POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Homework report

Homework2

Scientific computing tools for advanced mathematical modelling

Authors: Federica Botta, Simone Colombara and Michele Di Sabato

Academic year: 2021-2022

---

1. Mathematical formulation of the problem

1.1. Problem conceptualization

Thrombosis occurs when blood clots form inside a blood vessel, blocking its path. The development of these clots can be localized inside the left atrial appendage (LAA), because in these areas blood is more likely to remain still, becoming stagnant and thus hardening.
The probability of this event is linked with the shape of the LAA, which can be clustered into four groups, according to what shape the appendage resembles: cauliflower (from here on out, CF), chicken wing (CW), windsock (WS) and cactus (CS). The likelihood of developing thrombosis grows, if the shape of the appendage has a higher tendency of slowing down the blood flow.

1.2. Design of the shape model

For the implementation of the mesh, we used a radial basis function approach. Initially, we selected the coordinates of the points on the template mesh, called $\{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^N$, then we chose some control points on the new domain, $\{(x_i^c, y_i^c)\}_{i=1}^N$, and their correspondent displacements $\{(\delta_i^x, \delta_i^y)\}_{i=1}^N$ with respect to the reference mesh. As Figure 1 shows, the choice of the location of the control points on the reference domain varies according to the mesh: for example, since the CF geometry is characterized by many peaks located on the right side of the geometry, more control points need to be placed in that region.

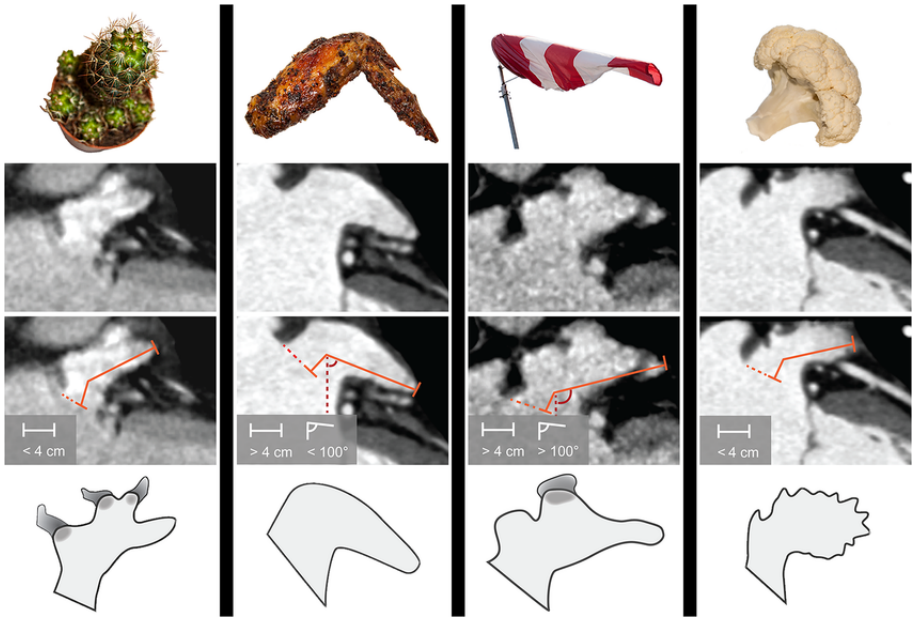


Figure 1: From left to right: cactus, chicken wing, windsock, cauliflower.

The general form of a radial basis function model is the following:

$$\tau(\boldsymbol{X};\epsilon) = \boldsymbol{X} + \boldsymbol{w}^T \begin{bmatrix} \phi(\|\boldsymbol{X}-\boldsymbol{X}_1\|) \\ \phi(\|\boldsymbol{X}-\boldsymbol{X}_2\|) \\ \phi(\|\boldsymbol{X}-\boldsymbol{X}_3\|) \\ \vdots \\ \phi(\|\boldsymbol{X}-\boldsymbol{X}_N\|) \end{bmatrix} \tag{1}$$

where $\boldsymbol{X}_i$ is the i-th control point on the reference domain, i.e. $\boldsymbol{X}_i = (\tilde{x_i}, \tilde{y_i})\ \forall i = 1, ..., N$.
In order to find the weight vector $\boldsymbol{w}$ we impose a condition on the control points, which is:

$$\tau(\boldsymbol{X}_i : \epsilon) = \boldsymbol{X}_i + \begin{bmatrix} \delta_i^x \\ \delta_i^y \end{bmatrix}$$

