POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Homework report

# Guided project 2:

Scientific computing tools for advanced mathematical modelling

**Authors: Simone Piazza, Giorgio Romano, Eugenio Varetti**

**Academic year: 2022-2023**

---

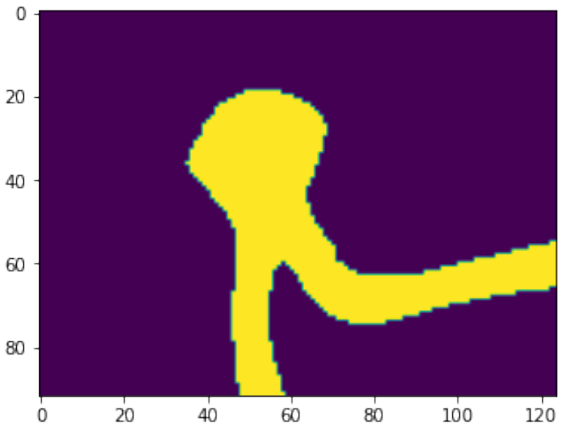
## 1. Mathematical formulation of the problem

This report addresses the problem of numerical approximation of blood velocity and pressure within a venous system affected by an aneurysm, using a two-dimensional (2D) medical image as a visualization. To carry out such goal, three key steps need to be performed subsequently: geometry segmentation, mesh construction and numerical approximation of the solution field.

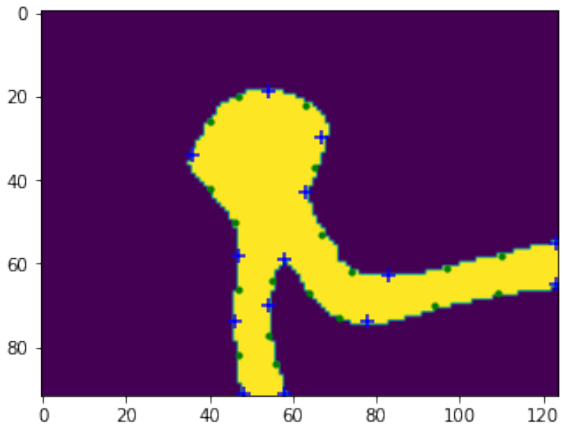
### 1.1. Checkpoint 1: segmentation of an Aneurysm from a 2D medical image

Initially, the 2D medical image needs to be segmented to identify the region of interest, the portion representing the venous system aneurysm. This step allows for the isolation of the study area for subsequent analyses. To achieve this goal, we decided to construct a shape model $\tau(\mathbf{x}, \mu)$, describing each test shape as a deformation of a properly chosen reference shape.

First, we identified K control points on the boundary of the reference and test shapes by studying notable points on the two boundary curves, as shown in Figure 2b (for more details, refer to Section 3).




(a) Reference image obtained by averaging all the training datasets.




(b) Blue points represent notable points, while green points extends this set uniformly.

Figure 1

Then, an optimization problem of matching the R notable points obtained from the reference figure and the T notable points obtained from the test figure was solved, resulting in $K' = \min\{R, T\}$ total control points. Given two sets of control points ($I$ the smaller one, $J$ the bigger one) and named $d_{ij}$ the distance between control points $i \in I, j \in J$, the optimization problem reads as follows:

$$\begin{aligned} min. \quad & \sum_{i \in I, j \in J} d_{ij} y_{ij} \\ subject\ to: \quad & \sum_{j \in J} y_{ij} \geq 1, \quad \forall i \in I \\ & \sum_{i \in I} y_{ij} \leq 1, \quad \forall j \in J \\ & y_{ij} \in 0, 1, \quad \forall i \in I, j \in J \end{aligned} \tag{1}$$

Basically, the optimization procedure matches each control point of $I$ with an unique point of $J$, in such a way that the cumulative distance between paired control points is minimized.

Since the $K'$ points obtained resulted in sparsely populated regions on the surface, it was decided to expand the set of control points by densifying the low-density areas, resulting in a total of $K$ control points (see Figure 2b).

