Curved mesh generation for fluid dynamics problems.

Ghina El Jannoun*, Cécile Dobrzynski*

* IMB - Université de Bordeaux and
Team Cardamom - INRIA Bordeaux Sud-Ouest, Bordeaux, France

Introduction

- Increase solution accuracy by using high order schemes
- True accuracy of these methods requires that the mesh boundary are represented with at least the same accuracy

$\Rightarrow$ Need to generate curved meshes.

- Bézier/Nurbs representation for curved element
- 3-order simplicial element

Guideline

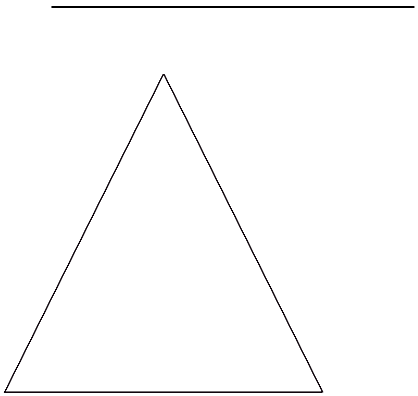
Curved element definition

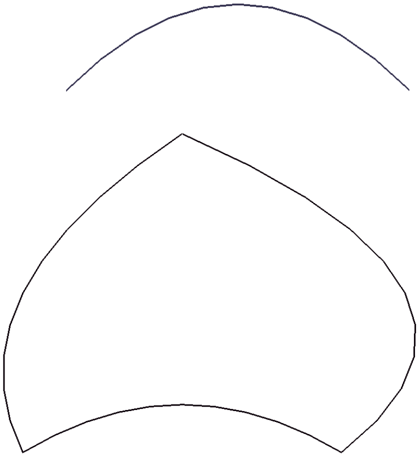
Curvilinear mesh generation

- validity criterion
- Computing curved mesh via linear elasticity analogy

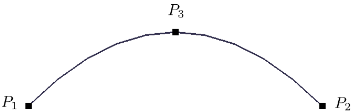
Boundary treatment

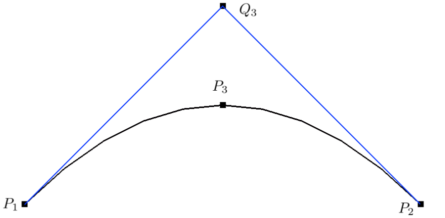
Curved element vs classical element





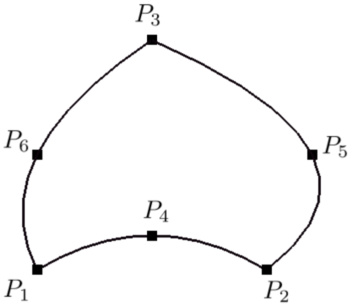
Curved element definition

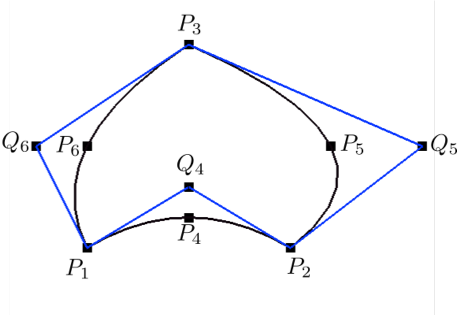




$(1-t)(1-2t)P_1 + 4t(1-t)P_3 + t(2t-1)P_2$

$(1-t)^2 P_1 + 2t(1-t)Q_3 + t^2 P_2$





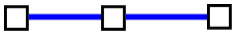
Bezier basis functions

Bézier basis functions of k+1 order over a simplex

- λ_i the barycentric coordinates,
- α multi-index of length k,
- the Bézier basis function is:

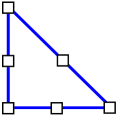
$$B_\alpha(M) := c(\alpha) \prod_{i=1}^{d+1} \lambda_i(M)^{\alpha_i}$$

