TatooineMesher: Anisotropic interpolation from 1D cross-sections and 2D channel mesher

17/09/2019, Toulouse (France)

L. DURON, F.-X. CIERCO, K. SAAD

Telemac User Conference 2019

CNR ENGINEERING

Table of contents

CNR ENGINEERING

1. Introduction

Introduction

CNR

- 1st producer of exclusively renewable energy in France
  - 18 hydroelectric facilities on the Rhône River (3000 MW)
  - Solar and wind energy (1000 MW)
- CNR Engineering Departement (for CNR and third party)

Introduction

CNR

- 1st producer of exclusively renewable energy in France
  - 18 hydroelectric facilities on the Rhône River (3000 MW)
  - Solar and wind energy (1000 MW)
- CNR Engineering Departement (for CNR and third party)

Objectives

1. **pre-treatment 1D models**: interpolate intermediate cross-sections
2. **pre-treatment 2D models**: interpolate the bathymetry and/or mesh the river bed
3. **post-treatment 1D models**: visualize the results in 2D (in a projected geographic coordinate system)

Developed tools

- Code : Python 3
- Command line scripts
- Usage:
  `https://github.com/CNR-Engineering/TatooineMesher/wiki`

Developed tools

- Code : Python 3
- Command line scripts
- Usage:
  `https://github.com/CNR-Engineering/TatooineMesher/wiki`

Installation and requirements

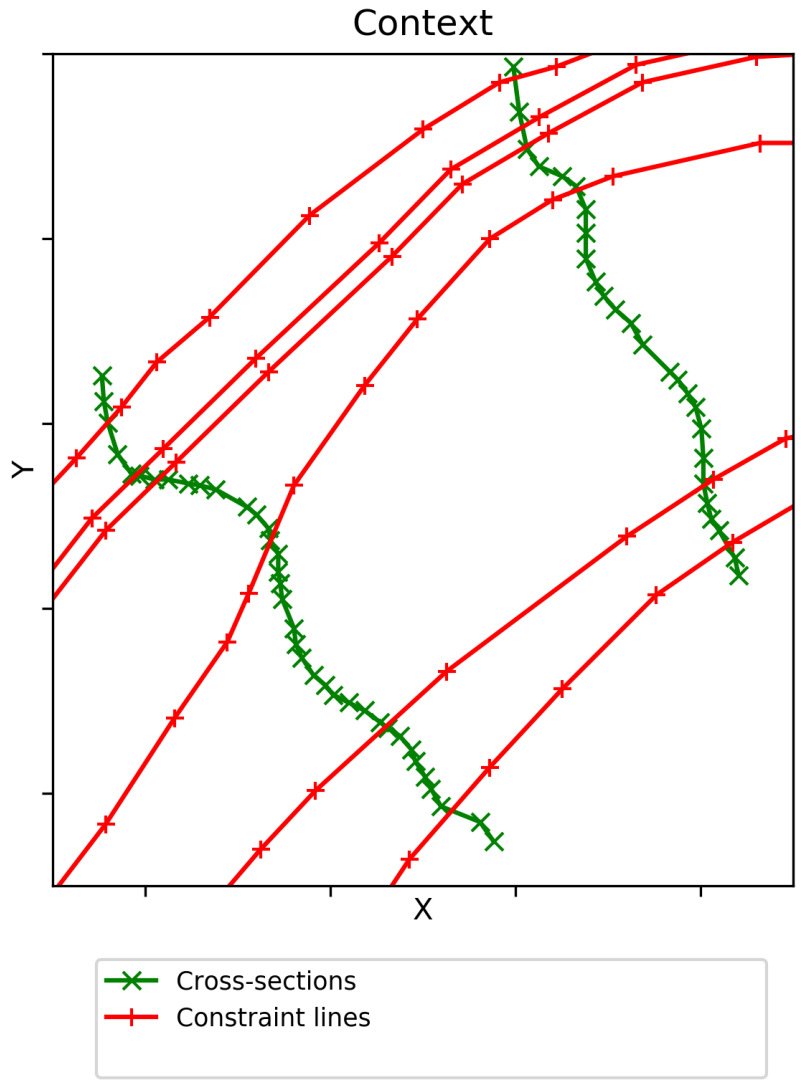
```
# Install required modules
pip install -r requirements.txt

# Install it as a module
pip install -e git://github.com/CNR-Engineering/TatooineMesher.git#egg=TatooineMesher --user
```

PyTelTools and Crue10_tools (contains Mascaret part of postel) are 2 packages required.

