POLITECNICO DI MILANO

Finite Element Simulation For Mechanical Design

Meshing techniques

Prof. Andrea Bernasconi

Prof. Michele Carboni

Introduction

A mesh is defined by:

- Coordinates of the nodes
- A connectivity table, there the nodes are associated to the elements
- Other useful info (list of edges, list of faces, …)

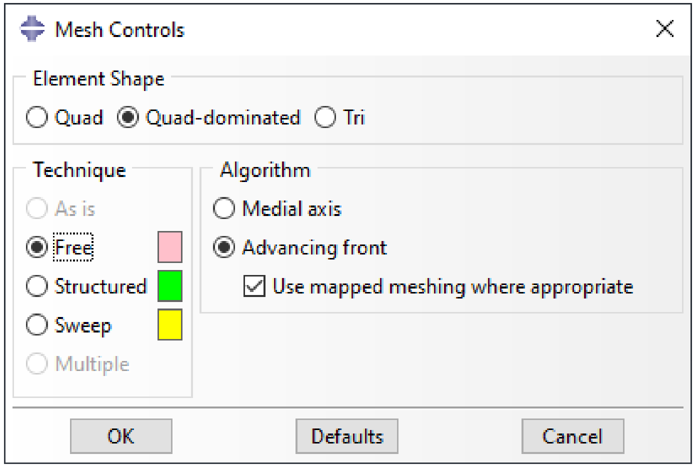
```
*Heading
** Job name: beam_pin_pin_3_elm Model name: Model-1
** Generated by: Abaqus/CAE 2018
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**
** PARTS
**
*Part, name=Part-1
*Node
      1,           0.,           0.
      2,          25.,           0.
      3,          50.,           0.
      4,          75.,           0.
      5,         100.,           0.
*Element, type=B23
1, 1, 2
2, 2, 3
3, 3, 4
4, 4, 5
```

Introduction

In general, meshes can be classified on the basis of:

- Geometry of the elements: triangle, rectangle, tetrahedra, hexahedra,…
- Type of mapping: e.g. isoparametric
- Topology: structured (or mapped), non-structured (or free), extruded, hybrid

A meshing technique consists of the application to the geometry of a particular combination of element geometry, mapping and topology.



Introduction

Once a mesh technique is chosen, different meshing algorithms are available and can be used.

It is worth noting that, given a certain mesh technique:

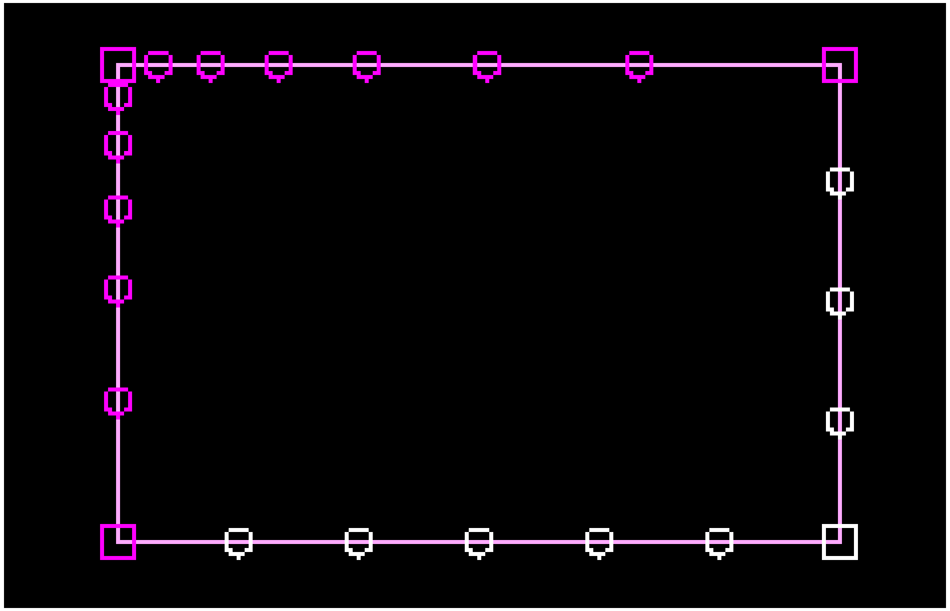
- It is not, in general, applicable to any possible case
- It might not be available, because not all the combinations of element geometry, mapping and topology are possible

Each mesh technique is characterized by a certain level of automation.

Two categories can be identified:

- Top-down (typically automatic or semi-automatic)
- Bottom-up (necessarily manual)

Seeding

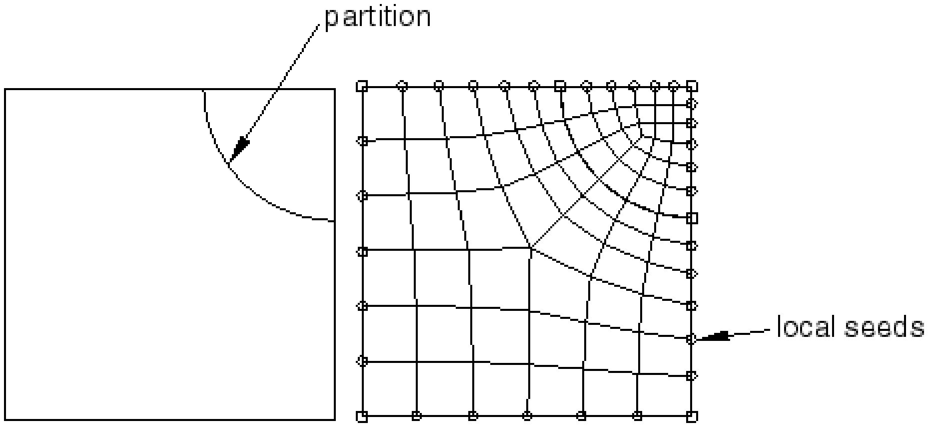


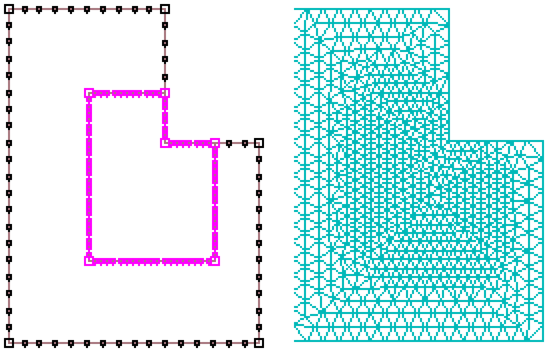
It is advisable to seed **all** edges

Seeds are **suggestions** for the position of nodes

If seeds are not compliant with the meshing techniques, they will be disregarded

Partitioning





Partitioning may improve mesh quality, by:

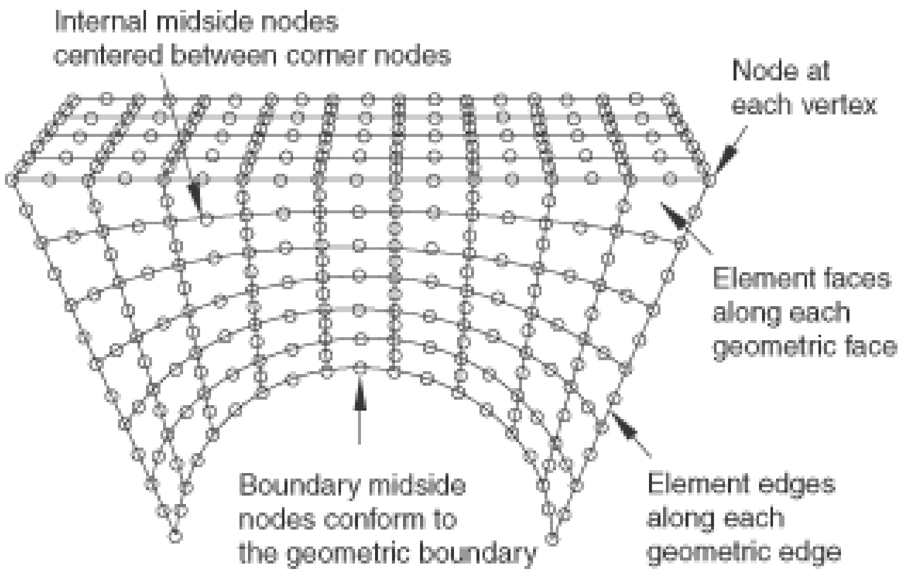
- Subdividing the geometry into portions that are easier to mesh by a regular meshing technique
- Controlling better the mesh generation process
- Defining regions where different types of element are used

Top-down technique

The mesh is generated starting from the geometry, down to each single element and node

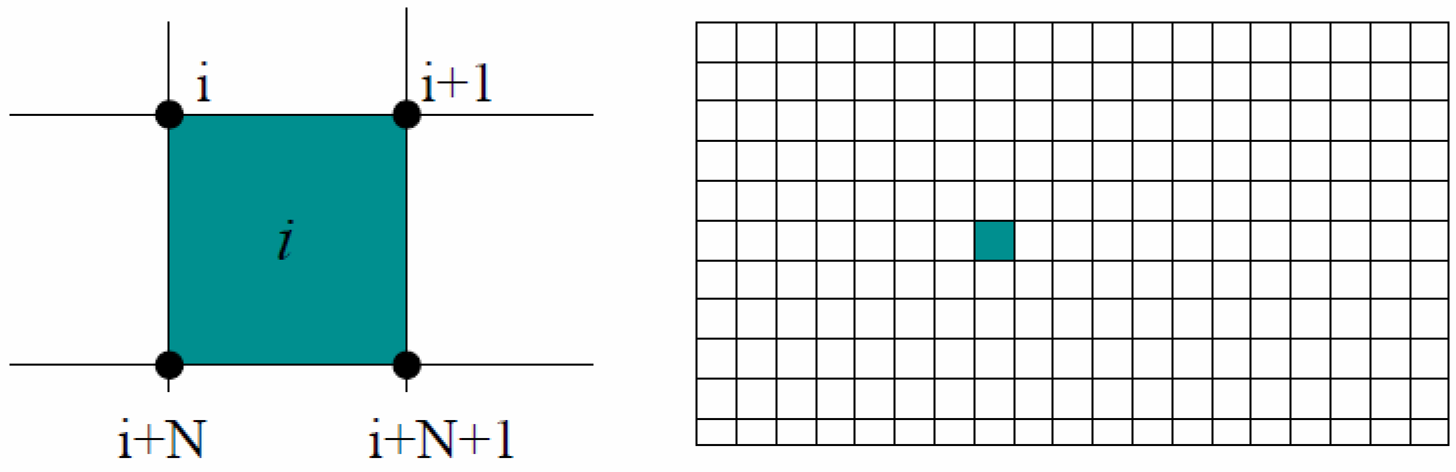
The mesh matches the original geometry, and meshing can be implemented into an automated process

However, it may lead to poor quality elements.



Top-down techniques: structured mesh

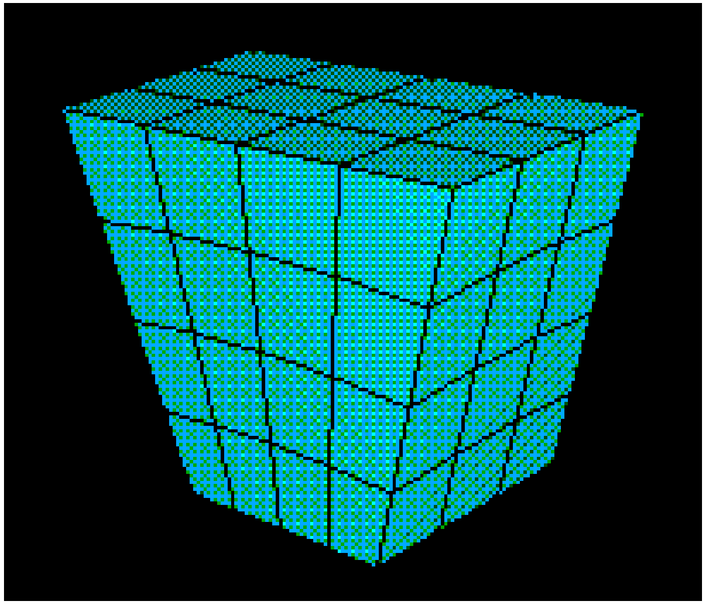
Structured = node numbering can be obtained by simple algebraic equations

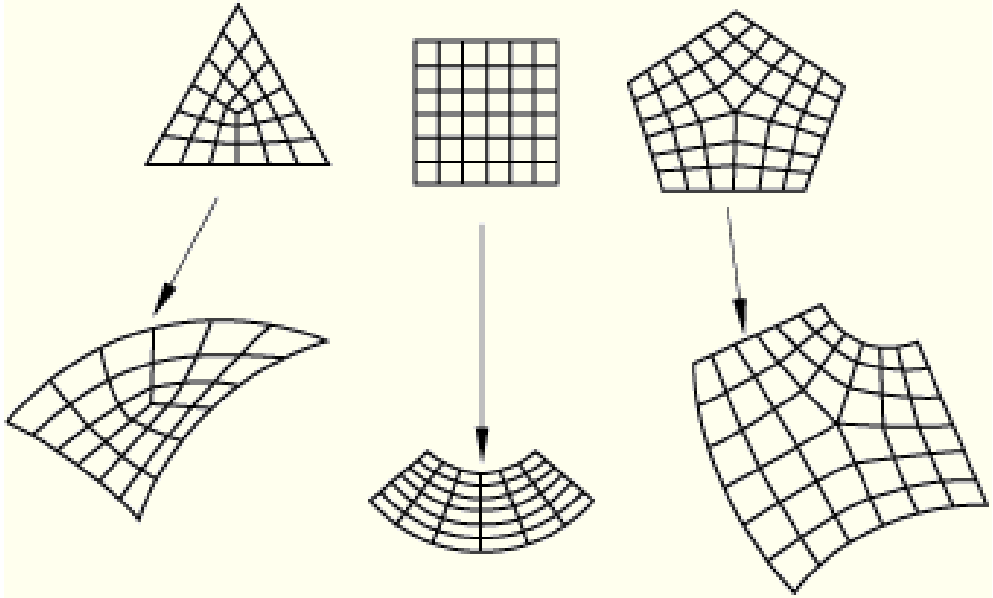


Pros: efficient algorithms

Cons: poor quality for complex geometries; not good for local mesh refinement

Top-down techniques: structured mesh





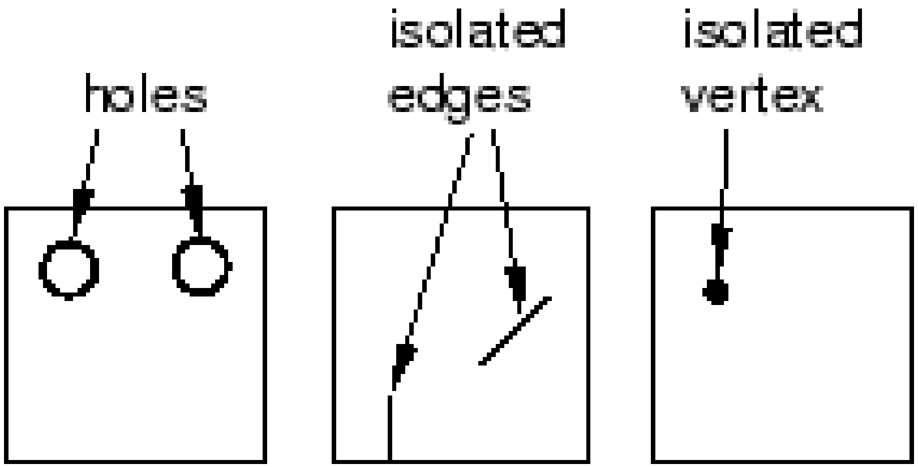
Structured meshes are based on predefined topologies, that are adapted to the actual geometry

By proper partitioning, i.e. by creating regions that match the predefined topologies, the user can take control of the meshing process.

Structured mesh of 2D regions

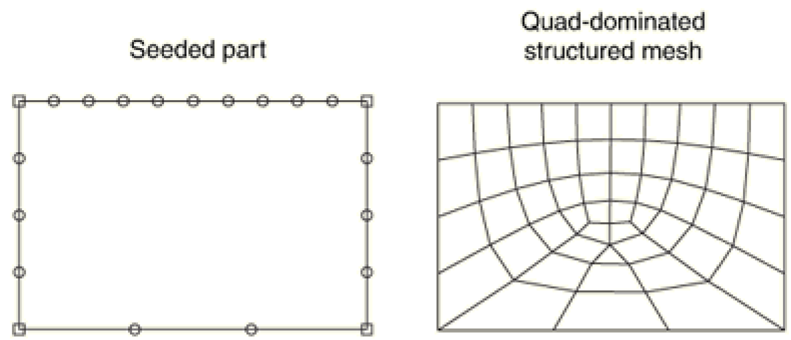
To mesh a 2D region with a structured mesh, the region needs:

- Being delimited by a number of logical edges comprised between 3 and 5 (a logical edge is a set of connected geometrical edges)
- Not have holes, isolated edges or isolated vertices

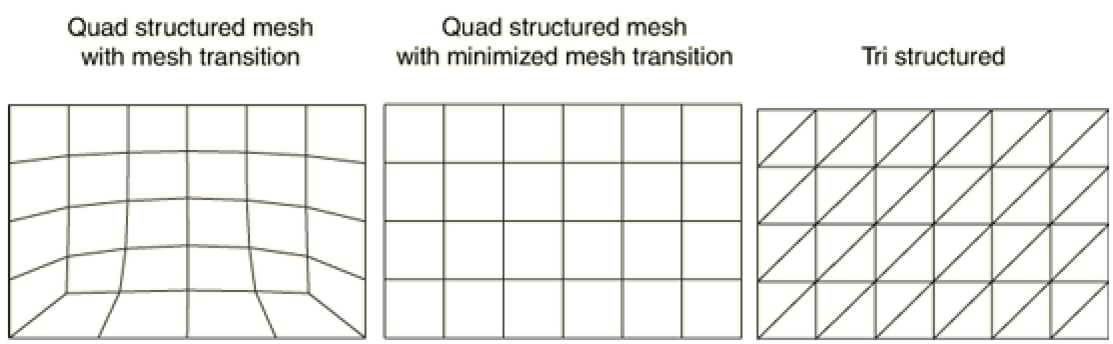


Structured mesh of 2D regions

- Mesh seeds are usually respected
- However, this may depend on the mesh control settings
- Moreover, the mesh needs to be conformal to that of the neighboring regions



Seeds are respected, but mesh is distorted

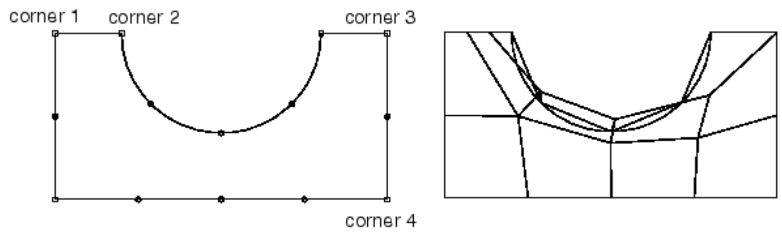


Elements are more regular, but seeds were not completely respected

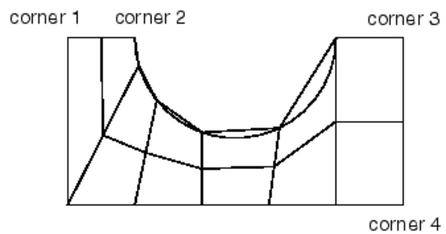
Elements are very regular, but seeds were disregarded

Structured mesh of 2D regions

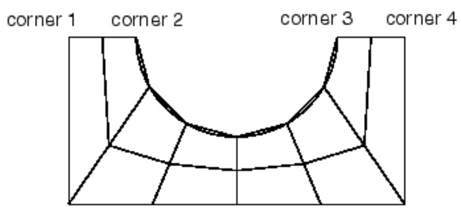
The shape of the region should not be too complex, otherwise it may result into invalid elements



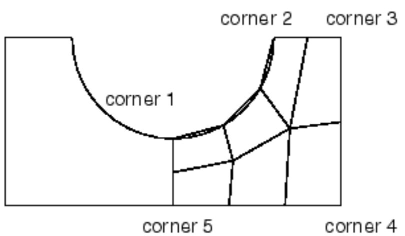
To solve the issue: adjust seeds; redefine logical edges; make partitions



adjust seeds



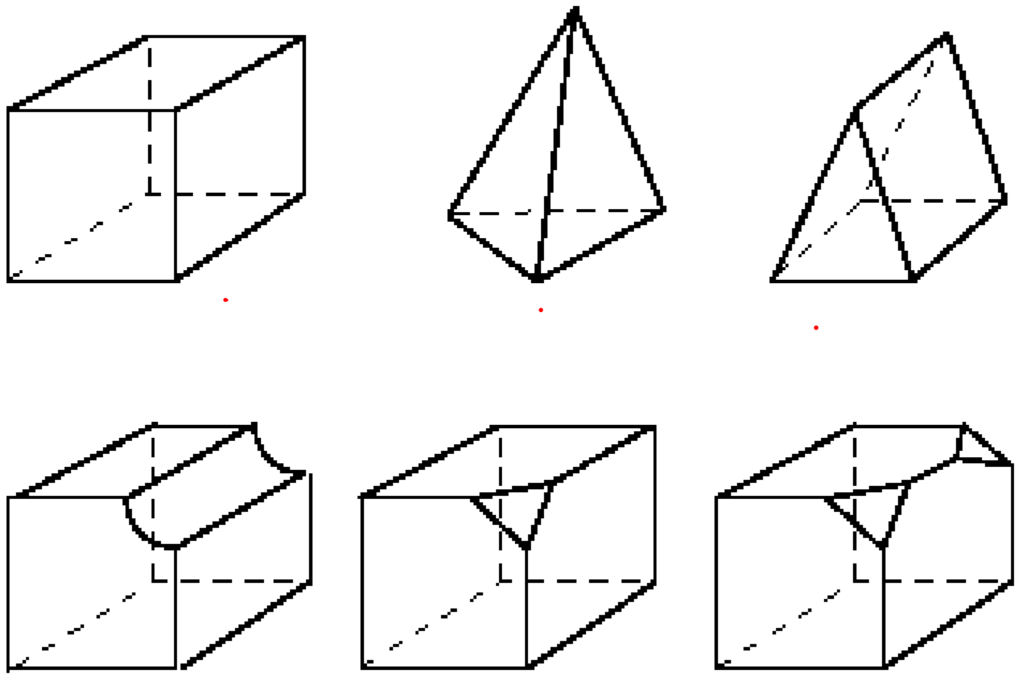
redefine logical corners



partition into two five-sided regions

Structured mesh of 3D regions

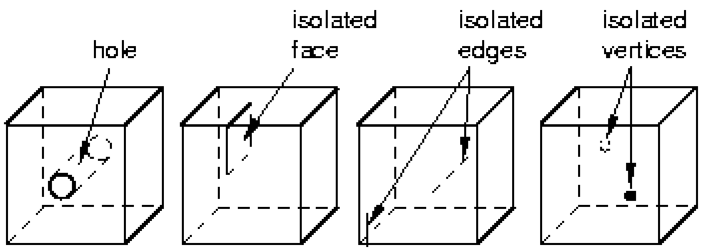
Define simple volumes, compliant with structured meshing

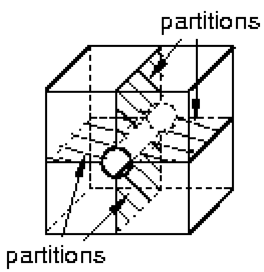


To mesh more complex geometries, partitions may be needed. Otherwise, if partitions cannot be introduced, non structured meshes with tetrahedra are required

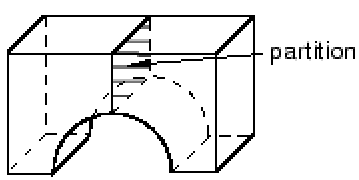
Structured mesh of 3D regions

Volumes must not have holes, isolated faces, isolated edges, isolated vertices

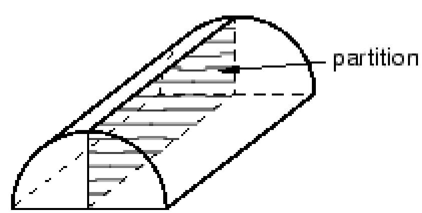




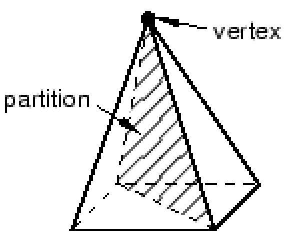
Holes needs being partitioned



Arcs should not exceed 90°



All faces must obey to rules for structured meshing of 2D regions



3 edges must meet at each vertex

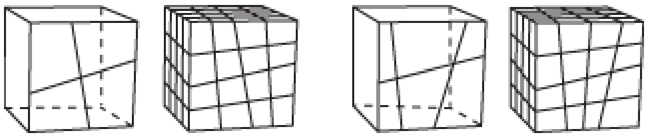
Structured mesh of 3D regions

The volume to be meshed needs being delimited by at least 4 faces (surfaces) and it is advisable that the angles between the faces (whatever their number) is as close as possible to 90° (angles larger than 150° must avoided)

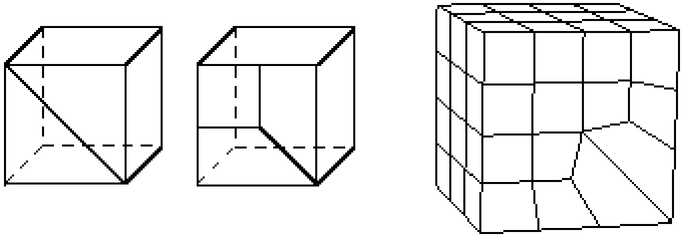
Every face must satisfy these conditions:

- If the volume is not cubic, each face must correspond to a single surface
- If the volume is cubic, each face can be composed of more surfaces (however, they must be at least 4 and must allow for regular hexahedral elements)

Acceptable



Not acceptable

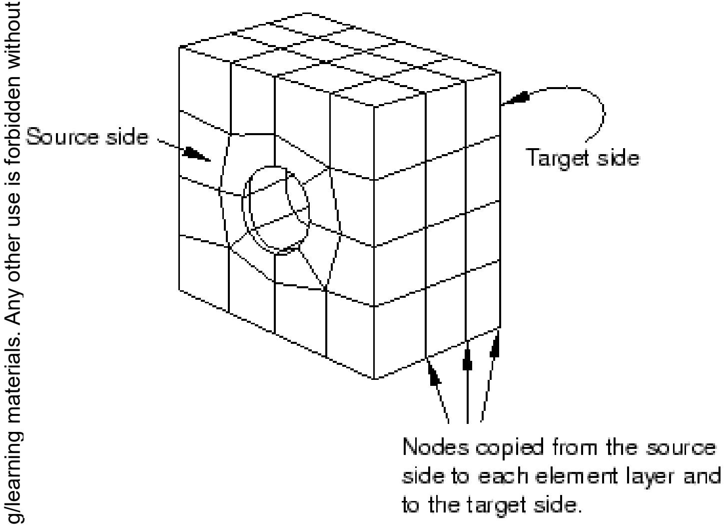


Extruded or swept mesh

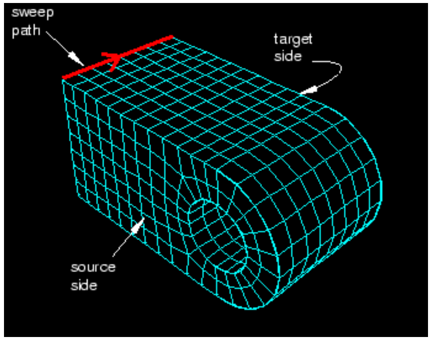
Extruded meshes requires two steps:

Creation of the mesh on one face (source side)

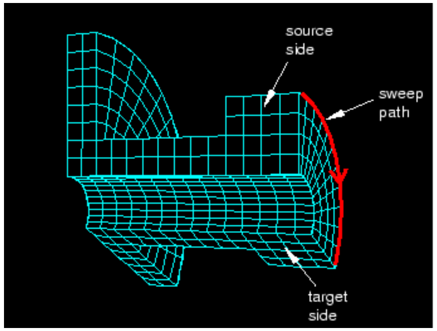
Copying the nodes of the source mesh along a sweep path, creating the mesh layer-wise, until the target side is reached



If the path is rectilinear or a spline, the mesh is defined as extruded



If the path is circular, the mesh is defined as revolved

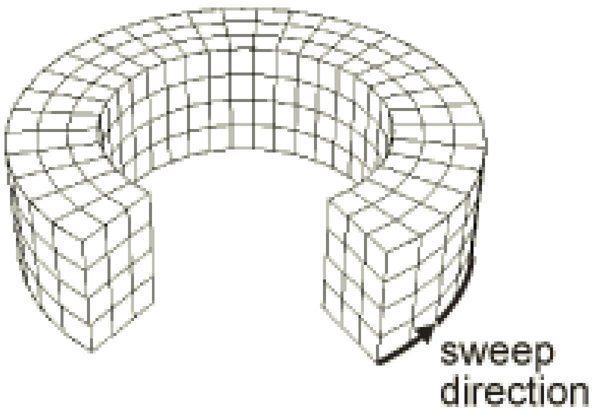


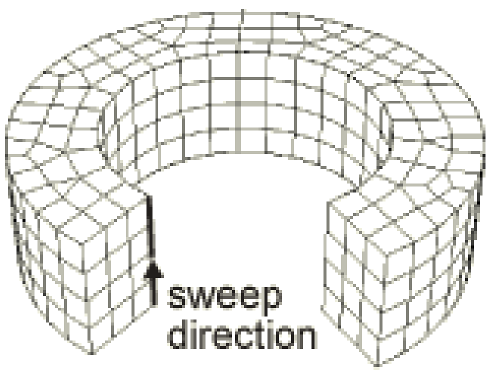
Extruded or swept mesh

Like structured meshes, extruded ones can only be applied to specific geometries and topologies.

If this is not possible, the geometry needs being partitioned until compatible topologies are obtained.