The weights $\boldsymbol{w}$ are thus the solution of the following linear system:

$$\begin{bmatrix} \phi(\|\boldsymbol{X}_1-\boldsymbol{X}_1\|) & \phi(\|\boldsymbol{X}_1-\boldsymbol{X}_2\|) \cdots \phi(\|\boldsymbol{X}_1-\boldsymbol{X}_N\|) \\ \phi(\|\boldsymbol{X}_2-\boldsymbol{X}_1\|) & \phi(\|\boldsymbol{X}_2-\boldsymbol{X}_2\|) \cdots \phi(\|\boldsymbol{X}_2-\boldsymbol{X}_N\|) \\ \phi(\|\boldsymbol{X}_3-\boldsymbol{X}_1\|) & \phi(\|\boldsymbol{X}_3-\boldsymbol{X}_2\|) \cdots \phi(\|\boldsymbol{X}_3-\boldsymbol{X}_N\|) \\ & \vdots \\ \phi(\|\boldsymbol{X}_N-\boldsymbol{X}_1\|) & \phi(\|\boldsymbol{X}_N-\boldsymbol{X}_2\|) \cdots \phi(\|\boldsymbol{X}_N-\boldsymbol{X}_N\|) \end{bmatrix} \boldsymbol{W} = \begin{bmatrix} \delta_1^x & \delta_1^y \\ \delta_2^x & \delta_2^y \\ \delta_3^x & \delta_3^y \\ \vdots \\ \delta_N^x & \delta_N^y \end{bmatrix} \tag{2}$$

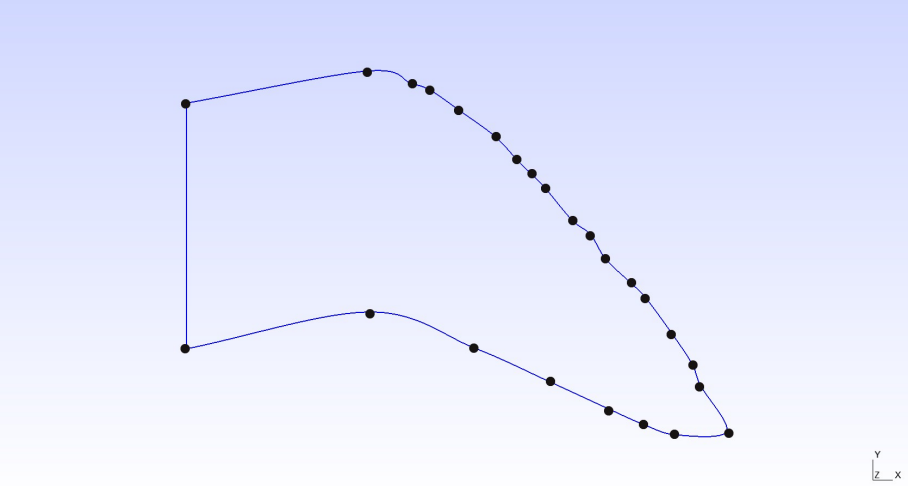
where $\boldsymbol{W} \in \mathbb{R}^{N\times 2}$.
The choice for ϕ depends on the geometry, while the hyperparameter ϵ has been tuned to avoid having overlapping elements in the mesh:

1. CS: multiquadratic, $\epsilon = 5$
2. CW: [already provided]
3. WS: multiquadratic, $\epsilon = 6$
4. CF: inverse quadratic, $\epsilon = 6$