2. Mesh generation

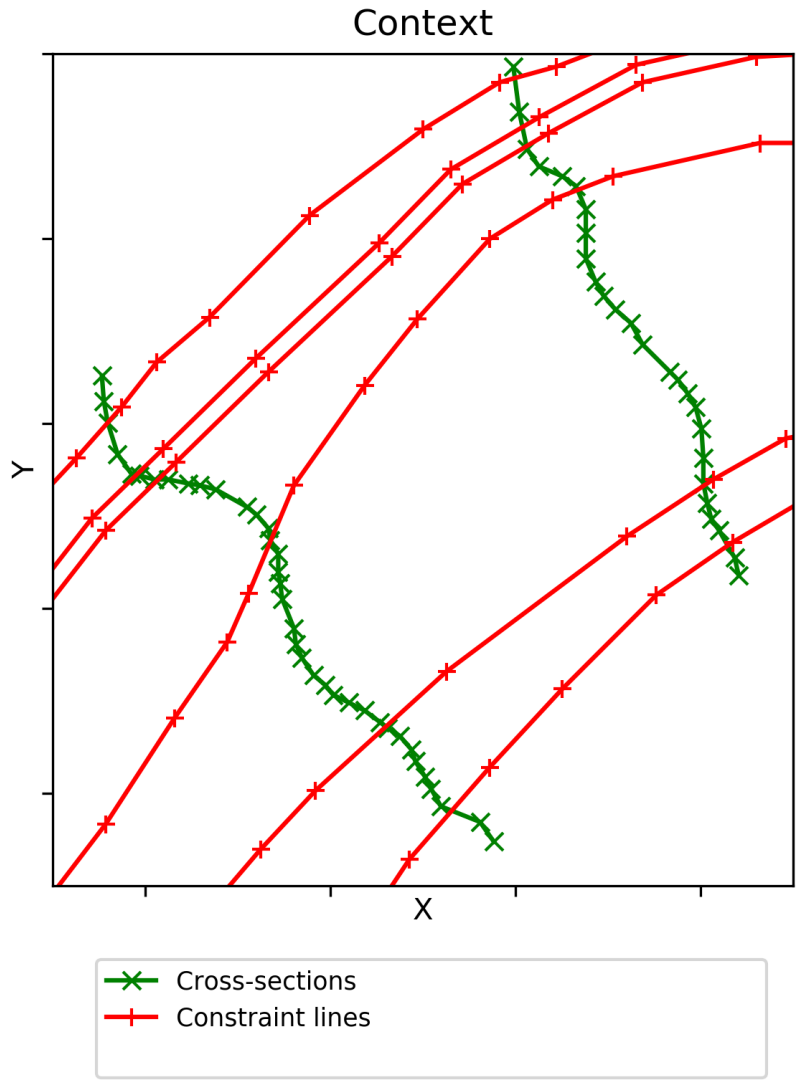
Principle - Step by step

1. Order cross-sections (CS)



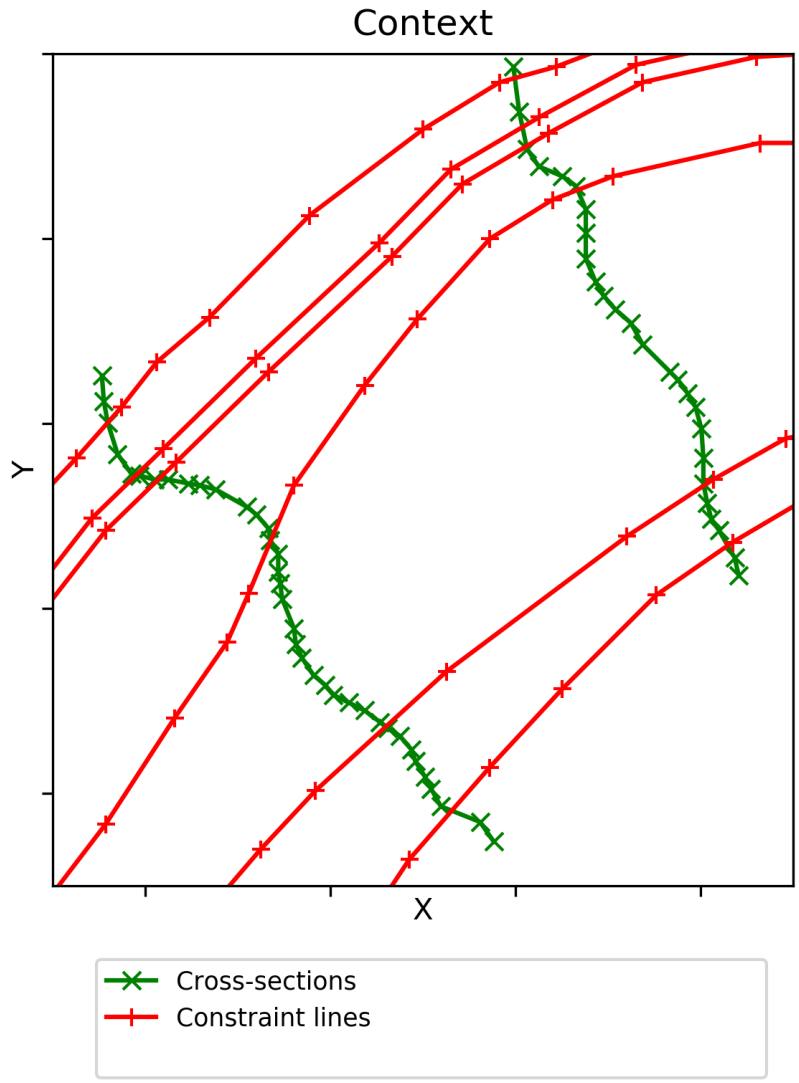
Principle - Step by step

1. Order cross-sections (CS)
2. Intersect CS and constraint lines



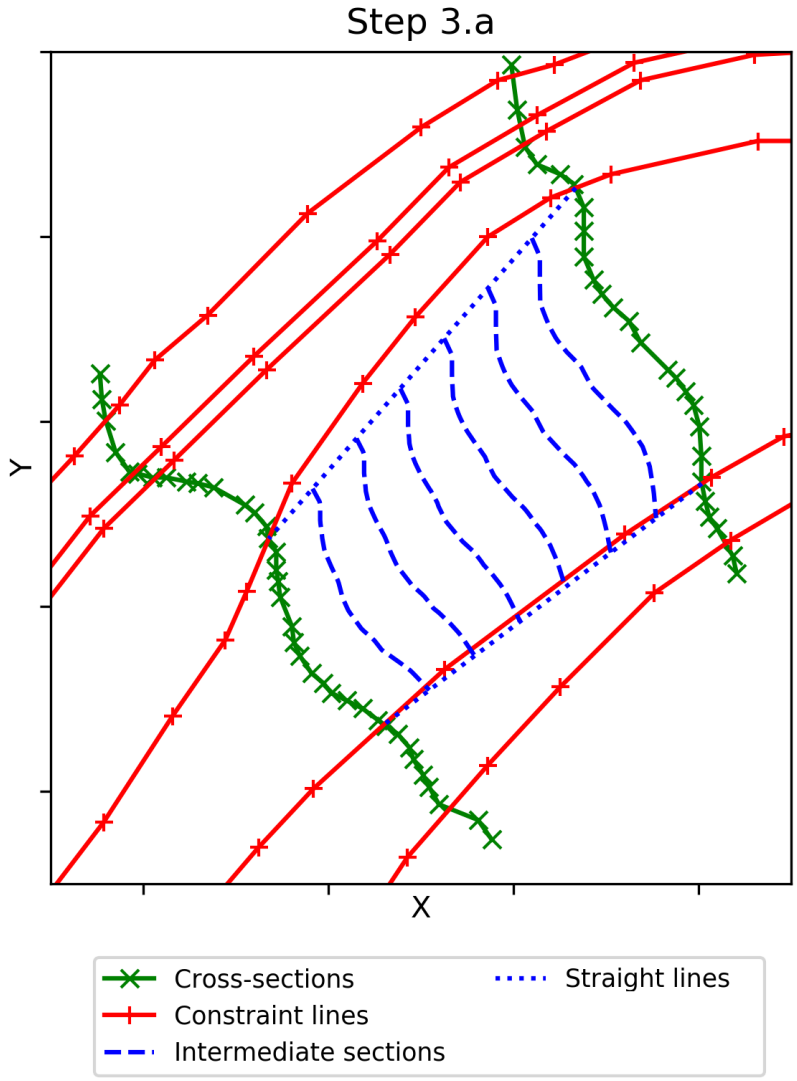
Principle - Step by step

1. Order cross-sections (CS)
2. Intersect CS and constraint lines
3. Generate nodes for each submesh



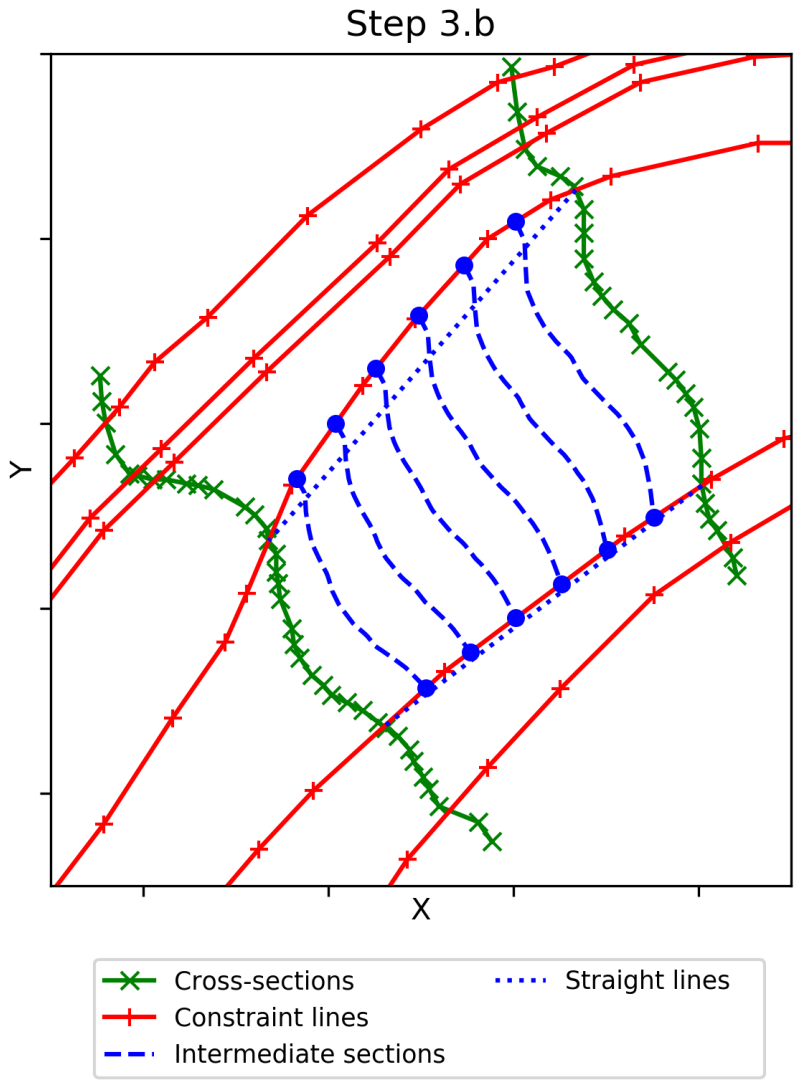
Principle - Step by step

1. Order cross-sections (CS)
2. Intersect CS and constraint lines
3. Generate nodes for each submesh
    a. Linear interp. for intermediate CS