1D basis function of 3-order



$$\lambda_1 = (1-t) \quad \lambda_2 = t$$
$$B_{20} = \lambda_1^2 \quad B_{02} = \lambda_2^2 \quad B_{11} = 2\,\lambda_1\lambda_2$$

2D basis function of 3-order



$$\lambda_1 = (1-\eta-\xi) \quad \lambda_2 = \xi \quad \lambda_3 = \eta$$
$$B_{200} = \lambda_1^2 \quad B_{020} = \lambda_2^2 \quad B_{002} = \lambda_3^2$$
$$B_{110} = 2\,\lambda_1\lambda_2 \quad B_{101} = 2\,\lambda_1\lambda_3 \quad B_{011} = 2\,\lambda_2\lambda_3$$

Curved element representation

Properties (φ : Bézier basis functions)

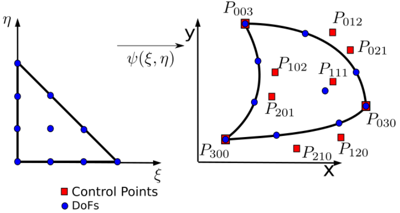
$$\begin{cases} B_\alpha(M) & \geq 0 \text{ for any } \alpha, |\alpha| = k \text{ and } M \in K \\ \displaystyle\sum_{\alpha,|\alpha|=k} & B_\alpha(M) = 1 \end{cases}$$

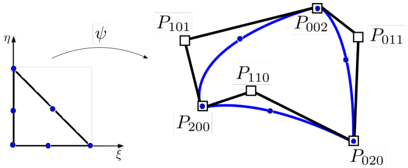
Curved element definition

Lets $\mathcal{P} = \{P_\alpha \in \mathbb{R}^p, \alpha \in \mathcal{I}\}$, be a family of control points, we approximate a function ψ by

$$\psi(M) \approx \sum_{\alpha,\alpha \in \mathcal{I}} P_\alpha B_\alpha^n(M).$$

Example of elements

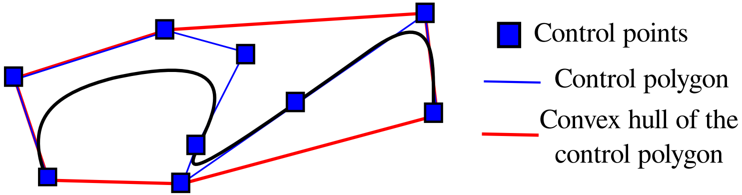




Properties of a Bézier element

Convexity of control polygon

A Bézier curve is contained in the convexe hull of its control polygon



Tangent property (for a segment)

At the ends of the curve, the curve is tangent to the control polygon.

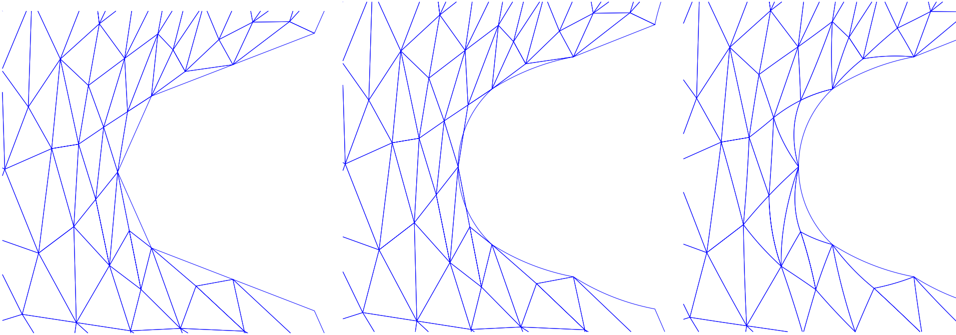
Curved mesh generation

Problematic

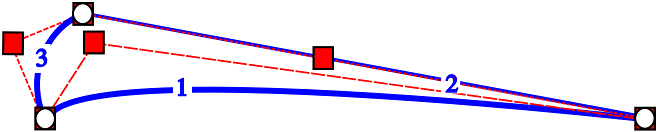
- How to generate curved meshes ?

