WATER QUALITY MANAGEMENT AND CONTROL OF POLLUTION IN LAKES

FACULTY OF MATHEMATICS

UNIVERSIDAD COMPLUTENSE
MADRID

XI Modelling Week UCM

Giulia Del Veccia
Ignacio Ferrero Melchor
Víctor Manuel González Vicente
Karina Haydeé Molina Martínez
Miriam Prieto Misol
Isabel Ruiz Arriaga

Madrid, June 2017

Contents

Introduction

Jaunay lake is a water reservoir of 3.700.000 m^3 located at the North-West of France and it has been exploited for recreational activities and human consumption.

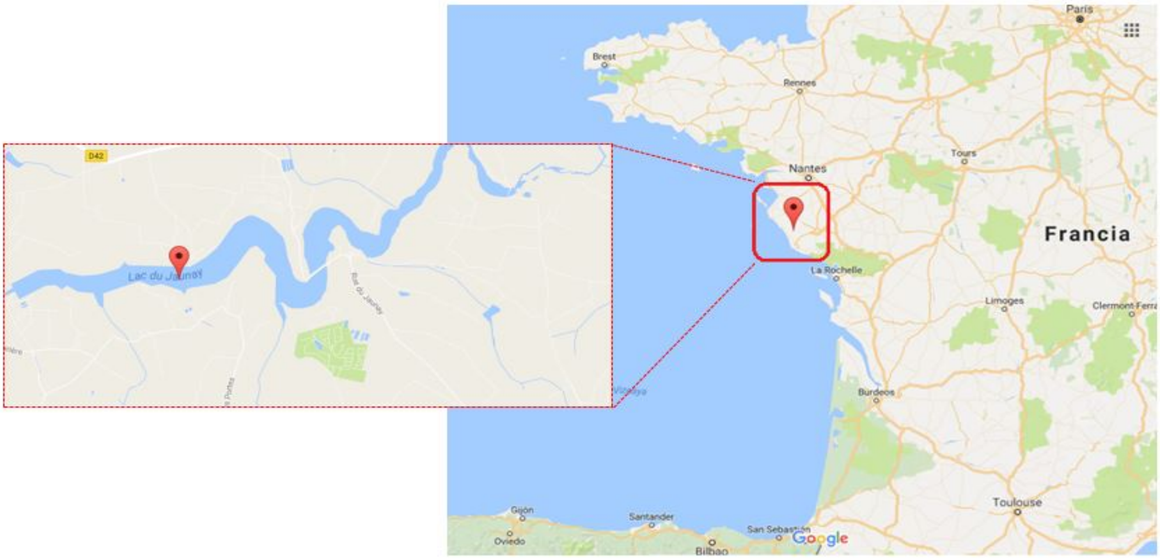

Figure 1: Geographic situation

Figure 2: Real aspect of the lake

The company Vendée Eau owns a water treatment plant located at a distance of 20 km from the lake, which is in charge of the water intake, its purification and its distribution to neighboring. This results in a reduction of the lake volume, which becomes alarming in dry seasons when this volume decreases to half of its capacity.

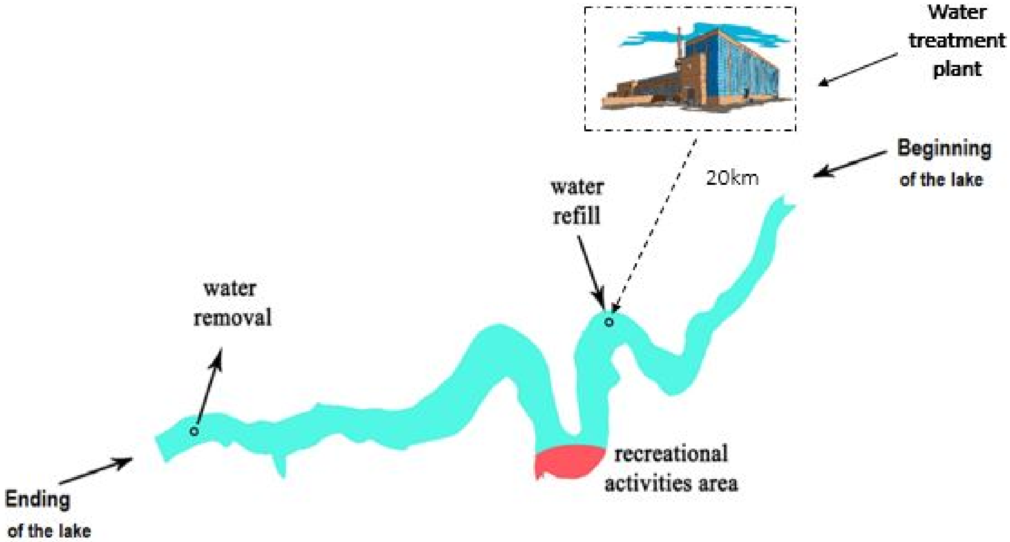

Figure 3: Situation of Company Vendée Eau

1. Problem description

Vendée Eau is trying to solve the volume problem refilling the lake with reused water coming from the water treatment plant.
The objective is to find the optimal location of the removal and refilling pipes in order to minimize the pollutant in the recreational area, taking into account that the value in this area must be lower than C_{max}. This value C_{max} is imposed by the company.