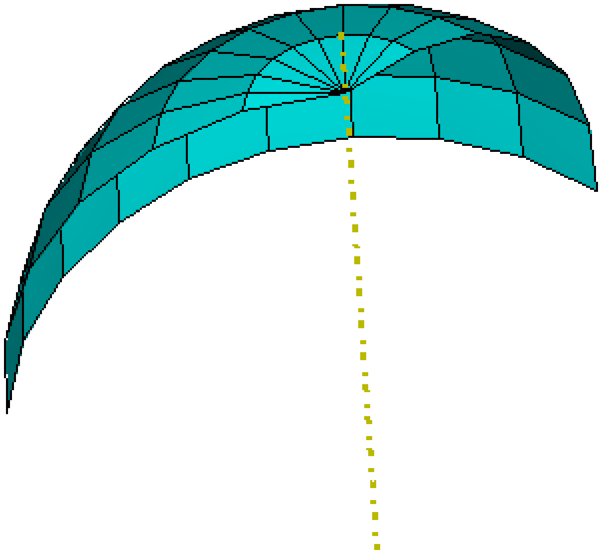
Sweep direction can be controlled by the user. However, the final quality can be significantly affected by the choice of the sweep path (i.e. by the quality of the source mesh)





Extruded or swept mesh

By extrusion, 2D meshes can be obtained from an edge, 3D ones from a surface



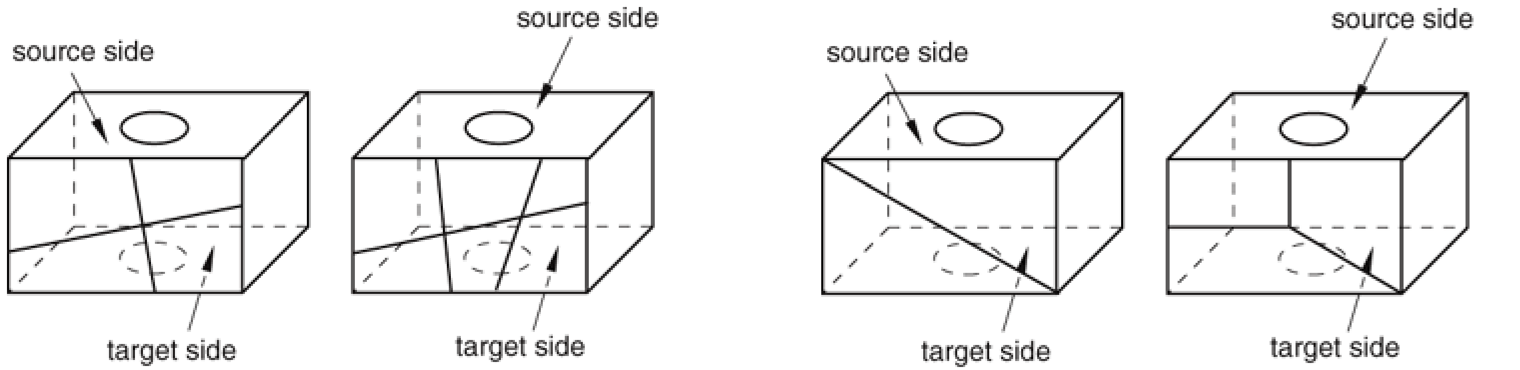
In some cases, to make it possible to generate meshes by revolution it can be required to allow the introduction of triangular elements (quad-dominated mesh)

Extruded or swept mesh

A volume can be meshed by extrusion if each face connecting the source side with the target side is composed of a single surface or by a set of surfaces allowing for regular meshing

Acceptable

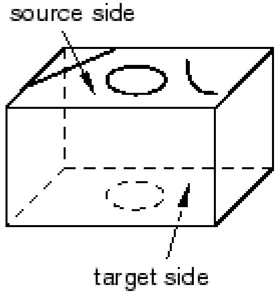
Not acceptable

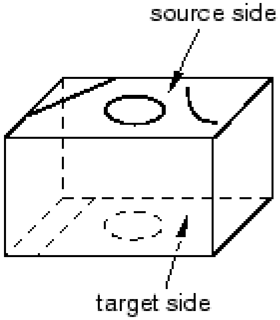


Extruded or swept mesh

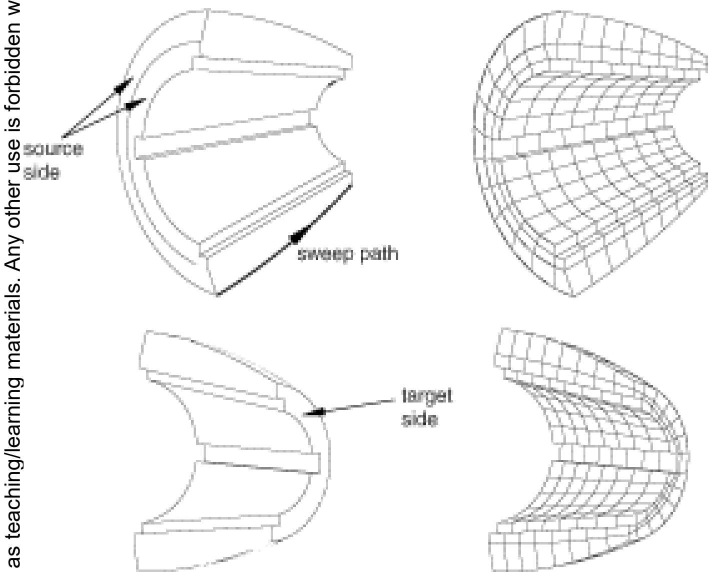
The target side cannot include isolated edges or vertices

Acceptable





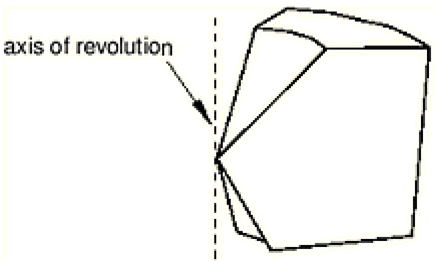
Not acceptable

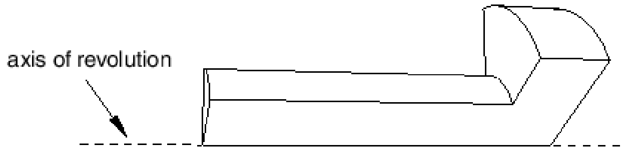


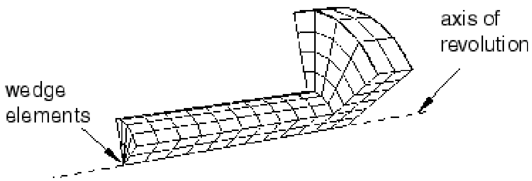
This is an example of a relatively complex geometry, with variable thickness, which however complies with the extruded mesh rules

Extruded or swept mesh

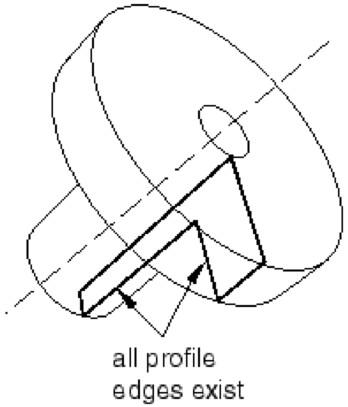
In a revolving mesh, the contour must not intersect the axis of revolution at one or more isolated vertices, unless tetrahedra are accepted



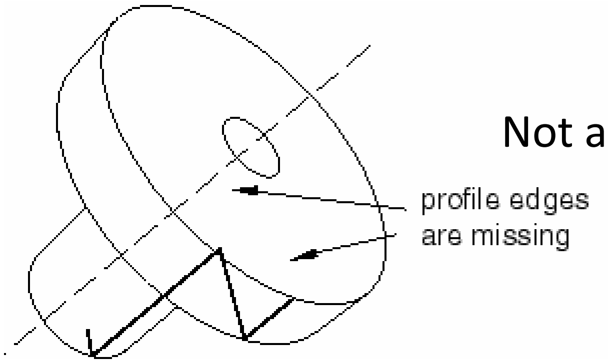




A volume of revolution which does not intersect the axis of revolution can be meshed only if all profile edges exist



Acceptable

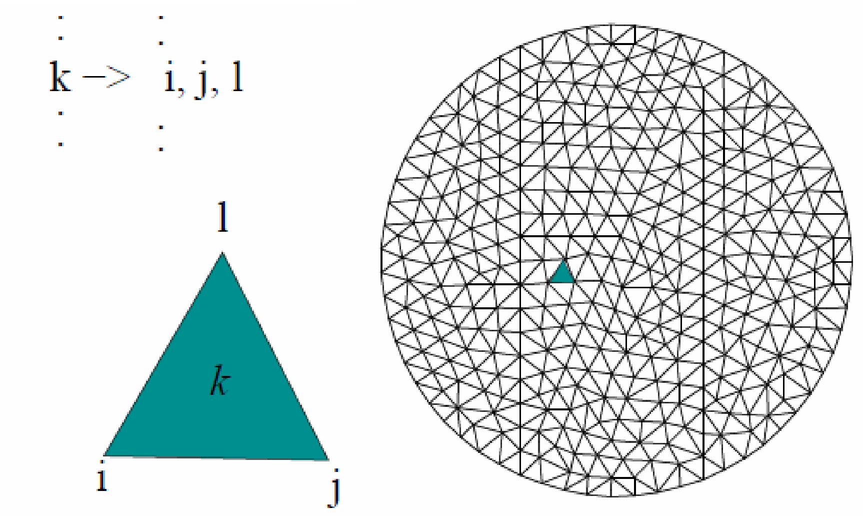


Not acceptable

Non structured, or free, mesh

A grid of elements is defined as non-structured when access to node data requires direct addressing, via the connectivity matrix

Connectivity matrix



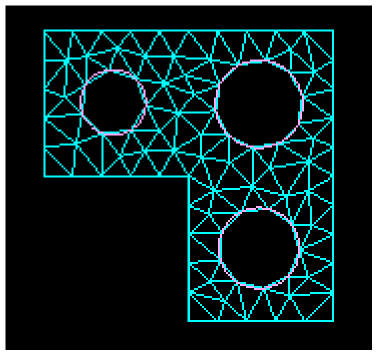
Pros: very flexible; allows for local refinement; can reach a high level of automation

Cons: more memory usage than structured meshes; less efficient because of direct addressing

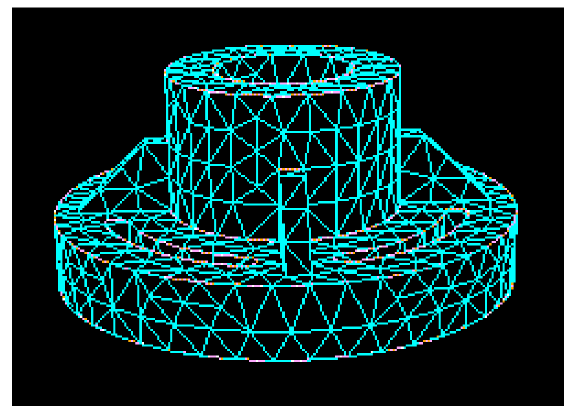
Non structured, or free, mesh

It’s more flexible and versatile. It can be applied to any geometry. However, it allows for low control by the user upon the result, which is strongly dependent upon the meshing algorithm

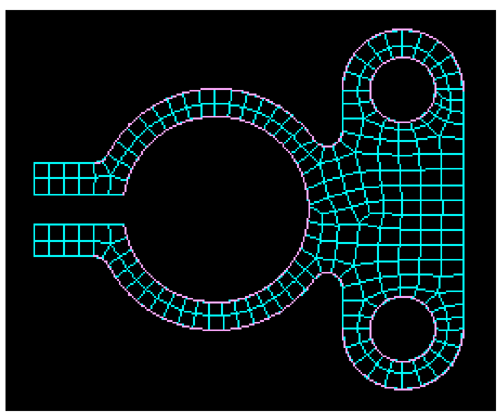
Triangular elements



Tetrahedral elements



Quadrangular elements



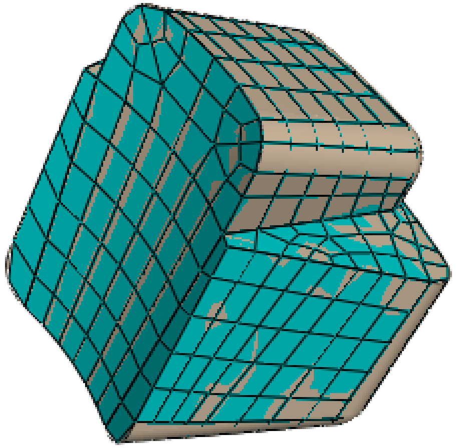
A symmetric object can result into a non symmetric mesh

Bottom-up techniques

By these techniques, 3D meshes are generated starting from 2D ones

Applicable to 3D geometries only, using hexahedra. The mesh needs not being conformal to the geometry edges.

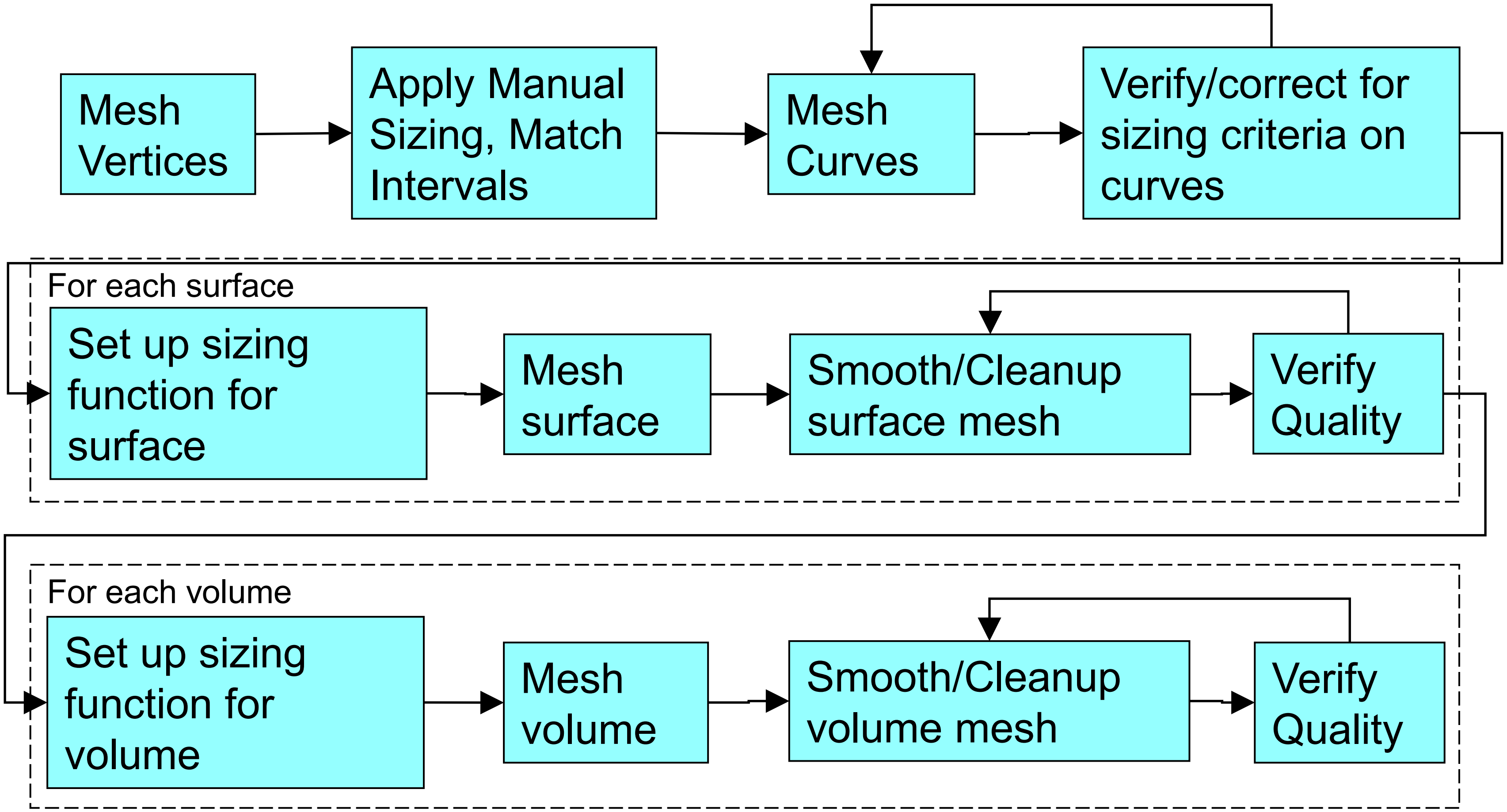
It is a manual process, that may lead to significative differences between the meshed region and the original geometry. However, they allow to obtain high quality meshes with hexahedral elements even in regions of very complex shape



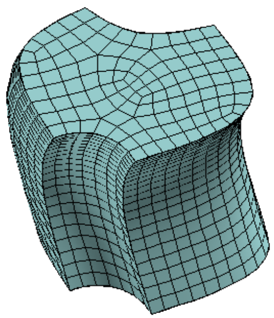
It allows for the maximum level of control by the user. However, the user must decide upon the quality of the final shape. If the quality is not acceptable, the geometry needs being partitioned, or the mesh needs to be redone.

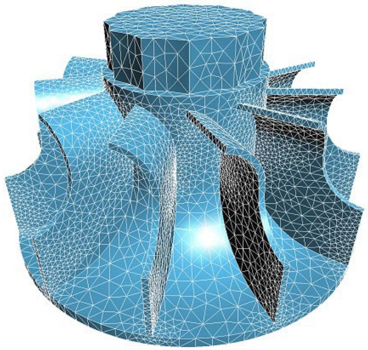
It is advisable to apply bottom-up techniques only when it is not possible to obtain a good quality mesh with a top-down approach

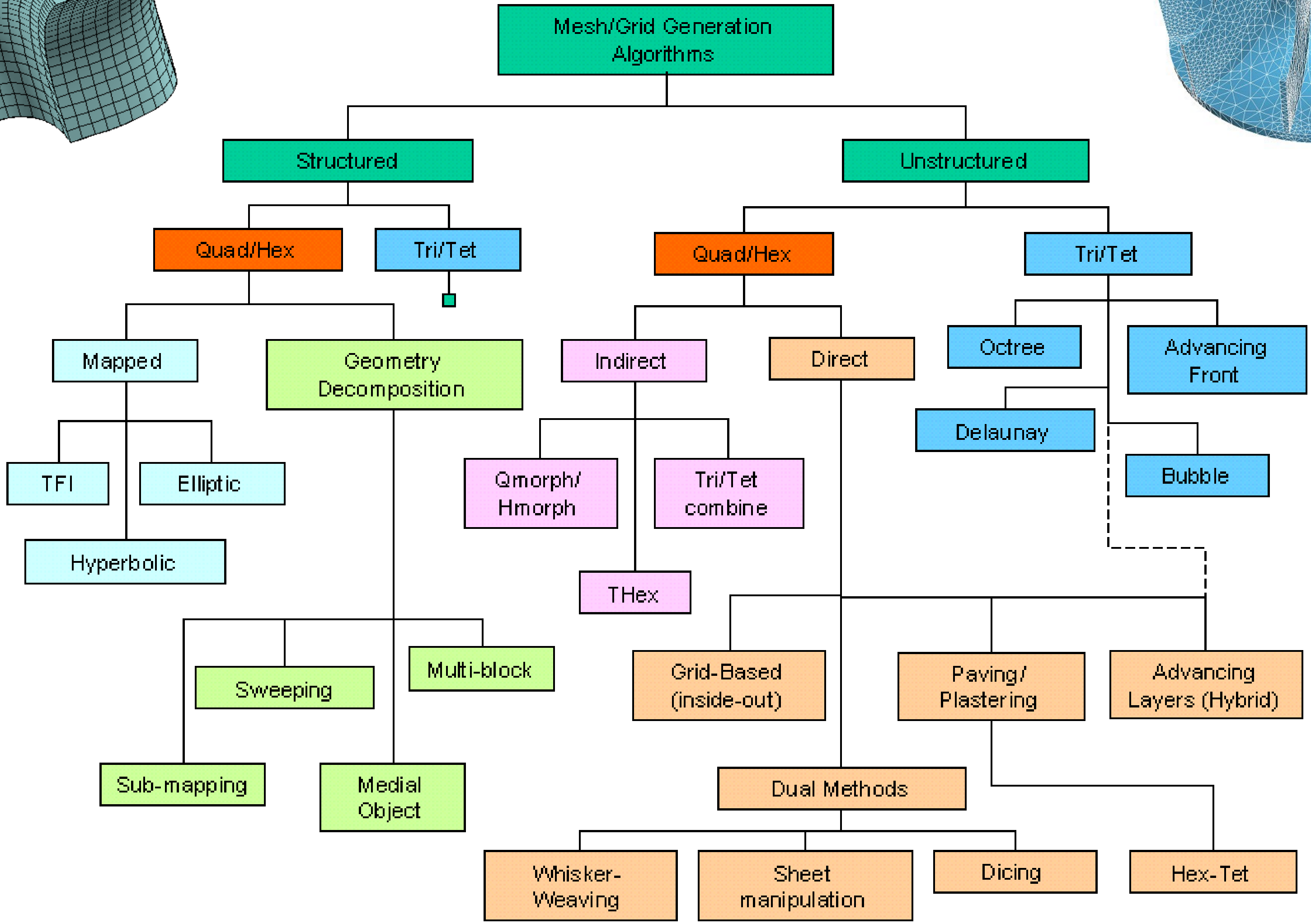
The meshing process



Meshing algorithms







Algorithms for non structured meshing

Generally, a structured mesh is difficult to apply automatically to complex geometries.

Alternatively, it is possible to use simpler elements, typically triangular or tetrahedral, because they allow for more flexibility to describe a given geometry and control the position of the vertices.

Consequently, the major part of automatic meshing algorithms are suitable for non structured meshes.

The definition of a non structured mesh consists of the creation of vertices and their relationships in terms of:

- Definition of the contour of the domain
- Definition of the distribution function of the dimensions of the elements
- Generation of a mesh which follows the geometric contour
- Optimization of the shape of the generated elements

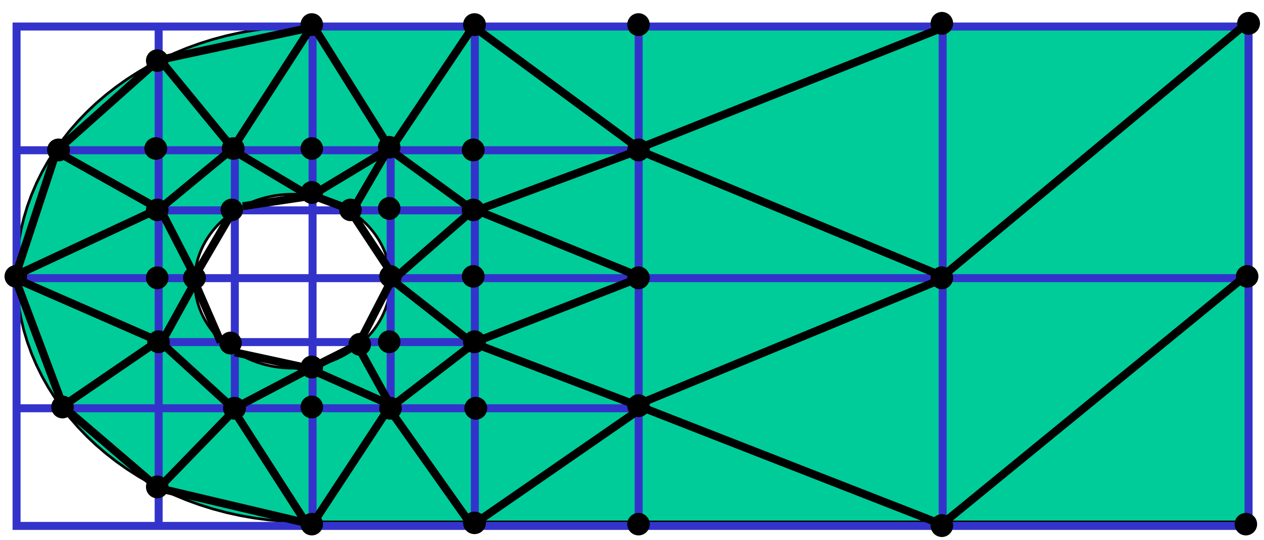
Algorithm "QuadTree/OcTree"

Suitable for the generation of non structured meshes using 2D triangular elements (QuadTree) or 3D tetrahedra (OcTree). In the 2D case:

Definition of an initial geometric shape (bounding box, also called root of the QuadTree) which includes the geometry to be meshed

Recursive subdivision in four leaves of each root, so that the geometry is resolved (by bisection steps)

Meshing each leave using its own vertices, adjacent vertices and intersections of the leaves with the edges of the geometry; deletion of all what remains outside the original geometry



Algorithm “QuadTree/OcTree”

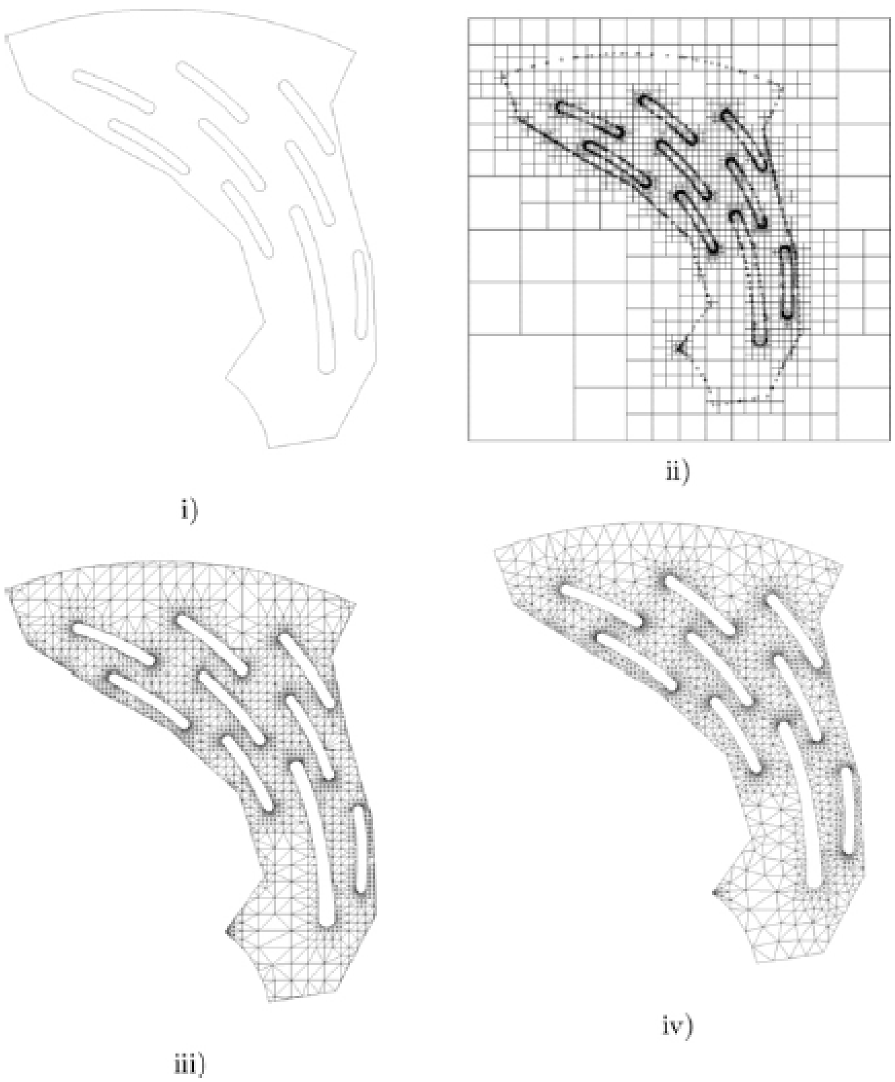


Figure 5.1: *A two-dimensional domain Ω i); spatial decomposition of B(Ω), the bounding box of Ω ii); the resulting mesh iii); and the final mesh of Ω iv).*

Triangles are obtained as partitions of the quadrangular elements that represent the QuadTree grid

The algorithm tend to refine the mesh at geometric discontinuities internal to the domain, characterized by sharp fillet radii

At the end of the procedure, the mesh is refined and optimized

Advancing front algorithm

Suitable to create non structured 2D and 3D meshes using triangular/tetrahedral or quadrangular/prismatic elements

The mesh is generated progressively from the exterior edges of the geometry

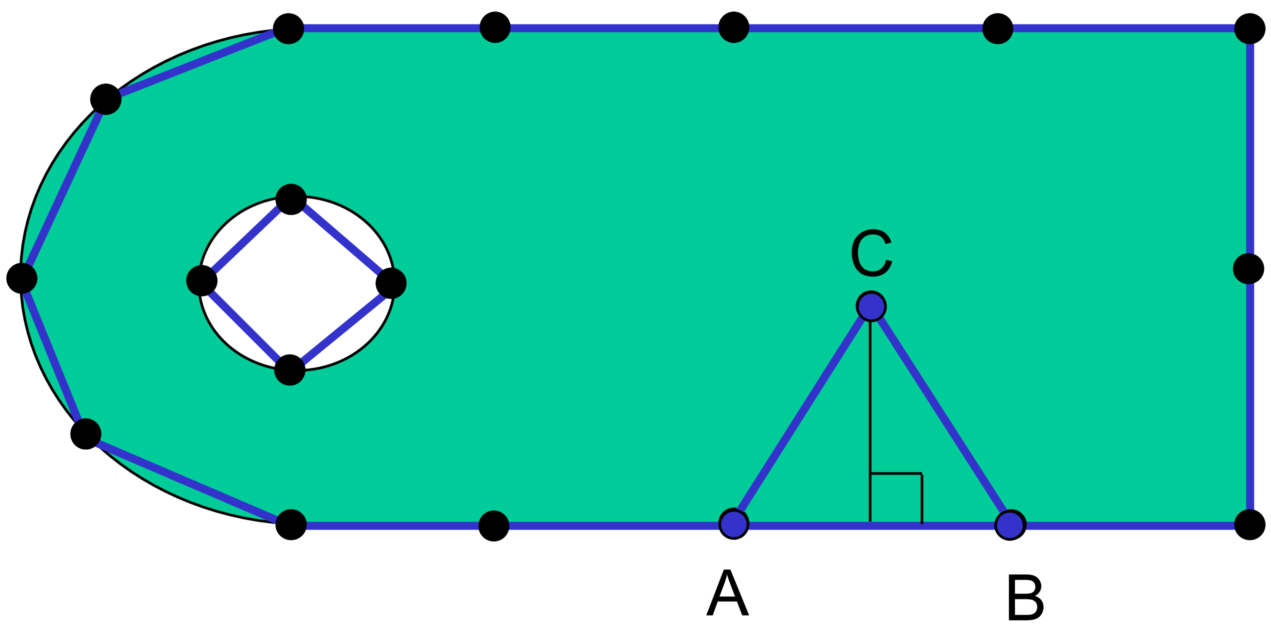
The iterative procedure results into the propagation, inside the domain, of a front that is the border between the meshed region and the one that still has to be meshed

The difficulty of the method consists of ensuring the consistency of the advancing front that needs to be assessed according to the following criteria:

- Quality of the resulting element
- Required mesh density
- Local constraints, e.g. other parts of the exterior edge or of the front
- All vertices must lay inside the geometric domain