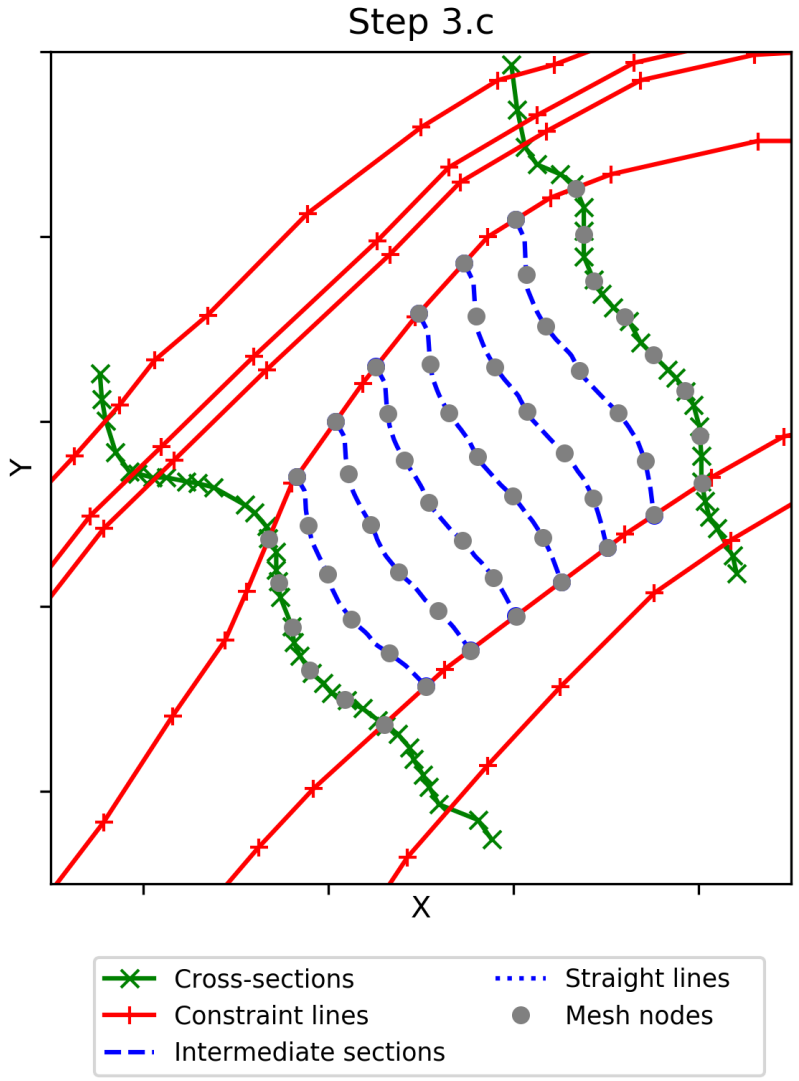
Principle - Step by step

1. Order cross-sections (CS)
2. Intersect CS and constraint lines
3. Generate nodes for each submesh
   a. Linear interp. for intermediate CS
   b. Application of an affine transformation



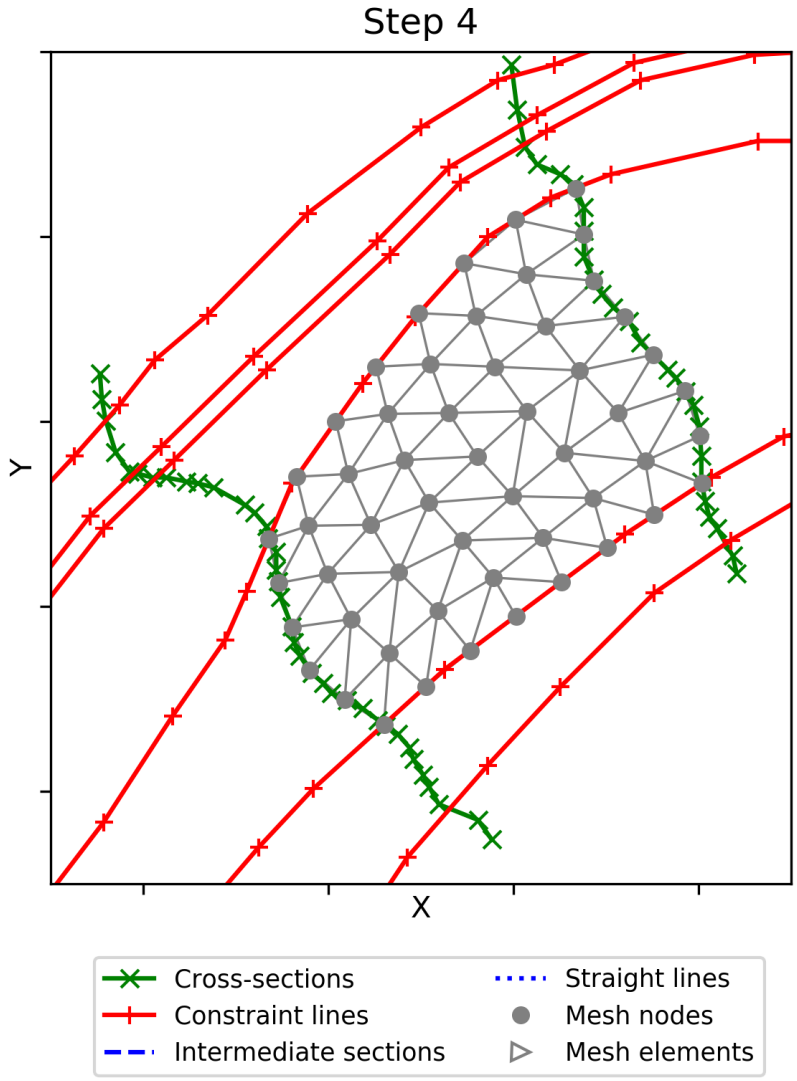
Principle - Step by step

1. Order cross-sections (CS)
2. Intersect CS and constraint lines
3. Generate nodes for each submesh
   a. Linear interp. for intermediate CS
   b. Application of an affine transformation
   c. Lateral sampling