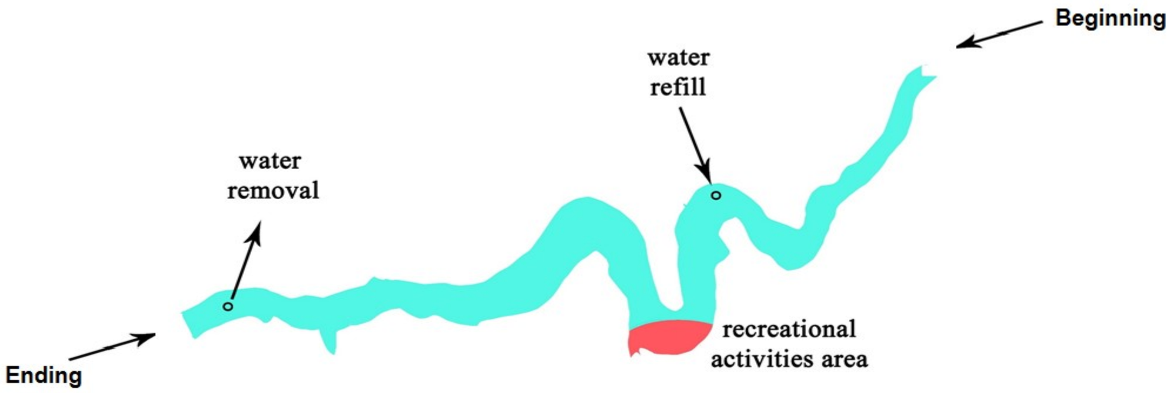

Figure 4: Lake layout - 1

Also, there are some restrictions related to the pipe's location. One of them must be located near the water treatment plant. Based on that, we suppose it must be in the area that is shown in the following figure.

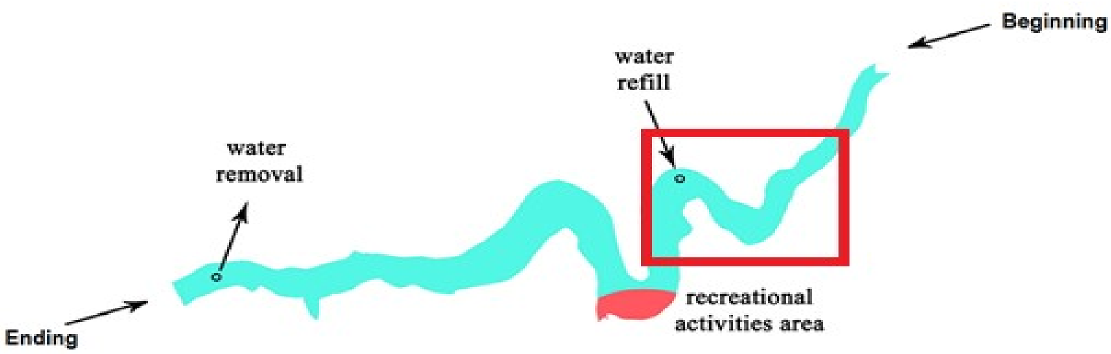

Figure 5: Lake layout - 2

1.1. Steps to solve the problem

For solving this problem, we will use the following steps:

1. Mathematical Modelling
2. Numerical Simulation (with COMSOL)
3. Optimization (with GOP)
4. Results
5. Conclusions

Using the tools *Comsol Multiphysics*, *Matlab* and *GOP*.

2. Mathematical modelling

In order to make the study easier, and observing that the pollutant is made in this model only of chlorine, which we know that remains in the surface of the lake, we considered a 2D model instead of a 3D one, supposing that the pollutant has a constant depth of $1m$.

2.1. Geometry of the domain

This is the first step in the mathematical model. The dimensions of the lake are the following ones:

- Length: 6150 m
- Mean width: 200 m

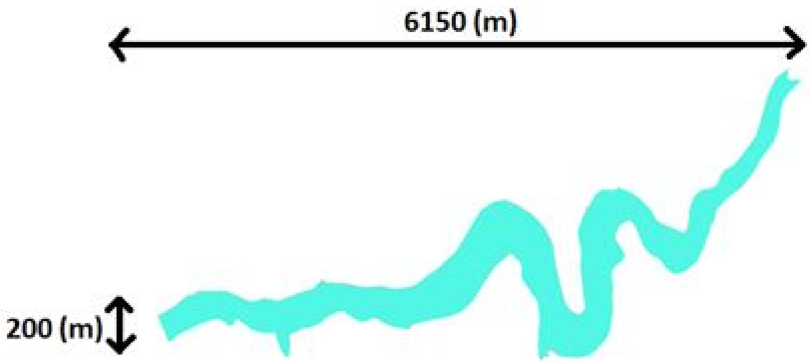

Figure 6: Dimensions of the lake

To obtain the geometry we start from a **Matlab code**, provided by our coordinators, which contains all the points of our domain. Using the Matlab function boundary we obtain those placed on the boundary.

The boundary function works as follows, executing k =*boundary(x,y)* Matlab returns a vector of point indices representing a single conforming $2-D$ boundary around the points (x, y).

Then we export that values from **Matlab** to **Comsol** and draw the geometry as an interpolation curve that passes throught all these points.