Our approach

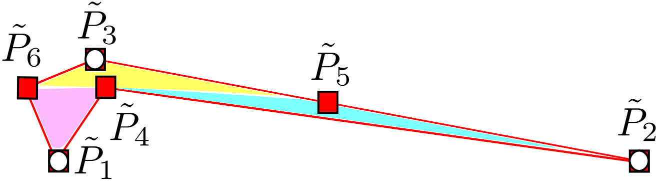
- Starting from a classical mesh of the domain (piecewise linear), we modify it to generate curved mesh.



Validity criteria

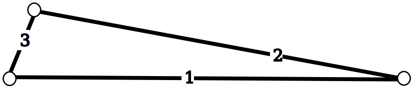


curved element

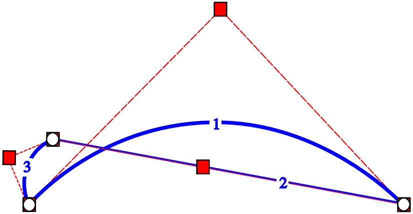


$$\begin{cases} \text{pink area} = det(\vec{v}_{16}, \vec{v}_{14}) & \geq \quad 0 \\ \text{blue area} = det(\vec{v}_{24}, \vec{v}_{25}) & \geq \quad 0 \\ \text{yellow area} = det(\vec{v}_{35}, \vec{v}_{46}) & \geq \quad 0 \end{cases}$$

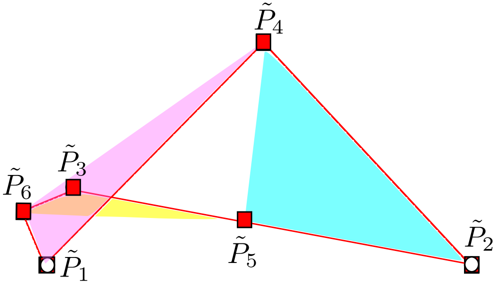
Example for invalid element



straight element



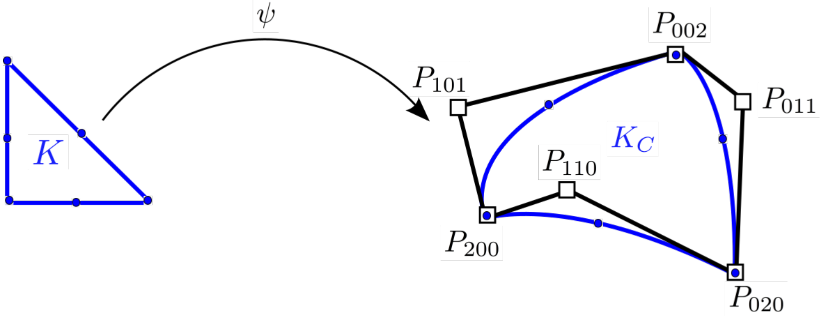
curved element



blue area $= det(\vec{v}_{24}, \vec{v}_{25}) \leq 0$

Remark on the element validity.

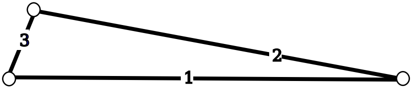
Curved element of 3-order



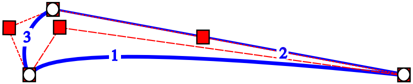
Previous validity criteria $\not\Rightarrow$ uniqueness of the mapping

- isogeometric mapping invertible $\rightarrow det(J_\psi) > 0$ on the element.