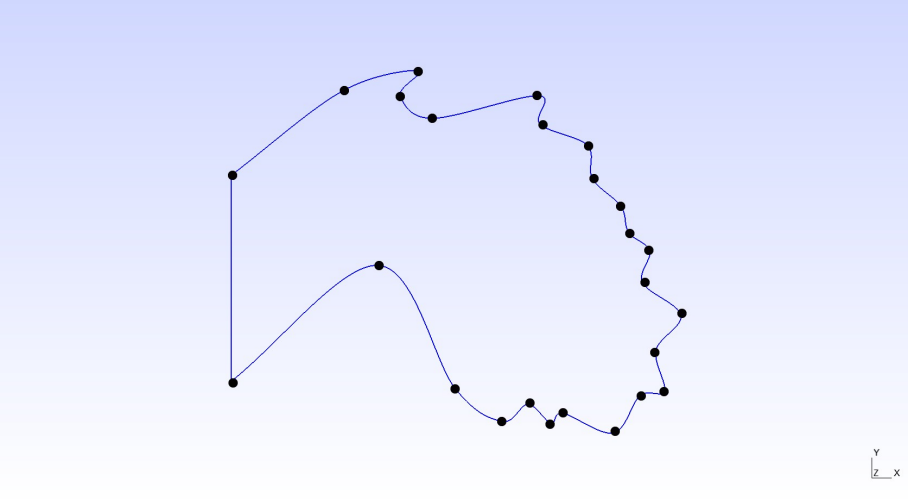
The mathematical expressions of the radial basis functions we used are:

- multiquadratic: $\phi(r) = \sqrt{1 + (\epsilon\ r)^2}$
- gaussian: $\phi(r) = e^{-(\epsilon\ r)^2}$
- inverse quadratic: $\phi(r) = \frac{1}{1+(\epsilon\ r)}$

The following plot displays the control points used for the cauliflower geometry:



(a) Template geometry



(b) Modified mesh

Figure 2: Radial inverse quadratic basis function applied to CF with $\epsilon = 6$

1.3. Numerical solver implementation

1.3.1 Dimension-full strong formulation

The Navier-Stokes equation formulation for this problem is the following:

Find u (velocity $[m/s]$) and p (pressure $[Pa]$) such that

$$\begin{cases} -\nu \cdot \Delta u + (u \cdot \nabla)u + \nabla \frac{p}{\rho} = 0 & \text{in } \Omega; \\ \text{div}(u) = 0 & \text{in } \Omega; \\ u = (0, -35) \text{ cm/s} & \text{on } \Gamma_{\text{inlet}}; \\ u = (0, 0) \text{ cm/s} & \text{on } \partial\Omega \setminus \Gamma_{\text{inlet}}. \end{cases} \tag{3}$$

where $\nu = \frac{\mu}{\rho} = 0.033 \ \frac{cm^2}{s}$ is the kinematic viscosity of blood.
Note that the boundary conditions are Dirichlet's, so it's a strong imposition of the value on the edges, for $\partial\Omega \setminus \Gamma_{\text{inlet}}$ the condition is homogeneous and for Γ_{inlet} is parallel respect to the axis y and directed downwards.

1.3.2 Dimensionless strong formulation

Defining the Reynolds' number as $Re = \frac{UL}{\nu}$, with U characteristic velocity (the velocity at inlet 35 cm/s) and L the characteristic length (the length of the inlet 1cm), we can rewrite equation (3) in the dimensionless form:

Find $u^* = \frac{u}{U}$ (dimensionless velocity) and $p^* = \frac{p}{\frac{1}{2}\rho U^2}$ (dimensionless pressure) with the dimensionless space variable $x^* = \frac{x}{L}$, such that:

$$\begin{cases} -\Delta^* u^*/Re + (u^* \cdot \nabla^*)u^* + \nabla^* p^* = 0 & \text{in } \Omega; \\ \text{div}^*(u^*) = 0 & \text{in } \Omega; \\ u^* = (0,-1)^T & \text{on } \Gamma_{\text{inlet}} \\ u^* = (0,0)^T & \text{on } \partial\Omega \setminus \Gamma_{\text{inlet}}. \end{cases} \tag{4}$$

For sake of simplicity, from now the on dimensionless variables will be used (u^*, p^*, x^*) but still written as (u, p, x).

