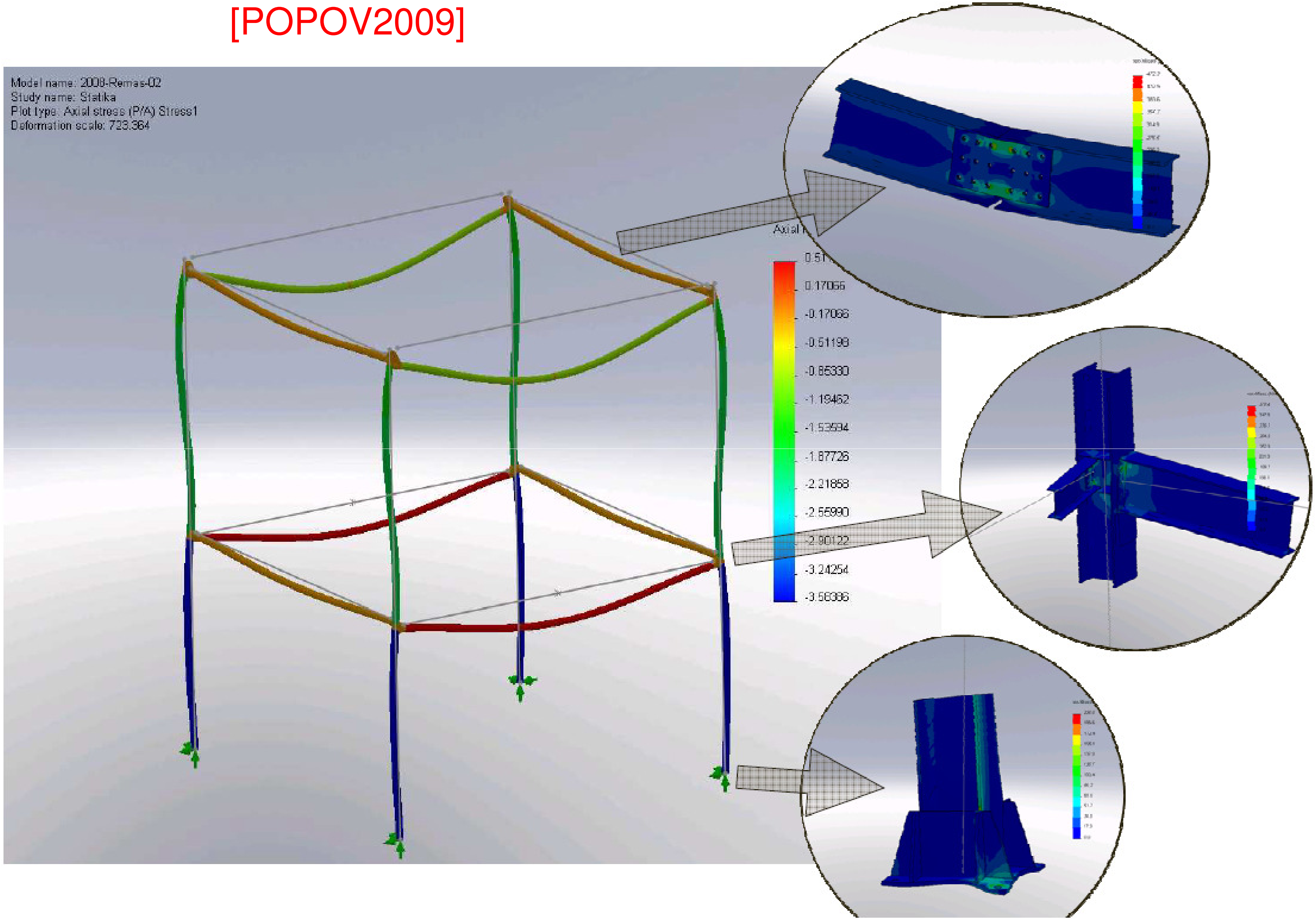
[POPOV2009]


Model name: Product 1_1
Study name: Study 1
Plot type: Static displacement Displacement1

[POPOV2009]