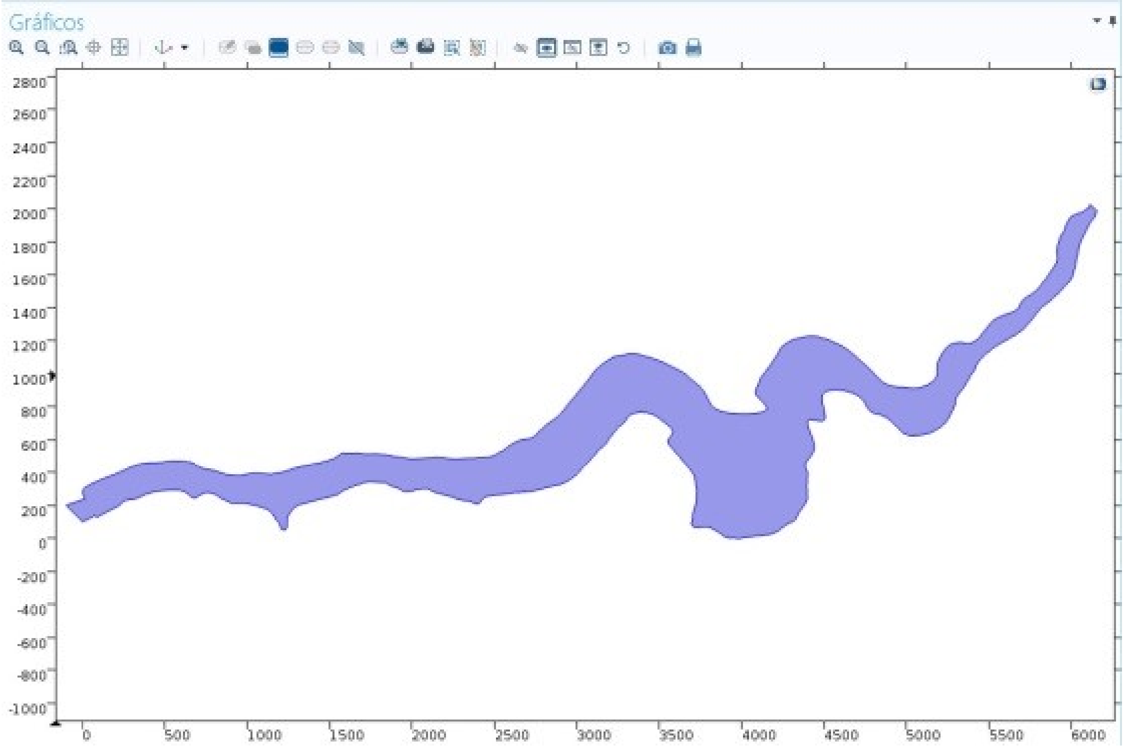

Figure 7: Lake in Comsol

2.2. Physical description of the pipes

- Both pipes have a 3 m radius.
- The speed at which water is removed and refilled is 0.1228 m/s.

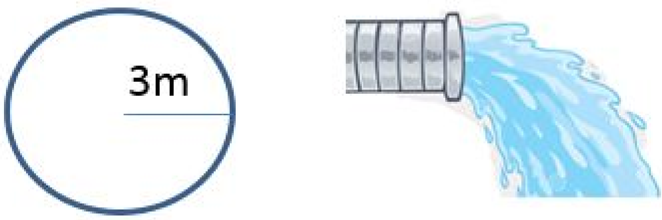

Figure 8: Radius and Speed of the Lake

2.3. Mathematical Model

The mathematical model is based on the following equation:

$$\frac{\partial c}{\partial t} = div(D\nabla c + \vec{u}c)$$

Which represents how the contaminant displace throught the lake in relation to:

- The value of it's coefficient difussion ($D\nabla c$). Bearing in mind that the pollution will tend to spread to less contaminated areas.
- A velocity field ($\vec{u}c$), known as advection, created by:
  - A laminar flow ($\vec{u}$), which is base on the *Navier Stokes equations.* In physics it describes the motion of viscous fluid substances. These balance equations arise from applying Newton's second law to fluid motion, together with the assumption that the stress in the fluid is the sum of a diffusing viscous term (proportional to the gradient of velocity) and a pressure term-hence describing viscous flow.
  - The wind ($\vec{v}$), which is a time-dependent function.
  - The velocity field can be written as:

$$u_x + drag \cdot v_x$$

$$u_y + drag \cdot v_y$$

It is important to take in account the drag factor, which represents the percentage of the wind that really affects the movement of the contaminant.

The model must be completed with:

- Initial conditions: $c = c_0$; where c_0 is the initial concentration of the contaminant in the lake.
- Boundary conditions:
  - $c = 0$. Condition for the beginning of the lake which represents that the water is entering from the river with no pollution.
  - $-n \cdot D\nabla c = 0$. Condition for the ending of the lake which represents that the water is going through, leaving the lake.
  - $-n \cdot N = 0$. Where $N = D\nabla c + \vec{u}c$. This last condition is for the rest of the boundary, considering it as a wall.
- Pump's restrictions:
  - $c = c_{in}$. Restriction for the refilling pump which represents the fact that the lake is being filled with contaminant water. Where c_{in} is the initial concentration.
  - $-n \cdot D\nabla c = 0$. Restriction for the removal pump which represents, as the same boundary condition, that the water is leaving the lake.

2.4. Data

- The data values used to solve the problem have been given by Vendée Eau.

Name	Value	Unit
Simulation Time	60	$days$
River's Velocity	0.0023148	m/s
Pump´s Radio	3	m
c_0	50	mg/m^2
D	0.04	m^2/s
drag	0.002	-
c_{in}	190	mg/m^2
C_{max}	250	mg/m^2

Figure 9: Data values

- The datawind have been extracted from the **Copernicus Marine Environment Monitoring Service**.

Figure 10: Copernicus Marine Environment Monitoring Service

Here it is an example of how this web site works:

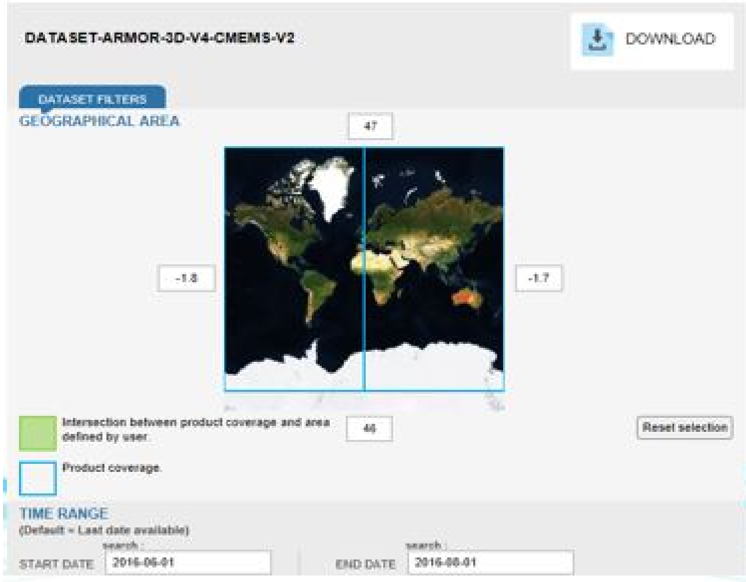

Figure 11: Example of Copernicus Marine Environment Monitoring Service

To obtain the data values of an specific area it is only necessary to enter its coordinates and a period of time, then the data values will be available for downloading.

