UNIVERSIDAD NACIONAL DE INGENIERIA
FACULTAD DE CIENCIAS

TESIS

"MODELLING AND SIMULATION OF THE CYANOBACTERIA BLOOMS DYNAMICS IN LAKE TAIHU, CHINA"

PARA OBTENER EL GRADO DE MAESTRO EN CIENCIAS
EN MATEMÁTICA APLICADA

ELABORADO POR
Joseph Luis Kahn Casapia

ASESOR:
Dr. Eladio Teofilo Ocaña Anaya

CO-ASESORES:
Dra. Céline Casenave
Dr. Antoine Rousseau

Lima, Perú

2018

Abstract

The present work deals with the modeling of the cyanobacteria blooms phenomenon from Lake Taihu in China by a 3D hydro-ecological model and the implementation of this model for the simulation.

The model is composed of: (1) a model of the lake hydrodynamics in two dimensions: we used the Shallow Water equations which are a particular case of Navier-Stokes equations, where the vertical dimension is neglected; (2) a Water Quality Model (WQM): we used the Water Quality Analysis Simulation Program(WASP) model in which are represented the reactions between ecological variables such as phytoplankton, oxygen, nitrogen and phosphorus, and the transport and diffusion of these substances by the fluid (in our case the water).

For the numerical resolution of the partial differential equations involved, the finite volume method was used with a non-uniform triangular mesh of the lake. The Navier-Stokes equations were solved independently in a first time to compute the current values. For this purpose, the free software called Finite Volume Coastal Ocean Model (FVCOM) was used. For the coupled hydro-ecological simulation, we developed a programme that performs the numerical resolution of the reaction-convection-diffusion equations, using the currents as inputs of the model. The simulations focus on the current effect and the coupling between hydrodynamics and water quality variables.

Finally, to analyze the model, we applied the Morris method, which is a Sensitivity Analysis method, to the water quality model. It gives us the most important parameters of the model which will be useful in the next step to calibrate the model for the specific case of the lake Taihu.

Contents

Chapter 1

Introduction

Lake Taihu, the third largest freshwater lake in China, is the main source of drinking water supply for several millions of people of the Yangtze Delta plain and also provides water for the agriculture. During a long time, its water was known for its high quality and cleanliness. However, due to the fast industrialization and urbanization of the region, the quality of the water is now decreasing tragically. Indeed, the human and industrial nutrient loadings are now so important that the lake doesn't manage to purify its water anymore. The pollution of the lake causes serious eutrophication problems. One of the consequences of the lake eutrophication is the proliferation of cyanobacteria (also known as "blue-green algae") whose periods of quick growth, called blooms, occur more and more often. During these blooms, the cyanobacteria form a thick green foam, disgraceful and foul, which floats on the surface of the water and which can have important economic consequences, especially in touristic areas. But the consequences may be more serious. Some cyanobacteria (like Mycrocystis) produce cyanotoxins that are toxic to animals and humans. In May 2007, in Taihu, the blooms led to a severe water crisis that let thousands of people without water for several days [1].

In the case of Lake Taihu, we know that the cyanobacteria proliferation is one of the consequences of the lake eutrophication, that is of the oversupply of nutrients (phosphorus and nitrogen). The low predation pressure, the heat and the high residence time also favor these phenomena. However, recent studies show that bacteria could play an important role in cyanobacteria population dynamics. By decomposing organic matter not assimilable by cyanobacteria into inorganic matter, bacteria promote the growth of cyanobacteria. In exchange, cyanobacteria provide a protected environment for bacteria when they agglomerate. Other assumptions are also made. It is commonly accepted that phosphorus is the limiting nutrient of cyanobacteria population in lakes. This hypothesis, a priori realistic in the case of nitrogen-fixing cyanobacteria, is questioned in other cases such as in lake Taihu where the dominant cyanobacteria are Microcystis, a non-nitrogen-fixing species. This cyanobacterium has also developed very efficient phosphorus acquisition strategies - storage of available phosphorus when it is abundant, enzyme secretion to degrade organic phosphorus - which could play an important role in its population dynamics.

To study this problem, the French National Institute of Agricultural Research

(INRA) in France and the Nanjing Institute of Geography and Limnology, Chinese Academy of Science (NIGLAS) in China have teamed up to carry out a project on the effect of anthropogenic changes in C/N/P ratios on the cyanobacteria proliferations in lakes: this project is called the ANSWER project (Analysis and Numerical Simulation of Water Ecosystems in Response to anthropogenic environmental changes). This project involved teams of 6 institutes which are: the French National Institute of Agricultural Research (INRA), the Nanjing Institute of Geography and Limnology, Chinese Academy of Science (NIGLAS), a French research institute dedicated to computational sciences (INRIA), the Water - Environment - Urban Systems Laboratory (LEESU), the Institute of Ecology and Environmental Sciences Paris (iEES), the Laboratory of Environmental Biotechnology (LBE).