Advancing front algorithm

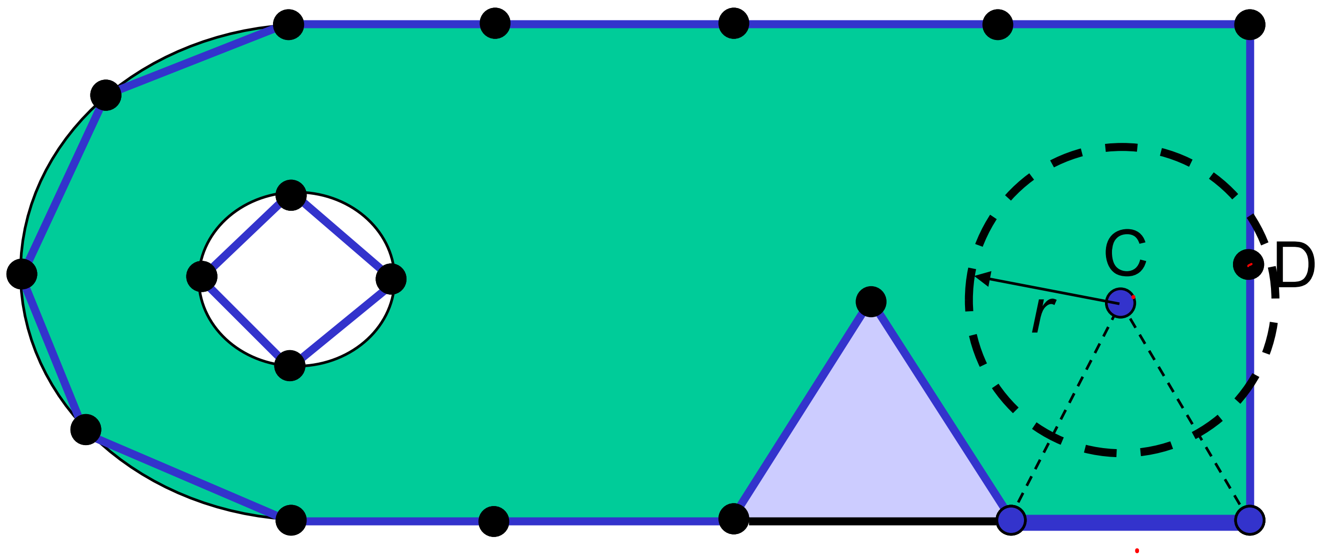
First, the exterior edges are meshed



For each segment AB of the contour, it is necessary to identify a point C which satisfies the criteria of topological consistency (in this case, an equilateral triangle)

Advancing front algorithm

Always check that any previously created node does not fall inside a circle of radius r centered on the point C being considered

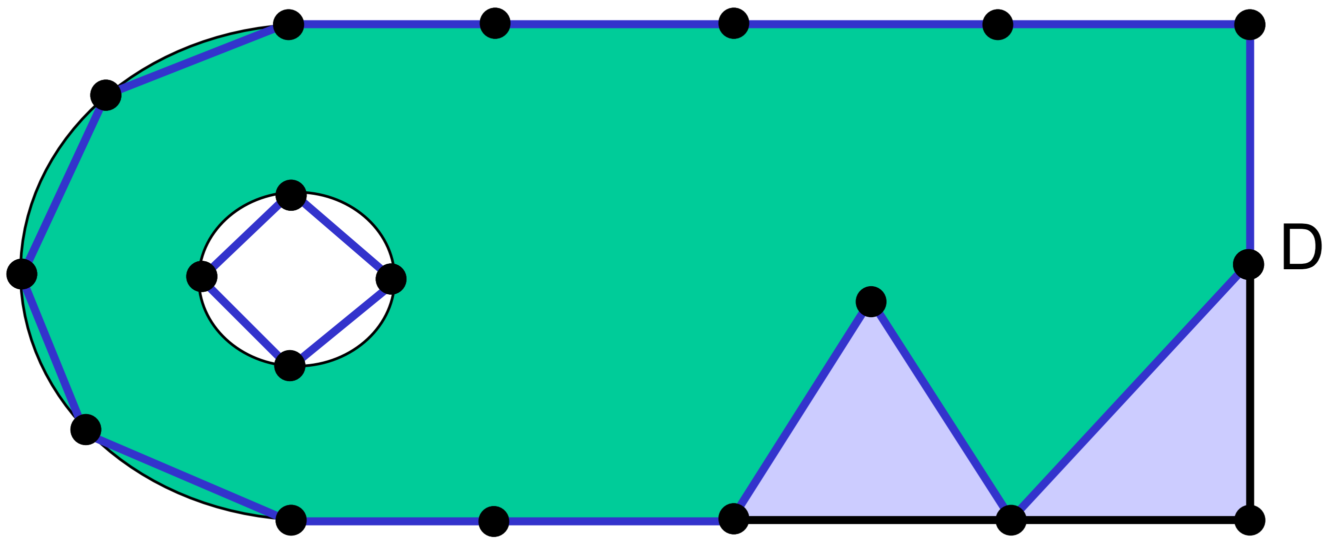


If so, point C is not created and D is chosen instead

The value of r depends on the criteria of consistency

Advancing front algorithm

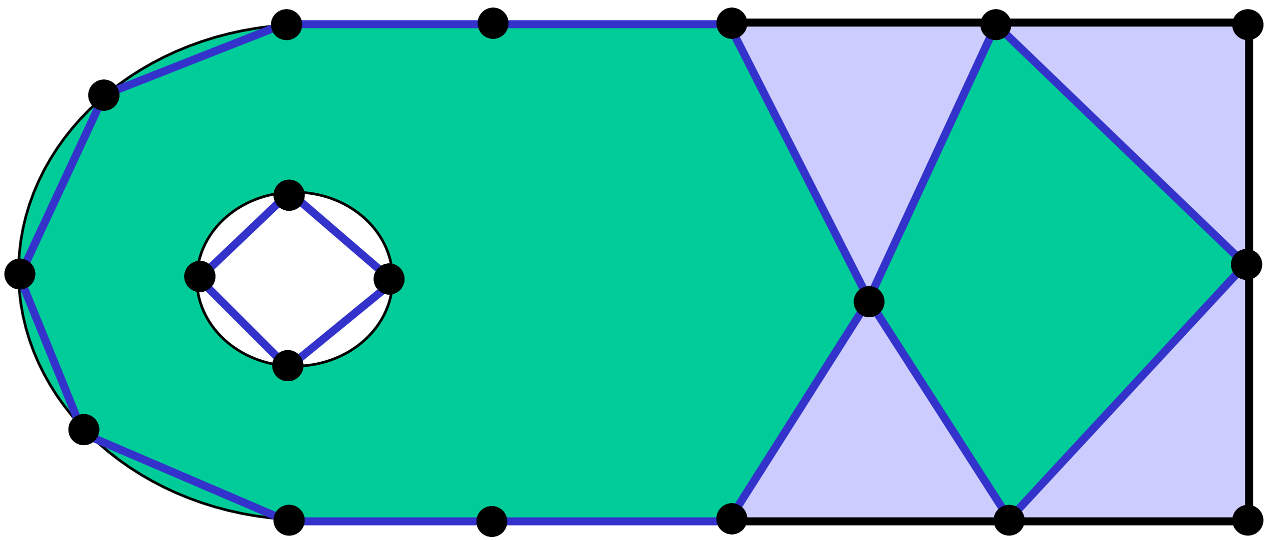
As elements propagate, the morphology of the front varies continuously, adding and removing edges



This part of the algorithm continues as long as all free edges are available for the construction of new elements

Advancing front algorithm

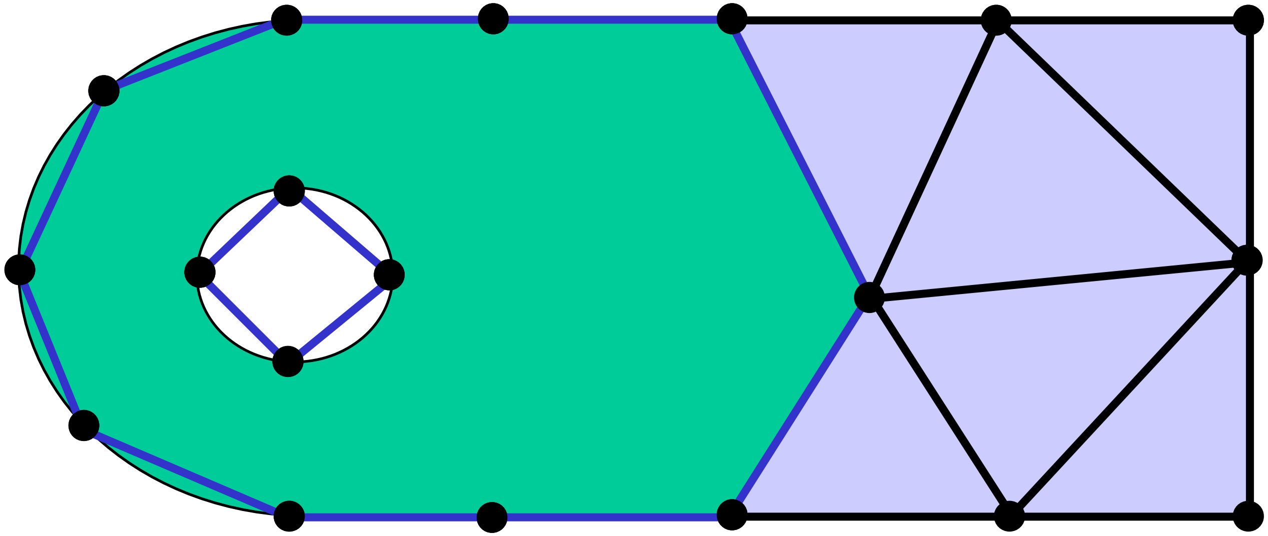
As elements propagate, the morphology of the front varies continuously, adding and removing edges



This part of the algorithm continues as long as all free edges are available for the construction of new elements

Advancing front algorithm

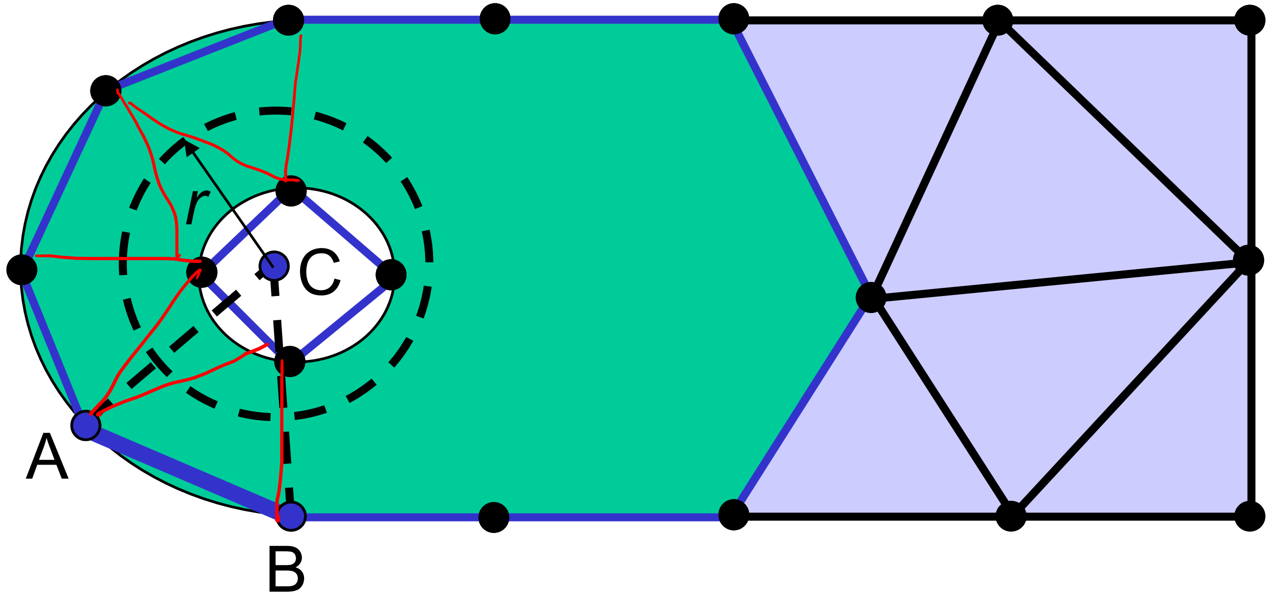
As elements propagate, the morphology of the front varies continuously, adding and removing edges



This part of the algorithm continues as long as all free edges are available for the construction of new elements

Advancing front algorithm

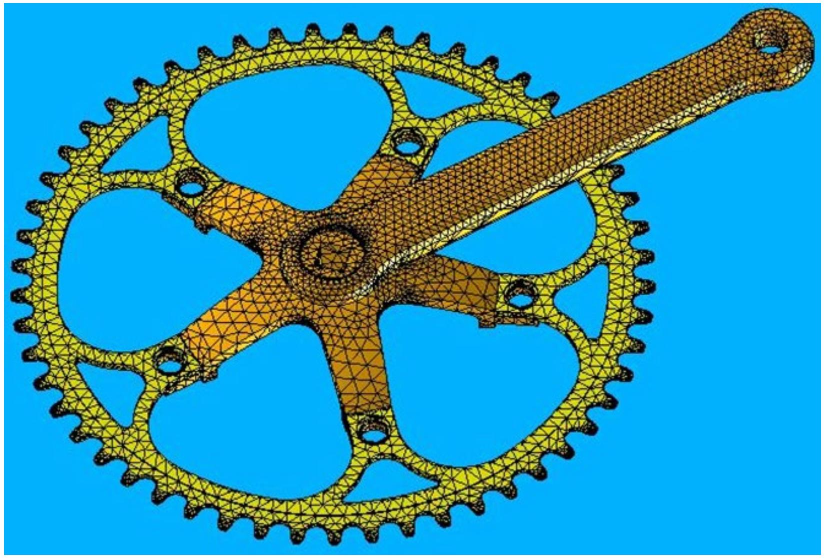
If multiple options are available, it is necessary to choose the one that satisfies better the criteria of consistency (e.g. the shape closest to an equilateral triangle)

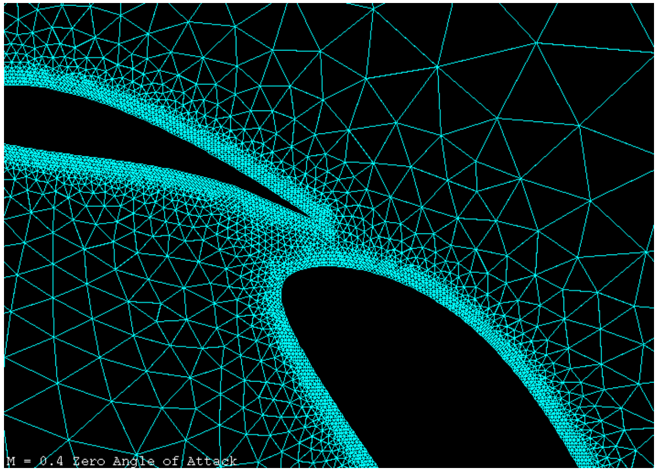


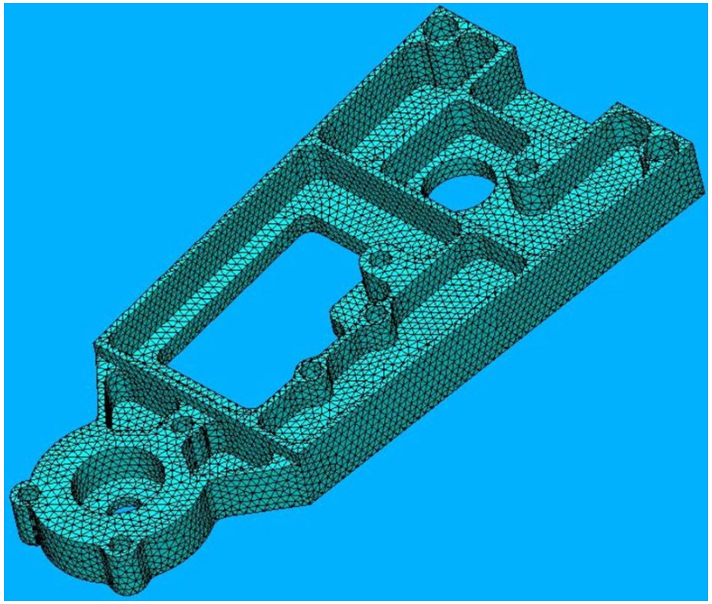
Intersections with already existing edges or fronts must be avoided

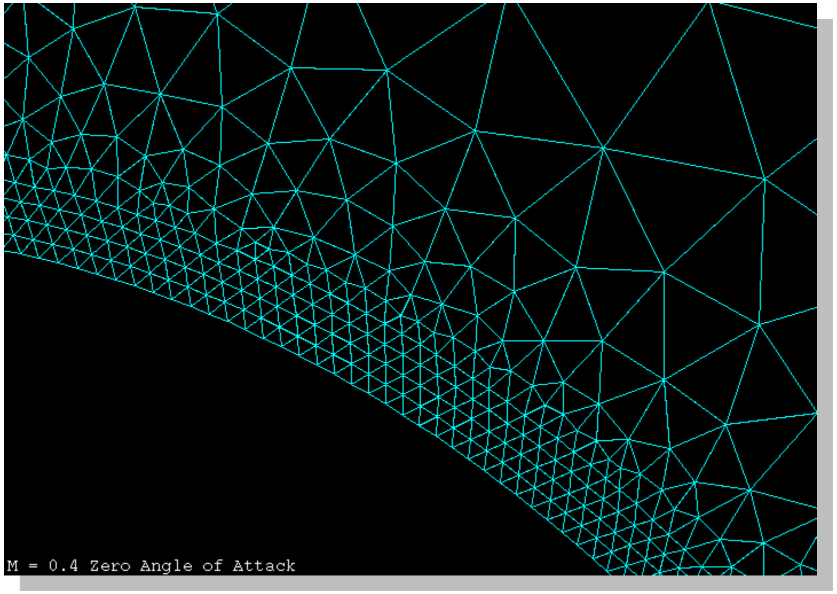
At the end of the procedure, the mesh is optimized

Advancing front algorithm





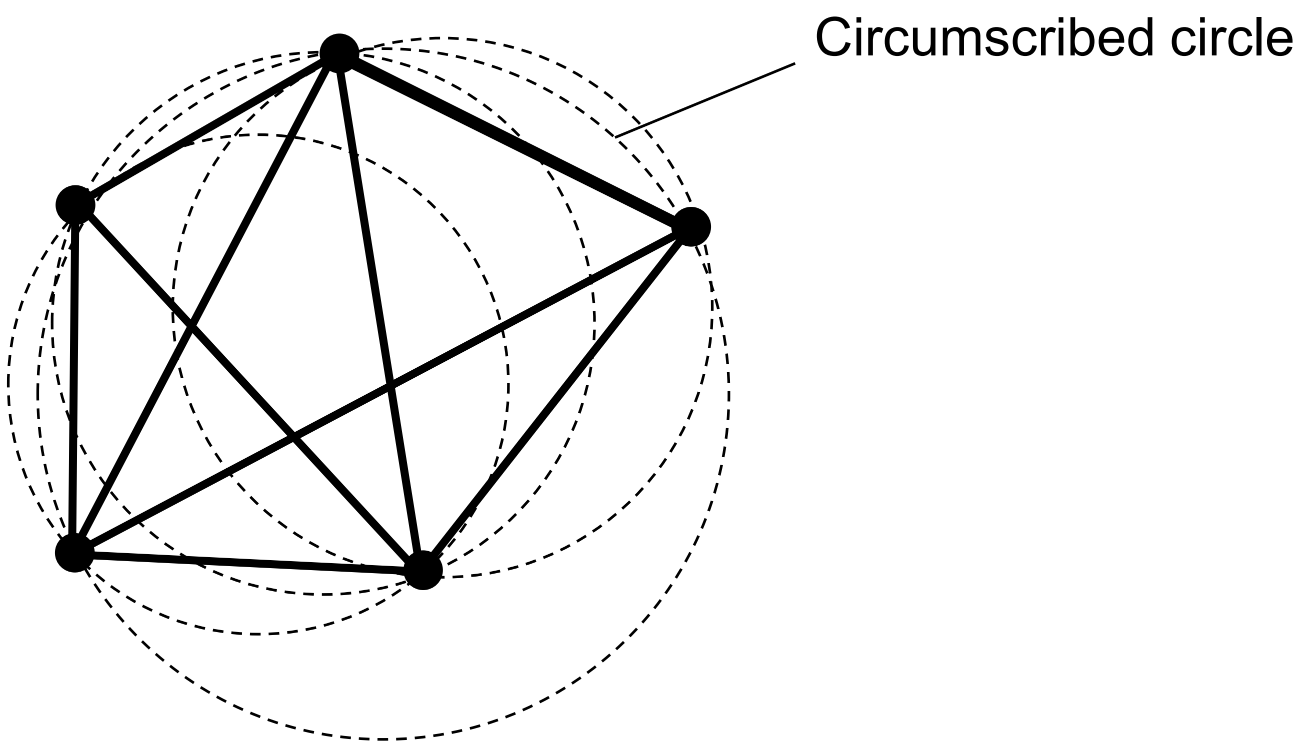




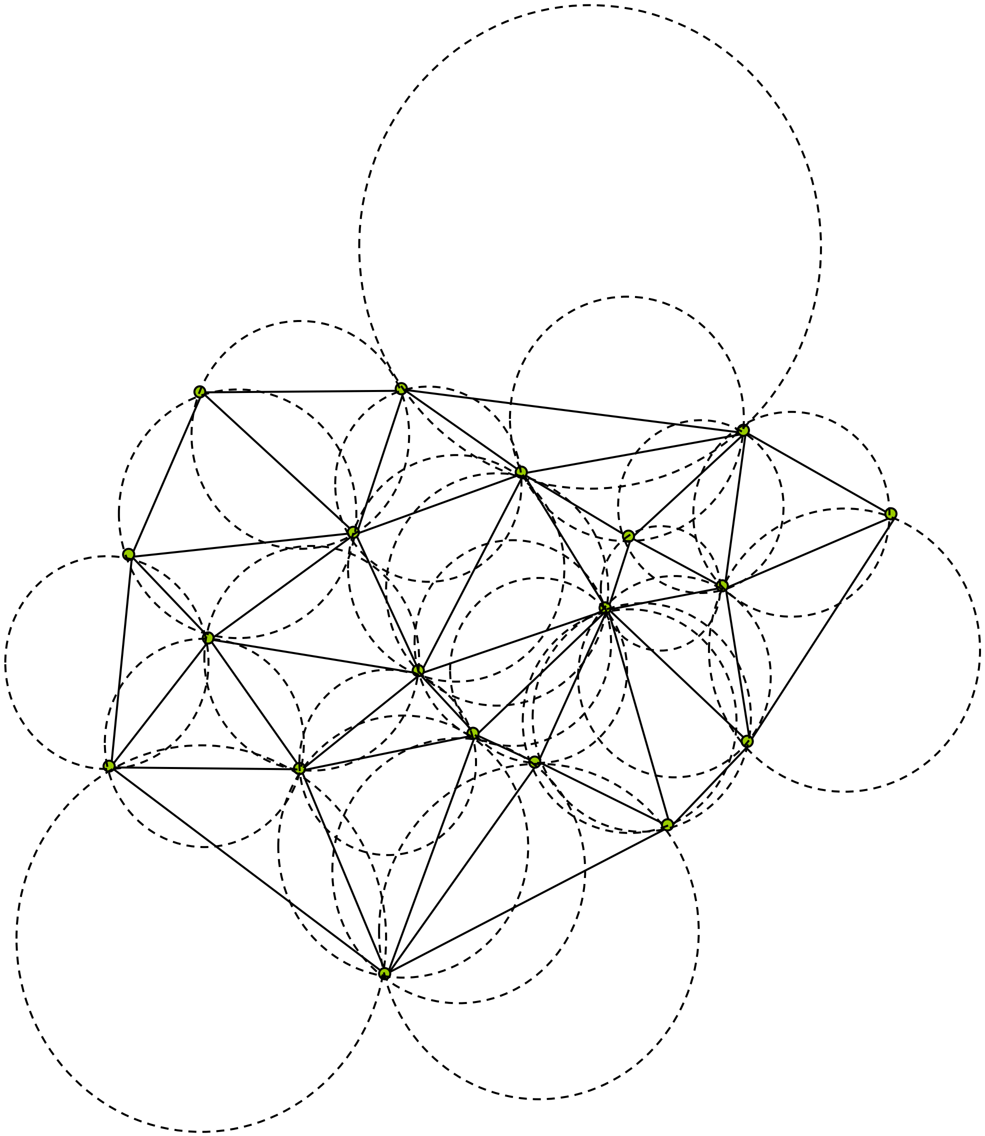
Delaunay triangulation

Suitable for 2D and 3D non-structured meshes using triangular/tetrahedral elements

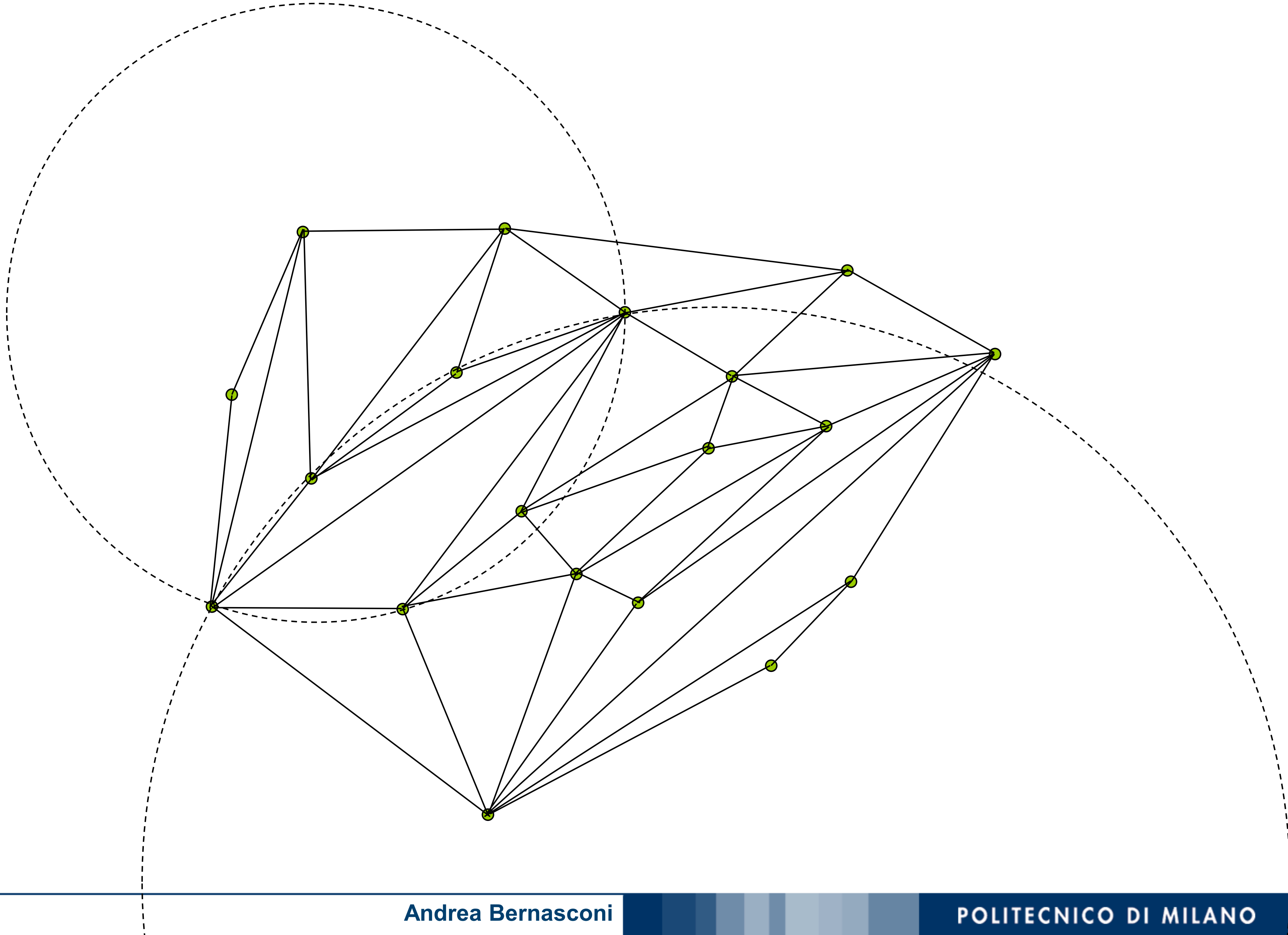
It is based on the empty circle/sphere principle: no other vertex can be inside the circle circumscribed to each triangle



This is a Delaunay triangulation

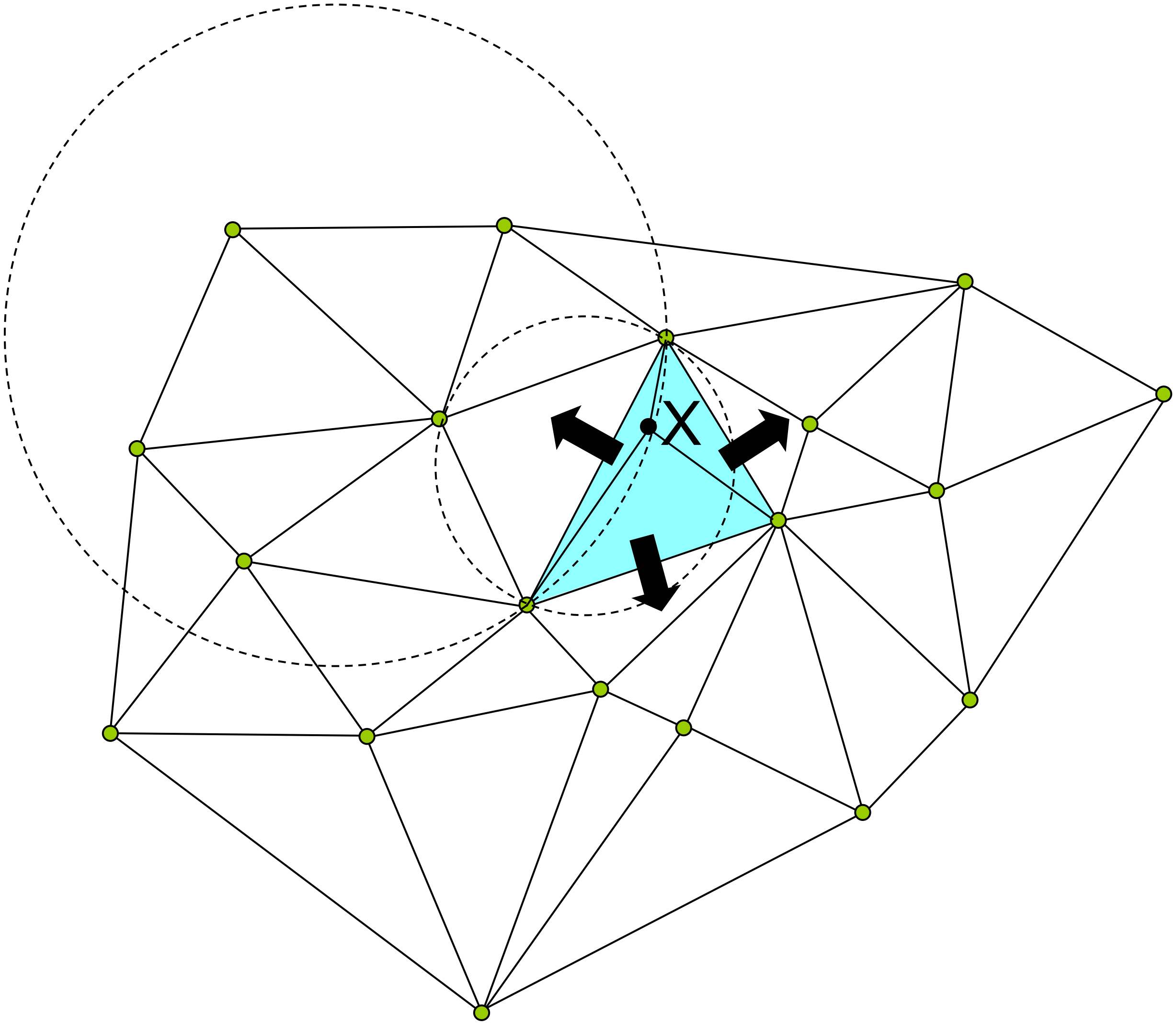


This is NOT a Delaunay triangulation



Delaunay triangulation

Given a Delaunay triangulation, how can a node be added?