Once the sets of control points are defined, the following shape model is obtained:

$$\tau(\mathbf{z}, \mathbf{w}) = \mathbf{z} + \sum_{i=0}^{K} \mathbf{w_i}\ \phi(||\mathbf{z} - \mathbf{x_i}||)$$

where $\{\mathbf{x_i}\}_{i=1}^{K}$ are the control points on the reference shape, $\{\mathbf{w_i}\}_{i=1}^{K}$ are the weights vectors (size $D = 2$) and $\phi$ is a Gaussian radial basis function, i.e:

$$\phi(r) = exp(\frac{r^2}{\sigma^2})$$

Equivalently, the model reads algebraically:

$$\tau(\mathbf{z}, \mathbf{W}) = \mathbf{z} + \mathbf{W}^T \begin{bmatrix} \phi(||\mathbf{z} - \mathbf{x_1}||) \\ \phi(||\mathbf{z} - \mathbf{x_2}||) \\ ... \\ \phi(||\mathbf{z} - \mathbf{x_K}||) \end{bmatrix} \tag{1}$$

with $\mathbf{z} \in R^2$, $\mathbf{W} \in R^{K \times 2}$.

The matrix $\mathbf{W}$ is obtained by first setting the constraints:

$$\mathbf{x_j} + \delta_\mathbf{j} = \mathbf{x_j} + \mathbf{W}^T \begin{bmatrix} \phi(||\mathbf{x_j} - \mathbf{x_1}||) \\ \phi(||\mathbf{x_j} - \mathbf{x_2}||) \\ ... \\ \phi(||\mathbf{x_j} - \mathbf{x_K}||) \end{bmatrix}, \ \forall\ j = 1, ..., K$$

and then solving the system:

$$\mathbf{\Sigma}\ \mathbf{W} = \mathbf{\Delta} \tag{2}$$

where $\delta_\mathbf{j}$ is the vector of displacement of the control point $\mathbf{x_j}$,

$$\Sigma = \begin{bmatrix} \phi(||\mathbf{x_1} - \mathbf{x_1}||) & \phi(||\mathbf{x_1} - \mathbf{x_2}||) & \phi(||\mathbf{x_1} - \mathbf{x_3}||) & ... & \phi(||\mathbf{x_1} - \mathbf{x_K}||) \\ \phi(||\mathbf{x_2} - \mathbf{x_1}||) & \phi(||\mathbf{x_2} - \mathbf{x_2}||) & \phi(||\mathbf{x_2} - \mathbf{x_3}||) & ... & \phi(||\mathbf{x_2} - \mathbf{x_K}||) \\ & & ... & & \\ \phi(||\mathbf{x_K} - \mathbf{x_1}||) & \phi(||\mathbf{x_K} - \mathbf{x_2}||) & \phi(||\mathbf{x_K} - \mathbf{x_3}||) & ... & \phi(||\mathbf{x_K} - \mathbf{x_K}||) \end{bmatrix}, \quad \Delta = \begin{bmatrix} \mathbf{t_1} - \mathbf{x_1} \\ \mathbf{t_2} - \mathbf{x_2} \\ ... \\ \mathbf{t_k} - \mathbf{x_k} \end{bmatrix}$$

with $\{\mathbf{t_i}\}_{i=1}^{K}$ the test's control points.
The test image is then reconstructed applying the model (1) for any $\mathbf{z}$ in the domain.

### 1.2. Checkpoint 2: mesh construction

Subsequently, it is crucial to construct a mesh within the surface identified by the segmented boundary. The mesh is a 2-dimensional representation of the vascular structures reported in the image. This procedure discretizes the space and creates a structure for performing numerical calculations.