Chapter 2

Hydrodynamic

The hydrodynamics of a lake is the dynamics of the water of the lake (speed, lines of current) which, in the case of Lake Taihu, is mainly affected by the effect of wind at the surface of the lake. Hydrodynamics is described by the Navier-Stokes equations composed of the continuity equation and the momentum equations for the current and the equations of the other state variables that are the temperature and the salinity. More details can be found in the following references: [2], [3], [4], [5], [6] and [7].

In the sequel, one part of the text was taken from the FVCOM user manual [2] and also of the Numerical Methods for Shallow Water flow [7].

2.1 Navier-Stokes equations

2.1.1 General equations

The governing equations of the hydrodynamics consist in the following momentum, continuity, temperature and salinity equations:

$$\begin{aligned}
\frac{\partial u}{\partial t} + \frac{\partial (u^2)}{\partial x} + \frac{\partial (uv)}{\partial y} + \frac{\partial (uw)}{\partial z} &= -\frac{1}{\rho_0}\frac{\partial p}{\partial x} + \nu \Delta(u) \\
\frac{\partial v}{\partial t} + \frac{\partial (uv)}{\partial x} + \frac{\partial (v^2)}{\partial y} + \frac{\partial (vw)}{\partial z} &= -\frac{1}{\rho_0}\frac{\partial p}{\partial y} + \nu \Delta(v) \\
\frac{\partial w}{\partial t} + \frac{\partial (uw)}{\partial x} + \frac{\partial (vw)}{\partial y} + \frac{\partial (w^2)}{\partial z} &= -\frac{1}{\rho_0}\frac{\partial p}{\partial z} + \nu \Delta(w)
\end{aligned} \tag{2.1}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{2.2}$$

$$\frac{\partial T}{\partial t} + \vec{U} \cdot \nabla T = \frac{\partial}{\partial z}\left(K_h \frac{\partial T}{\partial z}\right) + F_T \tag{2.3}$$

$$\frac{\partial S}{\partial t} + \vec{U} \cdot \nabla S = \frac{\partial}{\partial z}\left(K_h \frac{\partial S}{\partial z}\right) + F_S \tag{2.4}$$

where x and y are the east and north coordinates, z is the vertical coordinate in the cartesian coordinate system; u, v, and w are the x, y and z velocity components; T is the temperature; S is the salinity, ρ_0 is the fluid's density; p_a is the air pressure at fluid surface; p_H is the hydrostatic pressure; q is the nonhydrostatic pressure; g is the gravitational acceleration; ν is the viscosity coefficient; and K_h is the thermal vertical eddy diffusion coefficient. F_T and F_S represent the thermal and salt diffusion terms. The total water column depth is $\eta = h + z_b$ (see Figure 2.1), where z_b is the bottom depth (relative to $z = 0$) and η is the height of the free surface (relative to $z = 0$).

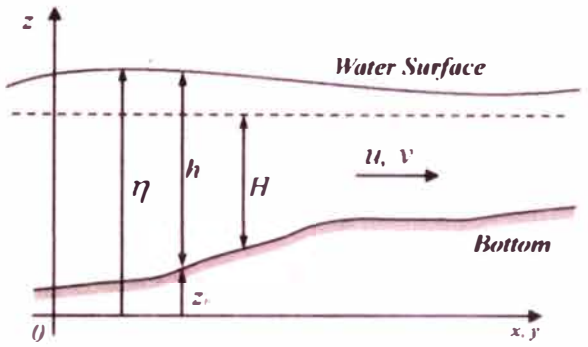


Figure 2.1. Water column in the lake; relationship between η, h and z_b: $\eta(x, y) = h(x, y) + z_b(x, y)$.

$p = p_a + p_H + q$ is the total pressure, it is express by a sum to remember what kinds of pressure compose the total pressure, in this context the following equation will use the total pressure. In the particular case of freshwater lake, the salinity S can be considered equal to zero.