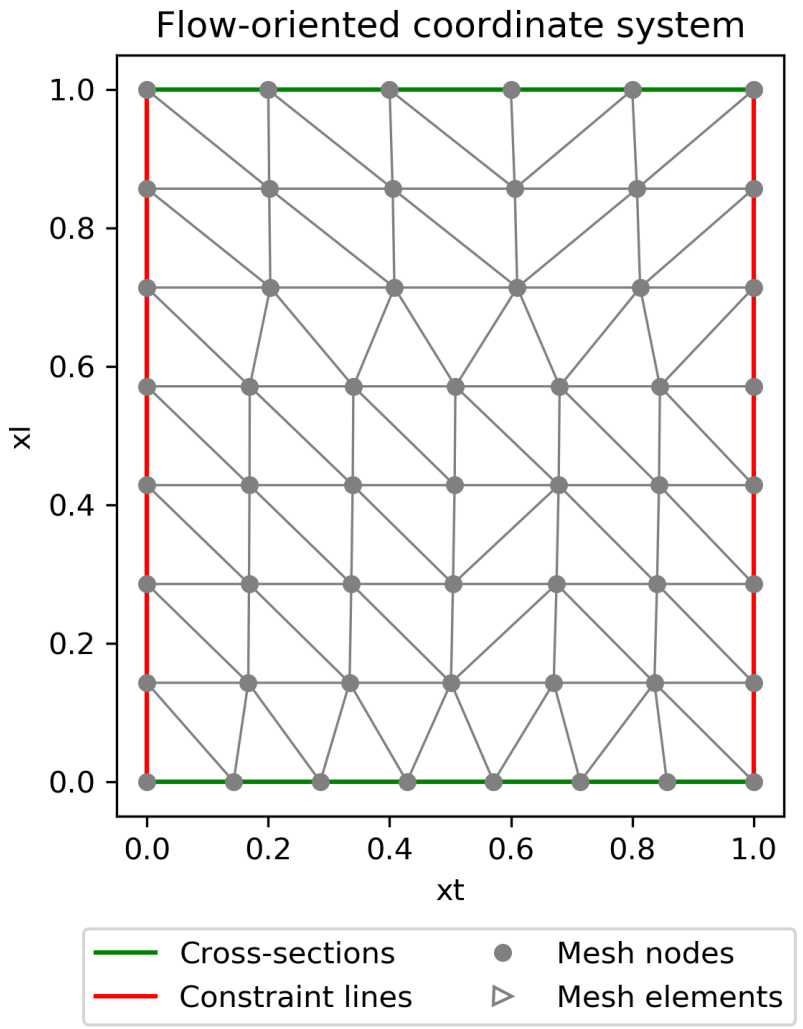
Principle - Step by step

1. Order cross-sections (CS)
2. Intersect CS and constraint lines
3. Generate nodes for each submesh
   a. Linear interp. for intermediate CS
   b. Application of an affine transformation
   c. Lateral sampling
4. Triangulate over the whole domain



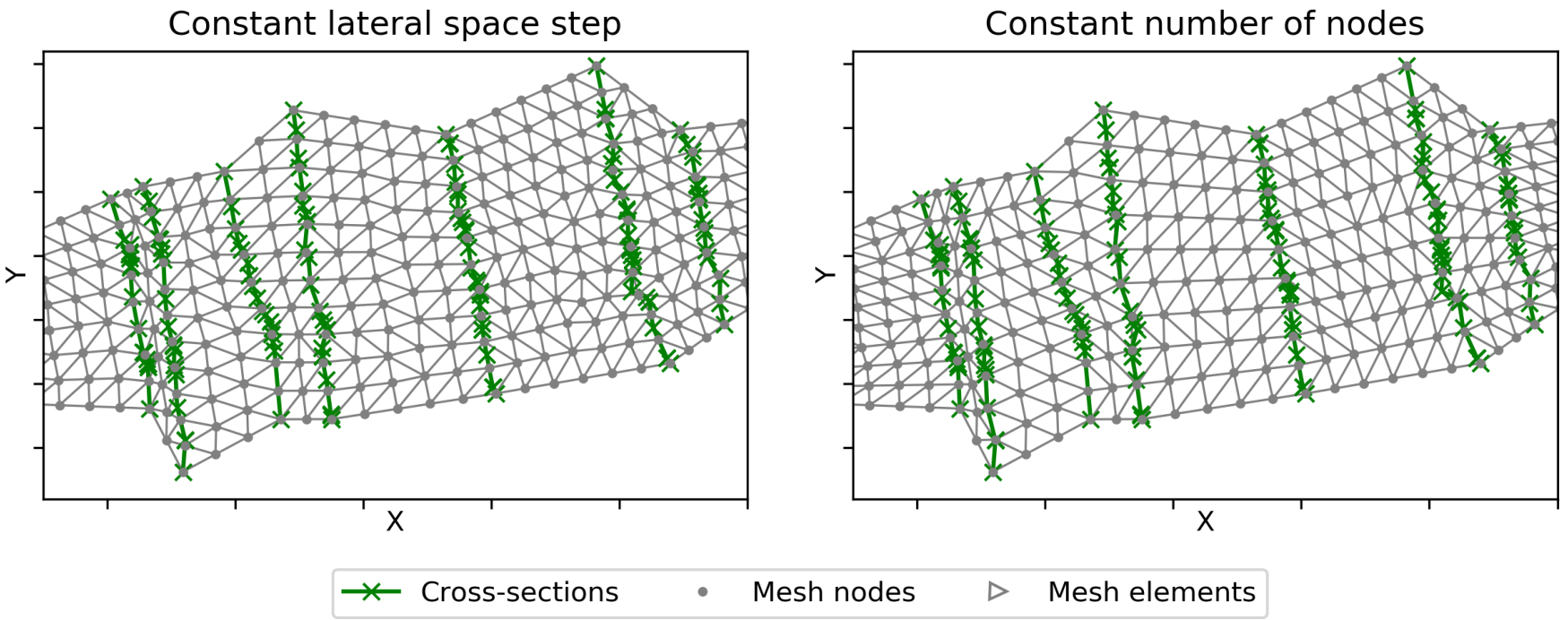
Principle - Step by step

1. Order cross-sections (CS)
2. Intersect CS and constraint lines
3. Generate nodes for each submesh
   a. Linear interp. for intermediate CS
   b. Application of an affine transformation
   c. Lateral sampling
4. Triangulate over the whole domain
5. Definition of a flow-oriented coordinate system



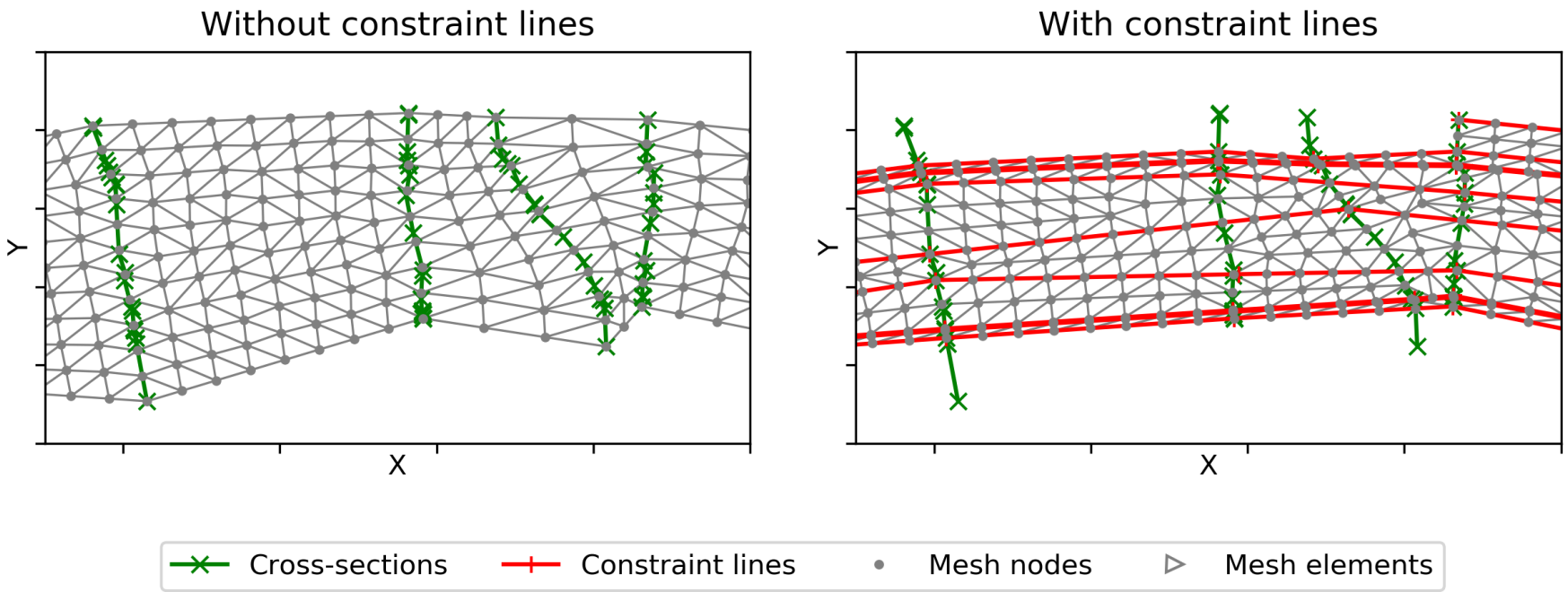
Feature 1 : Spatial discretization

- Longitudinal discretization
- Lateral discretization : structured or not (figure below)