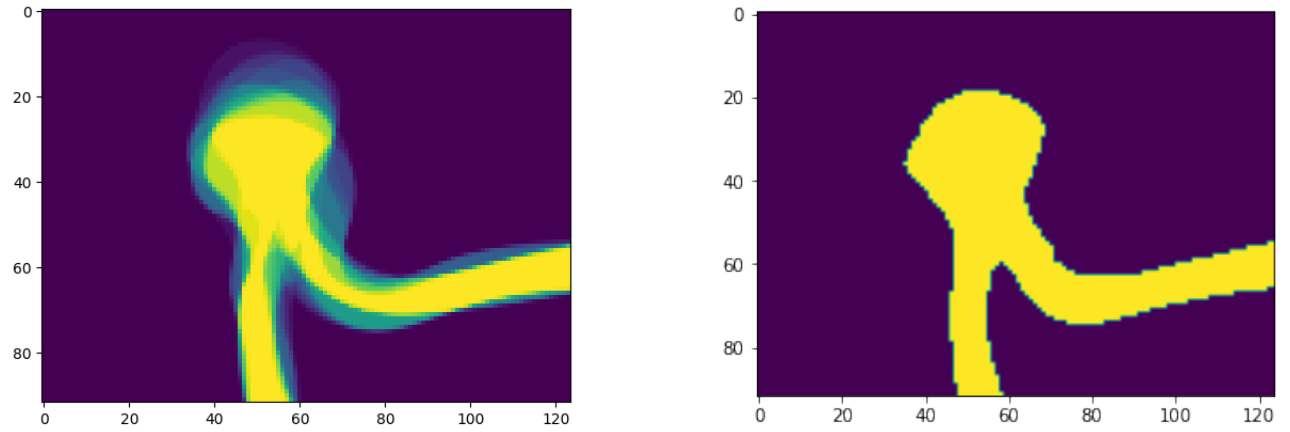
### 1.3. Checkpoint 3: numerical approximation of the physical system - pressure, velocity of the blood in the domain

Once the mesh is obtained, it becomes possible to reconstruct the behavior of blood flow within the structure. By using appropriate algorithms and models, the velocity and pressure of the blood are approximated in the region of the venous system affected by the aneurysm.

## 2. Methods

### 2.1. Checkpoint 1

To solve the segmentation problem, an **initial reference** figure was identified by averaging all the input training images. To determine its contour, a threshold was applied to the pixels of the resulting image: pixels with a value greater than 2 define the interior of the domain.



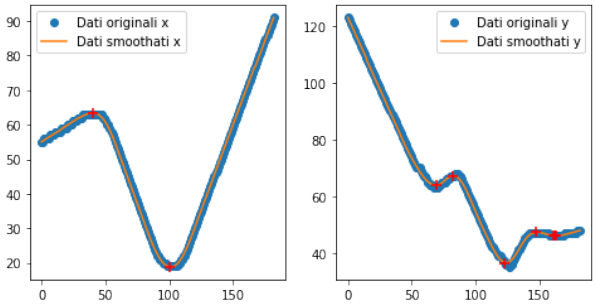

(a) Mean aneurism obtained by averaging all the training datasets.

(b) Reference image obtained by tresholding the average image.

Figure 2: Process to obtain initial image

Subsequently, by inputting the image into the function `get_control_points`, characteristic points were identified on the contour of the surface. Specifically, the function is capable of identifying the two curves defining the aneurysm surface (upper N and lower S), and subsequently locating the salient points.
This identification process involves studying the **stationary points** of the curves resulting from the parametrization along the x and y axes of each obtained curve. Thus, four function studies are performed for each image, as shown in Figure 3. It has been observed that many of the resulting points reflect recurrent characteristics in various aneurysms (see Figure 4).

Stationary points on North surface:




Stationary points on South surface:

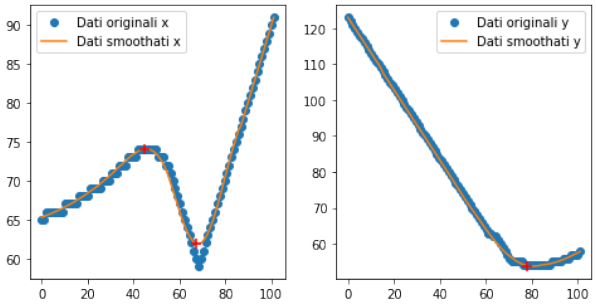
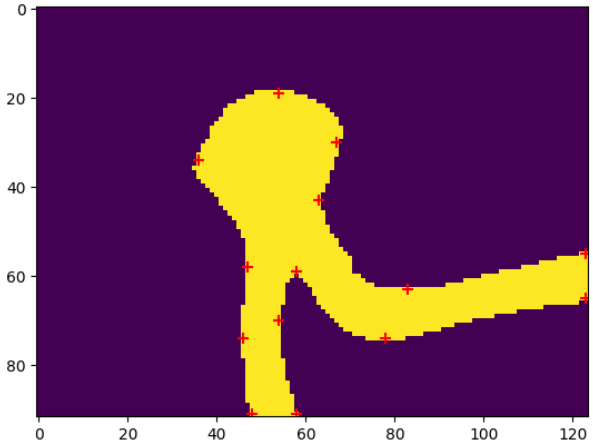
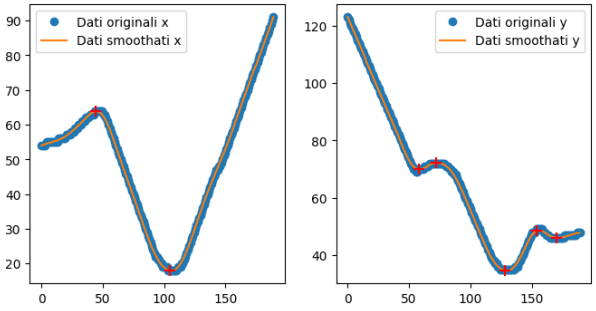