2.1.2 Boundary conditions

The Boundary Conditions are an important information that is necessary to solve a PDE with a given initial condition. In order to reduce the Navier Stokes equations in the next section to obtain the shallow water equations, we consider some conditions in the vertical axis: at the bottom of the lake and at the surface of the water. We use the kinematic conditions which say that water particles will not cross both of these boundaries. At the bottom, the normal velocity component must vanish. At the surface, it must be moving by itself, which means that the relative normal velocity must vanish. We can write these physical conditions as follows:

Bottom Condition ($z = z_b(x, y)$)

The normal flow condition at the bottom is expressed by:

$$u|_{z_b} \frac{\partial z_b}{\partial x} + v|_{z_b} \frac{\partial z_b}{\partial y} - w|_{z_b} = 0, \tag{2.5}$$

and the no slip condition at the bottom is expressed by:

$$u|_{z_b} = v|_{z_b} = 0. \tag{2.6}$$

Consider these conditions 2.5 and 2.6, we get also:

$$w|_{z_b} = 0$$

Surface Condition ($z = \eta(x, y)$)

The relative normal flow condition at the surface is expressed by:

$$\frac{\partial \eta}{\partial t} + u|_\eta \frac{\partial \eta}{\partial x} + v|_\eta \frac{\partial \eta}{\partial y} - w|_\eta = 0. \tag{2.7}$$

Pressure

The pressure condition at the surface is expressed by:

$$p(x, y, \eta) = p_a, \tag{2.8}$$

where p_a is the air pressure at surface.

2.1.3 Hydrostatic approximation

In the sequel, we will make the hydrostatic approximation to the total pressure which relies on the following condition:

$$\boxed{\frac{\partial p}{\partial z} = -\rho_0 g}. \tag{2.9}$$

and we will consider the ρ_0, g and p_a as parameters independent of the x, y and z components. Then, integrating equation (2.9) over depth $\eta(x, y) - z$ and $\eta(x, y)$ and using the pressure condition at water surface (2.8), we get:

$$p(x, y, z) = \rho_0 g(\eta(x, y) - z) + p_a$$

which after derivation with respect to x and y leads to:

$$\boxed{-\frac{1}{\rho_0}\frac{\partial p}{\partial x} = -g\frac{\partial \eta}{\partial x}, \quad -\frac{1}{\rho_0}\frac{\partial p}{\partial y} = -g\frac{\partial \eta}{\partial y}}. \tag{2.10}$$

Using (2.9) and (2.10), the momentum equations can be rewritten:

$$\begin{aligned}
\frac{\partial u}{\partial t} + \frac{\partial (u^2)}{\partial x} + \frac{\partial (uv)}{\partial y} + \frac{\partial (uw)}{\partial z} &= -g\frac{\partial \eta}{\partial x} + \nu\Delta(u) \\
\frac{\partial v}{\partial t} + \frac{\partial (uv)}{\partial x} + \frac{\partial (v^2)}{\partial y} + \frac{\partial (vw)}{\partial z} &= -g\frac{\partial \eta}{\partial y} + \nu\Delta(v) \\
\frac{\partial w}{\partial t} + \frac{\partial (uw)}{\partial x} + \frac{\partial (vw)}{\partial y} + \frac{\partial (w^2)}{\partial z} &= -g + \nu\Delta(w)
\end{aligned} \tag{2.11}$$

2.2 Shallow water model

In shallow waters, the vertical velocity is not as important as the horizontal velocities because of the small depth. The idea is, therefore, to take advantage of that to make some simplifications in order to obtain a model whose computational cost is less expensive than one of the Navier-Stokes equations. This is what is done in the Shallow Water Equations (SWE). Before explaining how we obtain the SWE, we explain the Leibniz's rule.

2.2.1 Leibniz's rule

The following theorem is a simple case of the Leibniz's rule for two variables which will be used to prove a version more complicated by the Leibniz's rule in Theorem 2.2.2.

Theorem 2.2.1. *Let $f : [a,b] \times [c;d] \to \mathbb{R}$ a continuous function such that $\frac{\partial f}{\partial y}$ exist and is continuous. Let*

$$F(y) = \int_a^b f(x,y)dx, \qquad \forall y \in [c,d].$$

Then for all $y \in [c,d]$ there exists $F'(y)$ and

$$F'(y) = \int_a^b \frac{\partial f}{\partial y}(x,y)dx, \qquad \forall y \in [c,d].$$

Proof. Consider $y_0 \in [c,d]$ and $\epsilon > 0$. By the uniform continuity of $\frac{\partial f}{\partial y}$, there exist $\delta > 0$ such that for all $y, y' \in [c,d]$ such that $|y - y'| < \delta$,

$$\left|\frac{\partial f}{\partial y}(x,y) - \frac{\partial f}{\partial y}(x,y')\right| < \frac{\epsilon}{(b-a+1)}, \qquad \forall x \in [a,b] \tag{2.12}$$

Let $0 < |h| < \delta$, $h \in \mathbb{R}$. From the mean value theorem, there exist $\theta \in]0,1[$ such that

$$f(x,y_0+h) - f(x,y_0) = \frac{\partial f}{\partial y}(x,y_0+\theta h)h, \qquad \forall x \in [a,b] \tag{2.13}$$