Lawson's algorithm:

Identify the triangle containing X

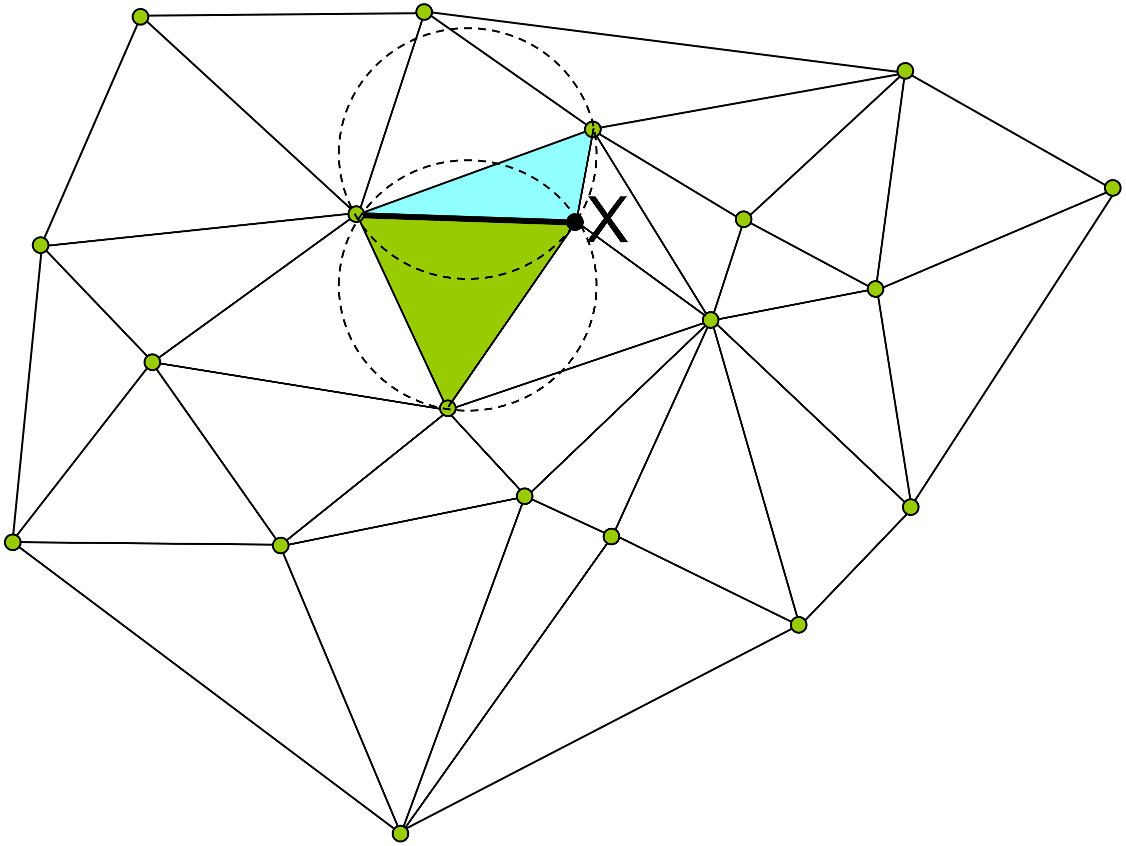
Subdivide the triangle

Check recursively the adjacent triangles to ensure that the empty circle principle is satisfied

Swap diagonals if necessary

Delaunay triangulation

Given a Delaunay triangulation, how can a node be added?



Lawson's algorithm:

Identify the triangle containing X

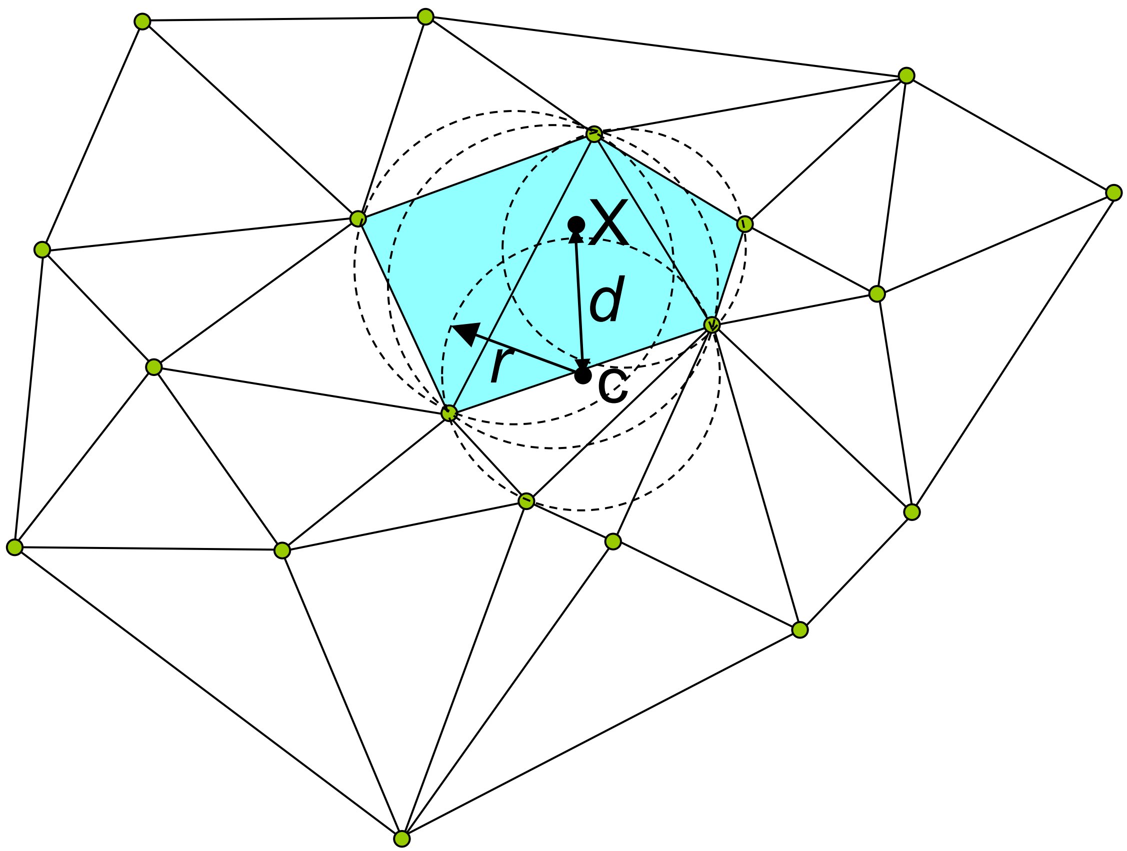
Subdivide the triangle

Check recursively the adjacent triangles to ensure that the empty circle principle is satisfied

Swap diagonals if necessary

Delaunay triangulation

Alternatively, the Bowyer-Watson algorithm can be applied

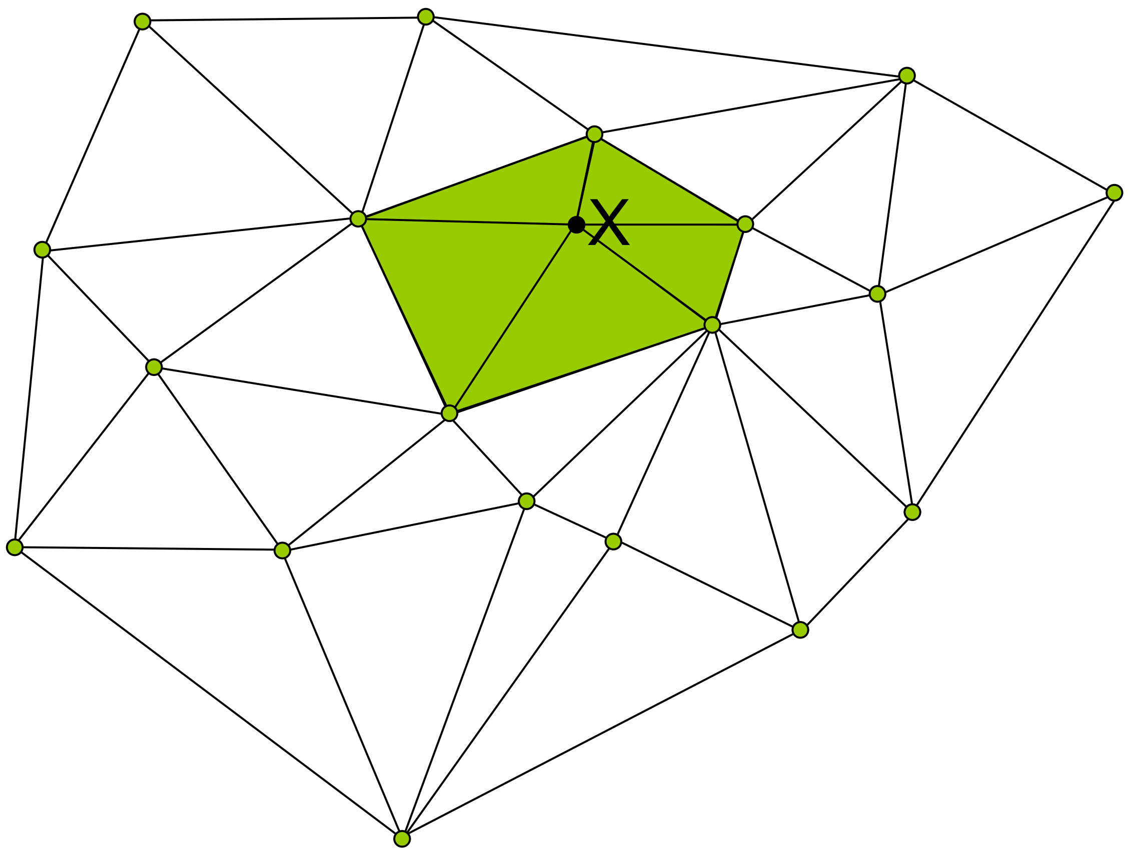


Identify the triangle containing X

Search for all triangles whose circumscribed circle includes X (d<r)

Delaunay triangulation

Alternatively, the Bowyer-Watson algorithm can be applied



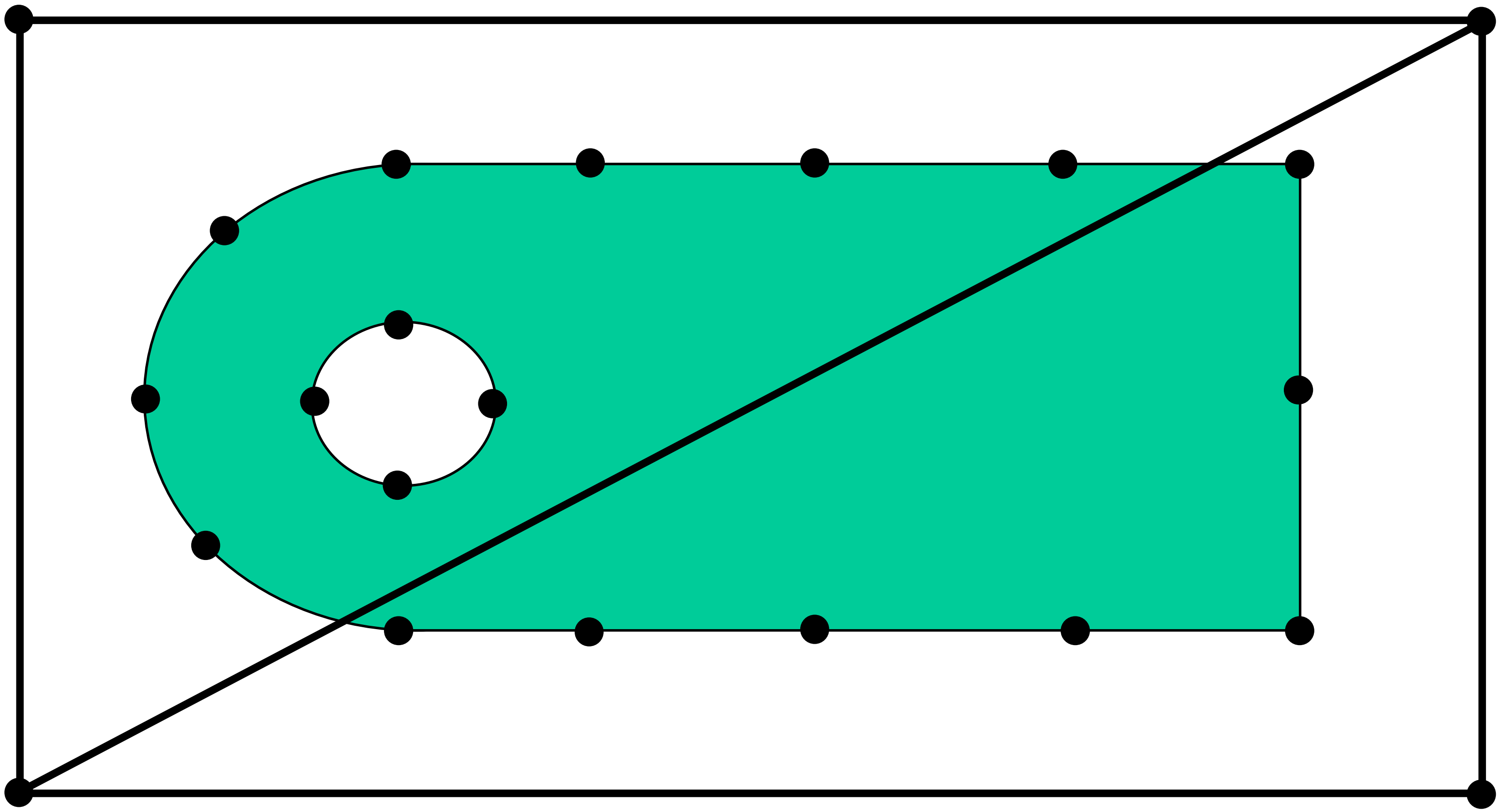
Identify the triangle containing X

Search for all triangles whose circumscribed circle includes X (d<r)

Delete these triangles (this leaves and empty space in the mesh)

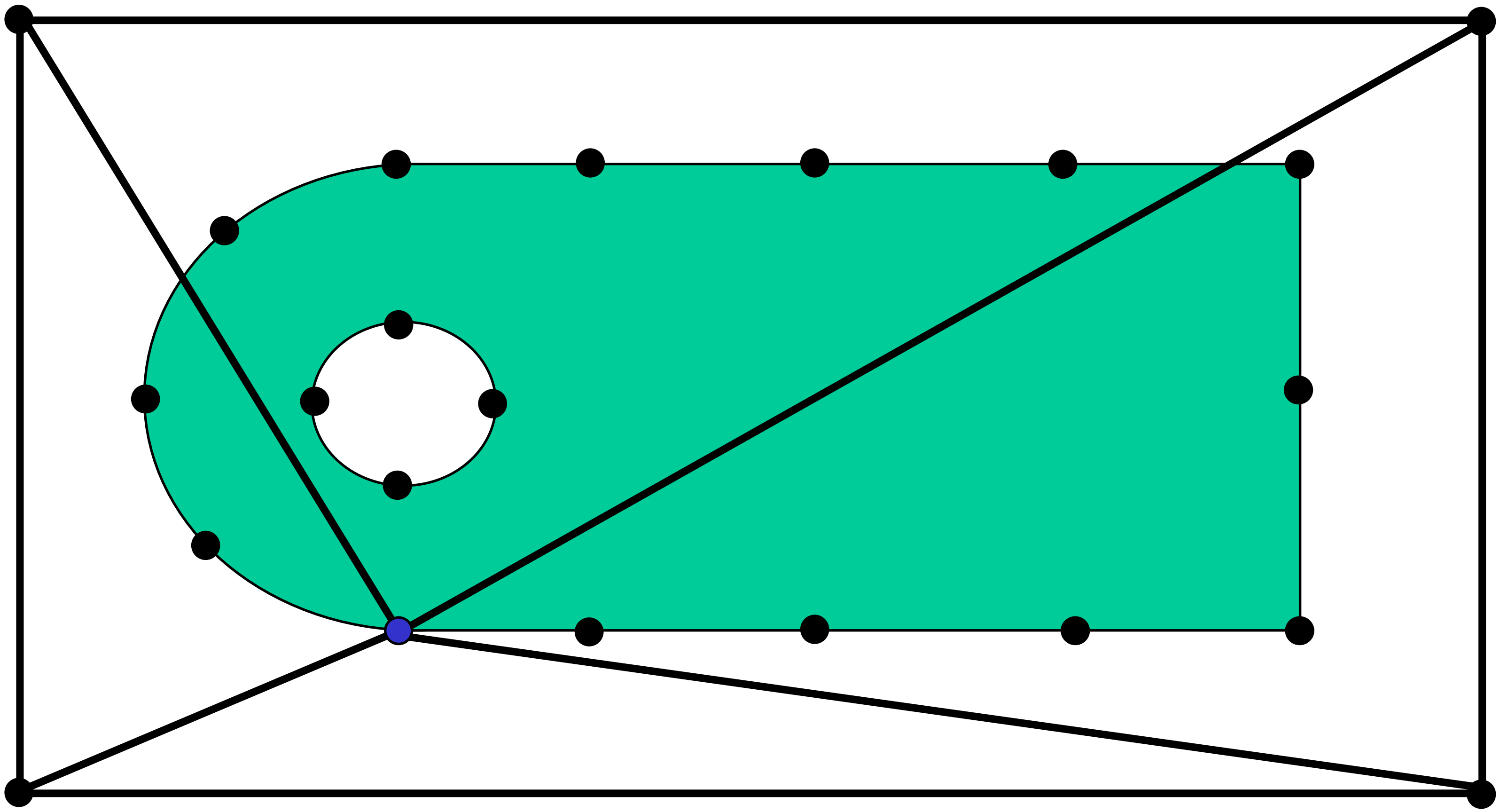
Create new triangles from X

Delaunay triangulation



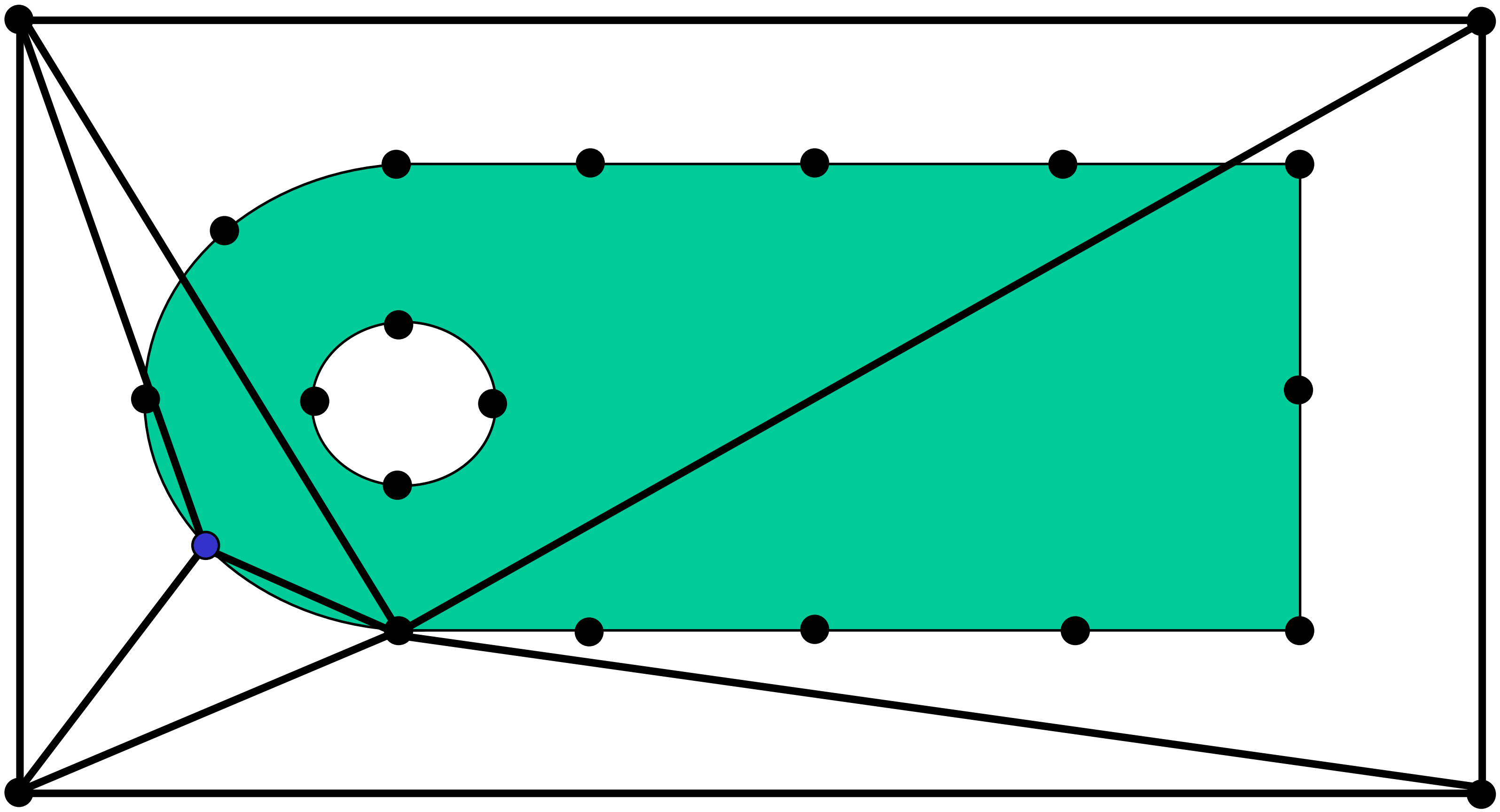
First, triangles containing the entire geometry are built

Delaunay triangulation



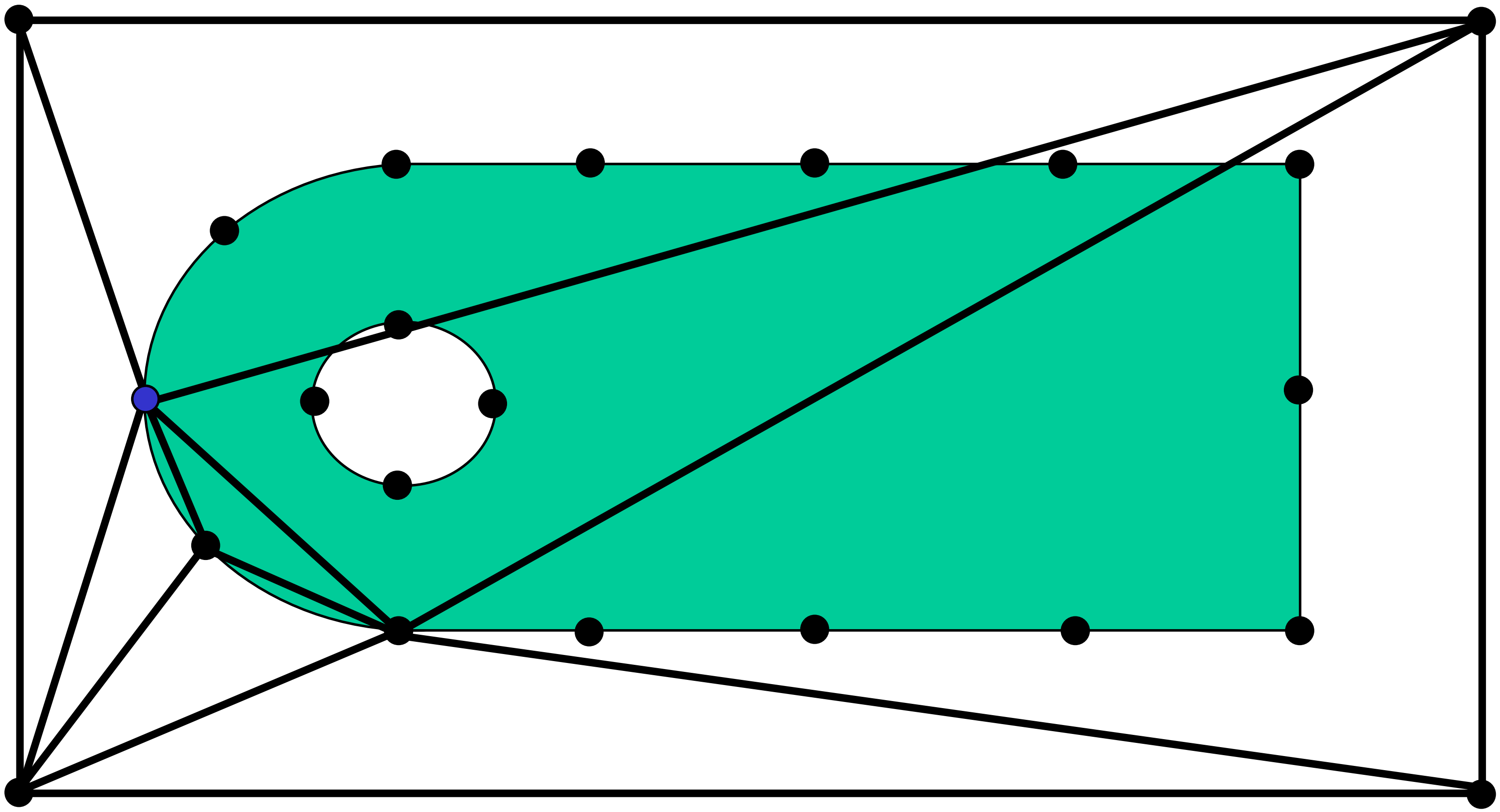
Edge node are created using Delaunay triangulation (by the Lawson or the Bowyer-Watson algorithm)

Delaunay triangulation



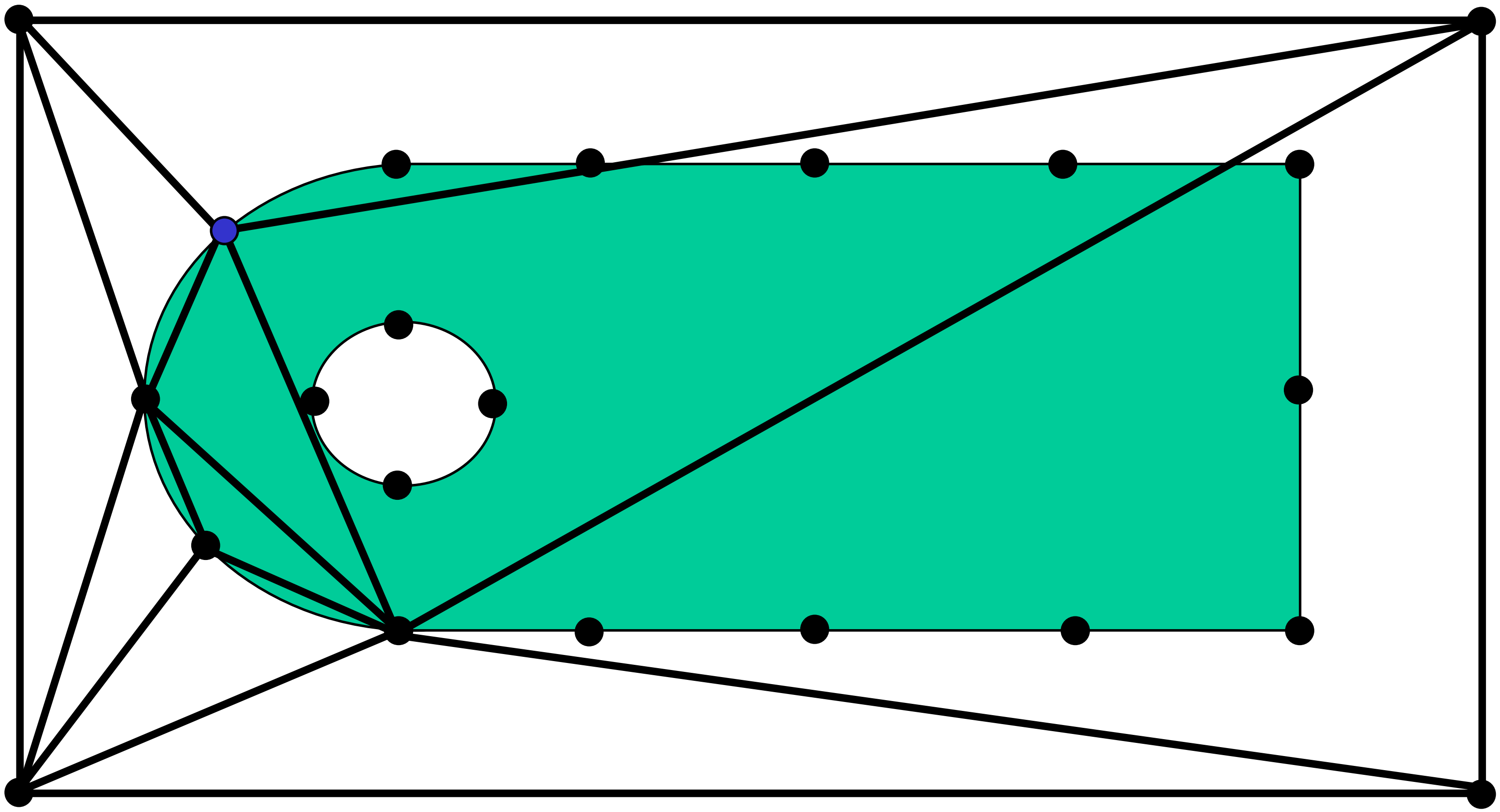
Edge node are created using Delaunay triangulation (by the Lawson or the Bowyer-Watson algorithm)