1.3.3 Weak formulation

Setting $V = [H_1(\Omega)]^2$, $V_0 = \{v \in V \text{ and } v|_{\partial\Omega} = 0\}$, the space of the test function for velocity, and $Q = L_2(\Omega)$, the space of the test functions for pressure, and defining the linear operators using any test function $v \in V_0, q \in Q$:

- $a : V \times V_0 \to \mathcal{R}$ such that $a(u,v) = \int_\Omega \nabla u \cdot \frac{\nabla v}{Re}\, dx$
- $b : V_0 \times Q \to \mathcal{R}$ such that $b(v,p) = -\int_\Omega p \cdot \text{div}(v) dx$
- $c : V \times V \times V_0 \to \mathcal{R}$ such that $c(w,u,v) = \int_\Omega (w \cdot \nabla)u \cdot v\, dx$

The weak formulation of (4) becomes:

Find $u \in V$ such that $u = (0,0)^T$ on $\partial\Omega \setminus \Gamma_{\text{inlet}}$ and $u = (0,-1)^T$ on Γ_{inlet} and $p \in Q$ such that:

$$\begin{cases} a(u,v) + b(v,p) + c(u,u,v) = 0 & \text{in } \Omega \quad \forall v \in V_0 \\ b(u,q) = 0 & \text{in } \Omega \quad \forall q \in Q. \end{cases} \tag{5}$$

This system is non linear because of the presence of the operator $c(u,u,v)$, so it needs to be linearized. We chose the fixed-point method which is an iterative algorithm, indeed the Navier-Stokes equation can be seen as an advection-diffusion problem with an additional incompressibility constraint. If we assume that the advection $\boldsymbol{w}$ is given and such that $\text{div}(\boldsymbol{w}) = 0$, the following problem, known as the Oseen's problem, is obtained:

$$\begin{cases} a(u,v) + b(v,p) + c(w,u,v) = 0 & \text{in } \Omega \quad \forall v \in V \\ b(u,q) = 0 & \text{in } \Omega \quad \forall q \in Q. \end{cases} \tag{6}$$

We can therefore define an operator $T : V \to V$ which associates to any function $w \in V$ the solution $u \in V$ of the Oseen's problem:

$$T(w) = u$$

It is evident that the solution of the steady Navier Stokes problem is a fixed point of T, namely:

$$T(u) = u$$

It can be proved that for sufficiently small data, the operator T is a contraction, that is:
$\exists \rho < 1$ such that

$$||u_1 - u_2||_{H^1(\Omega)} = ||T(w_1) - T(w_2)||_{H^1(\Omega)} \leq \rho ||w_1 - w_2||_{H^1(\Omega)}, \qquad \forall w_1, w_2 \in V$$

where the small data condition is:

$$\frac{C||f||_{H^1(\Omega)}}{\nu^2} < 1$$

which in our case, being the problem homogeneous ($f = 0$), is certainly fulfilled. This implies that the fixed point iteration:

$$u^{k+1} = T(u^k)$$

converges to the unique solution of the steady Navier Stokes equations for any initial data u^0. Hence, the fixed-point method for equation (5) is the following:

Given $u^0 \in V$, for $k = 1, 2, \ldots$, find $u^k \in V$ such that $u^k = (0,0)^T$ on $\partial\Omega \setminus \Gamma_{\text{inlet}}$ and $u^k = (0,-1)^T$ on Γ_{inlet} and $p^k \in Q$ such that:

$$\begin{cases} a(u^k, v) + b(v, p^k) + c(u^{k-1}, u^k, v) = 0 & \text{in } \Omega \quad \forall v \in V_0 \\ b(u^k, q) = 0 & \text{in } \Omega \quad \forall q \in Q. \end{cases} \tag{7}$$

2. Methods

2.1. Shape Model

The method we used to model the four different shapes is divided into the following steps:

Algorithm 1 Shape model algorithm

for each geometry {CS,WS,CF} **do**

Define the position of a suitable number of control points, i.e. we write a .geo file and compile it, to get a .msh file, called `original_template_*.geo` and `original_template_*.msh`[1]