Figure 3: Salient points



---

Stationary points on North surface:




Stationary points on South surface:

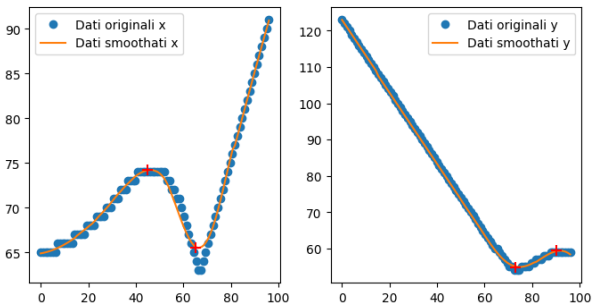
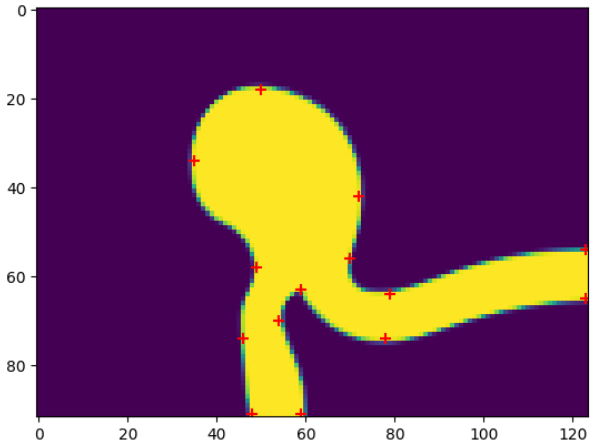



Figure 4: Salient points



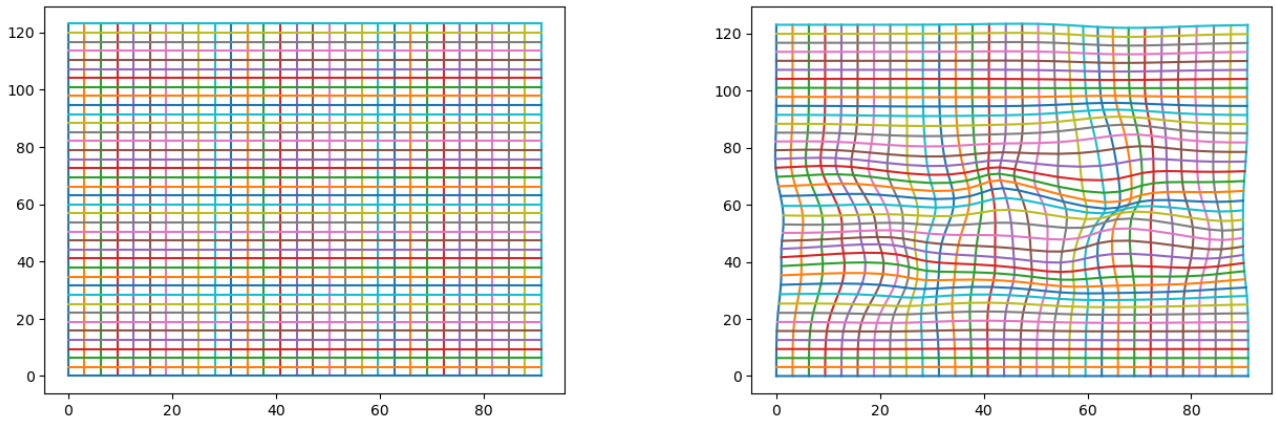
Applying the same function to the test images, T salient points are obtained, which need to be associated with the R salient points of the reference image.
At this point two cases can arise:

- **card(T) = card(R):** the two sets of points will be paired while maintaining the same number of points in each set;
- **card(T) ≠ card(R):** the set with the larger number of points should be reduced according to the number of points in the smaller set.