3. Numerical Simulation

3.1. Comsol

COMSOL Multiphysics is a software that models and simulates scientific and engineering problems.

Figure 12: Logo of Comsol Multiphysics software

First of all, as it can be seen in the Figure 13, the value of the parameters that had been provided by the company was included.

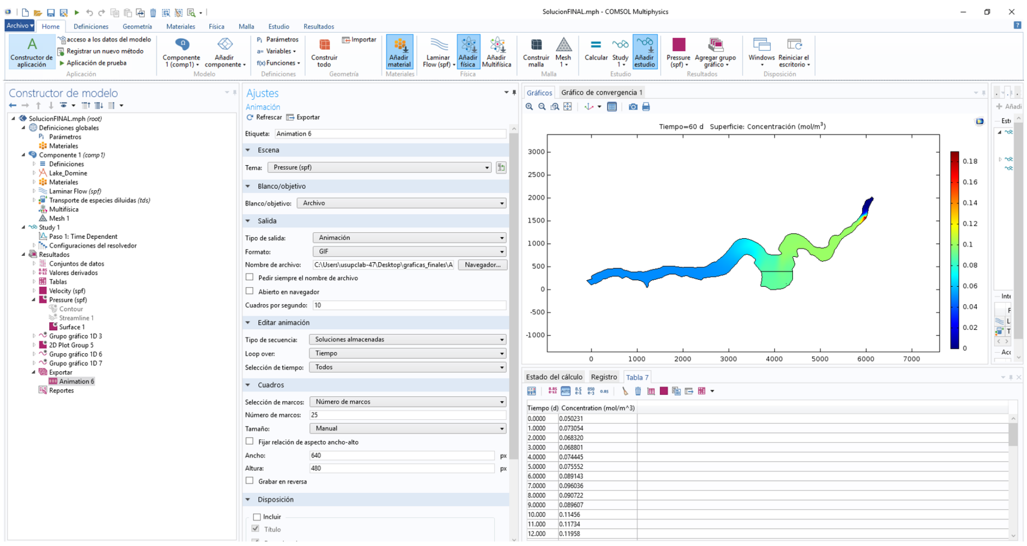

Figure 13: Graphical User Interface of Comsol

After the process that has been previously stated, regarding the definition of the domain, the coordinates have been imported and drawn. Additionally, since we do not know what will the solution to our problem be, the refilling and removal pumps have been randomly placed.

Secondly, the problem has been modeled. That is, we have defined the equations, the boundary conditions as well as the initial conditions.

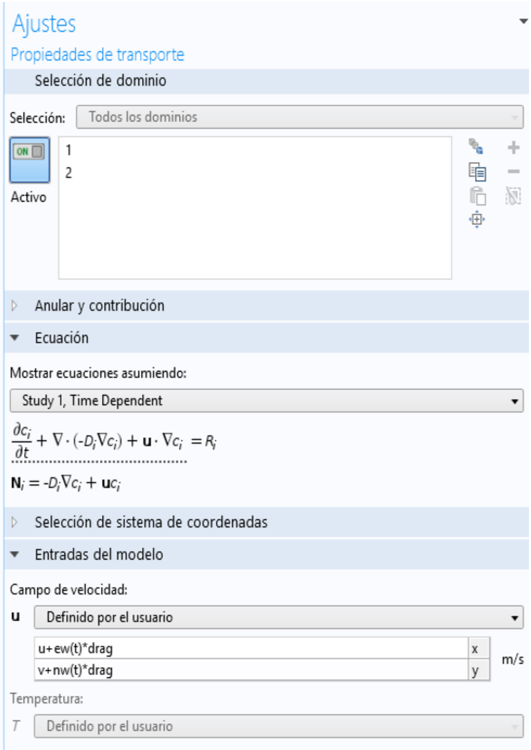

Figure 14: Modeling problem with Comsol

Finally, the mesh of the domain has been carried out. After defining the time unit, which in this case is the day, and the time interval, which was set in 2 months, we have proceeded to carry out the simulation.

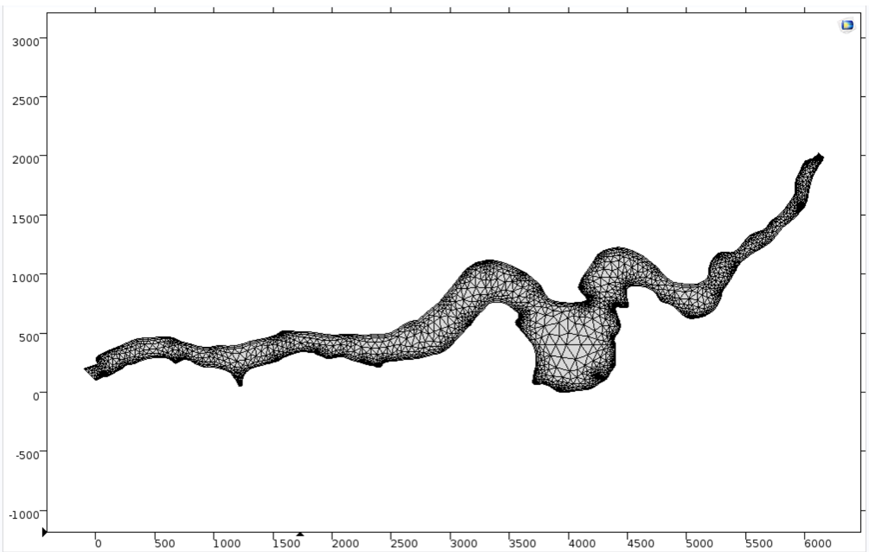

Figure 15: Mesh for solving numerically the problem

Since the bombs have been randomly placed, we have decided to show two situations in which it is shown that randomness does not provide a feasible solution. In the first example, it can be seen how the concentration exceeds the maximum value imposed by the company.