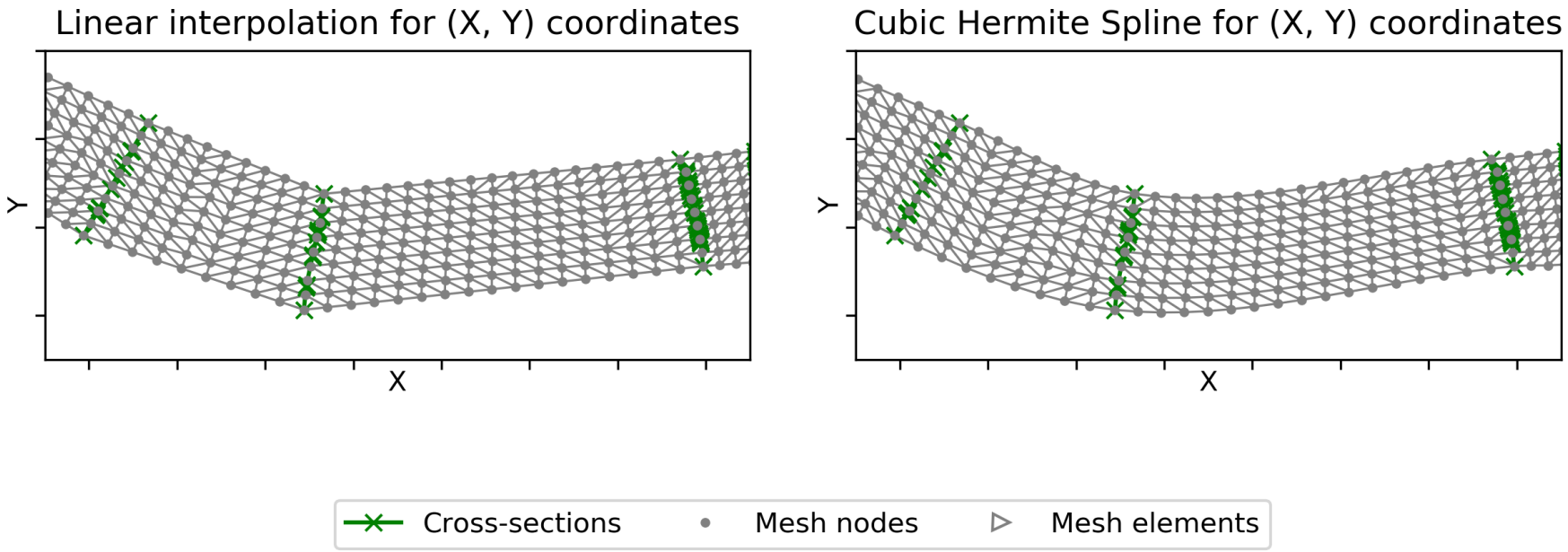


Feature 2: Constraint lines

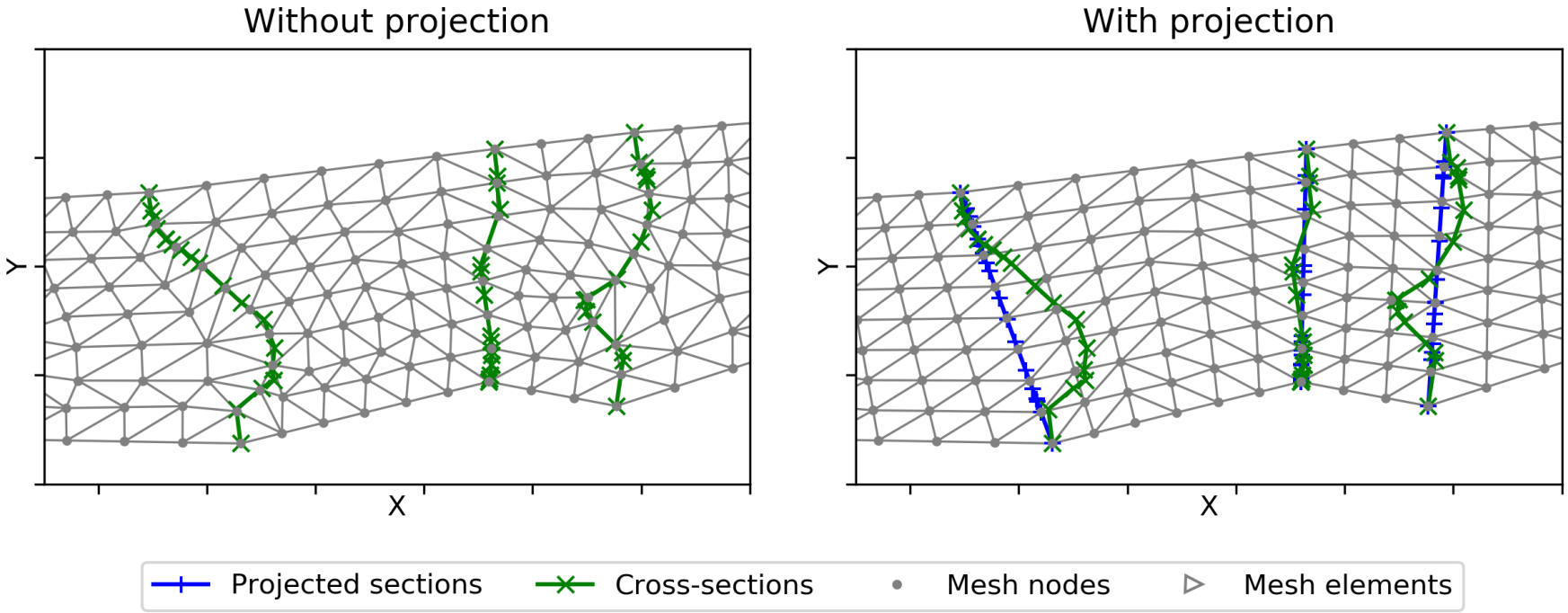
Guide the interpolation and follow topographic lines