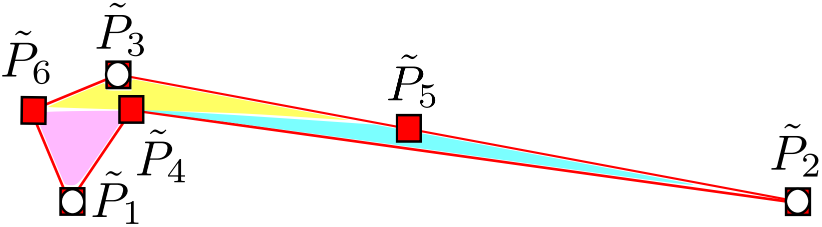
Validity criteria



straight element



curved element



$$\left\{ \begin{array}{rcl} \text{pink area} = det(\vec{v}_{16}, \vec{v}_{14}) & \geq & 0 \\ \text{blue area} = det(\vec{v}_{24}, \vec{v}_{25}) & \geq & 0 \\ \text{yellow area} = det(\vec{v}_{35}, \vec{v}_{46}) & \geq & 0 \end{array} \right.$$

$$\left\{ \begin{array}{rcl} det(\vec{v}_{16}, \vec{v}_{25}) + det(\vec{v}_{45}, \vec{v}_{46}) & \geq & 0 \\ det(\vec{v}_{35}, \vec{v}_{16}) + det(\vec{v}_{46}, \vec{v}_{43}) & \geq & 0 \\ det(\vec{v}_{35}, \vec{v}_{45}) + det(\vec{v}_{25}, \vec{v}_{36}) & \geq & 0 \end{array} \right.$$

How to curve straight mesh ?

Given

A piecewise linear mesh + Bézier curves on the boundary

Idea

Using linear analogy on control points subdivided mesh.

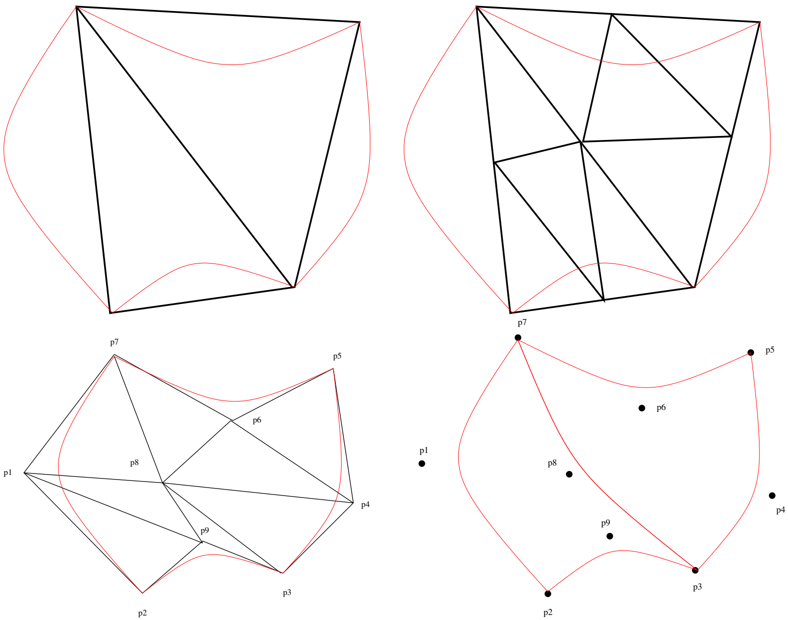
Thanks to the previous inequalities,

Subdivided mesh legal + other determinants $\geq 0 \Rightarrow$ curved mesh valid.

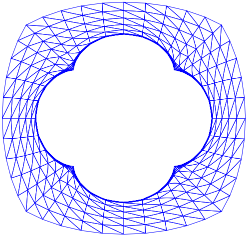
Steps:

- create mid-edge points and consider them as control points,
- subdivide the mesh with those points,
- deform the subdivided mesh using elasticity analogy to fit with $\mathcal{NURBS}$ boundary curves.