Then, for $0 < |h| < \delta$ in the following expression

$$R = \left|\frac{F(y_0+h) - F(y_0)}{h} - \int_a^b \frac{\partial f}{\partial y}(x,y_0)dx\right|, \tag{2.14}$$

using equations (2.12) and (2.13), we get

$$\begin{aligned} R &= \left|\int_a^b \left(\frac{f(x,y_0+h) - f(x,y_0)}{h} - \frac{\partial f}{\partial y}(x,y_0)\right)dx\right| \\ &\leqslant \int_a^b \left|\frac{\partial f}{\partial y}(x,y_0+\theta h) - \frac{\partial f}{\partial y}(x,y_0)\right|dx \\ &< \int_a^b \left|\frac{\epsilon}{(b-a+1)}\right|dx \\ &= \frac{(b-a)\epsilon}{(b-a+1)} < \epsilon \end{aligned}$$

■

Theorem 2.2.2. *Let $R := [a,b] \times [c,d] \subset \mathbb{R}^2$ and consider*

(i) a continuous function $f : R \to \mathbb{R}$ such that $\frac{\partial f}{\partial y}(x,y)$ exists for all $(x,y) \in R$.

(ii) two differentiable functions $\psi, \varphi : [c,d] \to [a,b]$.

Denote $F(y) = \int_{\varphi(y)}^{\psi(y)} f(x,y)dx$, $\forall y \in [c,d]$. *Then,* F *is differentiable on* $[c,d]$ *and satisfies*

$$F'(y) = \int_{\varphi(y)}^{\psi(y)} \frac{\partial f}{\partial y}dx + f(\psi(y),y)\frac{\partial \psi}{\partial y} - f(\varphi(y),y)\frac{\partial \varphi(y)}{\partial y}.$$

Proof. For all $y \in U$ the function $F(y) = \int_{\varphi(y)}^{\psi(y)} f(x,y)dx$ exists because f is continuous. Consider the following notation: $G : [a,b]^2 \times [c,d] \to \mathbb{R}$ a function defined by

$$G(t_1,t_2,t_3) = \int_{t_1}^{t_2} f(x,t_3)dx, \qquad \forall t_1,t_2 \in [a,b], \forall t_3 \in [c,d].$$

We have

$$F(y) = G(\varphi(y),\psi(y),y), \forall y \in [c,d].$$

Then, using the chain rule in the function F in $(\psi(u),\varphi(y),y)$

$$\frac{\partial F}{\partial y} = \frac{\partial G}{\partial t_1}(\psi(y),\varphi(y),y)\frac{\partial \varphi}{\partial y}(y) + \frac{\partial G}{\partial t_2}(\psi(y),\varphi(y),y)\frac{\partial \psi}{\partial y}(y) + \frac{\partial G}{\partial t_3}(\psi(y),\varphi(y),y) \quad (2.15)$$

Now, using the mean value theorem and continuity of f, we deduce

$$\begin{aligned}
\frac{\partial G}{\partial t_1}(\psi(y),\varphi(y),y) &= \lim_{h\to 0}\frac{G(\psi(y)+h,\varphi(y),y) - G(\psi(y),\varphi(y),y)}{h} \\
&= \lim_{h\to 0}\frac{\int_{\psi(y)+h}^{\varphi(y)} f(x,y)dx - \int_{\psi(y)}^{\varphi(y)} f(x,y)dx}{h} \\
&= -\lim_{h\to 0}\frac{\int_{\psi(y)}^{\psi(y)+h} f(x,y)dx}{h} \\
&= -\lim_{h\to 0}\frac{f(\xi,y)\cdot h}{h}, \quad \xi \in [\psi(y),\psi(y)+h] \\
&= -f(\psi(y),y) \qquad (2.16)
\end{aligned}$$

Similarity, by the same step we deduce

$$\frac{\partial G}{\partial t_2}(\psi(y),\varphi(y),y) = f(\psi(y),y). \qquad (2.17)$$

From Theorem 2.2.1, we get

$$\frac{\partial G}{\partial t_3}(\psi(y),\varphi(y),y) = \int_{\varphi(y)}^{\psi(y)} \frac{\partial f}{\partial y}(x,y)dx. \qquad (2.18)$$

Using equations (2.16), (2.17) and (2.18) in the equation (2.15) we get the result. ■

In the following theorem we will demonstrate a general version of the Leibniz's rule using Theorem 2.2.2:

Theorem 2.2.3. *Let U be an open subset of R^n, $\psi, \varphi : U \to \mathbb{R}$ some continuous functions and*

$$A := \{(x, y) \in \mathbb{R}^{n+1} : y \in U, x = t\psi(y) + (1-t)\varphi(y); t \in [0,1]\}$$

For a given continuous function $f : A \to \mathbb{R}$, considers

$$F(y) = \int_{\varphi(y)}^{\psi(y)} f(x, y)dx, \qquad \forall y \in U.$$

Then F is continuous on U.