The pairing procedure is performed by the `matching` function, which takes the two sets of points R and T as input and solves the **matching optimization problem** in 1.1, returning the solution matrix Y.

Subsequently, using the `enlarge_salienti` function, the density of control points on the surface border is analyzed, and the obtained set of points is expanded at points with low density.
An additional matching is then performed between the obtained points.
At this point, the mathematical problem stated above is solved by calculating the displacement between the two sets of points $\mathbf{\Delta}$, the matrix $\mathbf{\Sigma}$, and subsequently finding the weight matrix $\mathbf{W}$ by solving the System at 2.

The results of this procedure are shown below:




Original mesh Deformed mesh

Figure 5: Mesh deformation

As we can see from Figure 5, the original nodes (with their binary value 0 or 1) have been moved by applying the shape model, but they keep their original value. Therefore, in order to return the result on a regular grid, an **interpolation step** is necessary: this is done by exploiting the function `griddata` from the `"scipy.interpolate"` library by calling:

$$z_{\text{reg}} = \text{griddata}((y_{\text{def}}, x_{\text{def}}), z_{\text{def}}, (y_{\text{reg}}, x_{\text{reg}}), \text{method='nearest'})$$

obtaining the values of the reconstructed image on the nodes of the regular grid.

## 2.2. Checkpoint 2

Since the mesh will be then used jointly with the adimensionalized Navier-Stokes equations, the mesh construction is performed on a normalized domain.
Once normalized the space, an initial set of boundary points uniformly distributed over the aneurysm's frontier is computed using the function `find_points_mesh`. The density of the boundary points and therefore the characteristic length parameter `lc` of the mesh are empirically tuned analysing the resulting mesh.

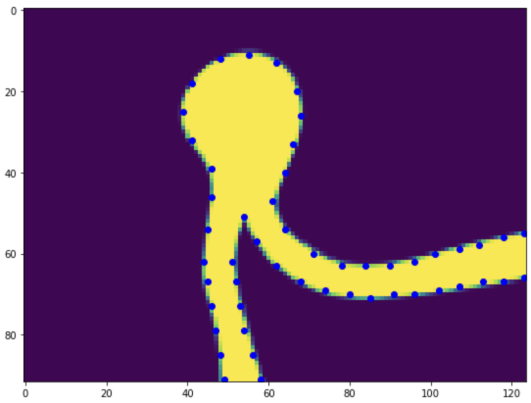



Figure 6: Mesh vertices

Subsequently, the mesh is built using the mesh-generation software Gmsh. Firstly, the previously obtained points are selected as the vertices of the mesh. Then the vertices are grouped into the corresponding portions of the boundary: the "inlet", the "outlet", and the sections of the "wall", which are the upper boundary curve and the left/right lower boundary curves. The inlet and the outlet are modeled using lines, while for the portions of the wall three different splines were used.

## 2.3. Checkpoint 3

The stationary dimensional Navier-Stokes equations for an incompressible fluid are:

$$\begin{cases} -\mu\Delta\mathbf{u} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p = 0 \\ \nabla\cdot\mathbf{u} = 0 \end{cases} \tag{2}$$

$$\textbf{adimensionalization} \downarrow$$

$$\begin{cases} -\frac{1}{Re}\Delta^*\mathbf{u}^* + (\mathbf{u}^*\cdot\nabla^*)\mathbf{u}^* + \nabla^* p^* = 0 \\ \nabla^*\cdot\mathbf{u}^* = 0 \end{cases} \tag{3}$$

where:
$\mathbf{u}$ is the velocity field of the fluid ($\mathbf{u}^*$ if adimensionalized),
$p$ is the pressure ($p^*$ if adimensionalized),
$\nu$ is the kinematic viscosity of the fluid,
$Re = \frac{UL}{\nu} = 162,5$ is the Reynolds number, since
- $\nu = 4 \times 10^{-6}\,\mathrm{m}^2/\mathrm{s}$
- $U = 0.1\,\mathrm{m/s}$ is the typical velocity (corresponding to the max velocity in the inlet region)
- $L = 6.5 \times 10^{-3}\,\mathrm{m}$ which is the typical length

The parameters' value has been chosen accordingly to the literature [4]. Since the equations have been normalized, the coordinates have also been adjusted accordingly. The original mesh, with dimensions [124x92 pixels], now has coordinates with values ranging from 0 to 1.35 (124/92) for the y-axis and from 0 to 1 for the x-axis (normalized to the original mesh dimensions).