Delaunay triangulation



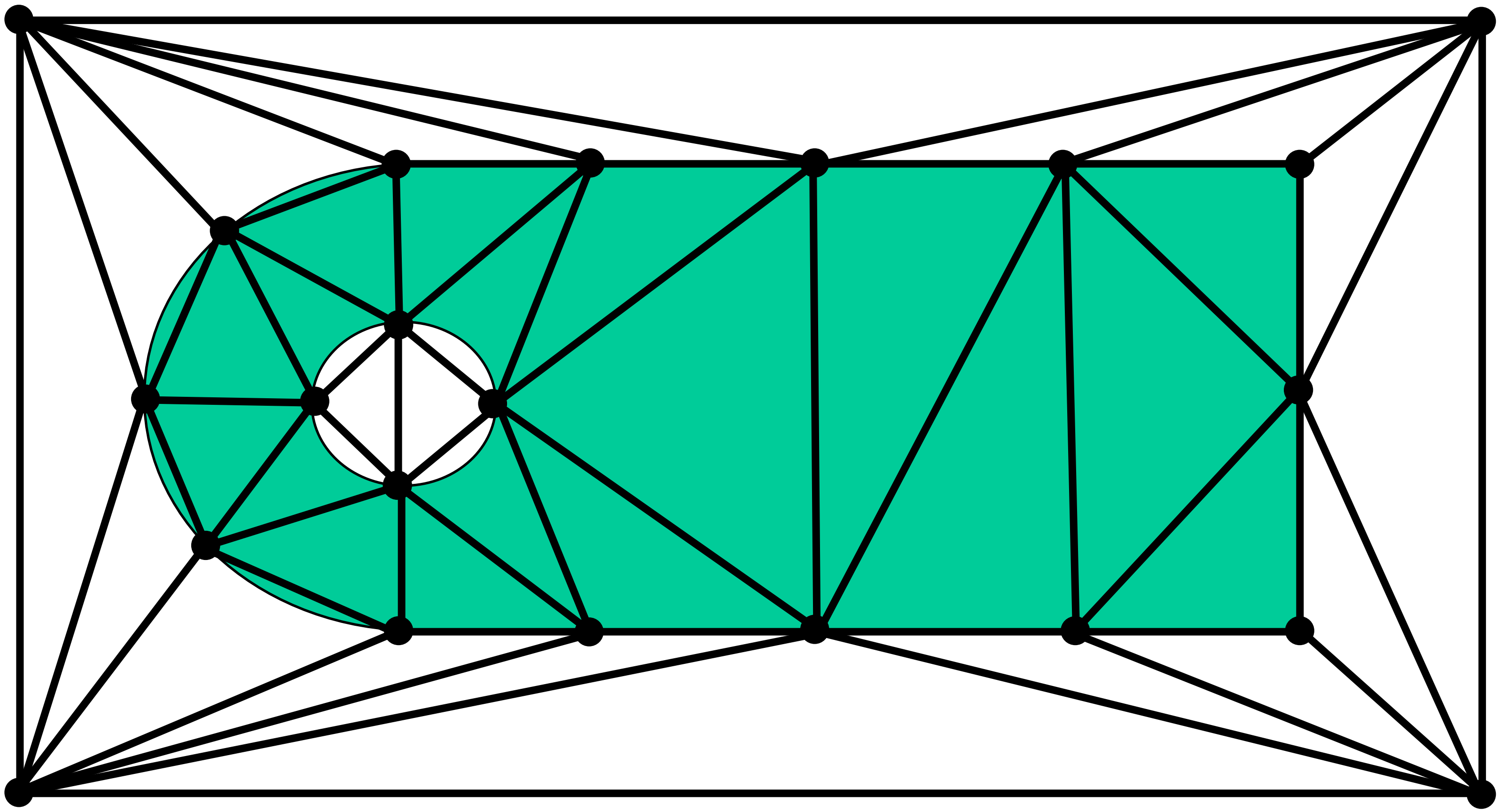
Edge node are created using Delaunay triangulation (by the Lawson or the Bowyer-Watson algorithm)

Delaunay triangulation



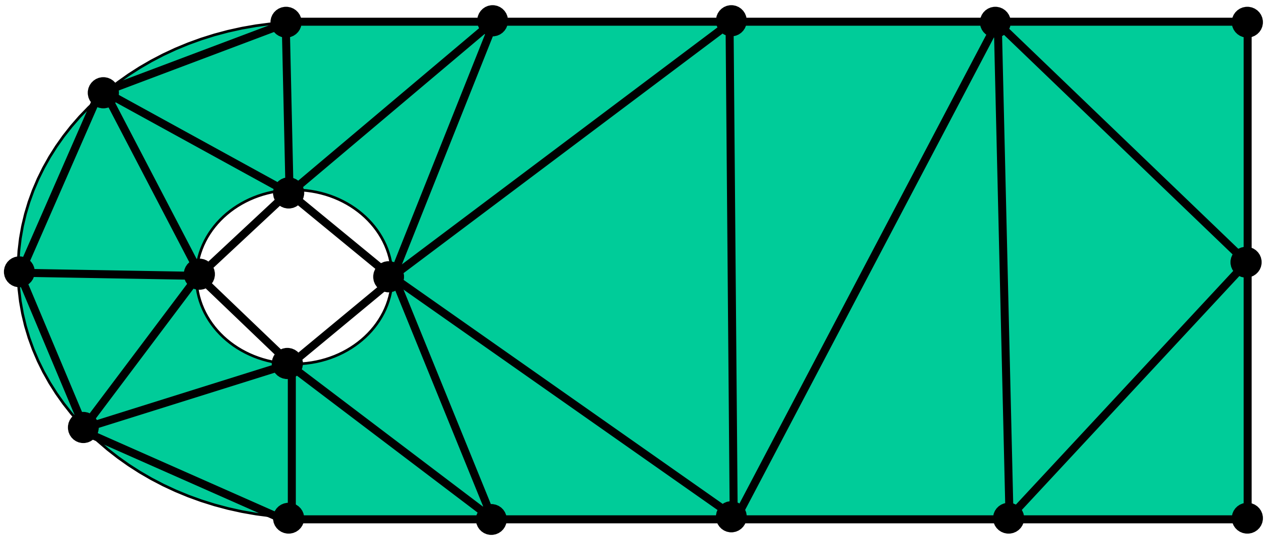
Edge node are created using Delaunay triangulation (by the Laowson or the Bowyer-Watson algorithm)

Delaunay triangulation



Edge node are created using Delaunay triangulation (by the Lawson or the Bowyer-Watson algorithm)

Delaunay triangulation

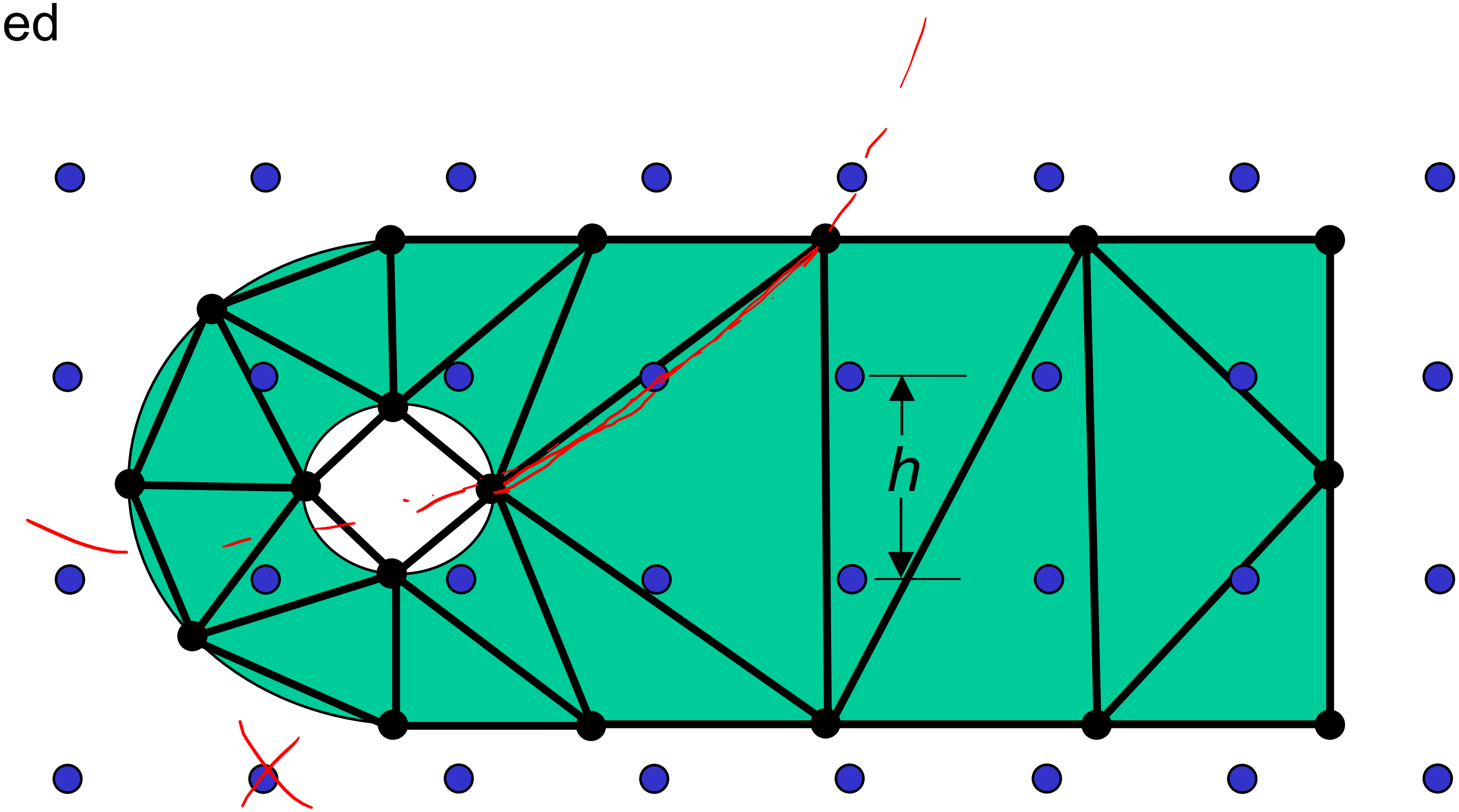


The contour is highlighted

Exterior triangles are removed

Delaunay triangulation

So far, only edge nodes have been defined. Now the interior nodes are added

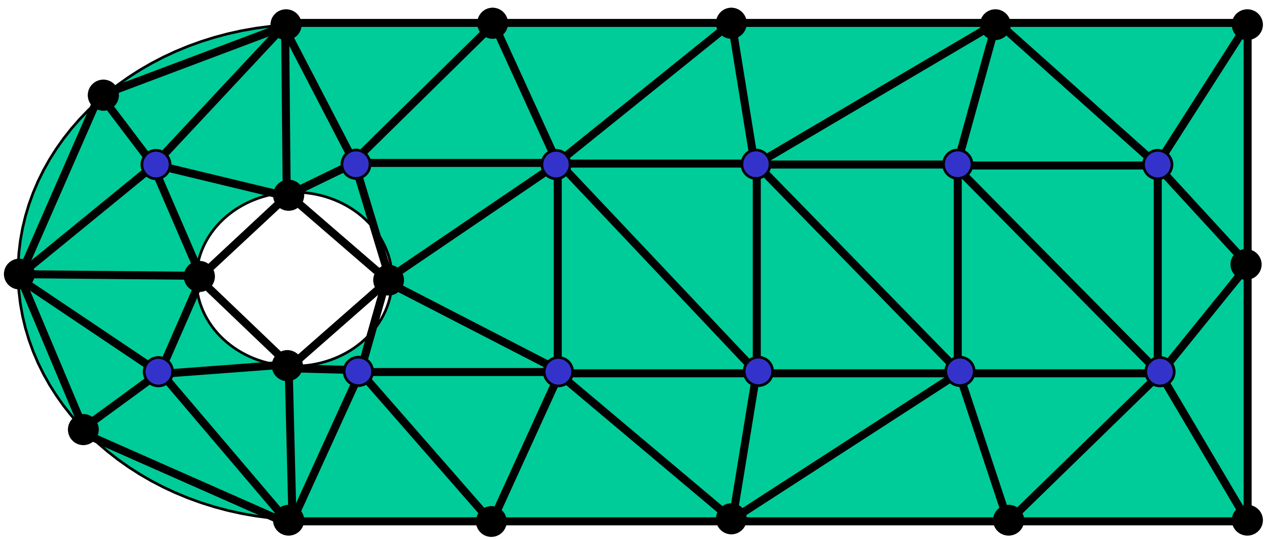


Grid method:

- A regular pattern of nodes is created
- The pattern can be rectangular, triangular, quadtree,
- Nodes outside the geometric domain are ignored

Delaunay triangulation

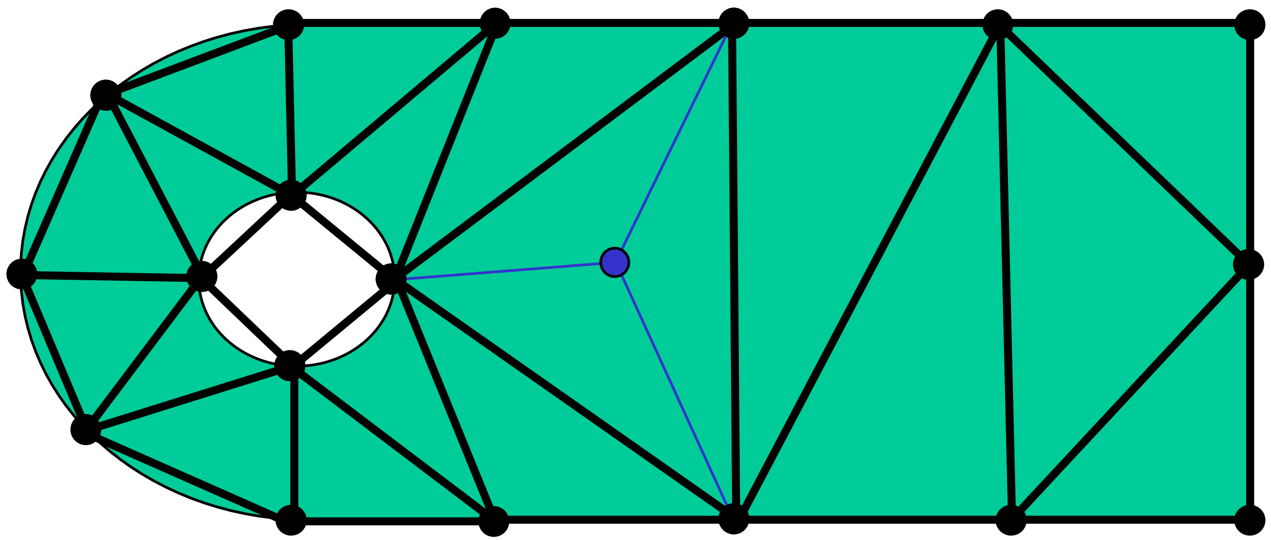
So far, only edge nodes have been defined. Now the interior nodes are added



Grid method:

- A regular pattern of nodes is created
- The pattern can be rectangular, triangular, quadtree,
- Nodes outside the geometric domain are ignored

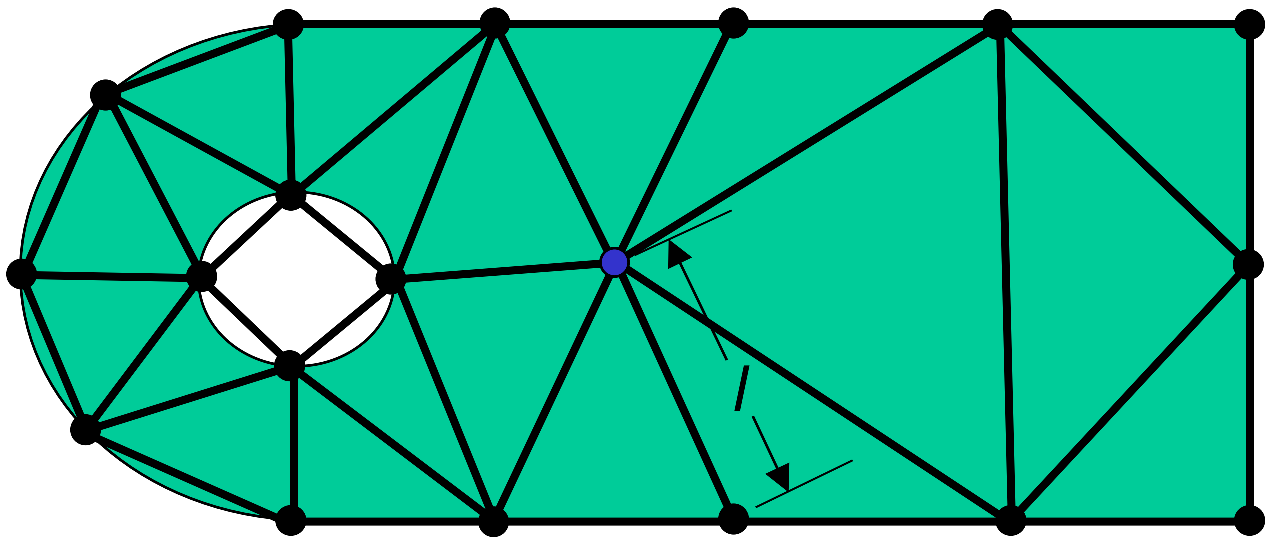
Delaunay triangulation



Centroid method

- Nodes are created at the centroid of each triangle
- The process continues iteratively until edge elements' length reaches the desired length ($l \approx h$)

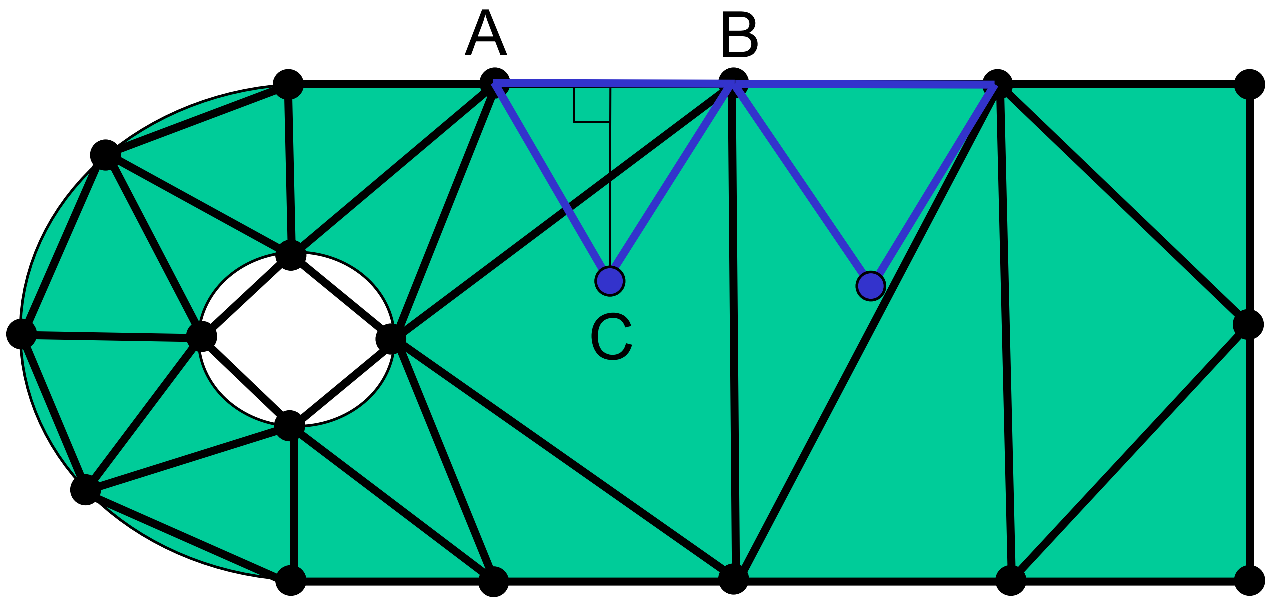
Delaunay triangulation



Centroid method

- Nodes are created at the centroid of each triangle
- The process continues iteratively until edge elements' length reaches the desired length (l ≈ h)

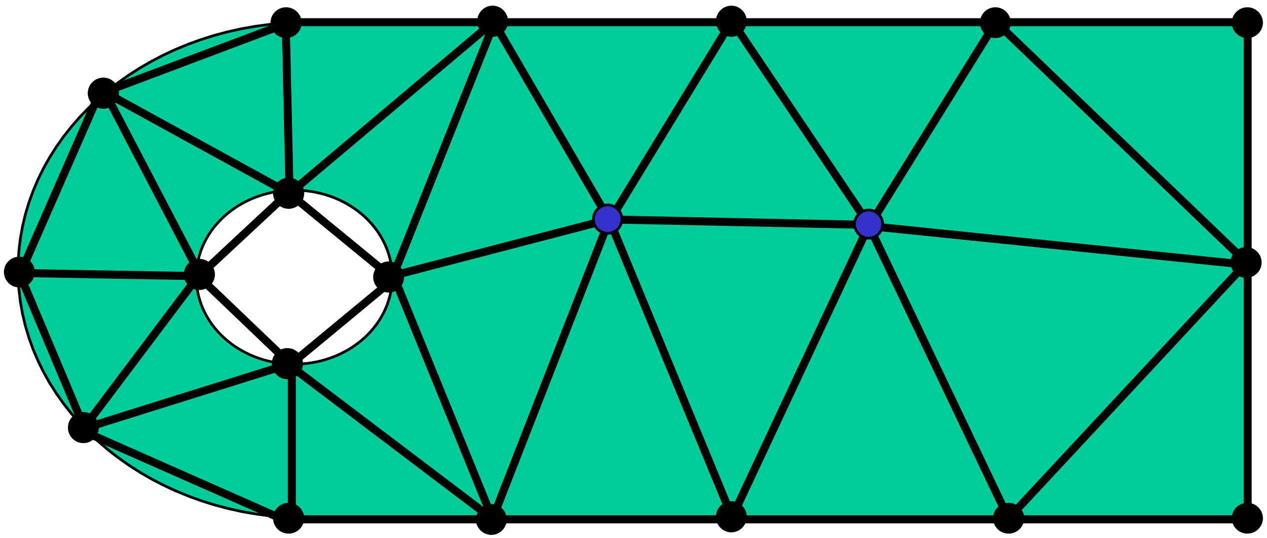
Delaunay triangulation



Advancing front method

Node are inserted in a consistent manner from the exterior edges

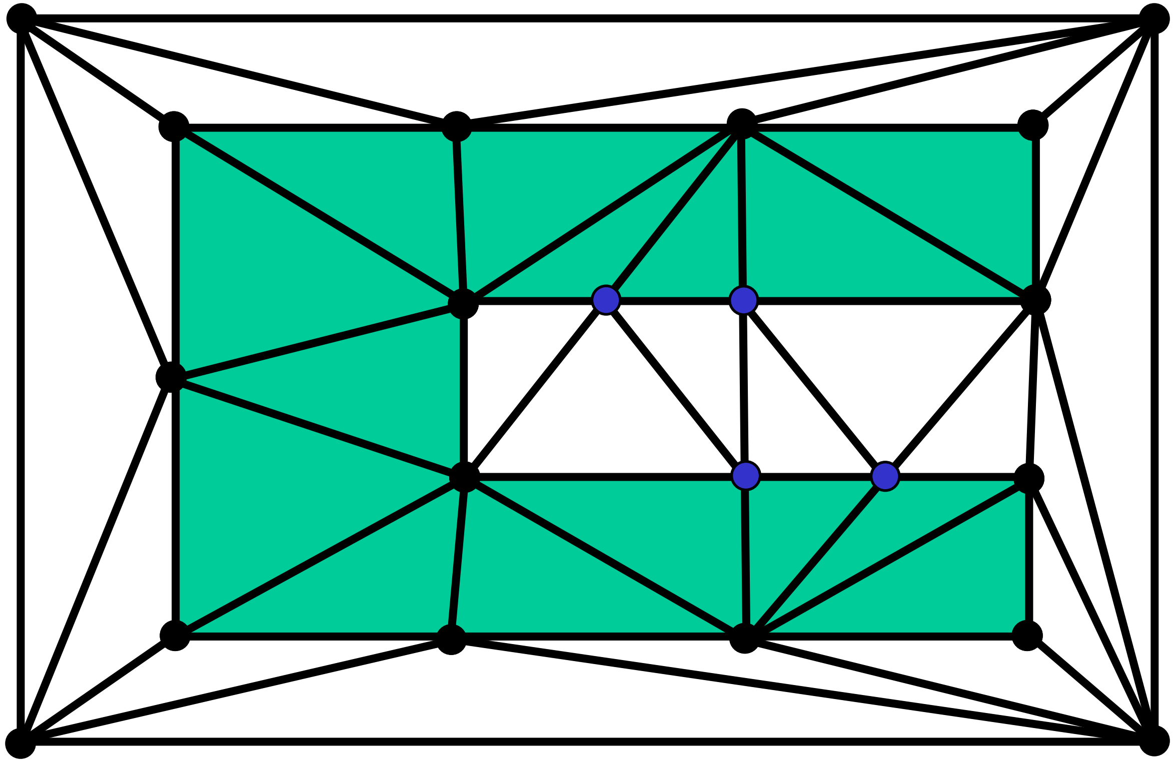
Delaunay triangulation



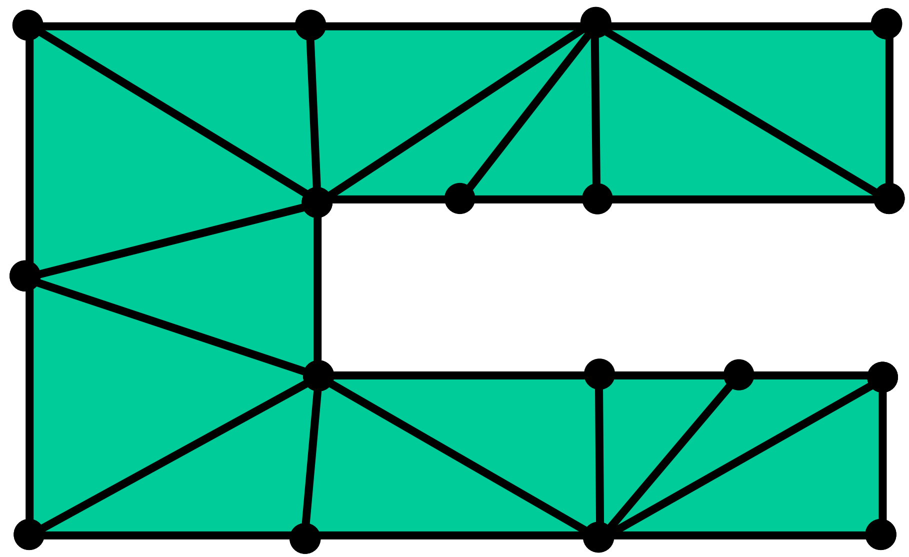
Advancing front method

Node are inserted in a consistent manner from the exterior edges

Delaunay triangulation: interior edges



Delaunay triangulation: interior edges



Delaunay triangulation

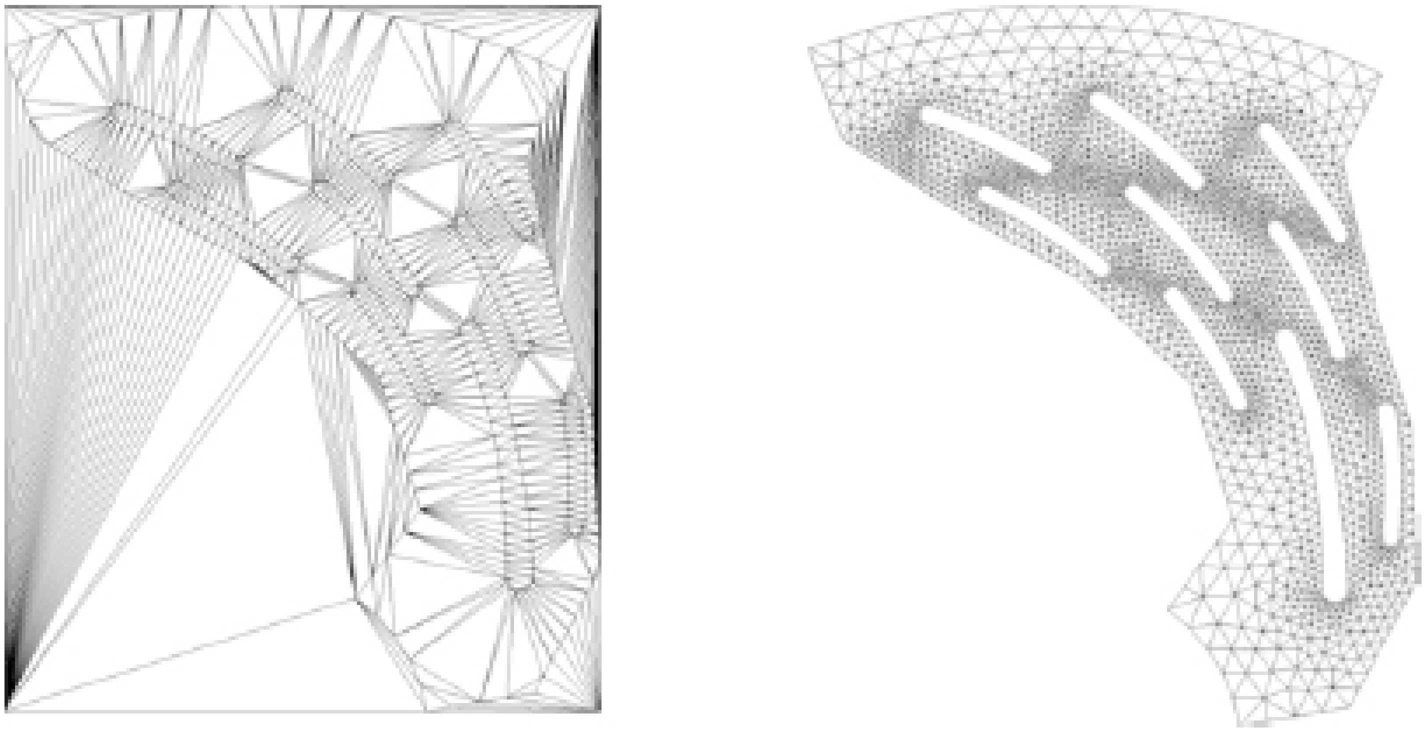


Figure 3.8: *Delaunay mesh of a mechanical device. Boundary mesh with no internal vertex (left-hand side) and resulting mesh after optimization (right-hand side).*

Comparison of methods

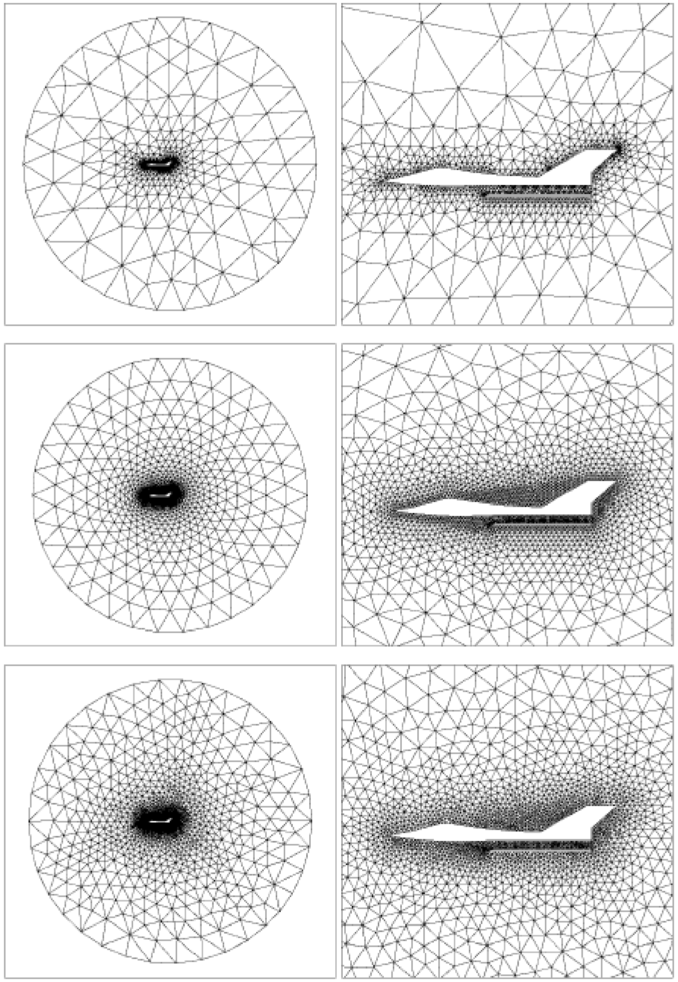


Figure 3.9: *Overview of the different mesh generation methods when applied in a domain (used, for instance, for a CFD problem). Quadtree type mesh (top), advancing-front type mesh (middle) and Delaunay type mesh (bottom) including a close-up view around the fuselage.*

method	np	ne	Q_M	Q_{worst}
quadtree	1,246	2,171	1.25	1.88
advancing-front	2,557	4,795	1.1	1.61
Delaunay	2,782	5,528	1.16	1.82

np, number of nodes

ne, number of elements

Q_M: shape quality of elements (best are close to 1)