Write another .geo file (called `modified_template_*.geo`), positioning the control points to match the final shape

Run a python file, which reads the two .geo and the .msh file, extracts the relevant information, such as the location of the control points, and computes the shifts

The same python file then computes the weights of the shape model and applies the radial basis function to all the points contained in `original_template_*.msh`

The same python file writes an entire .geo file (called `[GEOMETRY_NAME].geo`), containing the coordinates of the points in the `original_template_*.msh` file, shifted according to the radial basis function

Finally, we use GMSH to compile the `[GEOMETRY_NAME].geo` file just created and obtain the corresponding `[GEOMETRY_NAME].msh` file, which will be uploaded into the FreeFemm++ code

end for

Remark 1: e.g. the cauliflower should have more control points on the right side, since it is characterized by more spikes in that region with respect to the other geometries.

Thanks to the automation of this procedure, we were able to experiment with different shape models in an efficient and timesaving manner.

2.2. FreeFem++ Solver

We set $X_h^r = \{v_h \in C^0(\Omega) : v_h|_K \in P^r(K)\ \forall K \in \tau_h\}$, for the velocity we set $V_h = [X_h^2]^2$ and for the pressure $Q_h = X_h^1$ (we take P2-P1 to have the inf-sup condition[1] valid). The following is the FreeFem++ implementation of equation (7):

```
fespace Vh(Th, [P2, P2, P1]);
fespace Ah(Th,P2);


Vh [ux0h, uy0h, p0h],
   [uxh, uyh, ph],
   [vxh, vyh, qh],
   [incrx, incry, incrp];
```

[1] The inf-sup condition is valid when $\exists \beta \in \mathcal{R} : \inf_{q \neq 0 \in Q} \sup_{v \neq 0 \in V} \frac{b(v,q)}{\|v_V\| q_Q} \geq \beta > 0$, if not the uniqueness of q is not guaranteed

```
problem fixedpoint([uxh, uyh, ph], [vxh, vyh, qh]) =
    int2d(Th)(UGrad2(ux0h, uy0h, uxh, uyh)' * [vxh, vyh]
              + (Grad2(uxh, uyh)' * Grad2(vxh, vyh)) / Re
              - ph * Div(vxh, vyh)
              + Div(uxh, uyh) * qh)
    + on(2, uxh=0.0, uyh=-1.0)
    + on(1, uxh=0.0, uyh=0.0);
```

To reach convergence we used a residual-based criterion: the algorithm stops at the k-th iteration if the residual $r^k = a(u_h^k, v_h) + b(v_h, p_h^k) + c(u_h^{k-1}, u_h^k, v_h) - b(u_h^k, q_h)$ is such that $||r^k||_{L^2} \leq \epsilon$:

```
    varf residual([ux,uy,p],[vx,vy,q]) =
    int2d(Th)(UGrad2(uxh, uyh, uxh, uyh)' * [vx, vy]
              + (Grad2(uxh, uyh)' * Grad2(vx, vy)) / Re
              - ph * Div(vx, vy)
              + Div(uxh, uyh) * q)
              + on(2, ux=0, uy=0)
              + on(1, ux=0, uy=0);
```

Finally, our algorithm can be described as:

Algorithm 2 FreeFem solver for steady Navier-Stokes equations

Define data and functions of the problem
$[u^0, p^0] = [0.0, 0.0]$
$[u, p] =$ **fixed-point**(u^0, p^0) ▷ At least one iteration is done')
$[u^0, p^0] = [u, p]$
for n = 1, ..., N_{max} **do**
  $[u, p] =$ **fixed-point**(u^0, p^0)')
  err= **residual**(u,p) ▷ L_2-norm of the residual
  if err $< \epsilon$ **then**
    break
  end if
  $[u^0, p^0] = [u, p]$
end for

A key point is selecting an appropriate threshold ϵ in order to reach an accurate solution in the lowest number of iterations possible. After some trials, we chose $\epsilon = 10^{-5}$.