Feature 3: XY coordinates interpolation of constraint lines



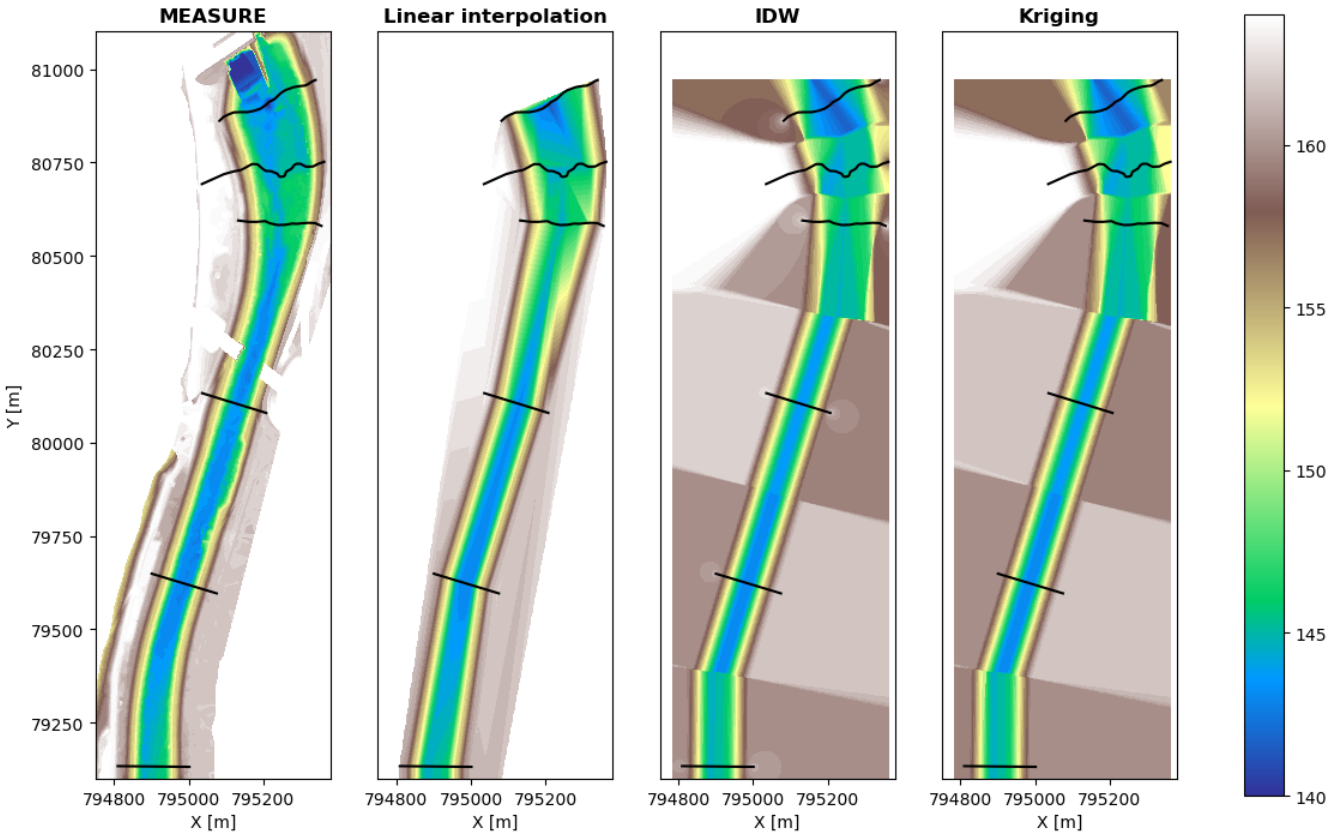
Feature 4: Flat projection of cross-section



This option makes the elements mesh adjacent to the cross-sections more organized.

3. Interpolation

Isotropic interpolation methods



Bathymetry measured (left) compared to interpolated bathymetry from elevation along cross-sections (3 interpolation methods: Linear, IDW (Inverse distance weighting) and Kriging)

Interpolation (of values at mesh nodes)

Consecutive 1D interpolators (lateral then longitudinal)

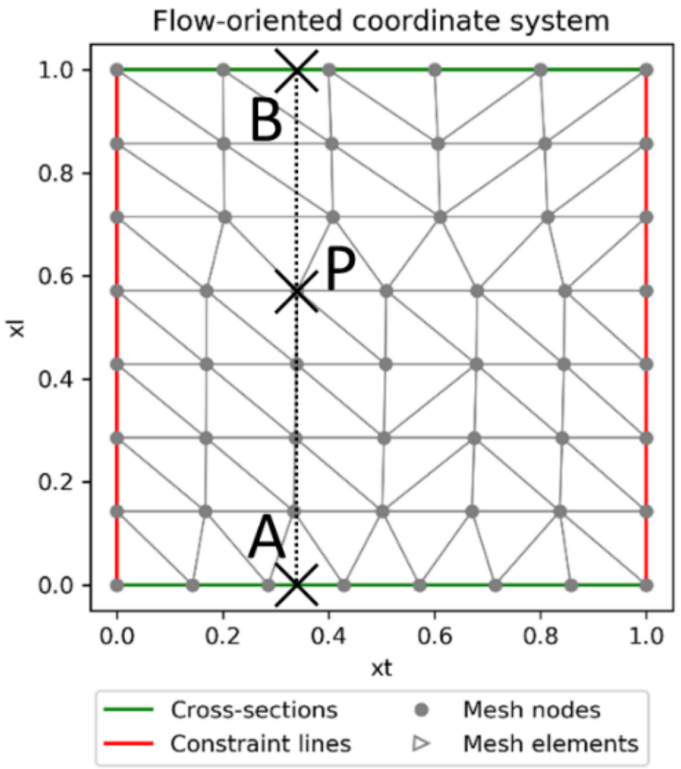
Lateral interpolation methods:

- **Linear**
- Akima spline
- Cubic spline
- PCHIP

Longitudinal interpolation method: **Linear**

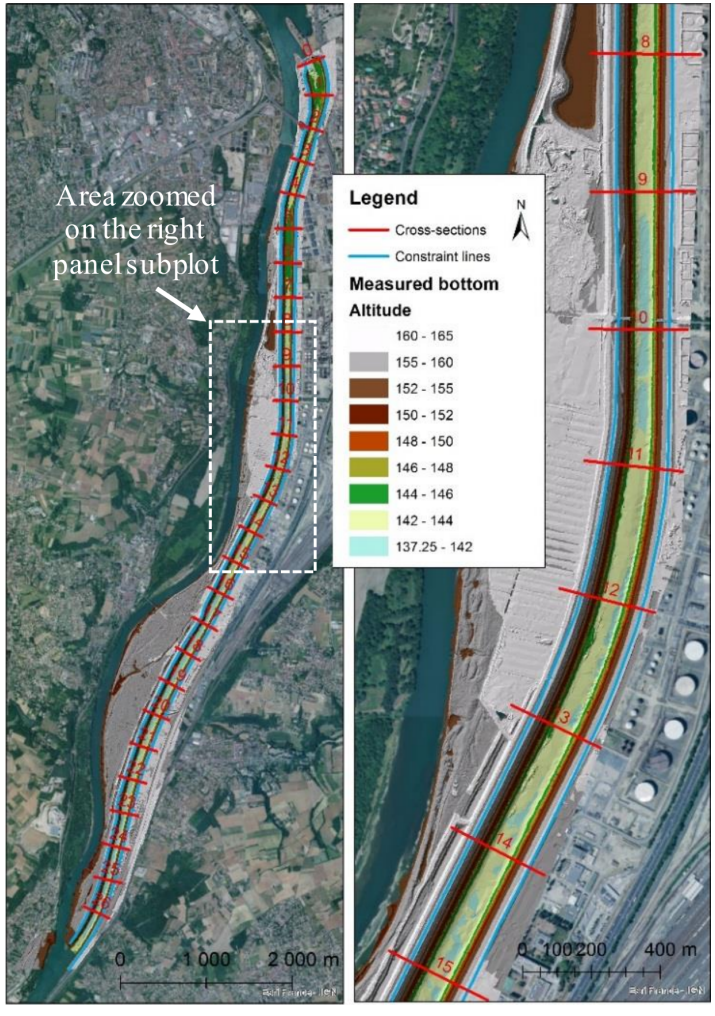
Global 2D interpolators

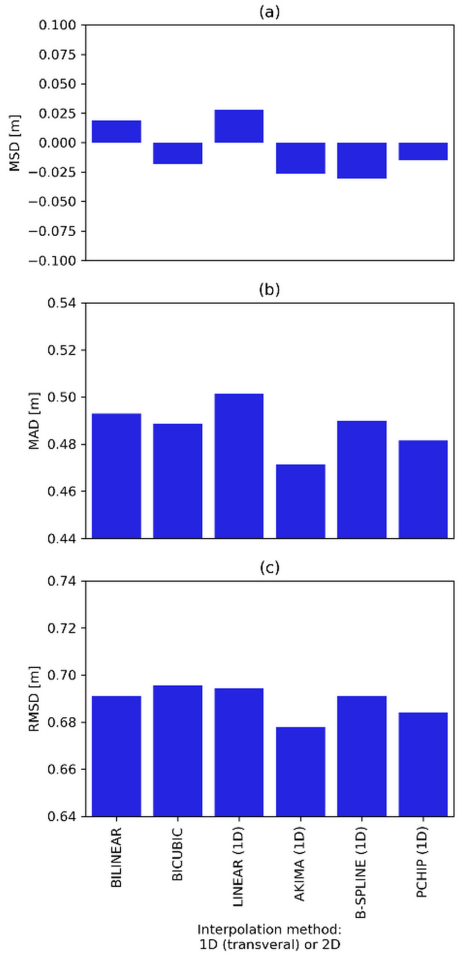
- Bilinear
- Bicubic



Consecutive 1D interpolations to have values at node P

Validation test cases



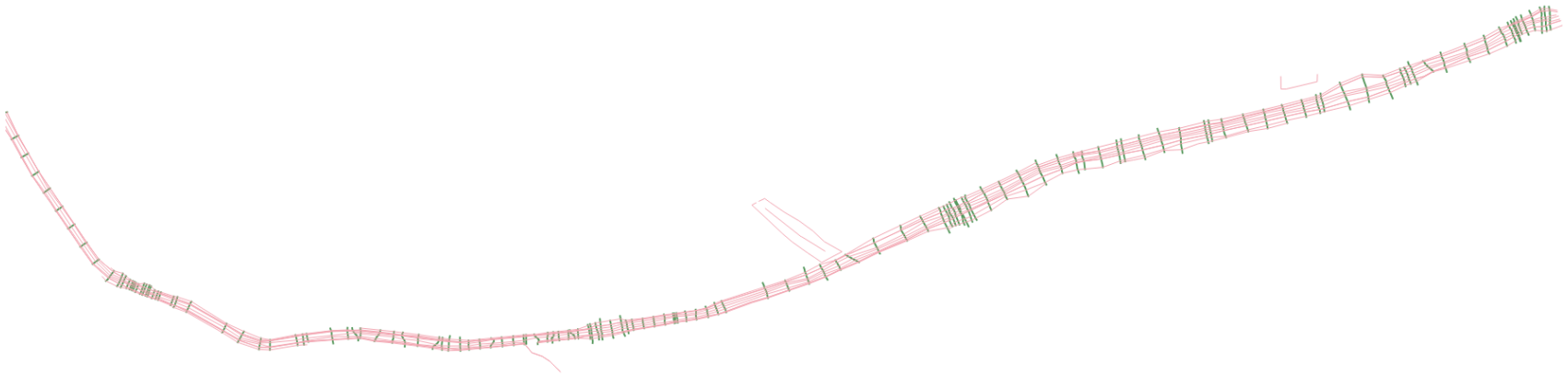


Study area: Vaugris (Rhône, France)

Criteria on difference in elevation (calculated - measured)

4. Applications

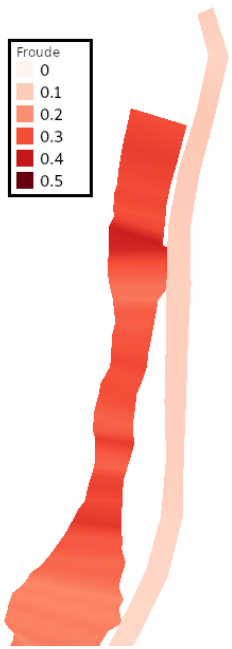
Channel mesher and intepolator



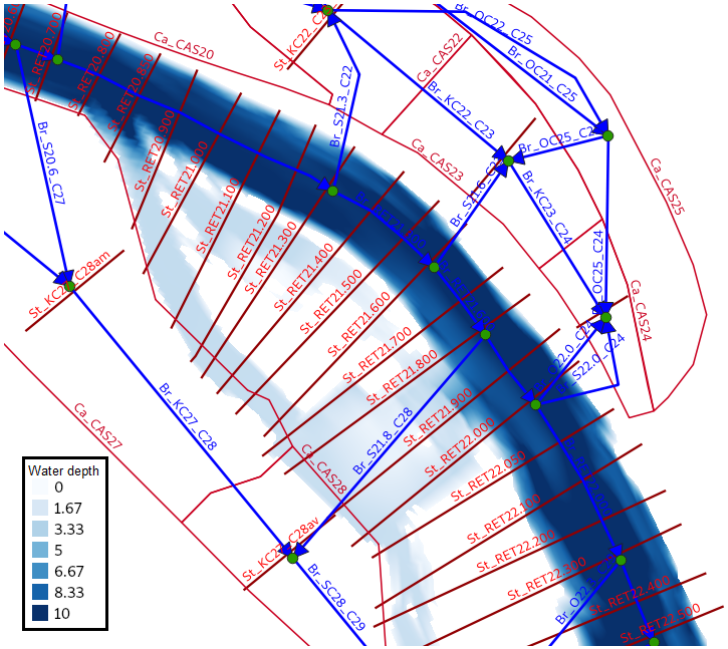
Leysse river example

Generate 2D surfaces from 1D model results

- Mesh over multiple branches
- Interpolation multiple frames and multiple variables:
  - 1D variables: free surface elevation, Froude number…
  - 2D variables: bottom elevation, friction coefficient, water depth, bed shear stress…



Froude Number (1D)



Water depth (2D)

5. Conclusion

Conclusion and perspectives

- Open-source Python package
- Multiple aims
- Usefull for Mascaret-T2D coupling?
- Use in combinaison with BlueKenue or GMSH?

Thank you for your attention !

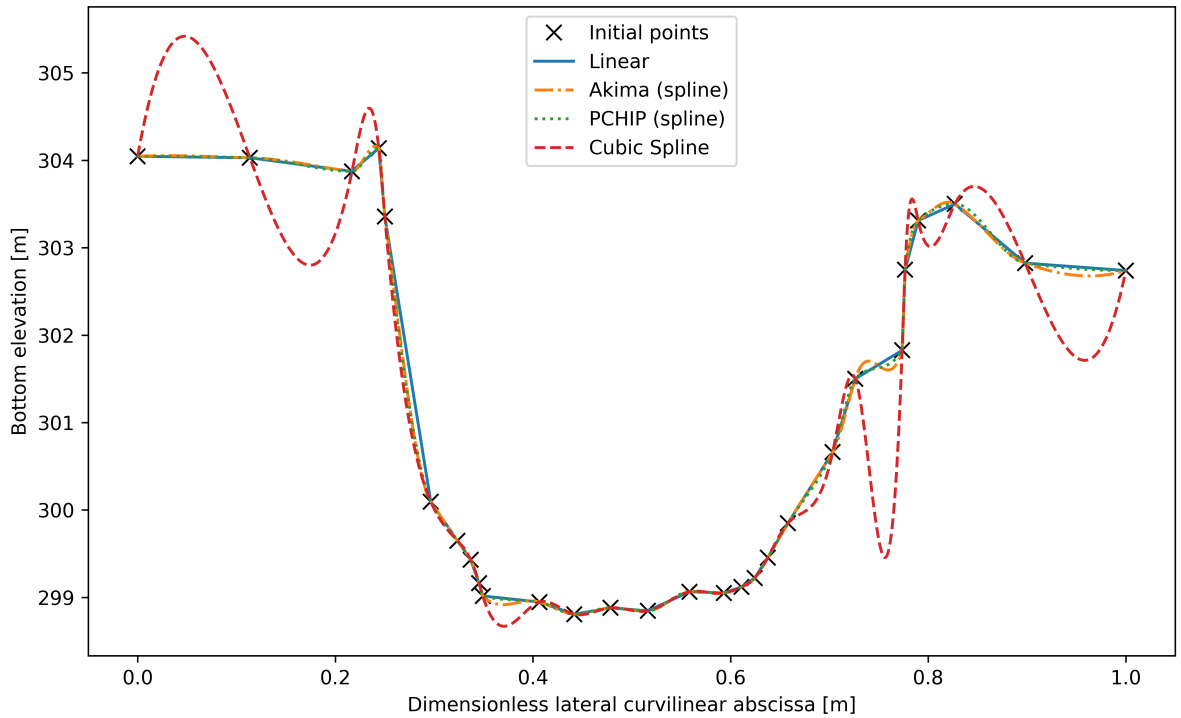
Any questions?