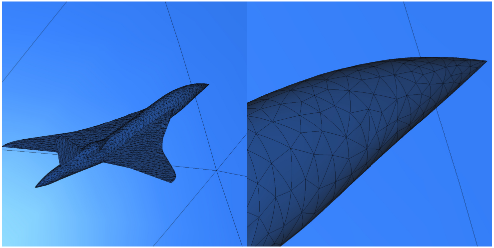
How to curve straight mesh?



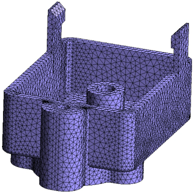
Curved mesh examples



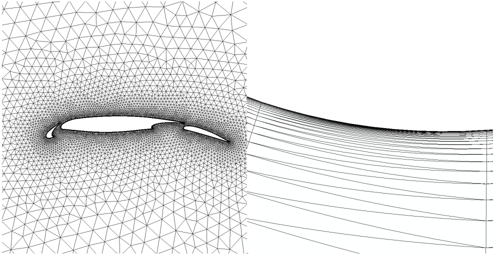
(a) 2d analytic: 6 iter



(b) Airplane: 9 iter

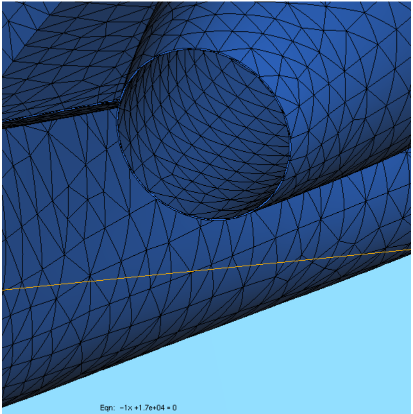


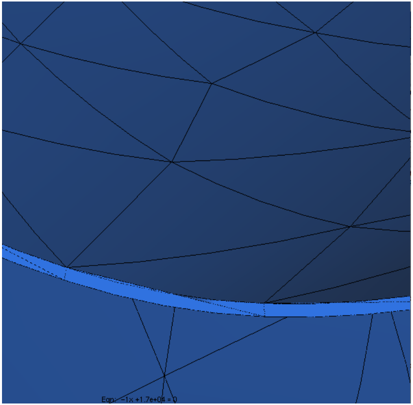
(c) Mechanical part: 2 iter



(d) Three components wing: 1 iter

Boundary problem





How to generate a valid mesh starting from a invalid curved boundary?

Given

A simplicial mesh + curved boundaries

1. Check mesh validity
2. How to untangle invalid elements while respecting the curved boundary and structure of the mesh?
   - node repositioning
   - curvature propagation

Untangling via local topological optimization

Given

A piecewise linear mesh + Bézier curves on the boundary

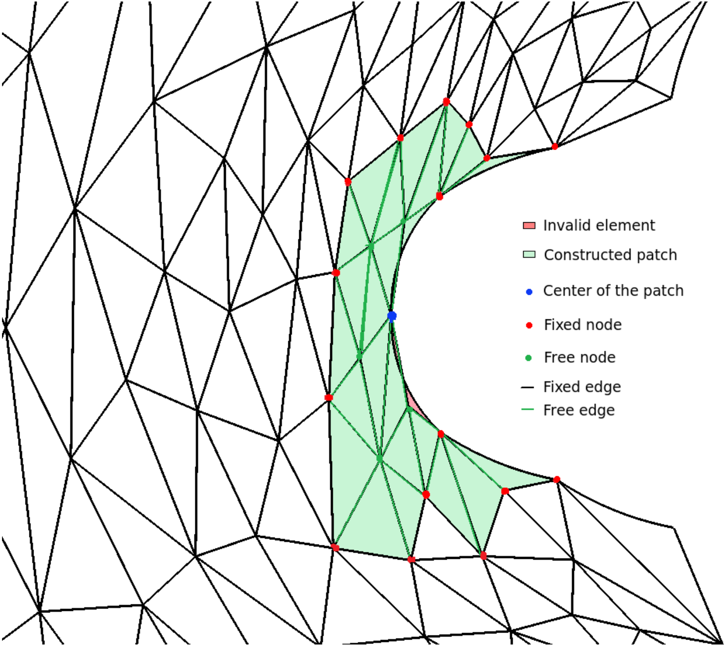
Idea

Solve a constrained optimization problem on a local patch around the invalid element. A patch of elements with determinants $\geq 0 \Rightarrow$ curved valid mesh.