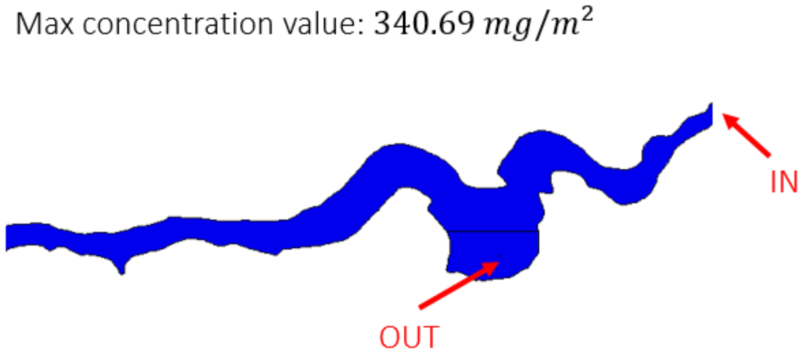

Figure 16: Random solution 1

Similarly, if we consider other coordinates for the pumps, the concentration again exceeds the allowed limit.

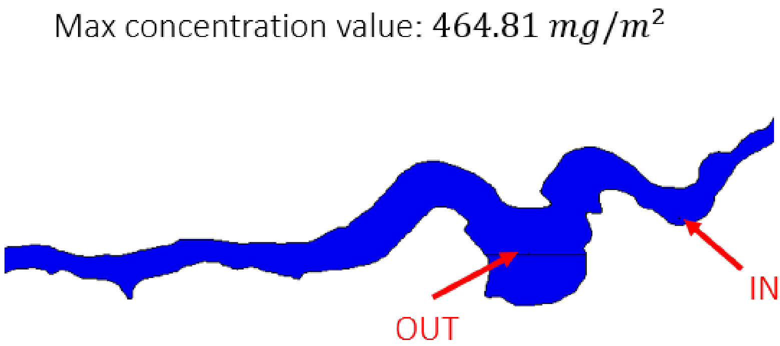

Figure 17: Random solution 2

3.2. Matlab

The next step is start working with MATLAB, converting the COMSOL code that contains the matemathical model into the MATLAB function

$$J = Lakes(X_{in}, Y_{in}, X_{out}, Y_{out}), \tag{3.1}$$

where the output J is an approximation of the following quantity

$$\int_0^T \iint_{\Omega_c} c(x, y, t) \; dx \; dy \; dt, \tag{3.2}$$

which is the integral of the concentration of the pollutant in the considered time (sixty days), and in space, specifically in the critical area Ω_c, that is the recreational area, shown in Figure 18.

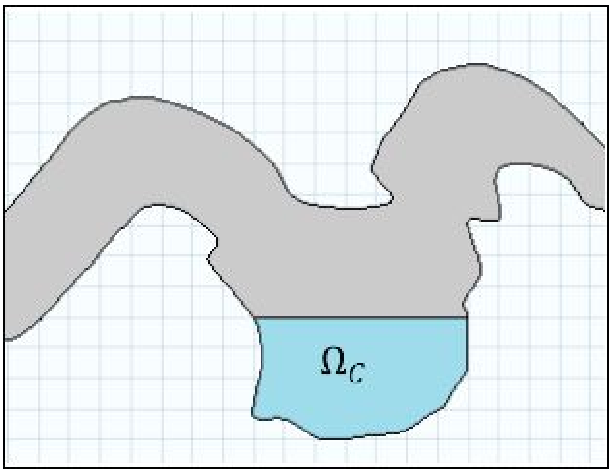

Figure 18: Image of the critical area Ω_c

As input, this MATLAB function needs the coordinates of the position of the two pumps, which are indicated with (X_{in}, Y_{in}) for the refilling pump, and (X_{out}, Y_{out}), for the removal pump.

To simplify the problem, we decided to locate the two pumps *inside* the lake; to do this, we created another MATLAB function, *inlake(xq,yq,X,Y)*, which controls if the point with coordinates (x_q, y_q) is inside the lake, whose boundary are descripted by the two vectors X and Y. The output of this function is actually the output of the MATLAB function *inpolygon(xq,yq,X,Y)*, on which the function inlake is based; this output is a boolean expression of 1, if the point (xq, yq) is inside the polygon expressed by the two vectors X and Y, or 0 otherwise.

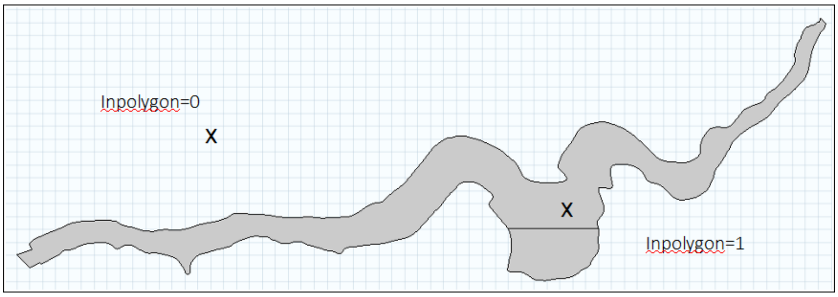

Figure 19: Example of how the function inpolygon works.