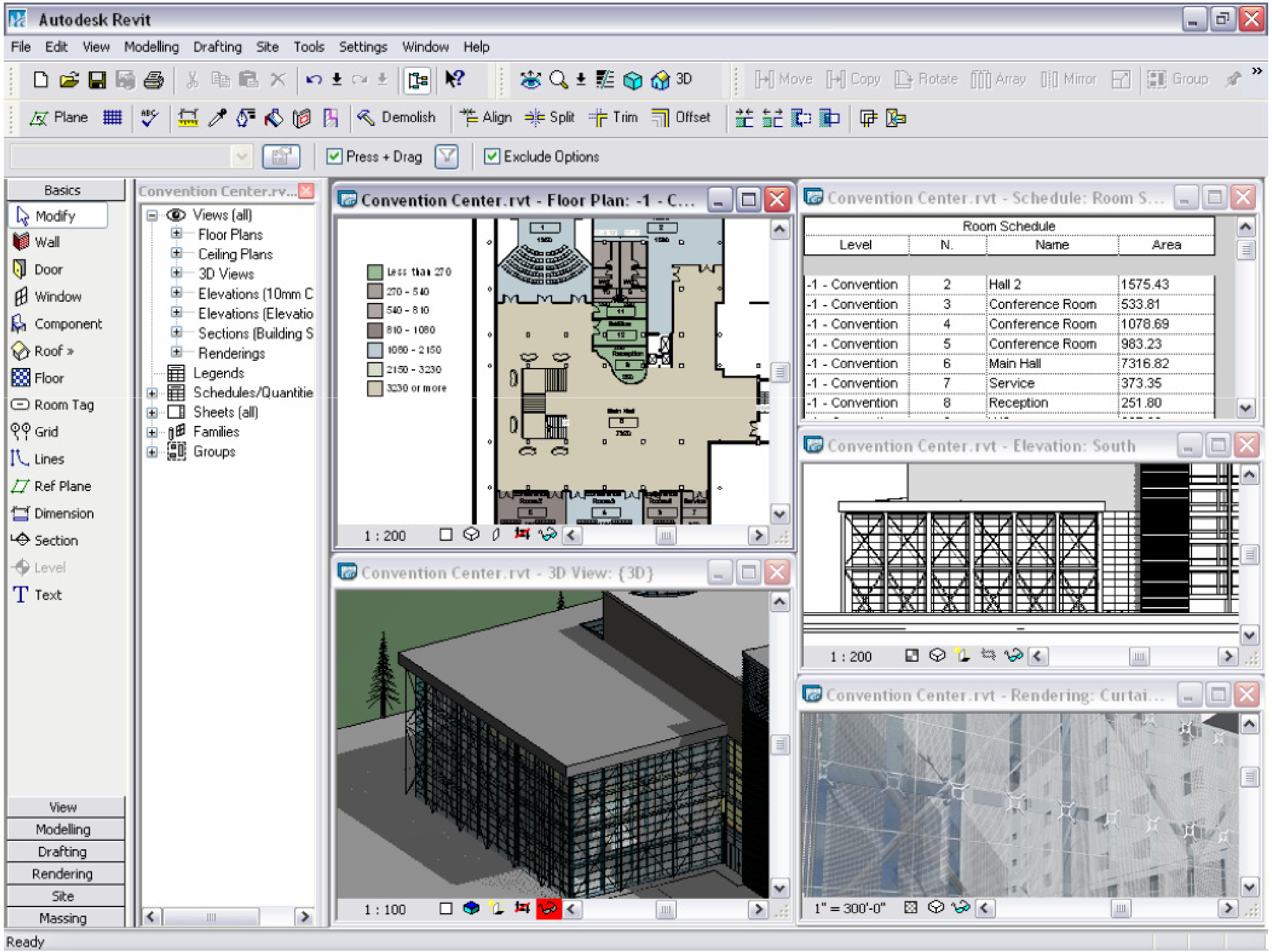

2008-Remas-02 (00-Default<Display State-2>)
Sensors
Annotations
Front Plane
Top Plane
Right Plane
Origin
(f) 2008-Remas-01<1> (Default-1<Display Sta
2008-Sija-02<1> -> (Default<As Machined>)
Mates
LocalLPattern1
LocalLPattern2
Cut-Extrude1
Cut-Extrude2
Plane1
Cut-Extrude3
Cut-Extrude4
Cut-Extrude5
Cut-Extrude6
Cut-Extrude7
Cut-Extrude8
Point2
Coordinate System1
Point4
Point6
Coordinate System2
Coordinate System2
Point8
Point10
Coordinate System4
Coordinate System5
Point12
Coordinate System6

[POPOV2009]