In addition, suppose that there exists an open set G such that $A \subset G \subset U \times \mathbb{R}$ such that for any $j \in \{1, \ldots, n\}$, the partial derivatives $\frac{\partial \varphi}{\partial y_j}$ and $\frac{\partial \psi}{\partial y_j}$ exist on U and that $\frac{\partial f}{\partial y_j}$ exists and is continuous on G. Then $\frac{\partial F}{\partial y_j}$ exists on U and takes the following expression:

$$\frac{\partial F}{\partial y_j} = \int_{\varphi(y)}^{\psi(y)} \frac{\partial f}{\partial y_j}(x, y)dx + f(\psi(y), y)\frac{\partial \psi(y)}{\partial y_j} - f(\varphi(y), y)\frac{\partial \varphi(y)}{\partial y_j}$$

Proof. For all $y \in U$ the function $F(y) = \int_{\varphi(y)}^{\psi(y)} f(x, y)dx$ exists because f is continuous. Consider $y_0 \in U$ and let's prove that F is continuous in y_0. Let $0 < \epsilon \leq 1$ and $\delta_1 > 0$ such that $B(y_0, \delta_1) \subset U$, where $B(y_0, \delta_1)$ denoting the open ball of center y_0 and radius δ_1.

Consider the following notations:

$$K_{\delta_1} = \{(x, y) \in A : y \in B(y_0, \delta_1)\}$$

which is a compact subset of U, and let

$$M = \sup\{|f(x, y)| : (x, y) \in K_{\delta_1}\} \quad \text{and} \quad M_1 = |\psi(y_0) - \varphi(y_0)|,$$

which satisfy $0 \leq M$ and $M_1 < \infty$. Since f is continuous on A, f is uniformly continuous on K_{δ_1} because it is compact. As functions ψ, φ are moreover continuous in y_0, then there exists $0 < \delta \leq \delta_1$ such that for all $h \in \mathbb{R}^n$ with $|h| < \delta$, we have for all $(x, y_0 + h), (x, y_0) \in K_{\delta_1}$ and $|h| < \delta$:

$$|f(x, y_0 + h) - f(x, y_0)| < \frac{\epsilon}{4M_1 + 1}. \tag{2.19}$$

$$|\psi(y_0 + h) - \psi(y_0)| < \frac{\epsilon}{4M + 4} \tag{2.20}$$

$$|\varphi(y_0 + h) - \varphi(y_0)| < \frac{\epsilon}{4M + 4} \tag{2.21}$$

Let $|h| < \delta$ and suppose without loss of generality that

$$\varphi(y_0) \leq \varphi(y_0 + h) \leq \psi(y_0) \leq \psi(y_0 + h).$$

Then we have:

$$\begin{aligned} F(y_0+h)-F(y_0) &= \int_{\varphi(y_0+h)}^{\psi(y_0+h)} f(x,y_0+h)dx - \int_{\varphi(y_0)}^{\psi(y_0)} f(x,y_0)dx \\ &= \int_{\varphi(y_0+h)}^{\psi(y_0)} [f(x,y_0+h)-f(x,y_0)]dx \\ &\quad + \int_{\psi(y_0)}^{\psi(y_0+h)} f(x,y_0+h)dx - \int_{\varphi(y_0)}^{\varphi(y_0+h)} f(x,y_0+h)dx \quad (2.22) \end{aligned}$$

From inequalities (2.19), (2.20), (2.21) and from the definition of M and M_1, we can then find some upper bounds for the three terms of the preceding expression:

- We have $\epsilon \leq 1$ then $\epsilon^2 \leq \epsilon \leq 1$

$$\begin{aligned} \left| \int_{\varphi(y_0+h)}^{\psi(y_0)} [f(x,y_0+h)-f(x,y_0)]dx \right| &< \frac{\epsilon}{4M_1+1} |\psi(y_0)-\varphi(y_0+h)| \\ &< \frac{\epsilon}{4M_1+1}\left[M_1 + \frac{\epsilon}{4M+4}\right] \\ &= \frac{M_1\epsilon}{4(M_1+\frac{1}{4})} + \frac{\epsilon^2}{4(4M_1+1)(M+1)} \\ &< \frac{\epsilon}{4}+\frac{\epsilon}{4} = \frac{\epsilon}{2} \end{aligned}$$

- $$\left| \int_{\psi(y_0)}^{\psi(y_0+h)} f(x,y_0+h)dx \right| < M\frac{\epsilon}{4(M+1)} < \frac{\epsilon}{4}$$
$$\left| \int_{\varphi(y_0)}^{\varphi(y_0+h)} f(x,y_0+h)dx \right| < M\frac{\epsilon}{4(M+1)} < \frac{\epsilon}{4}.$$