Q_{worst}: quality of the worst element

Major differences in terms of mesh size

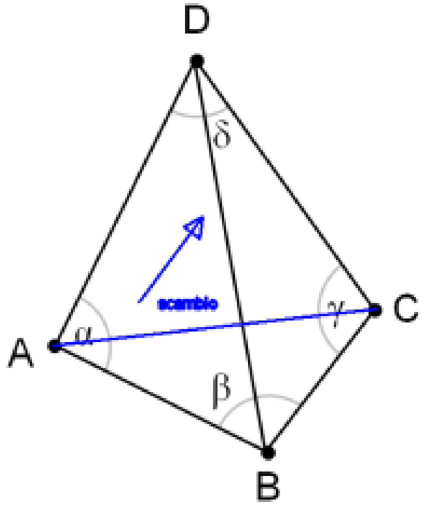
No significative differences in terms of mesh quality

Regularization techniques

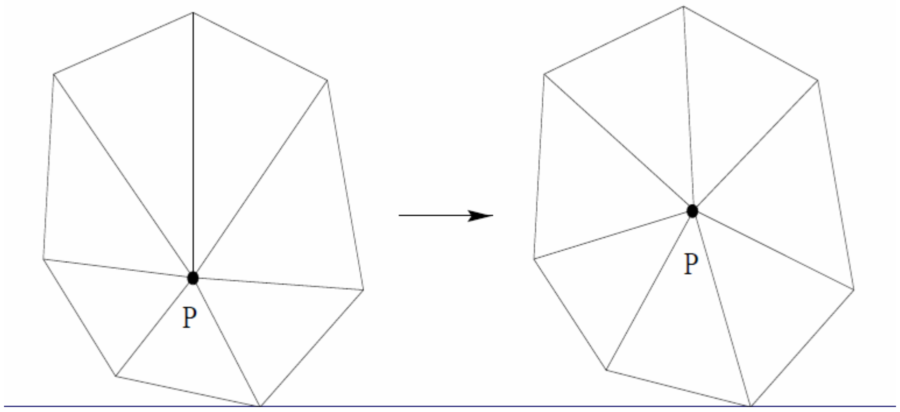
After generating the mesh, regularization/optimization is performed to improve the mesh quality

They are iterative techniques based on two approaches:

Swapping diagonals, to maximize the minimum angle (max-min optimization)



Local displacement of nodes, to move them closer to the centroid of selected patches of elements



Regularization techniques

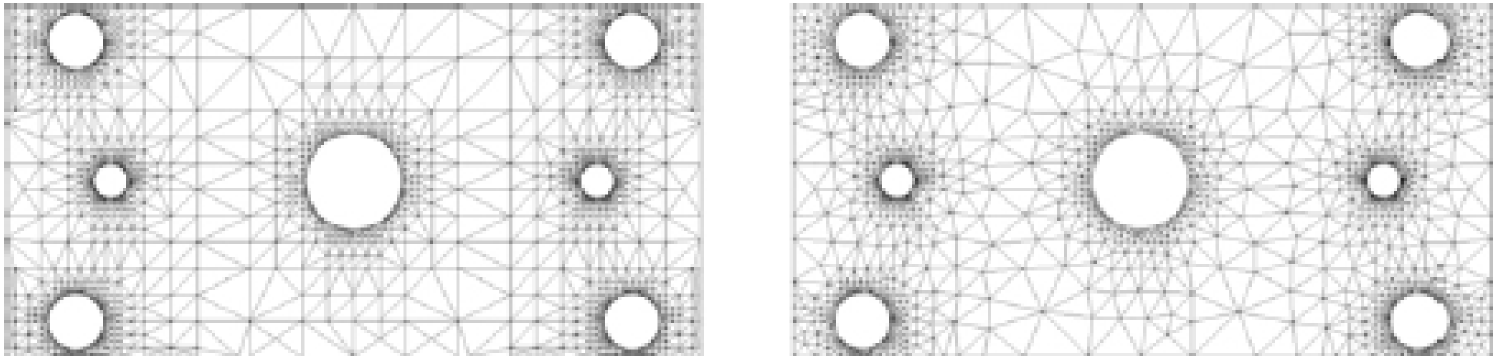
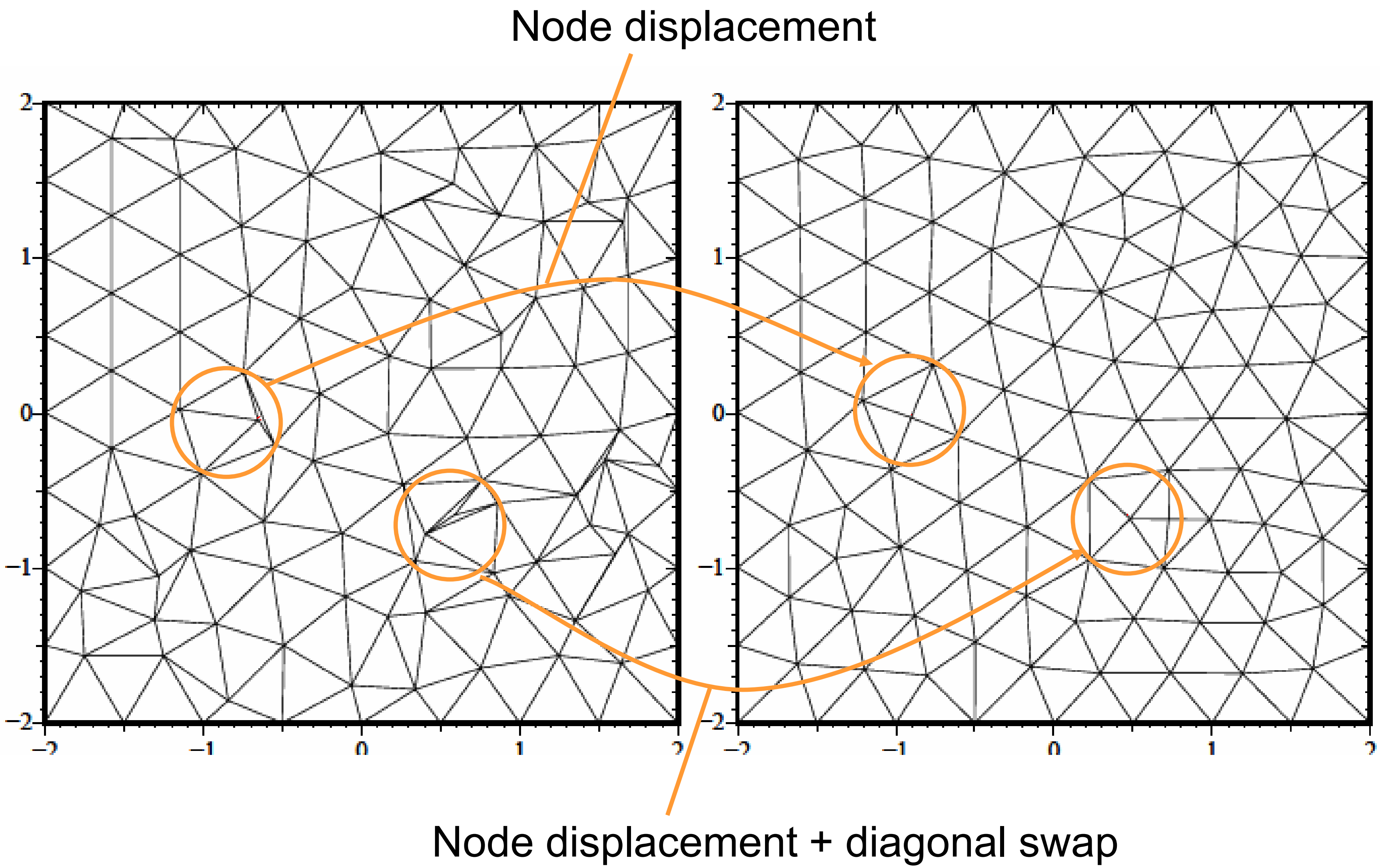


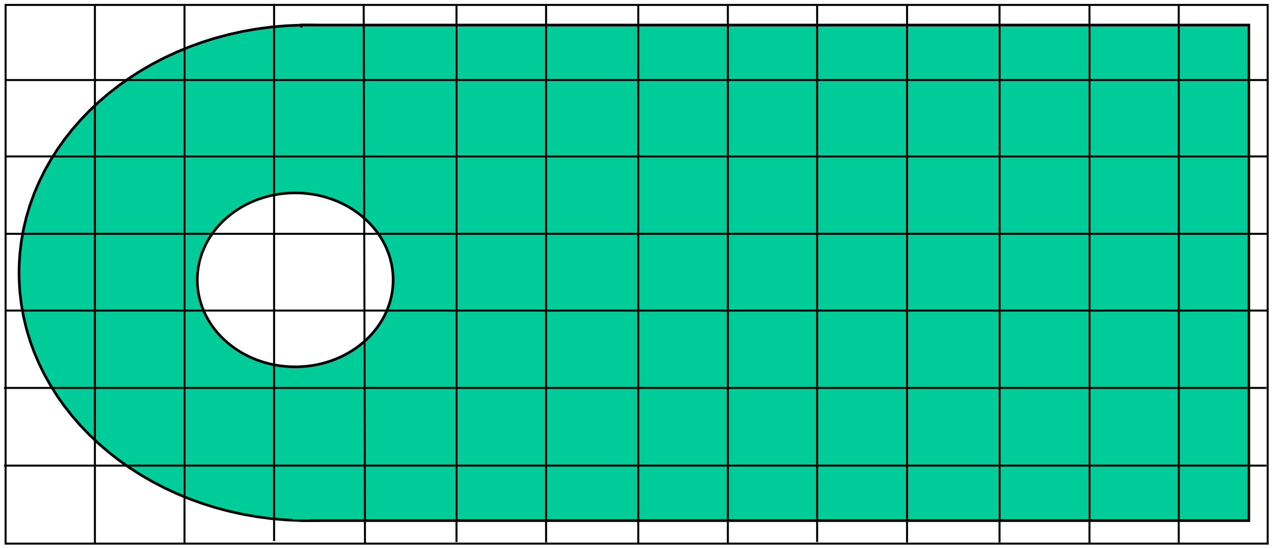
Figure 5.15: *Two dimensional mesh optimization. The initial raw quadtree mesh (left-hand side) and the final mesh after optimization (right-hand side).*

Regularization techniques



Direct and indirect methods

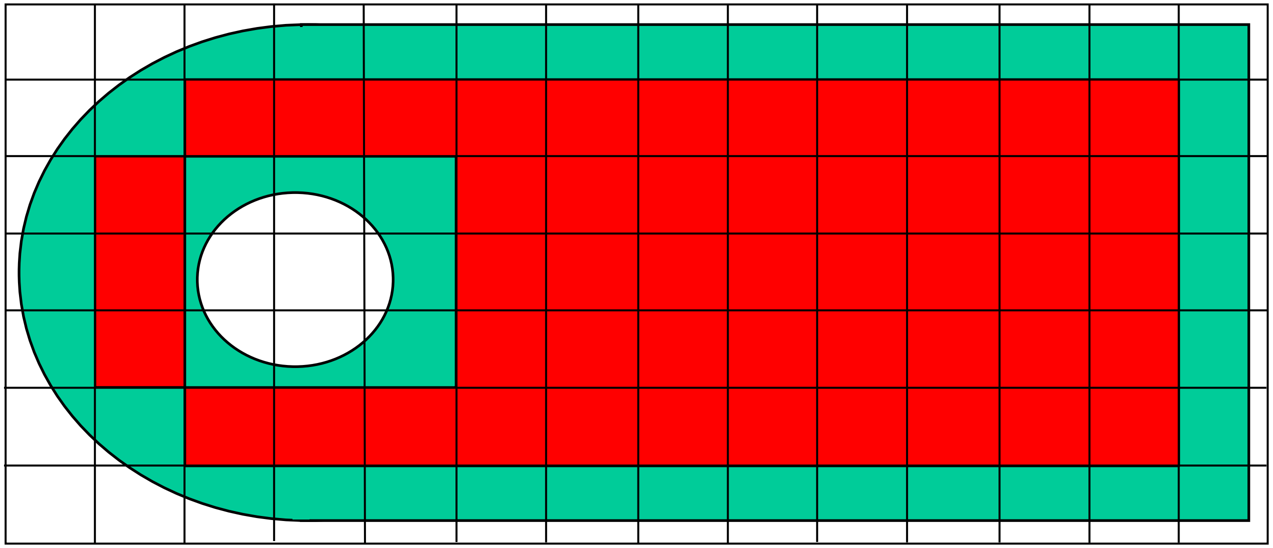
Non structured mesh methods based on quadrangular elements



Direct method: grid based

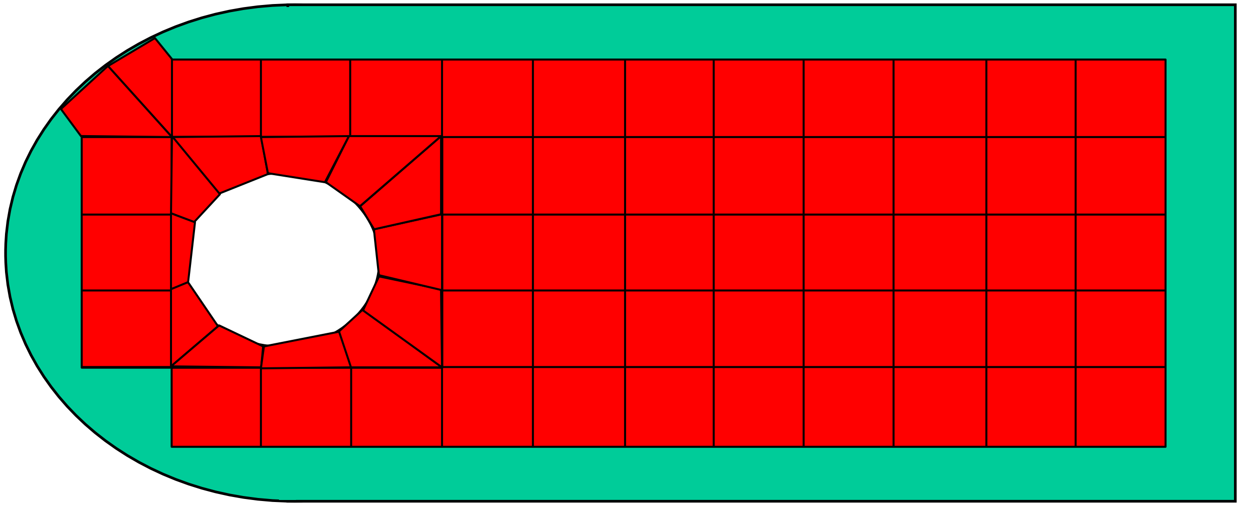
A grid of quadrangular elements (prisms in 3D) is superimposed to the model

Direct and indirect methods



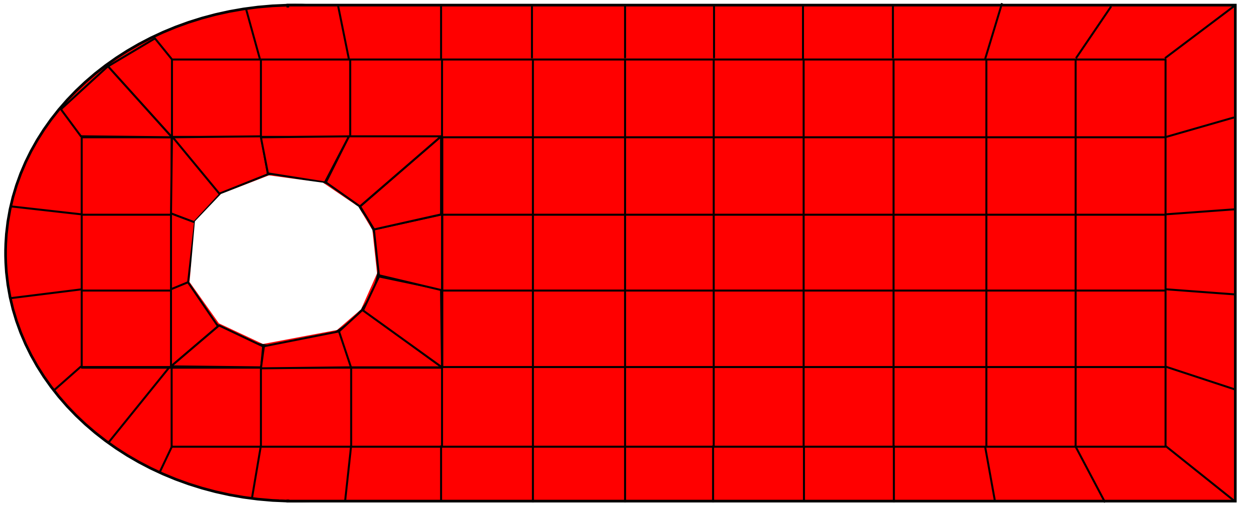
Interior elements are preserved

Direct and indirect methods



Edges of the interior elements are projected onto the contour edges

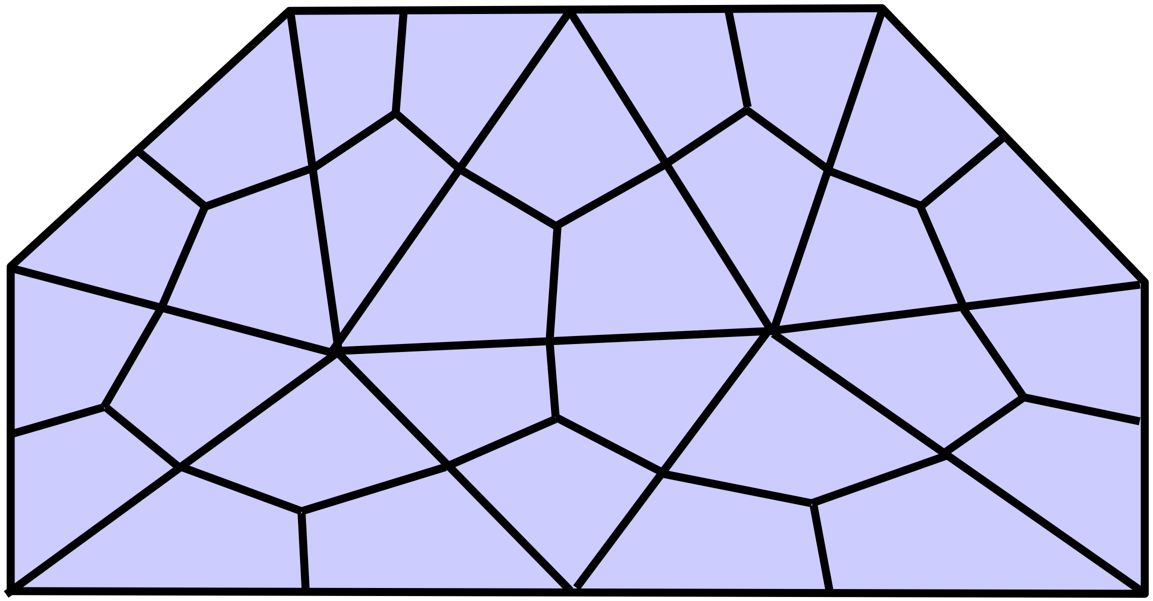
Direct and indirect methods



The quality of the contour elements may be poor and could be non conformal

Direct and indirect methods

Triangle and Tetrahedra splitting

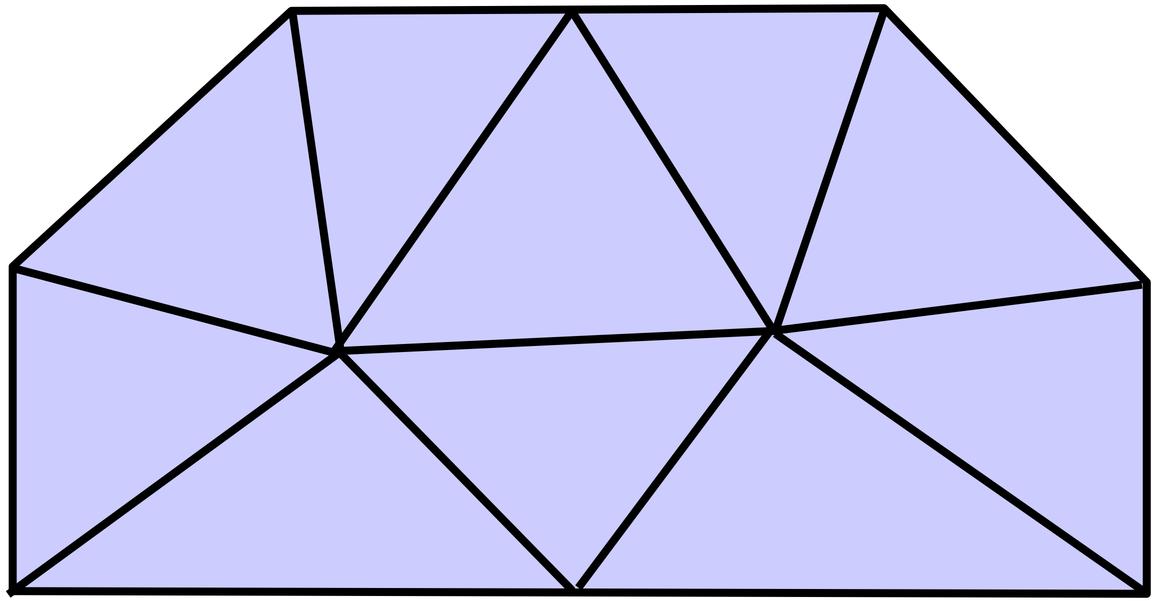


Every triangle is split into three quadrangular partitions

The mesh quality is often poor

Direct and indirect methods

Triangle or tetrahedra merging



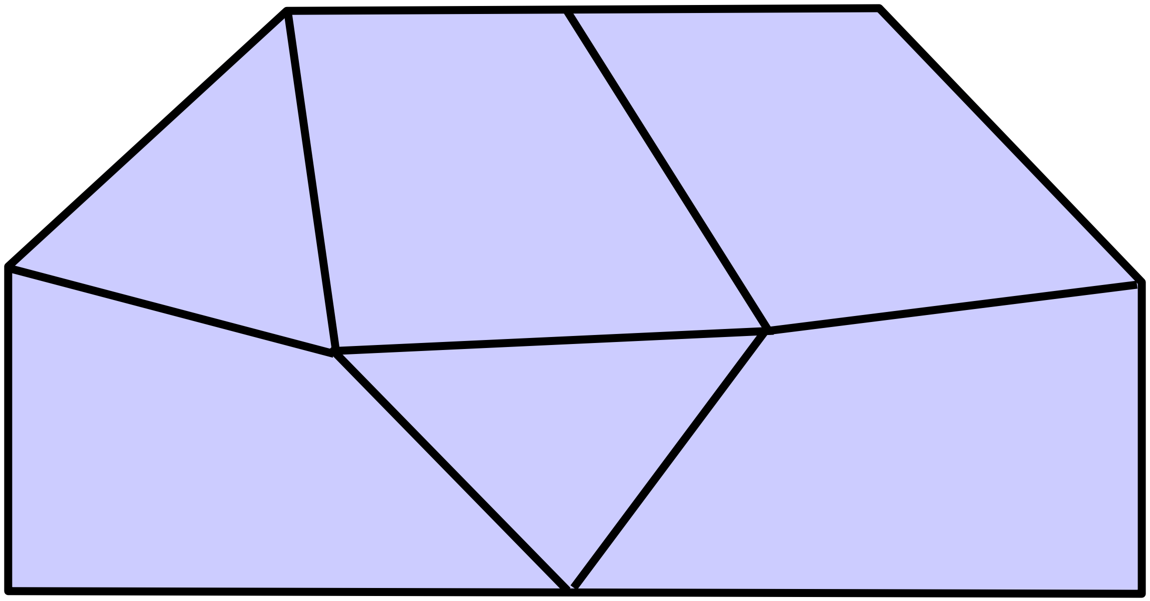
Neighboring triangles are merged, to generate a quadrangular element

The best combinations are searched for

Some triangles may remain

Direct and indirect methods

Triangle or tetrahedra merging



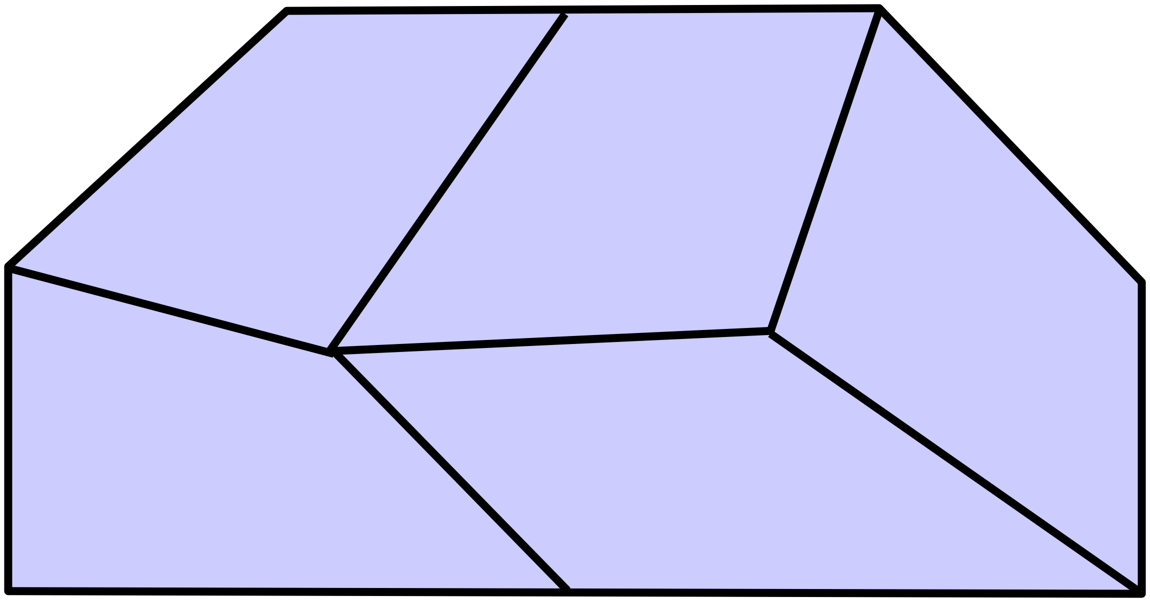
Neighboring triangles are merged, to generate a quadrangular element

The best combinations are searched for

Some triangles may remain

Direct and indirect methods

Triangle or tetrahedra merging



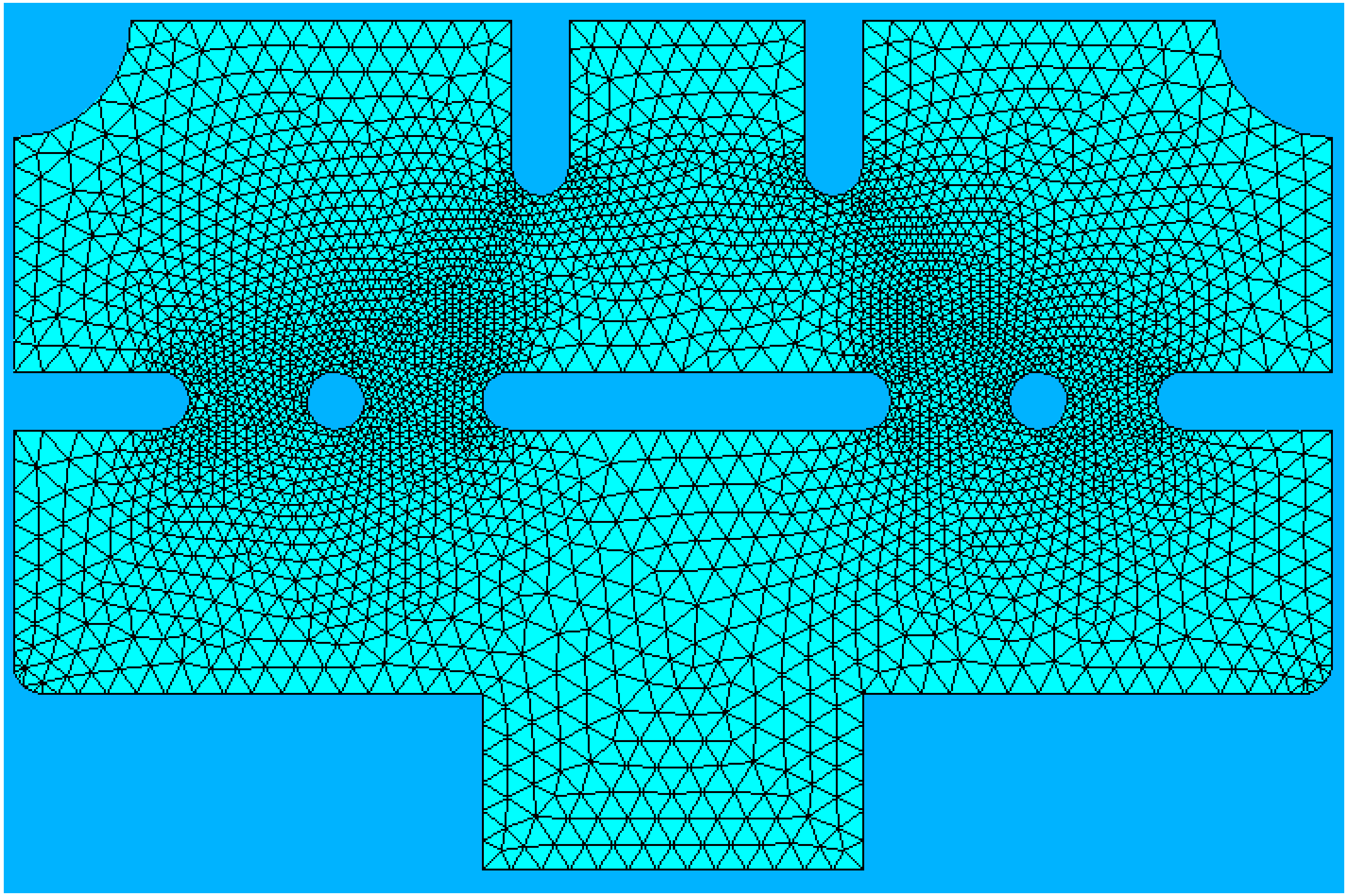
Neighboring triangles are merged, to generate a quadrangular element