### Boundary conditions

The BC of the problems have been set to:

**Inlet:** $\mathbf{u} = \mathbf{g}$ where $\mathbf{g}$ is a vertical parabolic profile, obtained as $\mathbf{g} = [0, \frac{4(x-x1)(x-x2)}{(x1-x2)^2}]$ with x1, x2 the two extrema of the inlet

**Wall:** $\mathbf{u} = 0$ (no-slip condition)

**Outlet:** $(\nabla \mathbf{u} - p) \cdot \mathbf{n} = 0$

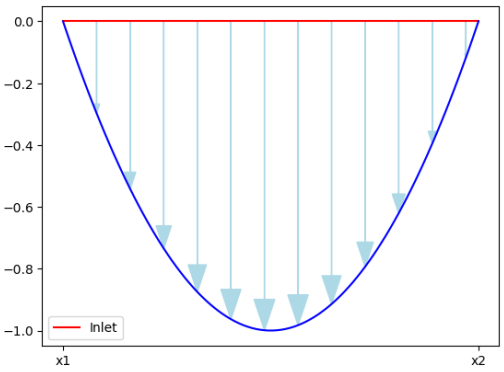



Figure 7: Inlet velocity profile

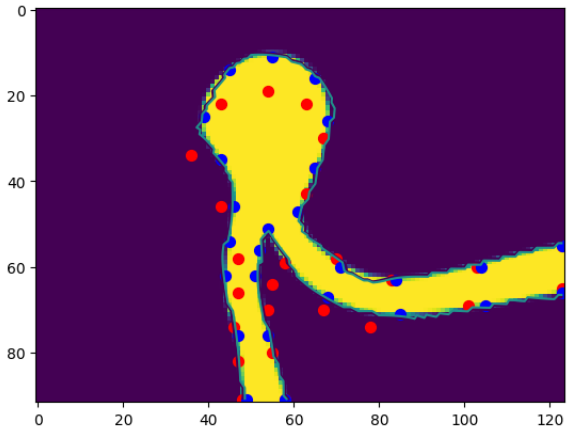
The equations were solved using the Newton method with P2 elements for velocity and P1 elements for pressure. Here is a snippet of the code, implemented in `FreeFem++` :

```
problem newton([uxh, uyh, ph], [vxh, vyh, qh]) =
  int2d(Th)(UGrad2(ux0h, uy0h, uxh, uyh)' * [vxh, vyh]
            + UGrad2(uxh, uyh, ux0h, uy0h)' * [vxh, vyh]
            + (Grad2(uxh, uyh)' * Grad2(vxh, vyh)) / Re
            - ph * Div(vxh, vyh)
            + Div(uxh, uyh) * qh)
  - int2d(Th)(UGrad2(ux0h, uy0h, ux0h, uy0h)' * [vxh, vyh])
  + on(1, uxh=0.0, uyh= 4*(x - x1)* (x - x2)/((x1-x2) * (x1-x2))
  + on(3, uxh=0.0, uyh=0.0);
```

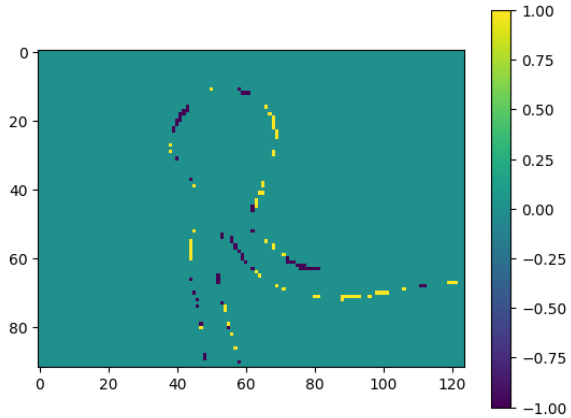
# 3. Numerical results

## 3.1. Checkpoint 1

In this section, numerical results for the best and worst reconstruction of the images from the 20 patients are provided.