By using these result in equation (2.22), we finally get:

$$|F(y_0+h)-F(y_0)| < \frac{\epsilon}{2}+\frac{\epsilon}{4}+\frac{\epsilon}{4} = \epsilon$$

For the addition of the theorem, we can apply the theorem (2.2.2) to the functions f to verify for all $y \in U$ and each $i = 1, \ldots, n$ the expression of the function $\dfrac{\partial F}{\partial y_i}$. ■

2.2.2 General equations

In order to obtain the SWE we will integrate over the depth $h(x,y) = \eta(x,y) - z_b(x,y)$ the continuity equation (2.2) and momentum equations (2.1). We will denote the mean velocities in the x and y components by $\bar{u}$ and $\bar{v}$ using the expressions:

$$\bar{u} = \frac{1}{h}\int_{z_b}^{\eta} u dz \quad \text{and} \quad \bar{v} = \frac{1}{h}\int_{z_b}^{\eta} v dz.$$

These mean velocities satisfies

$$\int_{z_b}^{\eta} u^2 dz = h\bar{u}^2 + \int_{z_b}^{\eta}(u-\bar{u})^2 dz \quad , \quad \int_{z_b}^{\eta} v^2 dz = h\bar{v}^2 + \int_{z_b}^{\eta}(v-\bar{v})^2 dz$$
$$\text{and} \int_{z_b}^{\eta} uv dz = h\bar{u}\bar{v} + \int_{z_b}^{\eta}(u-\bar{u})(v-\bar{v})dz \quad (2.23)$$

In the sequel, we will use the notation

$$u_d = u - \bar{u} \quad \text{and} \quad v_d = v - \bar{v}$$

Let us now integrate the continuity and momentum equations over the depth h:

- **Continuity equation:**

$$\int_{z_b}^{\eta} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) dz = 0$$

$$\Leftrightarrow \int_{z_b}^{\eta} \frac{\partial u}{\partial x} dz + \int_{z_b}^{\eta} \frac{\partial v}{\partial y} dz + w|_{\eta} - w|_{z_b} = 0$$

Using the Leibniz's theorem (Theorem 2.2.3), we get:

$$\frac{\partial}{\partial x} \int_{z_b}^{\eta} u dz - u|_{\eta} \frac{\partial \eta}{\partial x} + u|_{z_b} \frac{\partial z_b}{\partial x} + \frac{\partial}{\partial y} \int_{z_b}^{\eta} v dz - v|_{\eta} \frac{\partial \eta}{\partial y} + v|_{z_b} \frac{\partial z_b}{\partial y} + w|_{\eta} - w|_{z_b} = 0$$

$$\Leftrightarrow \frac{\partial}{\partial x} \int_{z_b}^{\eta} u dz + \frac{\partial}{\partial y} \int_{z_b}^{\eta} v dz + \left(w|_{\eta} - u|_{\eta} \frac{\partial \eta}{\partial x} - v|_{\eta} \frac{\partial \eta}{\partial y} \right) + \left(u|_{z_b} \frac{\partial z_b}{\partial x} + v|_{z_b} \frac{\partial z_b}{\partial y} - w|_{z_b} \right) = 0.$$

Using the conditions (2.5) and (2.7) we then obtain:

$$\frac{\partial}{\partial x} \int_{z_b}^{\eta} u dz + \frac{\partial}{\partial y} \int_{z_b}^{\eta} v dz + \frac{\partial \eta}{\partial t} = 0$$

which can be rewritten:

$$\boxed{\frac{\partial \eta}{\partial t} + \frac{\partial (h\bar{u})}{\partial x} + \frac{\partial (h\bar{v})}{\partial y} = 0.} \tag{2.24}$$

- **Momentum equation:** We only present here the integration of the equation of the velocity component u. We have:

$$\int_{z_b}^{\eta} \left(\frac{\partial u}{\partial t} + \frac{\partial (u^2)}{\partial x} + \frac{\partial (uv)}{\partial y} + \frac{\partial (uw)}{\partial z} \right) dz = \int_{z_b}^{\eta} \left(-g \frac{\partial \eta}{\partial x} + \nu \Delta(u) \right) dz$$

$$\underbrace{\int_{z_b}^{\eta} \frac{\partial u}{\partial t} dz}_{(I)} + \underbrace{\int_{z_b}^{\eta} \frac{\partial (u^2)}{\partial x} dz + \int_{z_b}^{\eta} \frac{\partial (uv)}{\partial y} dz + \int_{z_b}^{\eta} \frac{\partial (uw)}{\partial z} dz}_{(II)} = \underbrace{\int_{z_b}^{\eta} -g \frac{\partial \eta}{\partial x} dz}_{(III)} + \underbrace{\int_{z_b}^{\eta} \nu \Delta(u) dz}_{(IV)}$$