Remark. *The choice of ϵ was done by decreasing this value until the indices defined in (3.2) become constant, thus demonstrating to have reached convergence.*

Remark. *Note that we used the fixed-point method, instead of the more rapid Newton method, the reason why is that the Newton method to converge needs an initial solution (u^0, p^0) close to the real solution (unknown), so we should have done some iterations with the fixed-point method and then starts with the Newton method. In the end, in this case, the number of the total iterations is the same for both situations, so we opted for using only one algorithm.*

Remark. *Also, instead of a stopping criterion based on the normalized increment of the solution in two consecutive time steps, we implemented a stopping criterion based on the residual to accelerate the convergence of the fixed-point method.*

3. Numerical results

3.1. Numerical results of FreeFem++ solver

The following plots are the results of the algorithm for each geometry:

- Chicken wing
  For this shape the convergence is reached at iteration 32 and the plot of the velocity is:

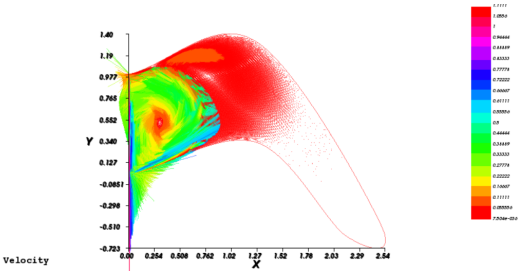


Figure 3: u_h plot for chicken wing

- Cactus
  For this geometry the convergence is reached at iteration 14 and the plot of the velocity is:

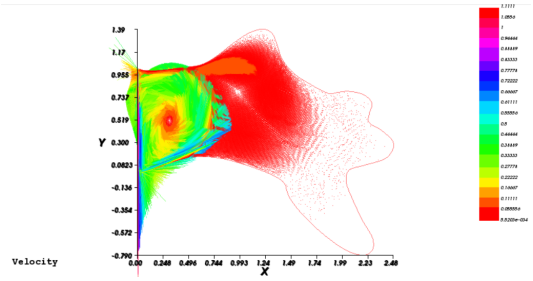


Figure 4: u_h plot for cactus

- Windsock
  Convergence is reached at iteration 45 and the plot of the velocity is:

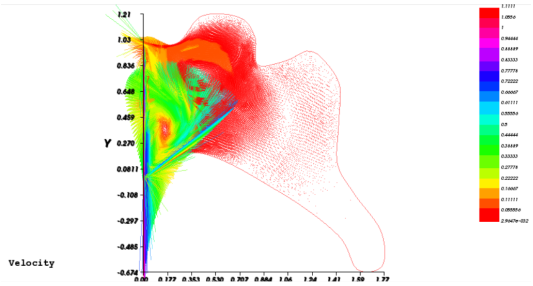


Figure 5: u_h plot for windsock

- Cauliflower
  For the final shape the convergence is reached at iteration 38 and the plot of the velocity is:

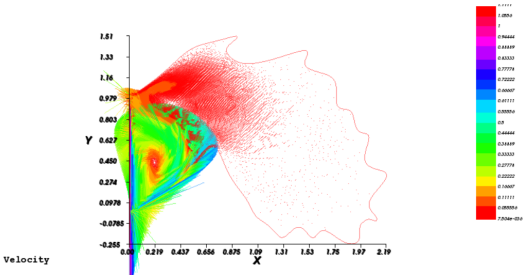


Figure 6: u_h plot for cauliflower

3.2. Definition of indicators

Concerning the computation of the probability of blood clots formation in the LAA for the different shape models, we designed the following indicator:

$$\frac{S_{still\ blood}}{S_T} \tag{8}$$

where $S_{still\ blood}$ and S_T represent the surface occupied by stationary blood (which may harden and form clots) and the total area, respectively.
This quantity can be rewritten as the integral over the domain of suitable indicator functions (Ω is one of the domains: WS, CW, CF, CS):

$$\frac{\int_\Omega \mathbb{I}_{x\in\Omega:\,|u(x,y)|<u_{max}}\, dxdy}{\int_\Omega \mathbb{I}_{(x,y)\in\Omega}\, dxdy} \tag{9}$$

where u_{max} is a threshold, which needs to be properly set. We set $u_{max} = 0.05556$, which seems appropriate based on the output of the FreeFem++ code. The final goal of this indicator is to rank the four geometries from least to most prone to blood clots. We remark that by slightly modifying the threshold u_{max} the numerical values of the indicators changed, but the ranking remained the same, indicating that the ranking itself is quite robust.
The FreeFem++ implementation is the following:

```
real ind1= int2d(Th)((abs(uh)<0.05556))/Th.area;
```

where `uh` is the total velocity defined in FreeFem++ as:

```
fespace Ah(Th,P2);
Ah uh;
uh=sqrt(uxh^2+uyh^2);
```