L'énergie au cœur des territoires

cnr.tm.fr

CNR ENGINEERING

Appendix

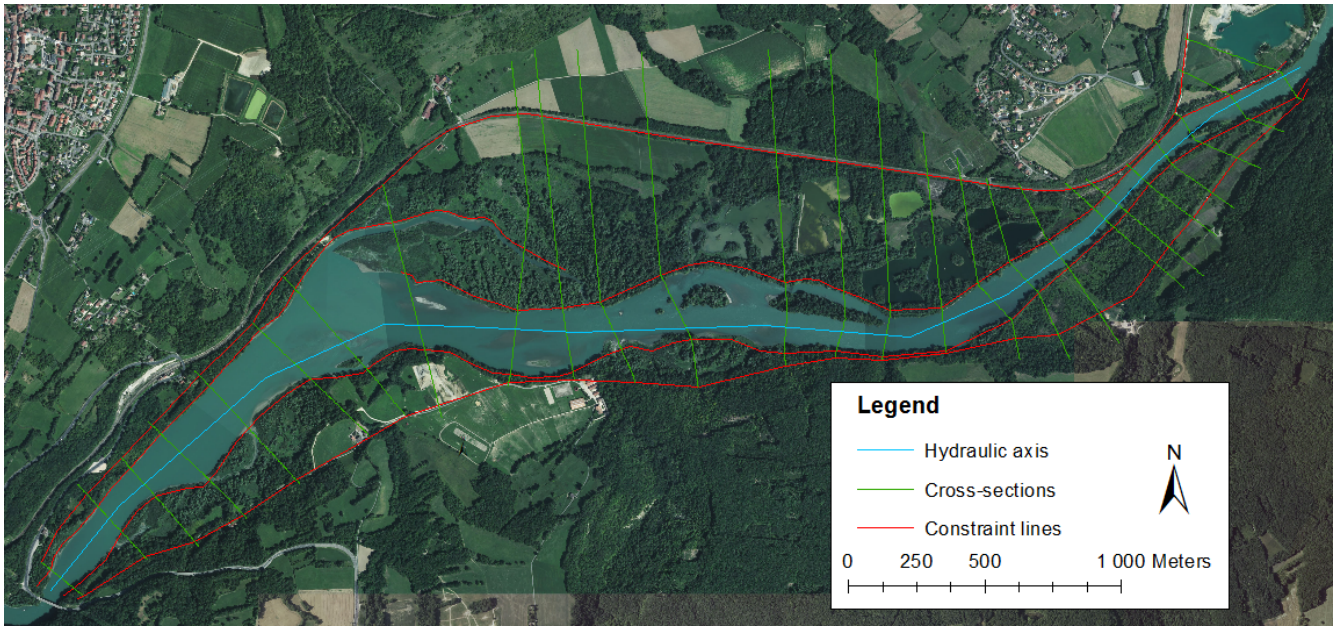


Comparison of linear, cubic spline, Akima and PCHIP interpolation at single cross section

Cubic spline is not robust. In the case of upsampling or non-equally spaced data, it creates over shooting at locations of abrupt changes in the slope.

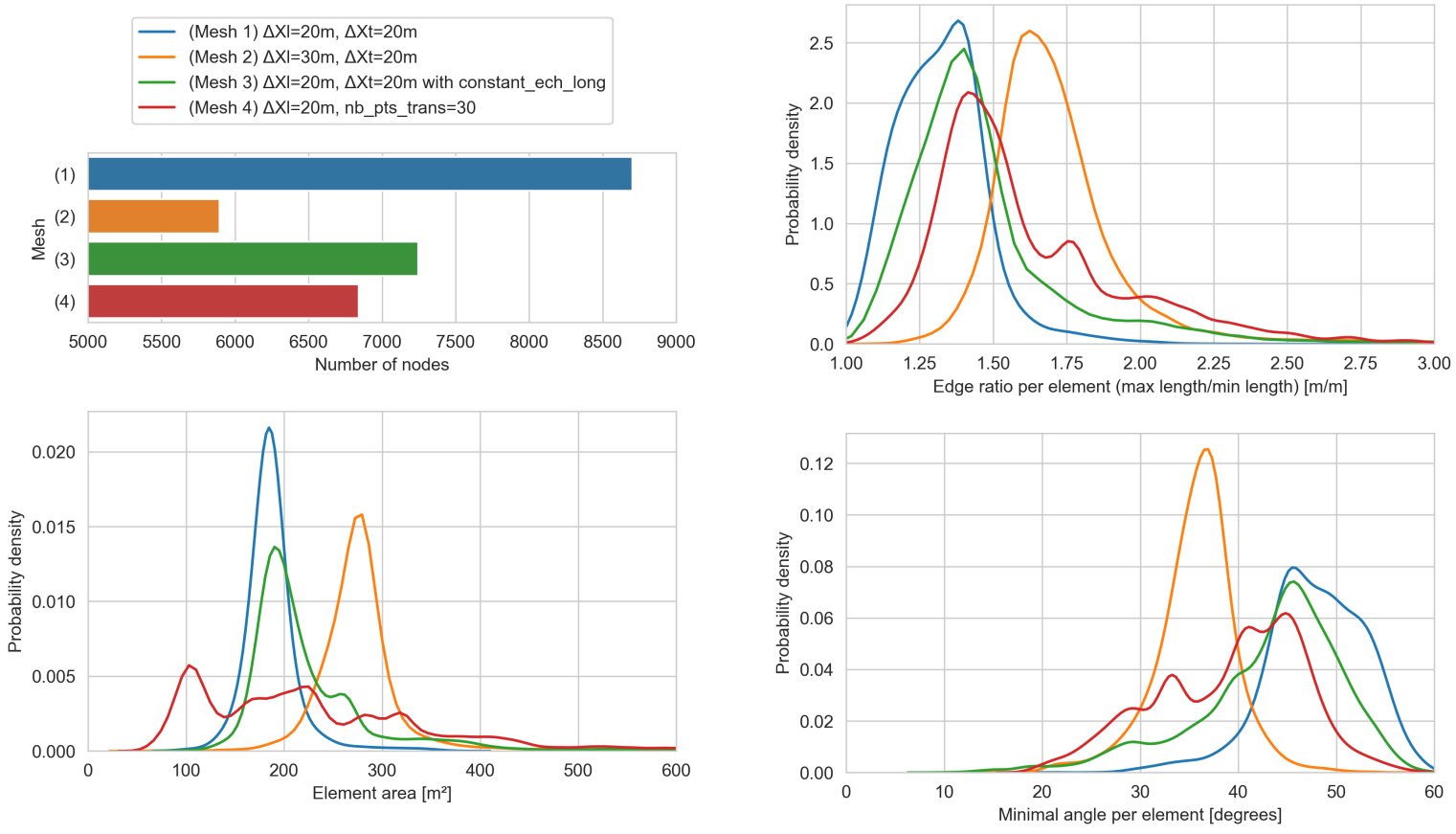
Mesh generation on L'Étournel site (1/2)

A limited domain on the Upper Rhône River (upstream Génissiat dam) called L'Étournel is chosen to compare meshes generated with TatooineMesher with different space discretization options. This simple data set, presented in Figure below, includes 25 cross-sections intersected by at most 5 constraint lines.



Geometrical data used to mesh "L'Étournel" site

Mesh generation on L'Étournel site (2/2)



Statistics on generated meshes

The End...