Steps:

- Construct a patch $\mathcal{P}_k$ of elements surrounding the invalid element T_k under investigation,
- Set the freedom of the nodes and control points inside the patch,
- deform the patch using local topological optimization to fit with the **NURBS** boundary curves.

Local topological optimization



Local topological optimization

Constrained optimization problem

$$
\begin{aligned}
\max_{\mathbf{x}_k, s} \quad & s \\
\text{s.t.} \quad & \\
& \forall T_i \in \mathcal{P}_k \begin{cases}
det(\vec{v}_{16}, \vec{v}_{14}) & \geq & s \\
det(\vec{v}_{24}, \vec{v}_{25}) & \geq & s \\
det(\vec{v}_{35}, \vec{v}_{46}) & \geq & s \\
det(\vec{v}_{16}, \vec{v}_{25}) + det(\vec{v}_{45}, \vec{v}_{46}) & \geq & s \\
det(\vec{v}_{35}, \vec{v}_{16}) + det(\vec{v}_{46}, \vec{v}_{43}) & \geq & s \\
det(\vec{v}_{35}, \vec{v}_{45}) + det(\vec{v}_{25}, \vec{v}_{36}) & \geq & s
\end{cases}
\end{aligned} \tag{1}
$$

Variables:

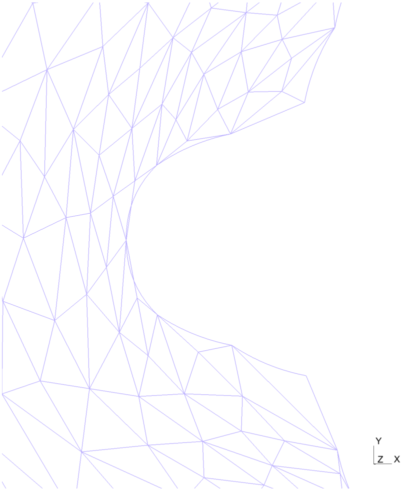
- $\mathbf{x}$ the coordinates of the free nodes and control points.
- s a lower bound on the determinants.

Curvilinear mesh untangling algorithm

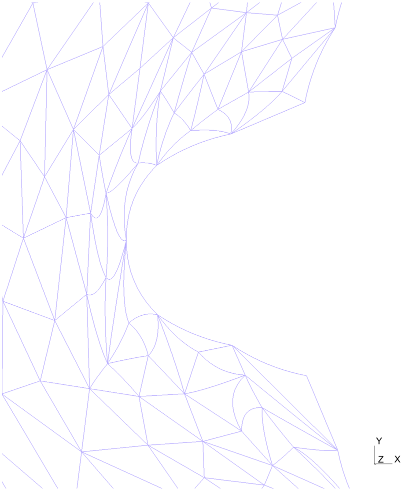
Algorithm 1 Curvilinear mesh optimization algorithm

1: **for** each invalid element k, **do**
2: construct the patch of elements $\mathcal{P}_k$
3: determine the free nodes in the patch with coordinates$\mathbf{x}_k$
4: **while** $s < 0$ **do**
5: Maximize s
6: **end while**
7: Reposition the nodes and control points.
8: **end for**=0

First example



Tangled



Untangled

Future work

- Test untangling optimization in 2d then on a surface
- Perform compressible simulation with the curved meshes

- Mesh adaptation