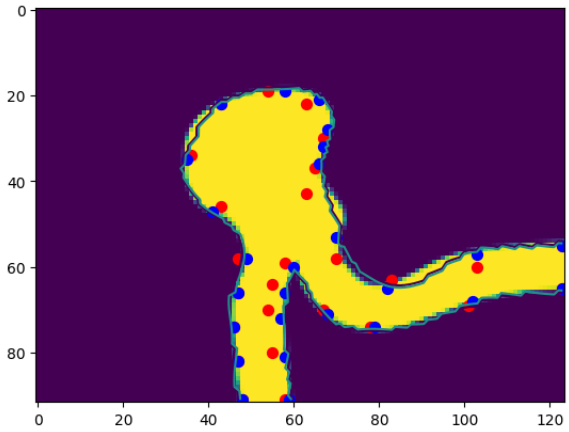



(a) Reconstruction (border) of the image. Red points are the original control points, blue points are the new control points associated to the image

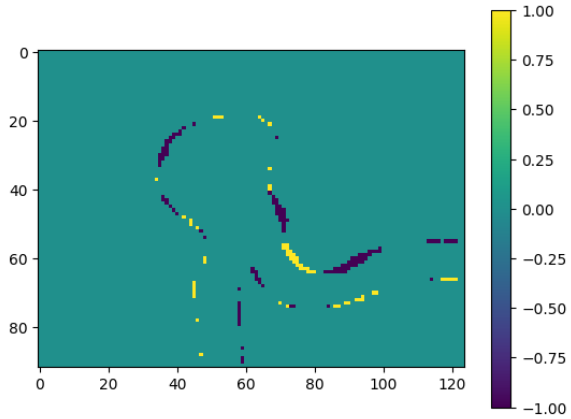



(b) Reconstruction error

Figure 8: Best case - Patient 18 aneurysm segmentation




(a) Reconstruction (border) of the image. Red points are the original control points, blue points are the new control points associated to the image

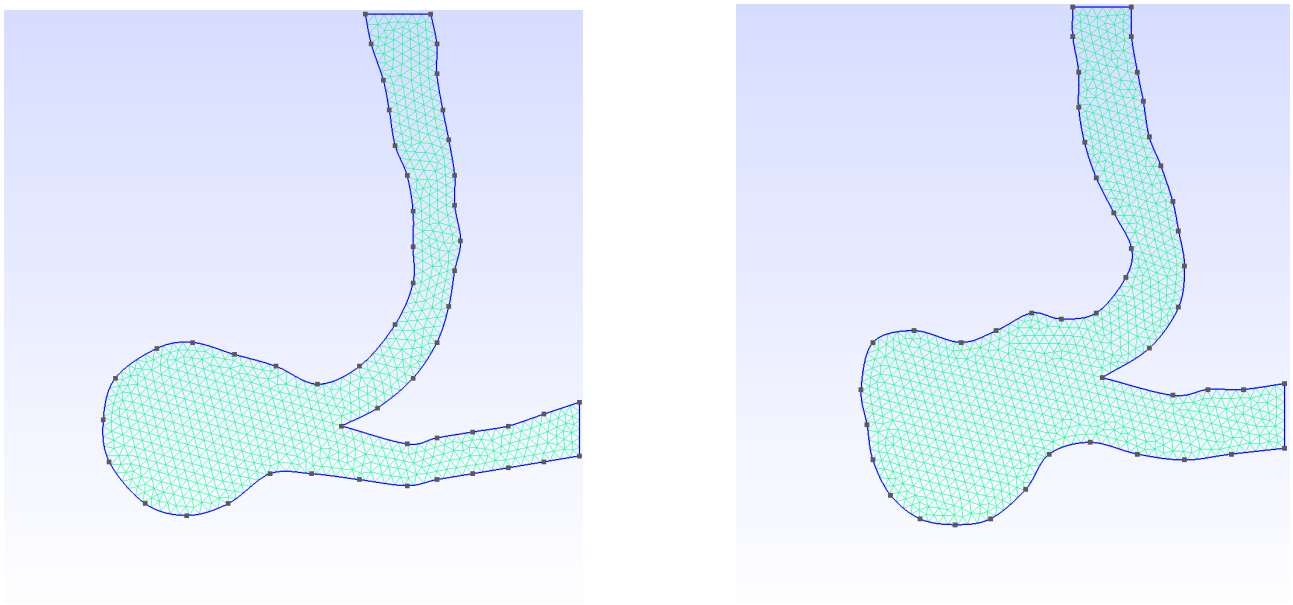



(b) Reconstruction error

Figure 9: Worst case - Patient 5 aneurysm segmentation

## 3.2. Checkpoint 2

Here the resulting meshes associated to the previously introduced patients 18 and 5 are reported.