4. Optimization

The concentration value c, as well as the MATLAB function *lakes*, strictly depends on where the two pumps are located, ie from the values of (X_{in}, Y_{in}), (X_{out}, Y_{out}). In this section, we will define, analize and solve the optimization problem that will give us back the best position of the two pumps in order to minimize the value of the concentration of the pollutant in the recreational area in two months.

4.1. Definition of the optimization problem

At first, we define the optimization problem, which can be written as follows:

$$\min_{p \in \Omega} J(p), \tag{4.1}$$

where

$$J(p) = \int_0^T \iint_{\Omega_c} c_p(x, y, t)\ dx\ dy\ dt, \tag{4.2}$$

with whom we search the minimum of the concentration of the pollutant depending from the position of the pipes.

We choose to find the solution in $\Omega = \Omega_1 \times \Omega_2 \subset \mathbb{R}^4$, the restricted domain represented in Figure 20, due to several conditions and observations: referring to Ω_1, the company itself expressely asked to locate one of the two pumps in the first part of the lake, near to the water treatment plant; regarding Ω_2, after several random tries, we saw that the pollutant accumulates mostly in this area, which is also the area object of our study. For this reasons, and due to the lack of time, we focus on this restricted domain, instead of study the problem in the whole lake.

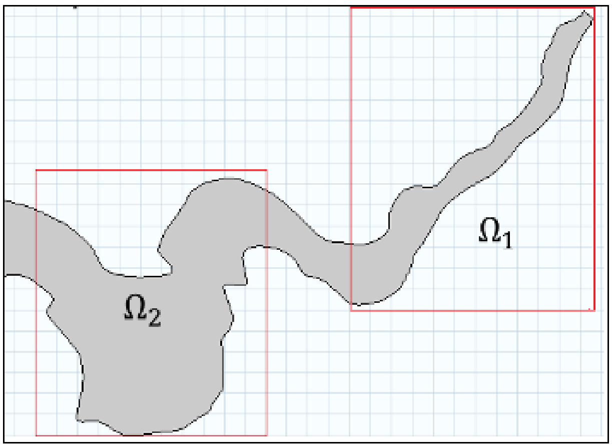

Figure 20: Image of the considered domain Ω.

4.2. Solve with GOP

Once we have defined the optimization problem to solve, we proceed to get a solution approach. We have donde this by using GOP (which stands for Global Optimization Platform), an optimization software developed by the group *MOMAT* from the UCM that regroups various class of determinist and stochastic optimization problems. In particular, we have used a genetic algorithm to get the approximation.

In order to use the genetic algorithm implemented in GOP, we have to create two matlab files, **userinit.m** and **userfunc.m**,

- userinit.m: Define initial conditions of the problem. In our case in this file we define the domain where the two pumps can be placed, $\Omega = \Omega_1 \times \Omega_2$.

- userfunc.m: Function to be optimized. In this case, as if we just define the function J to be estimated in each evaluation of the genetic algorithm, we would evaluate many points that are not inside our domain Ω, we first check if the coordinates of the pumps $p = (X_{in}, Y_{in}, X_{out}, Y_{out})$ chosen by the genetic algorithm are inside $\Omega = \Omega_1 \times \Omega_2$. Here is where we use the function *inpolygon* that we created. If so, $J(p)$ is evaluated by calling the function *lakes* previously defined, in other case, the output message PONT OUTSIDE THE LAKE is displayed.

  By using the function inpolygon we reduce the computacional cost, as the function J is only evaluated if the points are inside the domain.

After we have defined these two functions, we can open and run GOP. In the following figure the interface of GOP is shown.