The best combinations are searched for

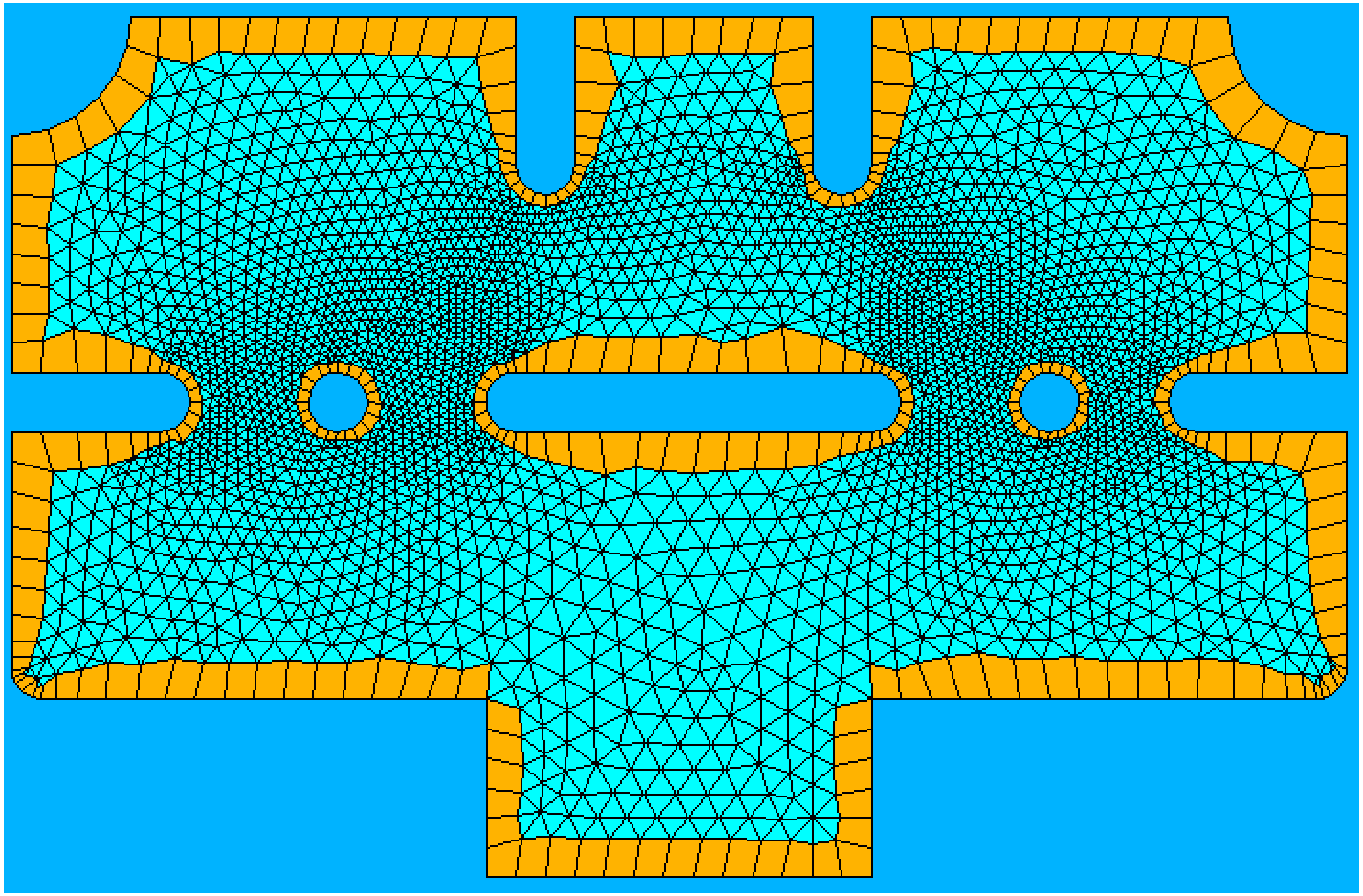
Some triangles may remain

Direct and indirect methods

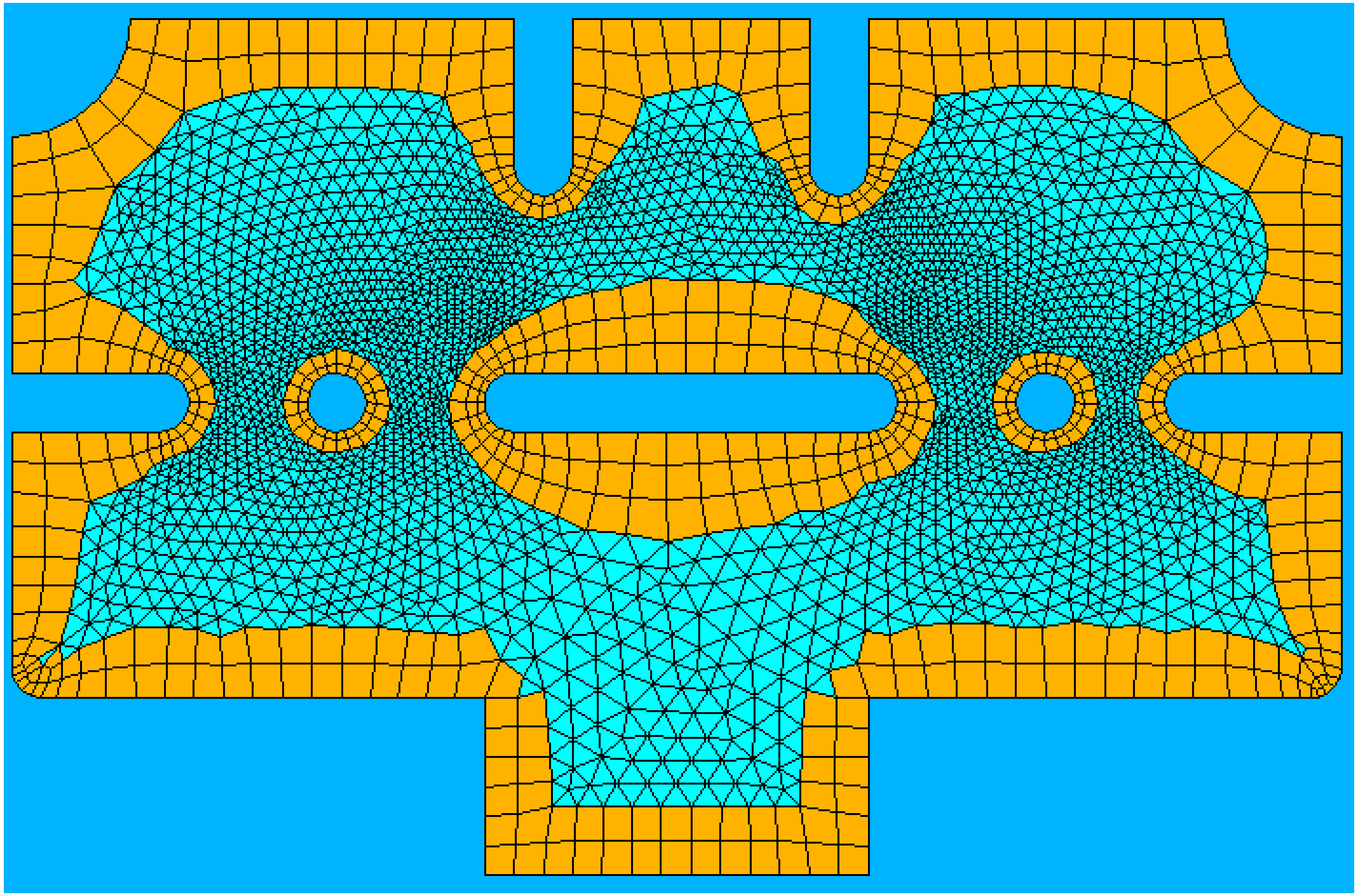
Q-Morph belongs to the class of the merging algorithms



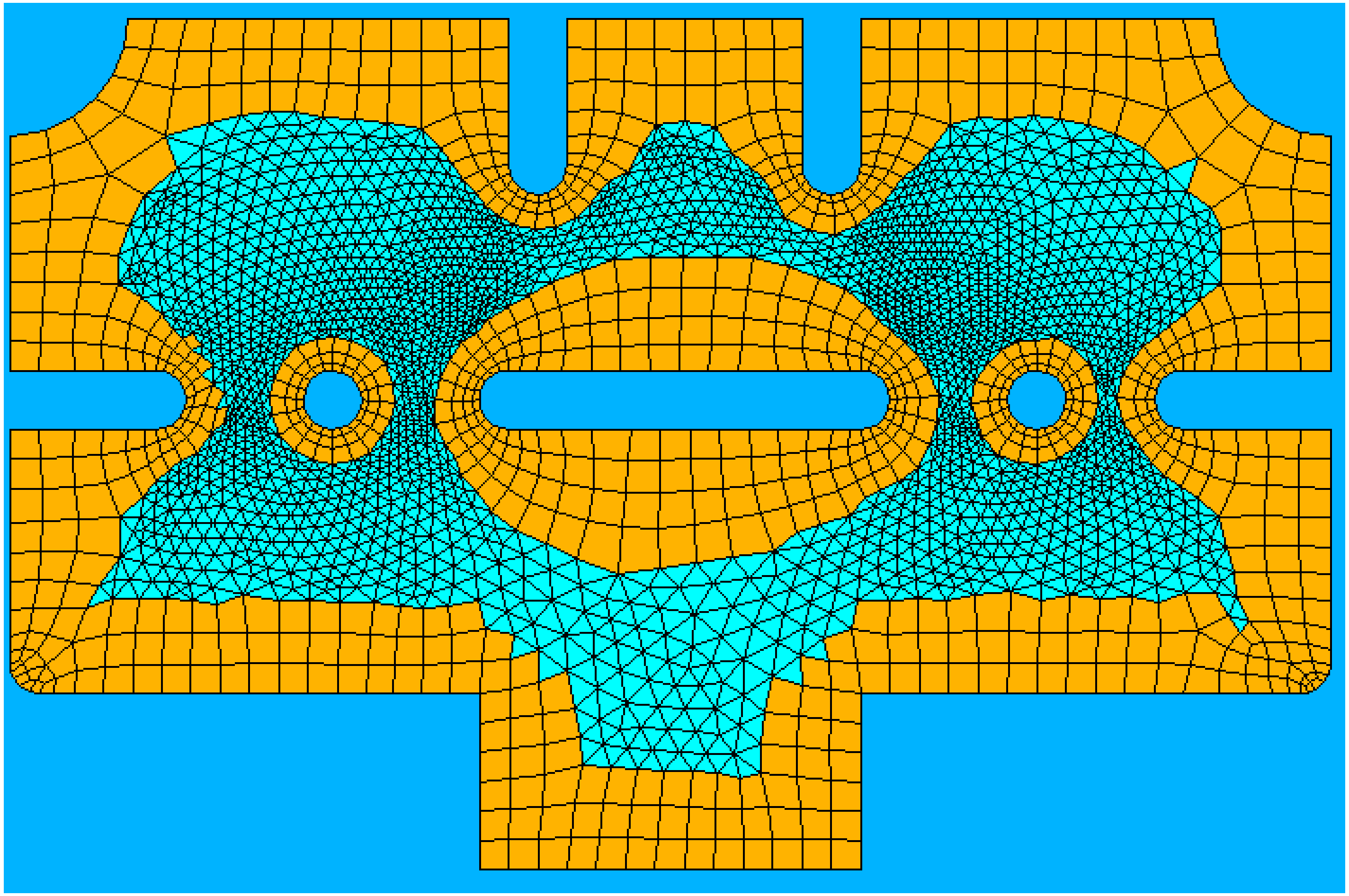
Direct and indirect methods



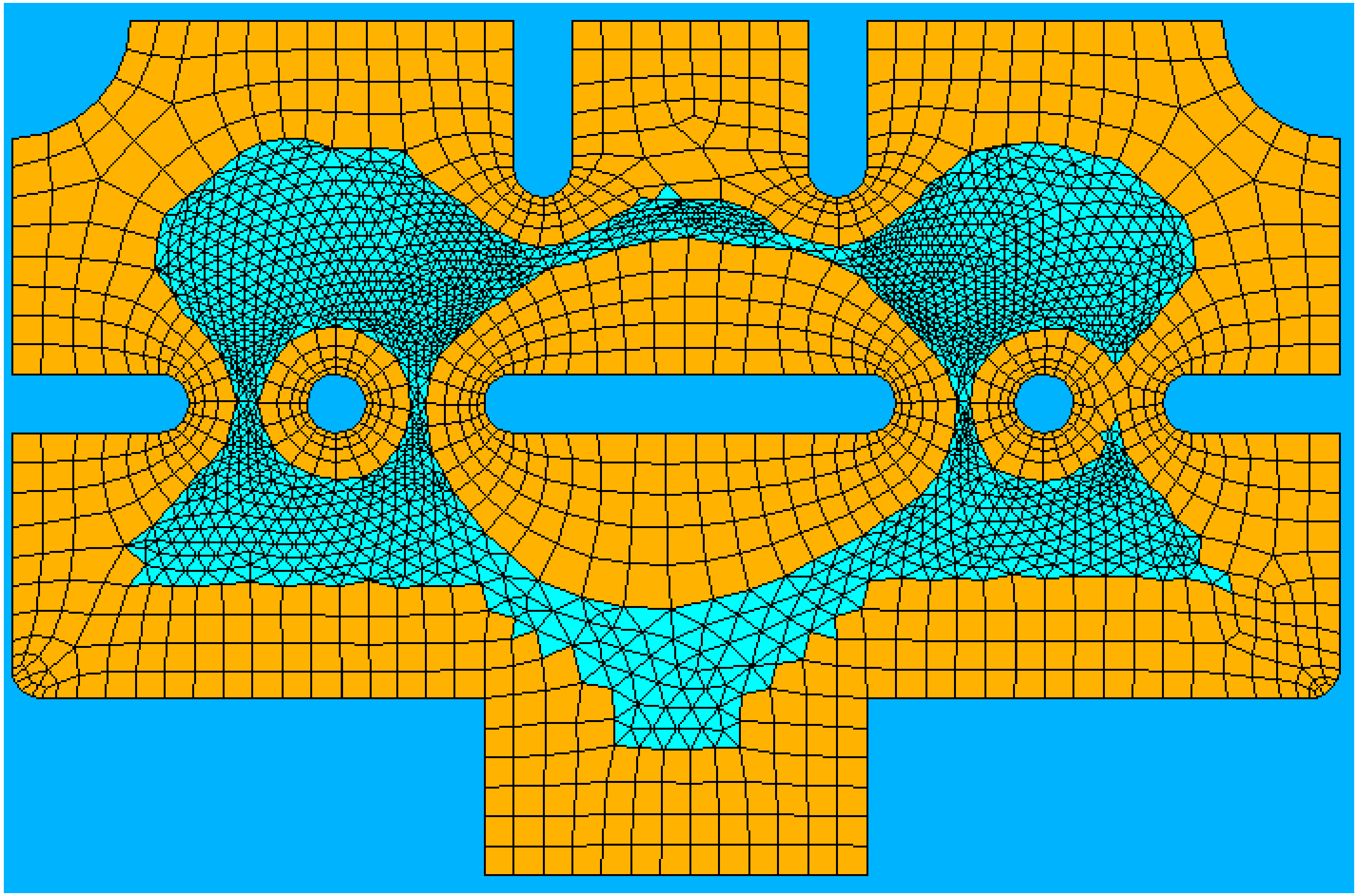
Direct and indirect methods



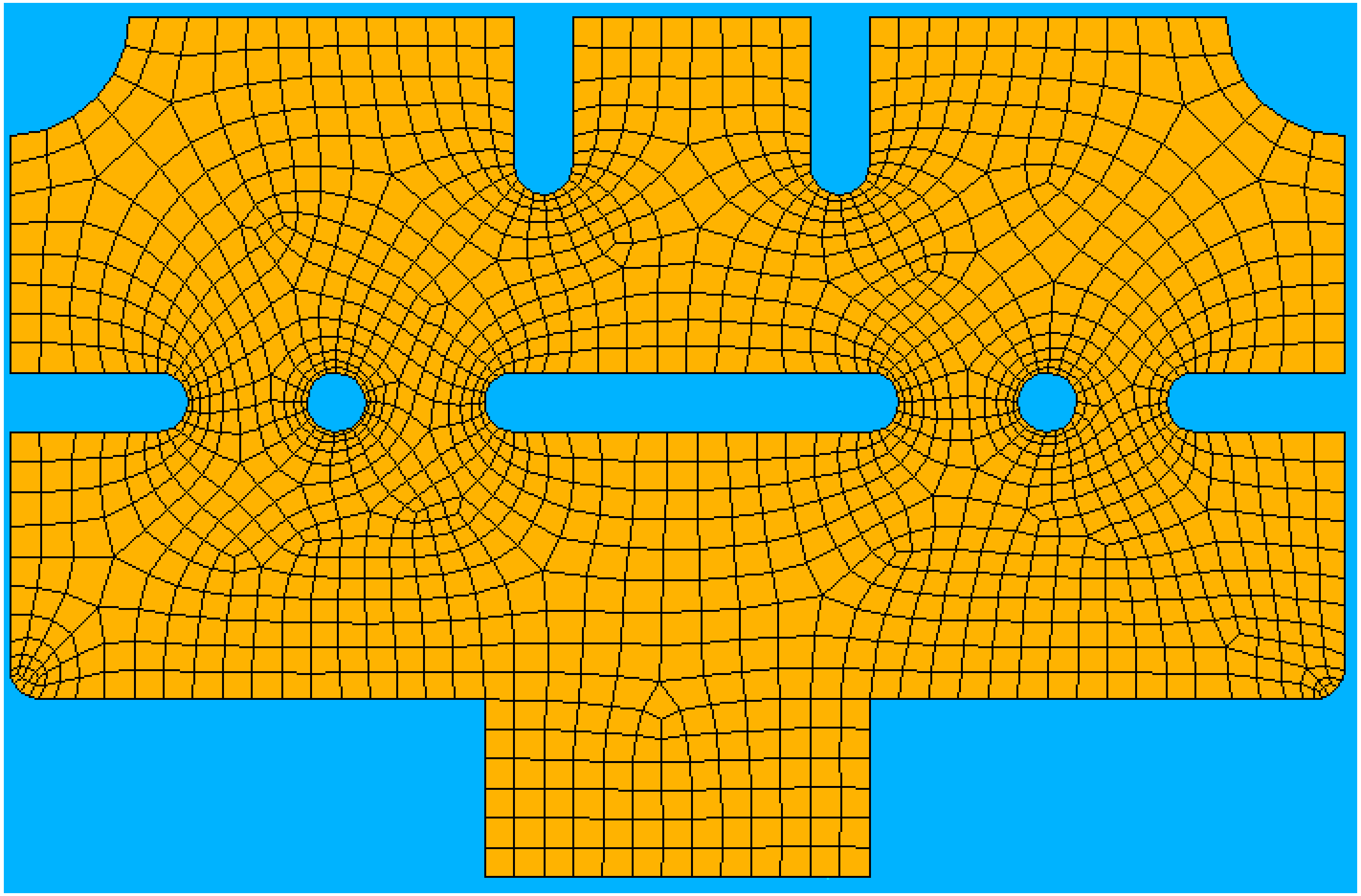
Direct and indirect methods



Direct and indirect methods



Direct and indirect methods



Algorithms for structured meshing

A procedure of structured meshing consist of mapping onto a physical domain the mesh defined on simple reference geometries.

It is similar to the isoparametric transformation, but in this case, it is applied to the entire region that needs to be meshed, instead of a single element.

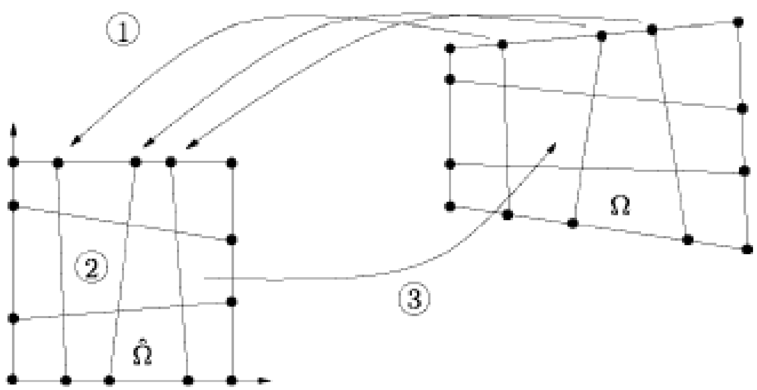
Two classes of methods exist:

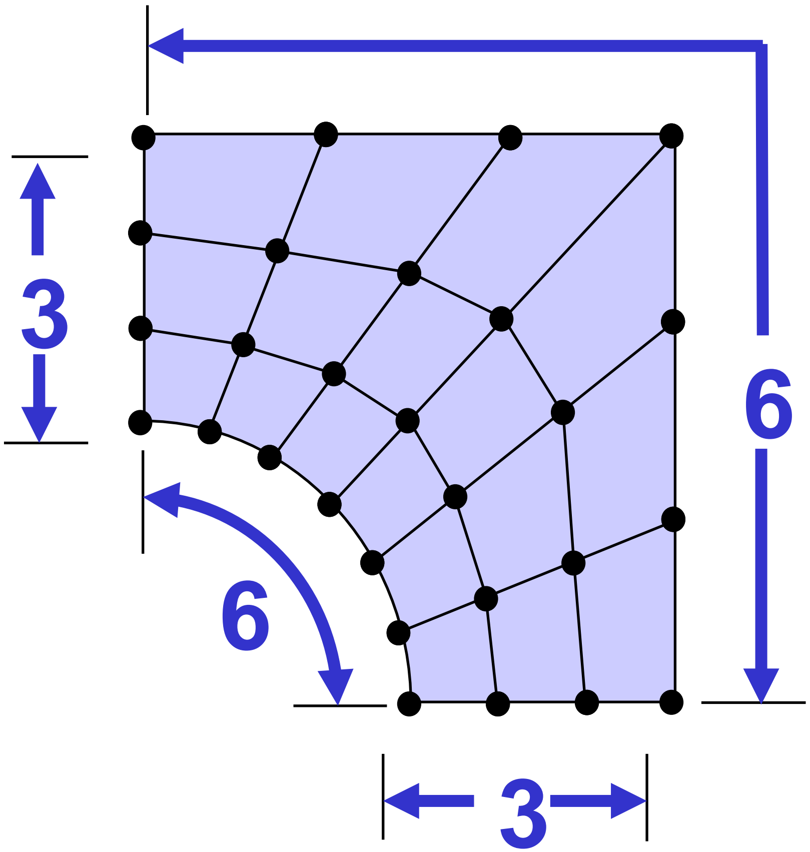
- Methods based on algebraic interpolation: after a mesh has been defined in the natural space, a mapping function is used to apply it to the true geometry in the physical space
- Methods based on partial differential equations

Crucial points are: 1) the definition of an appropriate mapping function; 2) the partitioning techniques

Trans-Finite Interpolation algorithm

It is the algorithm that resembles most the isoparametric transformation. It is based on the one-to-one transformation of a square or equilateral triangle of unit edges (natural space) into any 2D domain simply connected (physical space)



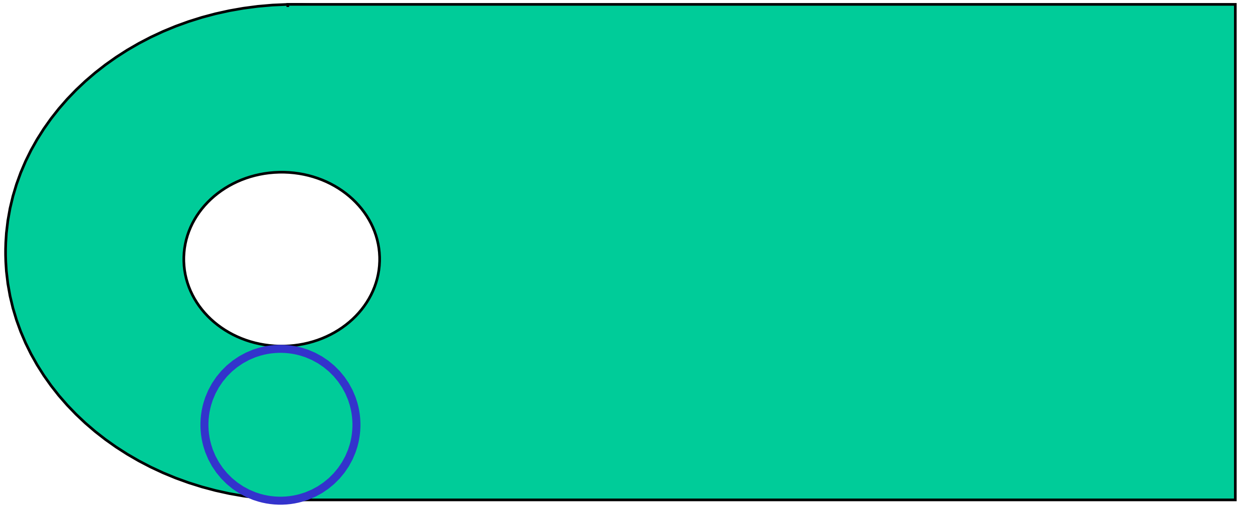


It consists of a topological distortion of the parent square or triangle into the physical shape

It works also in 3D

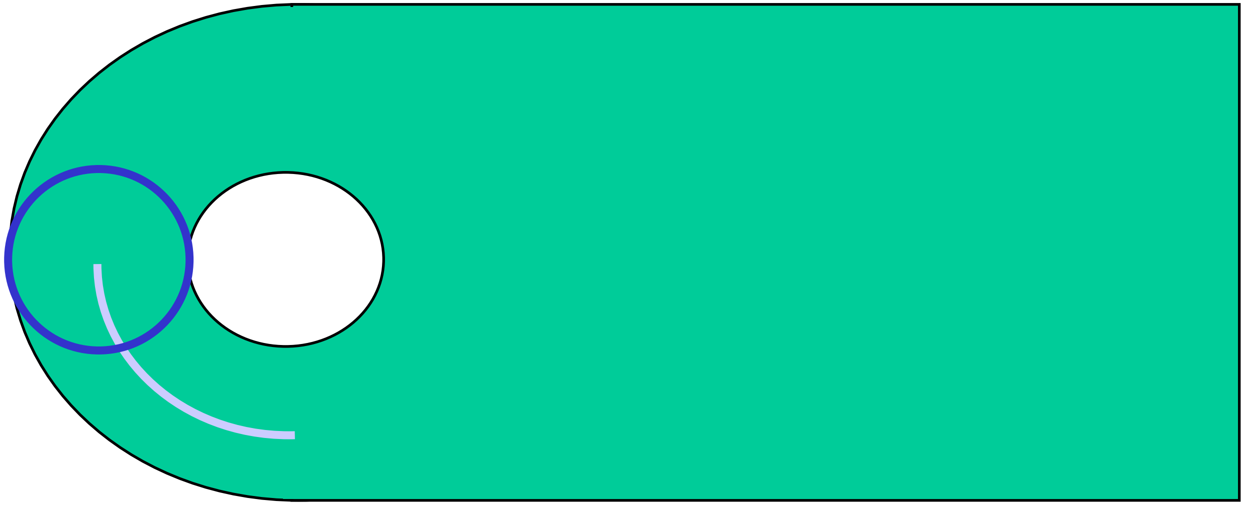
Medial axis algorithm

It is based on the concept of medial object: a circle (2D models) or a sphere (3D models) tangent to the contour is made to roll. The locus of the centres defines the medial object.