Again we use here the Leibniz's rule (Theorem 2.2.3) and get:

(I) **Unsteady term:**

$$\begin{aligned} \int_{z_b}^{\eta} \frac{\partial u}{\partial t} dz &= \frac{\partial}{\partial t} \int_{z_b}^{\eta} u dz - u|_{\eta} \frac{\partial \eta}{\partial t} + u|_{z_b} \frac{\partial z_b}{\partial t} \\ &= \frac{\partial (h\bar{u})}{\partial t} - u|_{\eta} \frac{\partial \eta}{\partial t} + u|_{z_b} \frac{\partial z_b}{\partial t}. \end{aligned}$$

(II) **Advection term:**

* x-component

$$\begin{aligned}\int_{z_b}^{\eta}\frac{\partial(u^2)}{\partial x}dz &= \frac{\partial}{\partial x}\int_{z_b}^{\eta}u^2dz - u^2\Big|_{\eta}\frac{\partial\eta}{\partial x} + u^2\Big|_{z_b}\frac{\partial z_b}{\partial x}\\ &= \frac{\partial(h\bar{u}^2)}{\partial x} + \frac{\partial}{\partial x}\int_{z_b}^{\eta}(u-\bar{u})^2dz - u^2\Big|_{\eta}\frac{\partial\eta}{\partial x} + u^2\Big|_{z_b}\frac{\partial z_b}{\partial x}\\ &= \frac{\partial(h\bar{u}^2)}{\partial x} + \frac{\partial}{\partial x}\int_{z_b}^{\eta}u_d^2dz - u^2\Big|_{\eta}\frac{\partial\eta}{\partial x} + u^2\Big|_{z_b}\frac{\partial z_b}{\partial x}. \quad (2.25)\end{aligned}$$

* y-component:

$$\begin{aligned}\int_{z_b}^{\eta}\frac{\partial(uv)}{\partial y}dz &= \frac{\partial}{\partial y}\int_{z_b}^{\eta}uvdz - uv|_{\eta}\frac{\partial\eta}{\partial y} + uv|_{z_b}\frac{\partial z_b}{\partial y}\\ &= \frac{\partial(h\bar{u}\bar{v})}{\partial y} + \frac{\partial}{\partial y}\int_{z_b}^{\eta}(u-\bar{u})(v-\bar{v})dz - uv|_{\eta}\frac{\partial\eta}{\partial y} + uv|_{z_b}\frac{\partial z_b}{\partial y}\\ &= \frac{\partial(h\bar{u}\bar{v})}{\partial y} + \frac{\partial}{\partial y}\int_{z_b}^{\eta}u_dv_ddz - uv|_{\eta}\frac{\partial\eta}{\partial y} + uv|_{z_b}\frac{\partial z_b}{\partial y} \quad (2.26)\end{aligned}$$

* z-component

$$\int_{z_b}^{\eta}\frac{\partial(uw)}{\partial z}dz = uw|_{\eta} - uw|_{z_b} \quad (2.27)$$

(III) **Pressure term:**

$$\int_{z_b}^{\eta}-g\frac{\partial\eta}{\partial x}dz = -gh\frac{\partial\eta}{\partial x} \quad (2.28)$$

(IV) **Momentum term:**

$$\begin{aligned}\int_{z_b}^{\eta}\nu\triangle(u)dz &= \int_{z_b}^{\eta}\nu\vec{\nabla}\cdot(\vec{\nabla u})dz\\ &= \vec{\nabla}\cdot\int_{z_b}^{\eta}\nu\vec{\nabla}udz - \nu\,\vec{\nabla}u\Big|_{\eta}\frac{\partial\eta}{\partial x} + \nu\,\vec{\nabla}u\Big|_{z_b}\frac{\partial z_b}{\partial x}\\ &= \vec{\nabla}\cdot\left(\nu\vec{\nabla}(h\bar{u})\right) - \nu\cdot\vec{\nabla}u\Big|_{\eta}\frac{\partial\eta}{\partial x} + \nu\cdot\vec{\nabla}u\Big|_{z_b}\frac{\partial z_b}{\partial x}\\ &= \nu\triangle(h\bar{u}) - \nu\,\vec{\nabla}u\Big|_{\eta}\cdot\vec{\nabla}\eta + \nu\,\vec{\nabla}u\Big|_{z_b}\cdot\vec{\nabla}z_b \quad (2.29)\end{aligned}$$

Let us also make the following remarks:

- The following terms $\int_{z_b}^{\eta}u_d^2dz$ and $\int_{z_b}^{\eta}u_dv_ddz$ in the (II) Advection term the parameterization of these fluctuation of velocities is based on the concept of dispersion:

$$\int_{z_b}^{\eta}u_d^2dz = \nu_d\frac{\partial(h\bar{u})}{\partial x} \quad \text{and} \quad \int_{z_b}^{\eta}u_dv_ddz = \nu_d\frac{\partial(h\bar{u})}{\partial y},$$

where ν_d is the dispersion coefficient (see [6]).