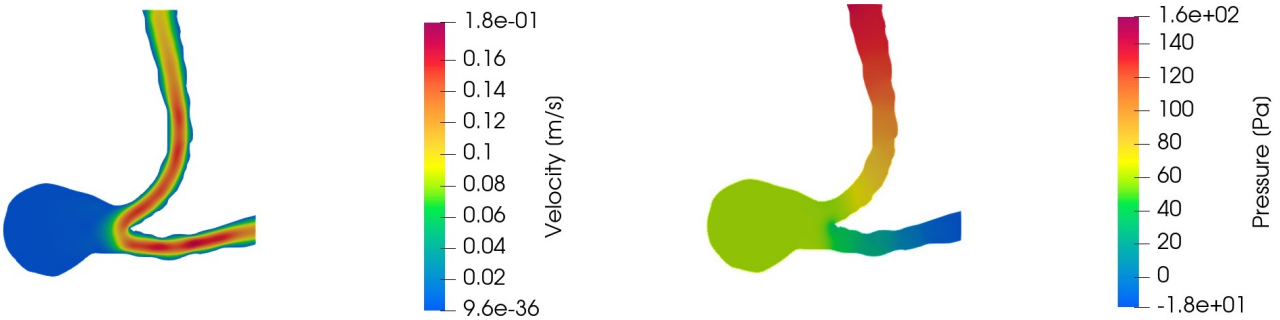


(a) Patient 18 (b) Patient 5

Figure 10: Meshes of aneurysm geometries

## 3.3. Checkpoint 3

In this section, the numerical results presented are the ones of patient 18, since it has been already introduced. The velocity field returned from the `FreeFem++` code is adimensional: then the real velocity field is reconstructed by multiplying the output by the factor U (characteristic velocity). The same holds for the pressure, where the factor is $\rho U^2$ (characteristic pressure).




Velocity field Pressure field

Figure 11: Patient 18 - Velocity and pressure fields

# 4. Conclusions

## 4.1. Checkpoint 1

The obtained results are satisfactory: the implemented algorithm is completely **automated**, and the user does not need to intervene (except for providing the image to be segmented as input). Furthermore, the achieved error is acceptable in all but rare cases, such as the one shown in Figure 9. In pathological cases, it may be necessary to sample more densely in certain areas.

## 4.2. Checkpoint 2

The mesh obtained seems to discretize quite accurately the input geometries, and represents a good structure on which performing the numerical approximation of velocity and pressure fields.

## 4.3. Checkpoint 3

The obtained pressure and velocity fields appear to be quite smooth, which is expected for low Reynolds numbers (~150) as the flow is characterized by slow velocities and high-viscosity fluids. In cases of higher Reynolds numbers, additional stabilization methods may be required to mitigate oscillations in the resulting fields

# References

1) Cardiac Ion Channels, Augustus O. Grant (https://doi.org/10.1161/CIRCEP.108.789081)
2) https://webhome.phy.duke.edu/ qelectron/pubs/BullMathBio65$_7$67.*pdf*
3) Brunton, S. L., Kutz, J. N. (2022). Data-driven science and engineering: Machine learning, dynamical systems, and control. Cambridge University Press.
4) Nader E, Skinner S, Romana M, Fort R, Lemonne N, Guillot N, Gauthier A, Antoine-Jonville S, Renoux C, Hardy-Dessources MD, Stauffer E, Joly P, Bertrand Y, Connes P. Blood Rheology: Key Parameters, Impact on Blood Flow, Role in Sickle Cell Disease and Effects of Exercise. Front Physiol. 2019 Oct 17;10:1329. doi: 10.3389/fphys.2019.01329. PMID: 31749708; PMCID: PMC6842957.