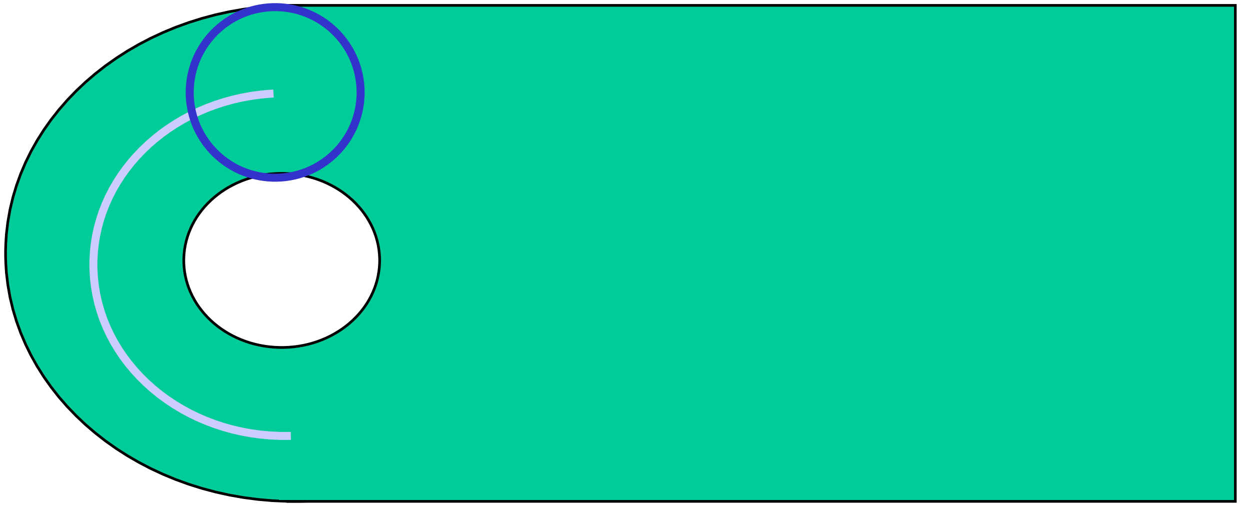
Medial axis algorithm

It is based on the concept of medial object: a circle or a sphere tangent to the contour is made to roll. The locus of the centres defines the medial object



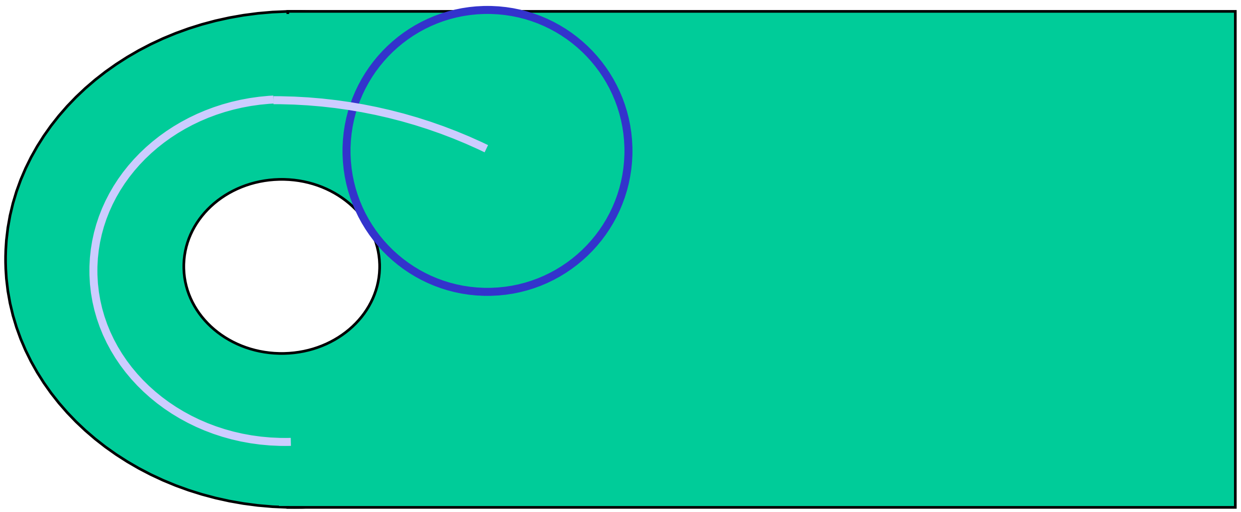
Medial axis algorithm

It is based on the concept of medial object: a circle or a sphere tangent to the contour is made to roll. The locus of the centres defines the medial object



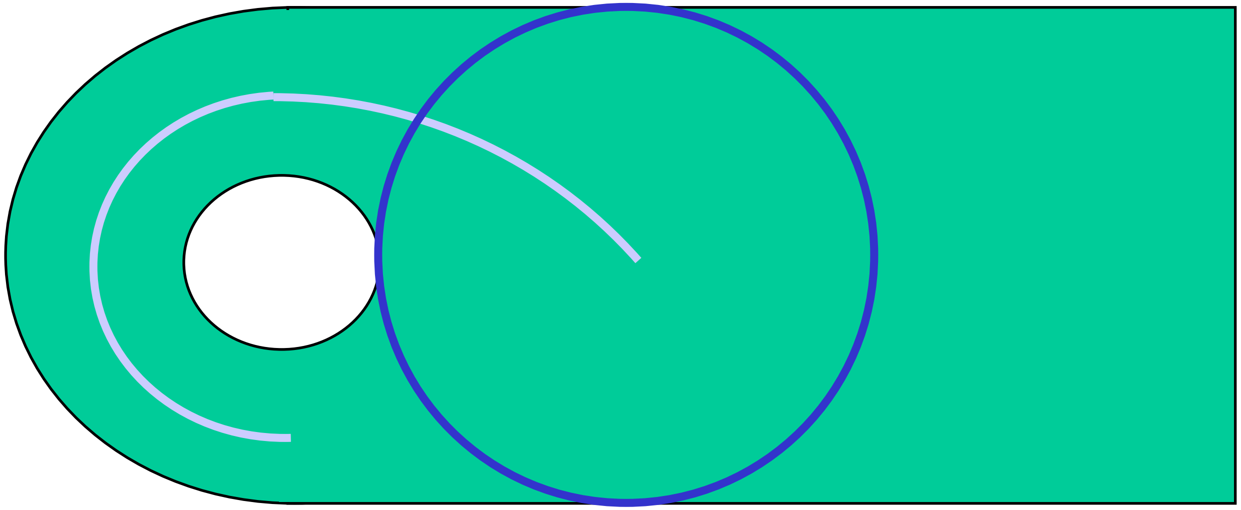
Medial axis algorithm

It is based on the concept of medial object: a circle or a sphere tangent to the contour is made to roll. The locus of the centres defines the medial object



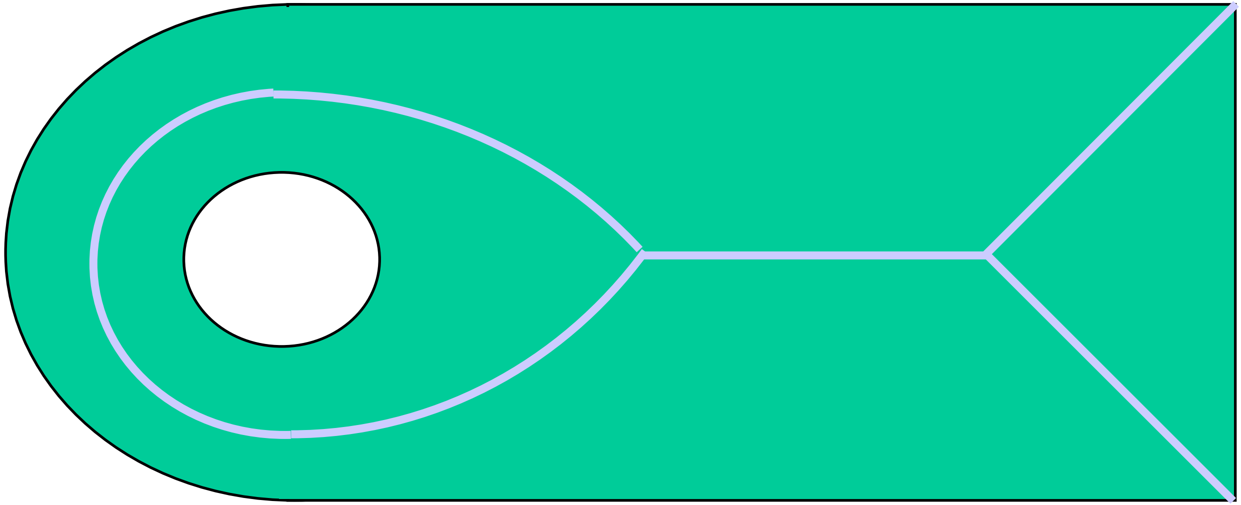
Medial axis algorithm

It is based on the concept of medial object: a circle or a sphere tangent to the contour is made to roll. The locus of the centres defines the medial object



Medial axis algorithm

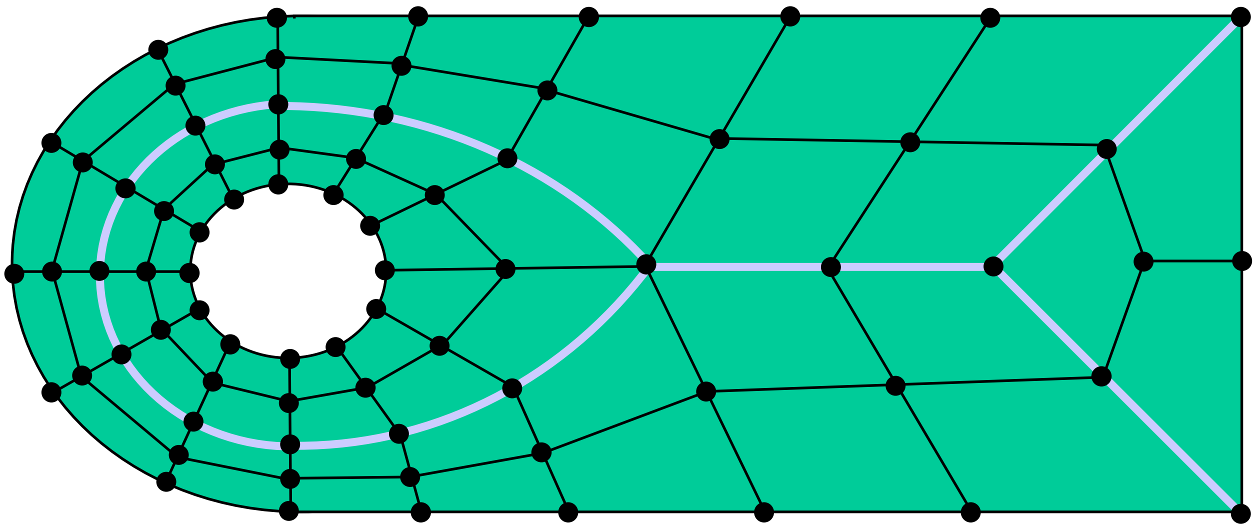
It is based on the concept of medial object: a circle or a sphere tangent to the contour is made to roll. The locus of the centres defines the medial object



The medial object is used to partition automatically the body into simpler regions that can be meshed by structured meshes.

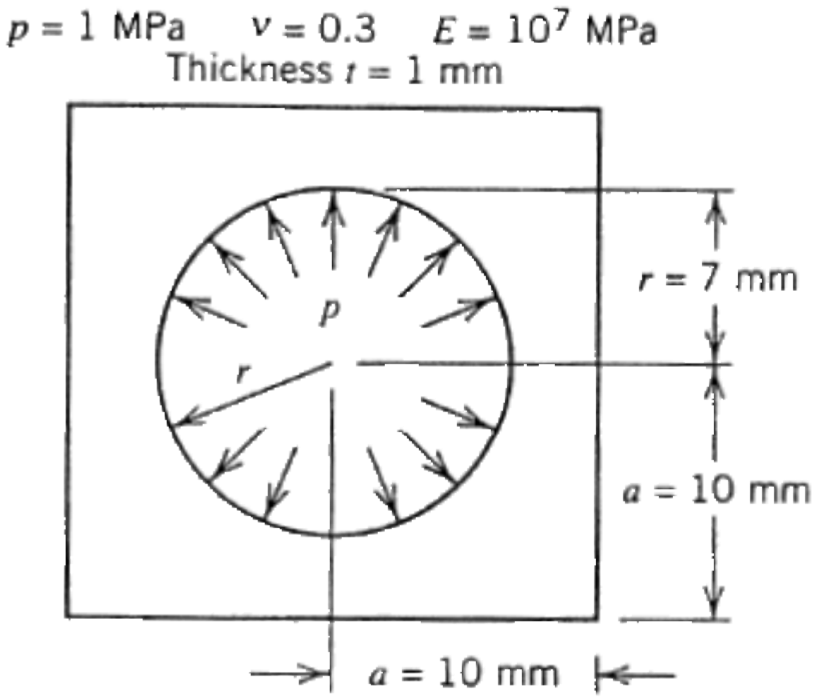
Medial axis algorithm

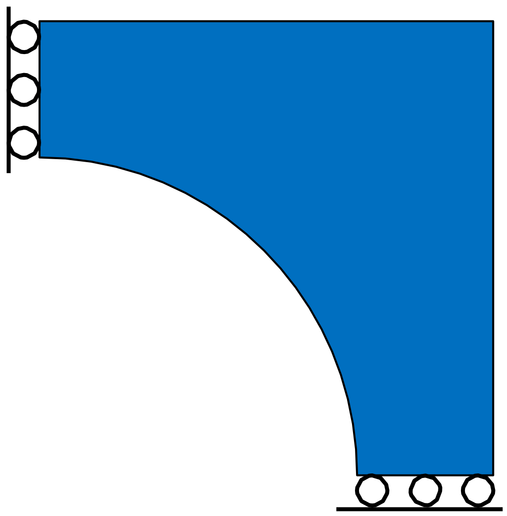
It is based on the concept of medial object: a circle or a sphere tangent to the contour is made to roll. The locus of the centres defines the medial object



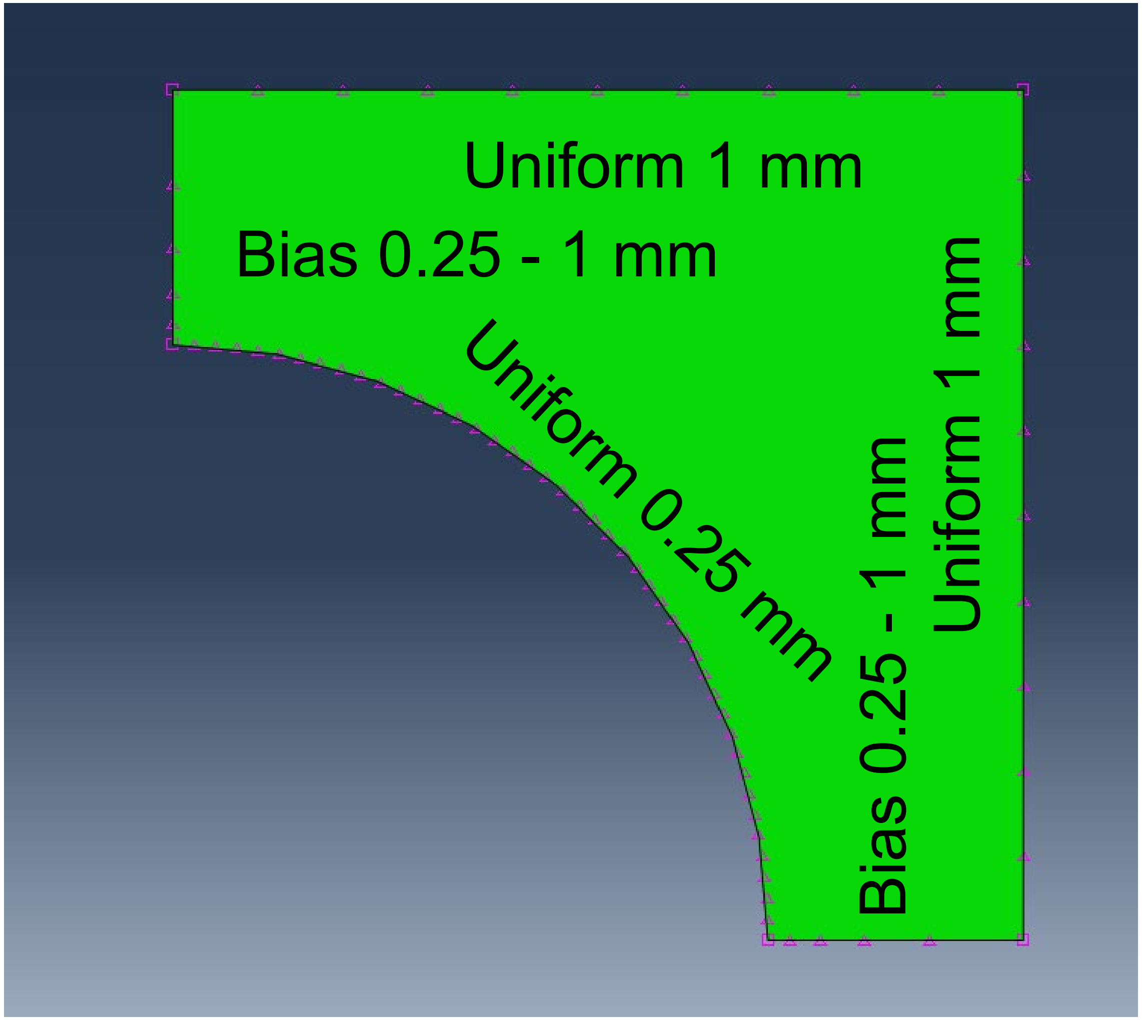
The medial object is used to partition automatically the body into simpler regions that can be meshed by structured meshes.

Example: plate with a hole

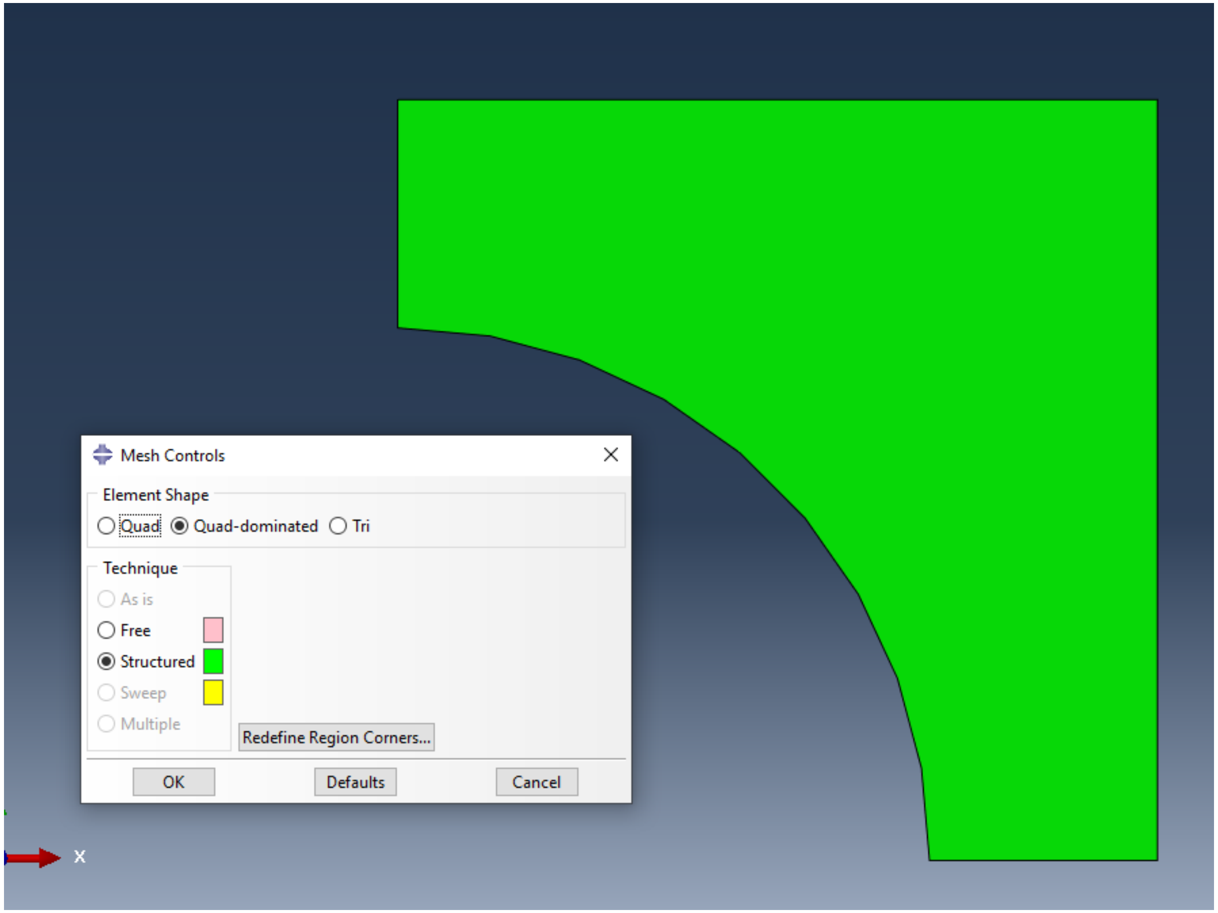




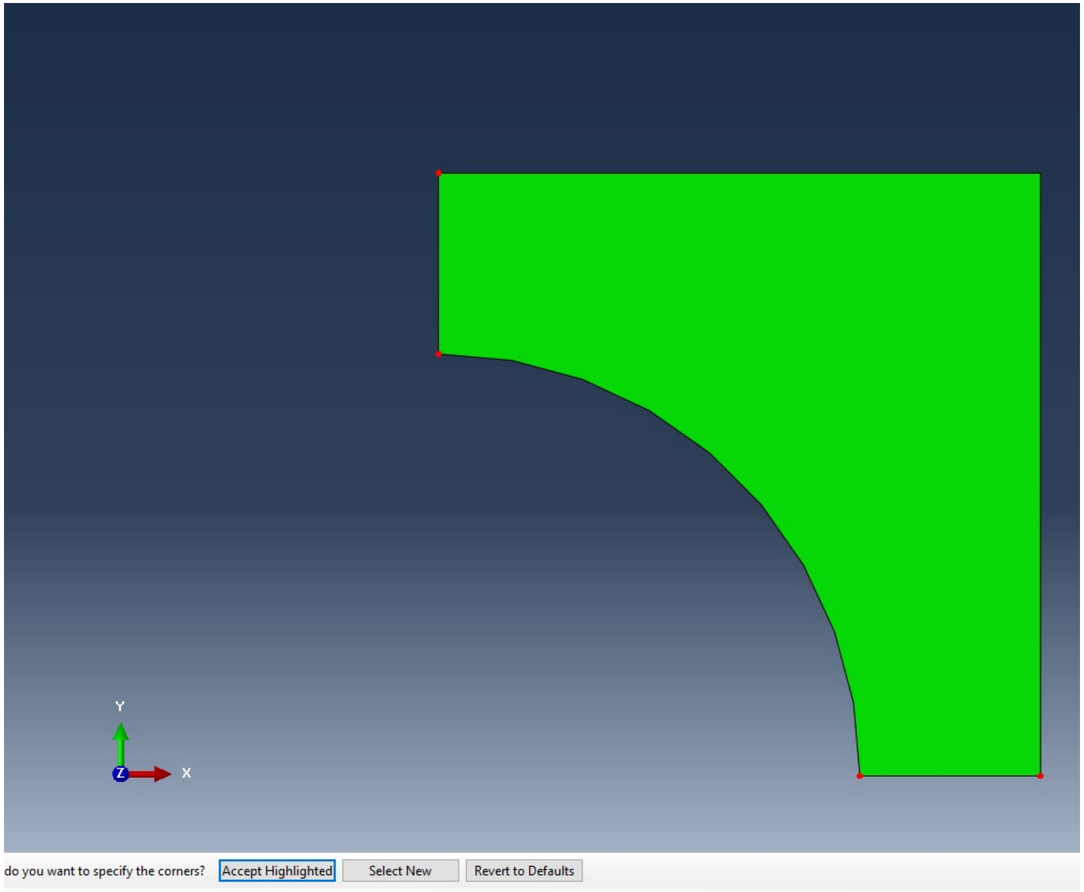
Carefully define seeds



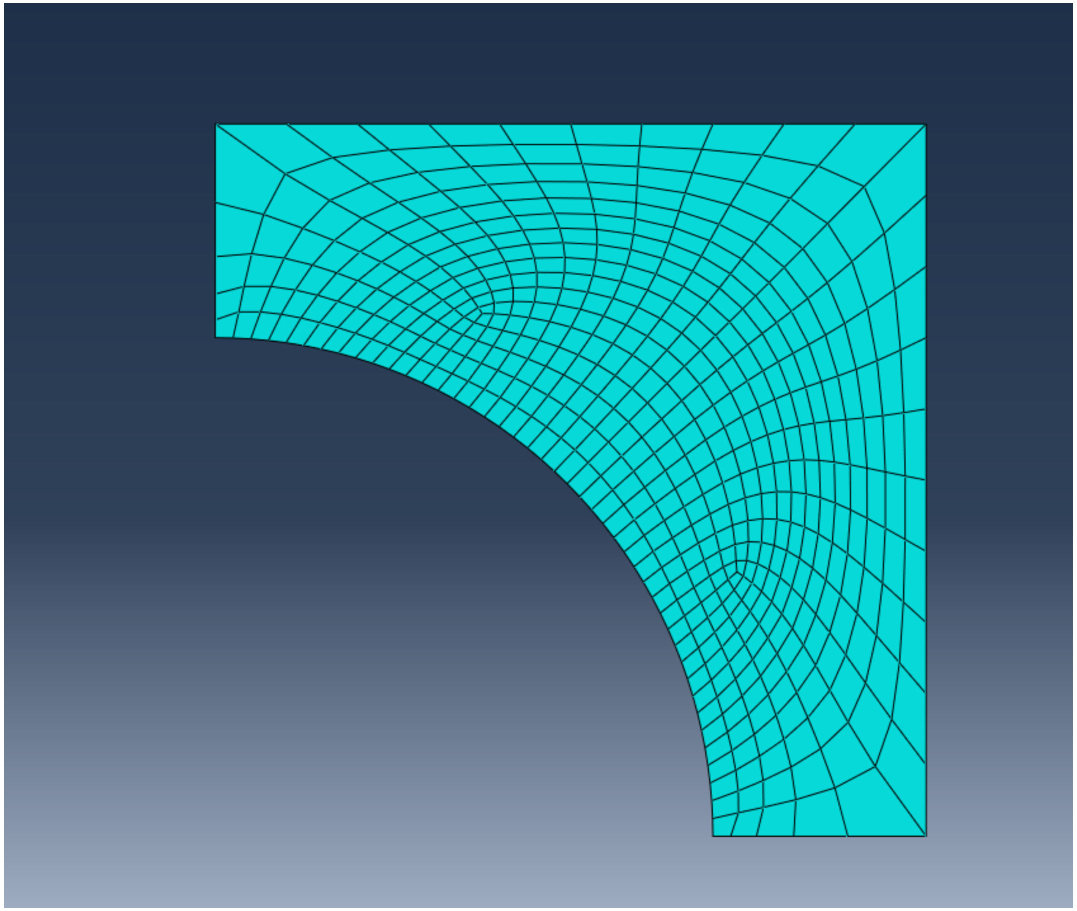
Define mesh controls



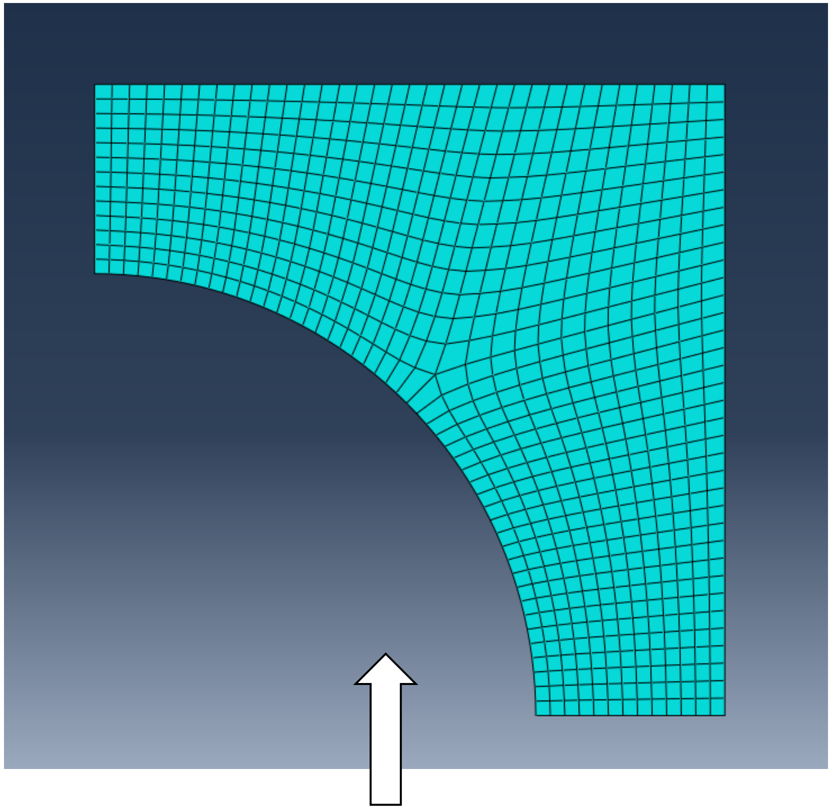
Define logical the edges for the structured mesh



Final mesh



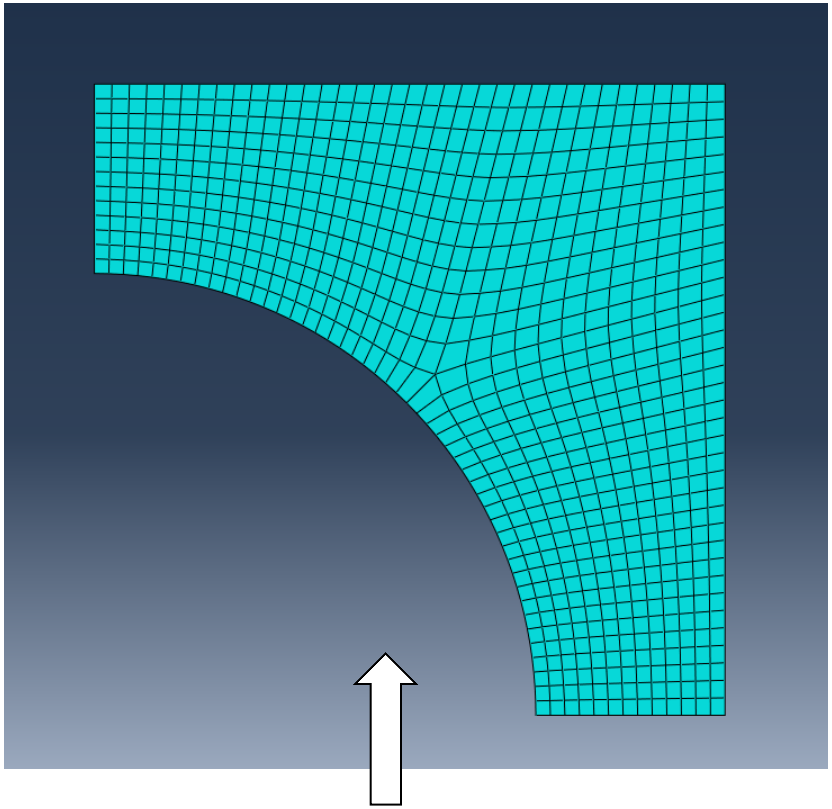
Comparison with a structured mesh with global seeds of 0.25 mm and 5 default logical edges

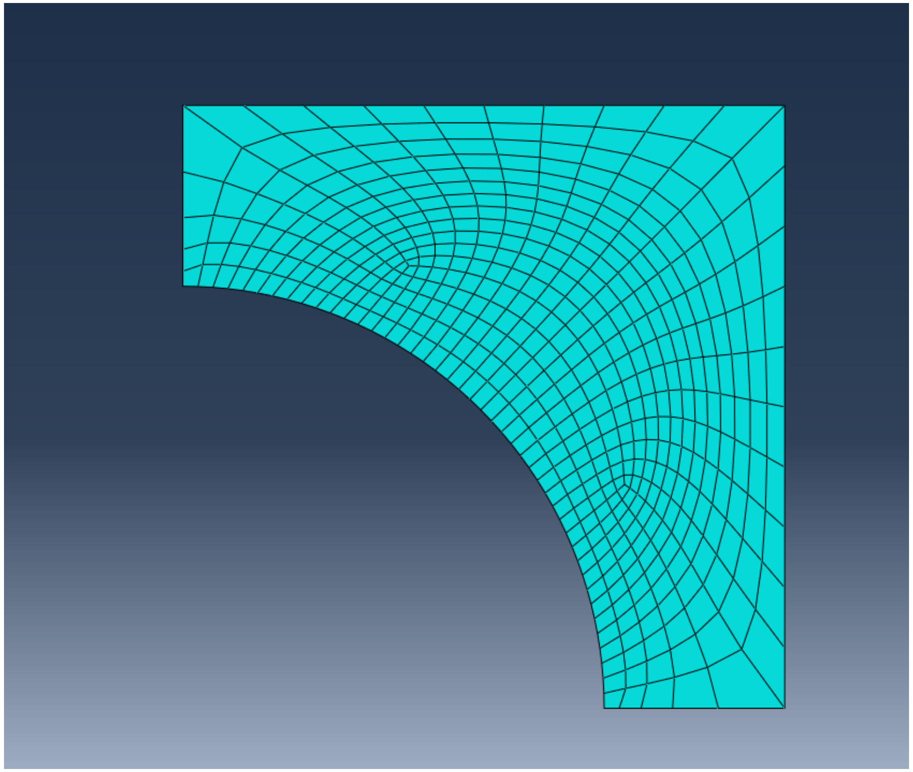


Effortless, but poor quality:

- highly distorted elements where higher stresses are expected
- fine mesh everywhere, also where it is not needed

Comparison with a structured mesh with global seeds of 0.25 mm and 5 default logical edges





Effortless, but poor quality:

- highly distorted elements where higher stresses are expected
- fine mesh everywhere, also where it is not needed