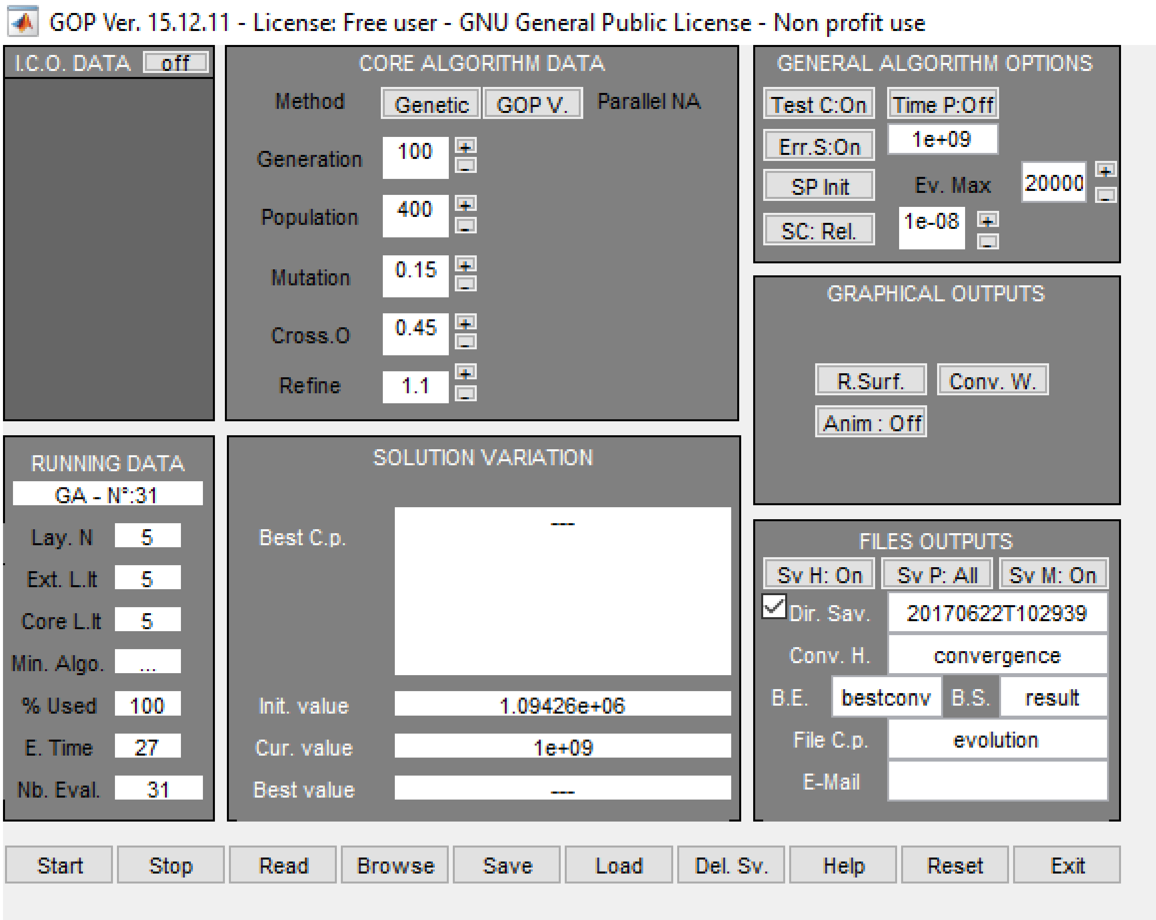

Figure 21: Interface of GOP

As we can see in the Figure 21, we have chosen in the Genetic Method and also fixed some parameters as the number of generations and population or the percentage of mutation. Once all the settings required are fixed, we only have to browse the folder where userinit and userfunc are and press start to run the genetic algorithm.

After running the algorithm for several hours, the best solution obtained is locating the pipes as in the following figure:

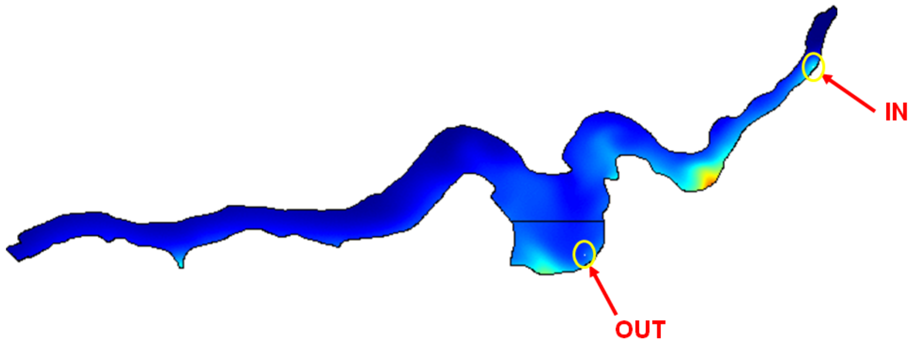

Figure 22: Best solution found with GOP

For this solution, the maximum concentration of the pollutant is $234.68mg/m^2$, which is below the maximum concentration allowed ($250mg/m^2$), so the solution fulfills the restriction imposed.

5. Results

Now, we are going to analyse the best solution we have obtained.

- If we do not consider the effect of wind, we get these results:

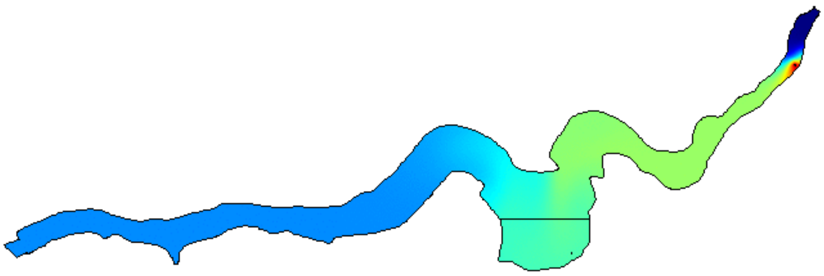

Figure 23: Evolution of pollutant without wind

- As we can see, the pollutant follows the natural course of the river.
- As might be expected, in this case drag factor is 0.

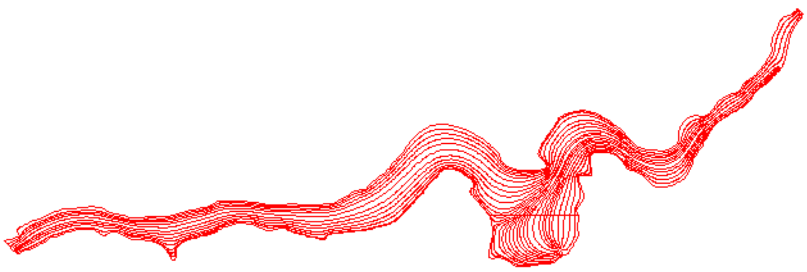

Figure 24: Streamlines without wind effect

- The streamlines show the trajectories of the Lagrangian particles of the lake in a particular moment. Some of them are absorbed by the output pump, and the others leave the lake.

- If we include the effect of the wind in our model:

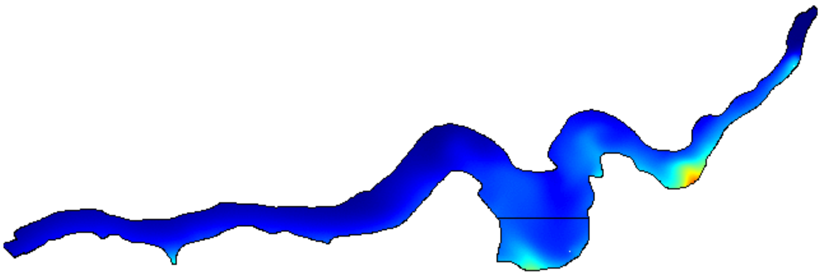

Figure 25: Evolution of pollutant considering wind force

- We can see that its influence is very relevant as we observe that the pollutant acummulates in these parts of the lake.
- This is the reason why the algorithm locates the out pump in this area.
- The drag factor is 0.0002. We have taken it from an article of a lake in Ontario (Canada) (see [3]) because we do not know this data of our river.