Note that `Ah` has the same polynomial degree as `Vh` for velocity.
The following table represents the final ranking obtained with the previously described index:

Table 1: Physical indicator

Shape	**Indicator**
Chicken wing (CW)	0.710269
Cactus (CS)	0.744876
Windsock (WS)	0.788661
Cauliflower (CF)	0.842357

The geometry most prone to blood clots is the cauliflower, while chicken wing seems to be the safest geometry. This is consistent with the literature, indeed the shape of the chicken wing is very regular with less lobes, hence

it is less capable to hold back the blood, unlike the cauliflower, which is characterized by many "spikes" and irregularities.
Finally, we tried to reproduce the same procedure using a different index. In [1] the problem of ranking the geometries was tackled advancing in time the Navier-Stokes equations and moving to a 3D domain. The authors of [1] modeled the flow of blood inside the LAA mixed with a colouring agent. They developed an indicator which depends on the amount of colouring agent still present in the cavities after the four heart cycles have finished: the more colouring agent left in the LAA, the more likely that type of appendage is to allow blood to stagnate and, therefore, harden. As a proxy[2] for this indicator, we used the ratio of stationary and total area, as already detailed. In the same paper, the following indicator was mentioned:

$$\frac{\mathbb{E}[u]}{S_T} = \frac{\int_\Omega u(x,y)dxdy}{\int_\Omega \mathbb{I}_{(x,y)\in\Omega}dxdy} \tag{10}$$

Which corresponds to the average velocity normalized with respect to the surface of the LAA.
According to this new index, the ranking becomes:

Table 2: Alternative physical indicator

Shape	**Indicator**
Chicken wing (CW)	0.0896409
Cactus (CS)	0.0803797
Windsock (WS)	0.060858
Cauliflower (CF)	0.0499455

Now, the lowest value indicates the shape with the higher probability for the clot formation. This ranking is coherent with the one of the previous indicator, which confirms the robustness of the model.

4. Bibliography

To get acquainted with the problem and the physics behind the equations, we studied some papers on the matter, namely: [1] and [2].

5. Conclusions

Our results are coherent with scientific literature. Indeed, the riskiest left atrial appendage shape should be Cauliflower, having lots of spikes that facilitate the formation of blood clots, while the Chicken wing, being the most regular shape model, should be the safest. The risk ranking obtained suggests a dependence on the LAA shape: the higher the number of irregularities in the geometry, the higher the probability of developing thrombosis.

We focused on the development of an **efficient** python tool that allows us to quickly and simply draw the shape models necessary to run the FreeFem++ simulations, as previously explained in Section 2.1. On the other hand, regarding the numerical solver, the choice of a fixed point method achieves satisfactory results in terms of the accuracy of the solution in a few iterations.

It is important to note that the results may vary according to the geometries obtained by the four shape models. This is not surprising nor a reason for suspicion, as the geometries themselves are an approximation of the real shape of the left atrial appendage.

The results proposed by our method are robust to the choice of index: by comparing the ranking resulting from using indicator (8) and indicator(10), we can observe the same ordering.

[2]Our model is in 2D and time is fixed, so we are not able to appropriately describe the flow of the agent inside the LAA.

References

[1] Giorgia Maria Bosi et al. Computational fluid dynamic analysis of the left atrial appendage to predict thrombosis risk. *Frontiers in Cardiovascular Medicine*, 2018.

[2] Miika Korhonen et al. Left atrial appendage morphology in patients with suspected cardiogenic stroke without known atrial fibrillation. 2018.