Model name: 2008-Remas-02
Study name: Statika
Plot type: Axial stress (P/A) Stress1
Deformation scale: 723.364
Axial
0.51
0.17066
-0.17066
-0.51198
-0.85330
-1.19462
-1.53594
-1.87726
-2.21858
-2.55990
-2.90122
-3.24254
-3.58386

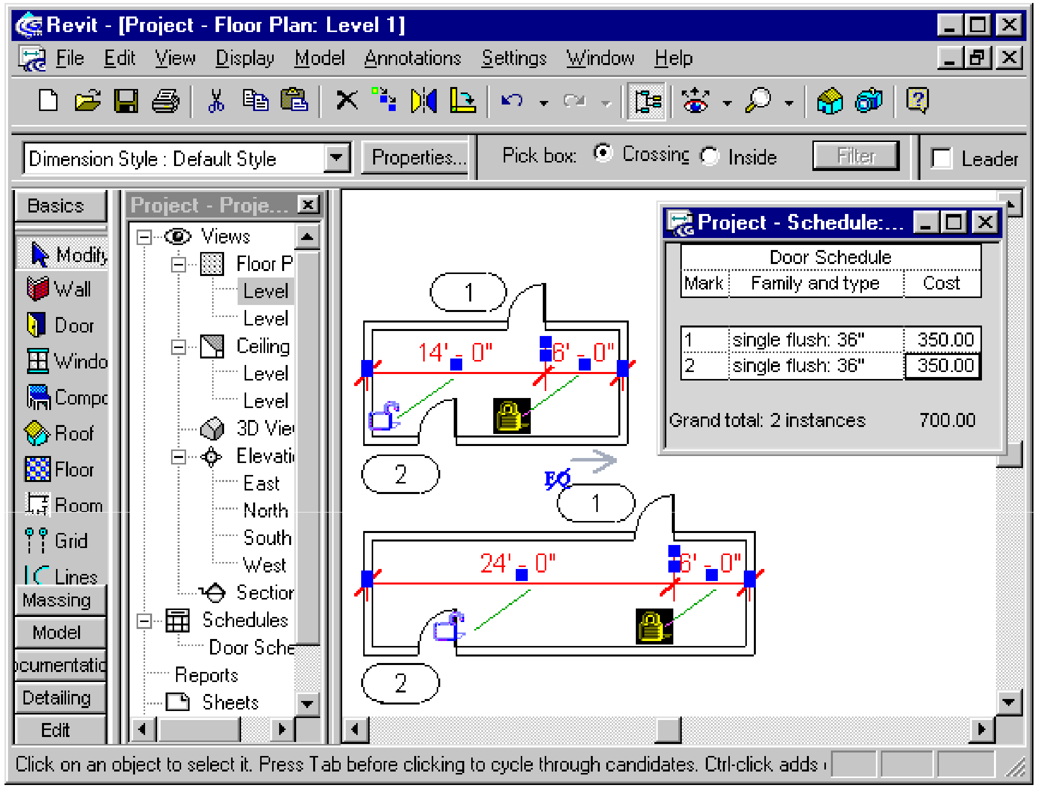
Modelagem Paramétrica



BIM

NEWS / OPINIONS / FEATURES / DEALS / HOW-TO / BUSINESS / SUBSCRIBE

ALL REVIEWS / LAPTOPS / TABLETS / PHONES / APPS / SOFTWARE / SECURITY / PRINTERS / CAMERAS / HDTVS



ENCYCLOPEDIA

SEARCH ENCYCLOPEDIA

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0-9 MISC

Definition of: parametric modeling

[PCMag2014]

The door in this room has been "locked" to four feet from the right wall. When the wall is dragged to the right to make the room larger, the door maintains its relationship with the wall. This screen shot is in Autodesk Revit, the first parametric building modeler to tie together all component views and annotations parametrically for the A/E/C industry. In addition, the program maintains automatic interaction between graphic and schedule views (note door schedule at right). If either one is changed, its counterpart is updated. (Screen shot courtesy of Autodesk, Inc., www.autodesk.com)

Referências

[PCMag2014]
PC Magazine 2014
Encyclopedia: Parametric Modeling
http://www.pcmag.com/encyclopedia/term/48839/parametric-modeling

[SHIH2006]
Randy Shih 2006
Parametric Modeling: The new CAD Paradigm for Mechanical Designs

[POPOV2009]
Vladimir Popov, Andrej Jarmolajev 2009
Integrated Design and Analysis Applications for Structural Steelwork and Plant Systems