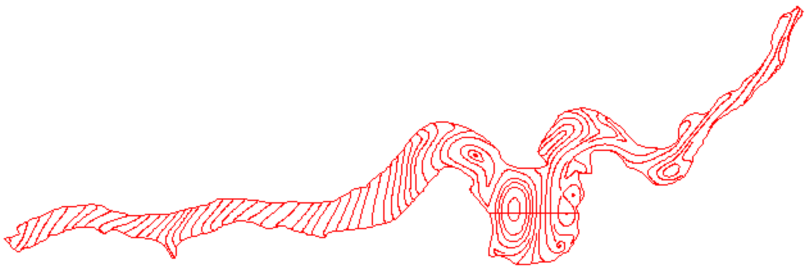

Figure 26: Streamlines considering wind effect

- In this case, because of the wind force, quiet and calm water zones (vortexes) appear.
- This fact can be used to improve our solution, for instance, restricting the localitation of the in pump.

- Finally, in the next graphic we see the temporal evolution of the maximum concentration of pollutant in the critical area.

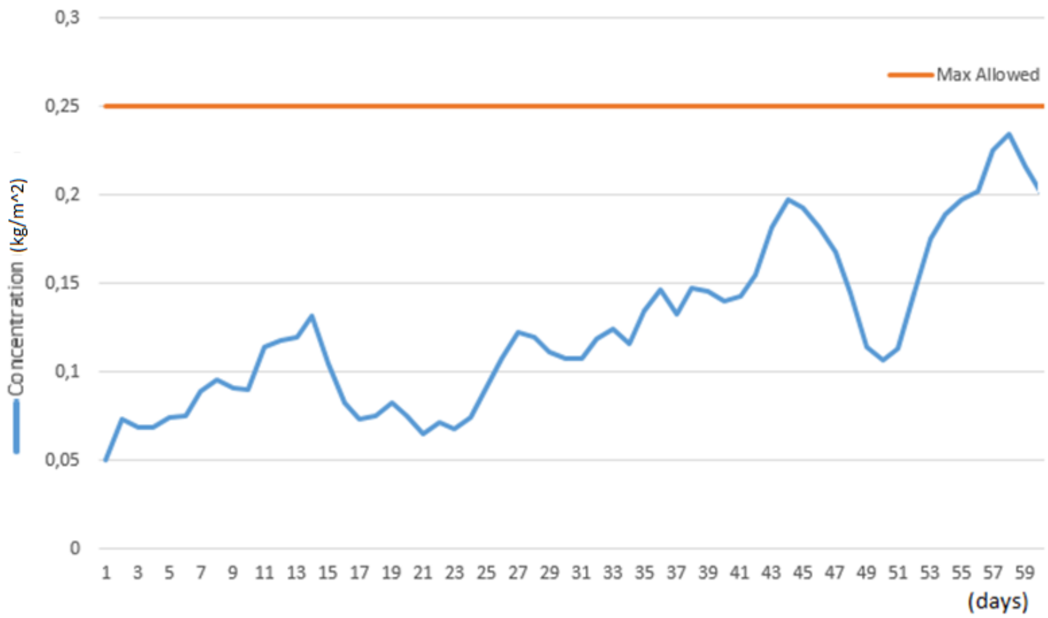

Figure 27: Temporal evolution of the maximum concentration in Ω_C

It can be seen that our solution never reaches the maximum value permitted, so the problem is solved.

6. Conclusions

Recalling the main steps we have followed to obtain a final solution, we first studied the geografic situation of the Jaunay lake and described the problem proposed: to refill the lake with reused water. In order to do that, our objective was to find the optimal location of two pipes in the lake, one to refill the lake with the contaminated water and another to remove the pollutant, minimizing the pollutant in the recreational area of the lake. We also had to take into account that the maximum concentration of the pollutant could not reach a fixed maximum value. The next step was to make the mathematical model of the problem, defining the geometry, the data and the equations involved in COMSOL. After this we converted the COMSOL code into a MATLAB function with the coordinates of the pumps as parameters. Finally, we stated the optimization problem, and obtained an approximation of the optimal solution by using a genetic algorithm implemented in GOP.

6.1. Recomendations

According to the best solution we have obtained, our final recomendations are to locate the refilling pump at the beginning of the lake and the removal pump in the recreational area and near the border.

6.2. Future work

As future work, a more complete and realistic model could be created including the space variation of the wind. Also, a better solution of the optimization problem would probably be obtained running the algorithm for longer, as our time for this was very limited.

References

[1] IVORRA, B. , GÓMEZ, S. , RAMOS, A. M. y GLOWINSKI, R. (2015) *Nonlinear Advection-Diffusion-Reaction Phenomena Involved in the Evolution and Pumping of Oil in Open Sea: Modeling, Numerical Simulation and Validation Considering the Prestige and Oleg Naydenov Oil Spill Cases.*

[2] WEIQJ, L. , LAWRENCE, S. , SUTTLES, S. (2002) *Drag Coefficients with Fetch-Limited Wind Waves.*

[3] ZACHRY, B. , LETCHFORD, C. , DELONG, Z. , SCHROEDER, J. , KENNEDY, A. (2009) *Surface Drag Coefficient Behavior Hurricane Ike.*