- Let the fluid shear stress τ_{bx} and τ_{by} acting on the bottom and determined by Newton's law:

$$-\nu\ \vec{\nabla}u\Big|_{z_b}\cdot\vec{\nabla}z_b=\frac{\tau_{bx}}{\rho_0}=C_Bu\sqrt{u^2+v^2}, \tag{2.30}$$

$$-\nu\ \vec{\nabla}v\Big|_{z_b}\cdot\vec{\nabla}z_b=\frac{\tau_{by}}{\rho_0}=C_Bv\sqrt{u^2+v^2}. \tag{2.31}$$

where C_B is the bottom drag coefficient. Also Wind stress τ_{hx} and τ_{hy} are usually expressed with a similar formulas to (2.30) and (2.31) except that the velocity of the fluid is replaced by the wind speed. We note the wind stress by the equation:

$$\nu\ \vec{\nabla}u\Big|_{\eta}\cdot\vec{\nabla}\eta=\frac{\tau_{sx}}{\rho_0}=C_Du_w\sqrt{u_w^2+v_w^2}\ , \tag{2.32}$$

$$\nu\ \vec{\nabla}v\Big|_{\eta}\cdot\vec{\nabla}\eta=\frac{\tau_{sy}}{\rho_0}=C_Dv_w\sqrt{u_w^2+v_w^2}\ , \tag{2.33}$$

where $(u_w\ ,v_w\ )$is the wind velocity and C_D is the wind stress coefficient. For more details see [7] and [8].

Based on the preceding computations, we can rewrite the momentum equation given by (I) + (II) = (III) + (IV). The first member (I) + (II) is the sum of the unsteady and advection terms:

$$\begin{aligned}
&\frac{\partial(h\bar{u})}{\partial t}+\frac{\partial(h\bar{u}^2)}{\partial x}+\frac{\partial(h\bar{u}\bar{v})}{\partial y}-u|_{\eta}\left(\frac{\partial\eta}{\partial t}+u|_{\eta}\frac{\partial\eta}{\partial x}+v|_{\eta}\frac{\partial\eta}{\partial y}-w|_{\eta}\right)\\
&+u|_{z_b}\left(\frac{\partial z_b}{\partial t}+u|_{z_b}\frac{\partial z_b}{\partial x}+v|_{z_b}\frac{\partial z_b}{\partial y}\right)+\frac{\partial}{\partial x}\int_{z_b}^{\eta}u_d^2dz+\frac{\partial}{\partial y}\int_{z_b}^{\eta}u_dv_ddz\\
=\ &\frac{\partial(h\bar{u})}{\partial t}+\frac{\partial(h\bar{u}^2)}{\partial x}+\frac{\partial(h\bar{u}\bar{v})}{\partial y}+\frac{\partial}{\partial x}\int_{z_b}^{\eta}u_d^2dz+\frac{\partial}{\partial y}\int_{z_b}^{\eta}u_dv_ddz.\\
=\ &\frac{\partial(h\bar{u})}{\partial t}+\frac{\partial(h\bar{u}^2)}{\partial x}+\frac{\partial(h\bar{u}\bar{v})}{\partial y}+\frac{\partial}{\partial x}\left(\nu_d\frac{\partial(hu)}{\partial x}\right)+\frac{\partial}{\partial y}\left(\nu_d\frac{\partial(hu)}{\partial y}\right)\\
=\ &\frac{\partial(h\bar{u})}{\partial t}+\frac{\partial(h\bar{u}^2)}{\partial x}+\frac{\partial(h\bar{u}\bar{v})}{\partial y}+\nu_d\triangle(hu)
\end{aligned} \tag{2.34}$$

The second member (III) + (IV) is given by:

$$\begin{aligned}
&=-gh\frac{\partial\eta}{\partial x}+\nu\triangle(h\bar{u})-\nu\ \vec{\nabla}u\Big|_{\eta}\cdot\vec{\nabla}\eta+\nu\ \vec{\nabla}u\Big|_{z_b}\cdot\vec{\nabla}z_b\\
&=-gh\frac{\partial\eta}{\partial x}+\nu\triangle(h\bar{u})+\frac{\tau_{hx}}{\rho_0}-\frac{\tau_{z_bx}}{\rho_0}
\end{aligned} \tag{2.35}$$

Using the equations (2.34) and (2.35) we obtain for the equation of the x-component of the velocity:

$$\boxed{\frac{\partial(h\bar{u})}{\partial t}+\frac{\partial(h\bar{u}^2)}{\partial x}+\frac{\partial(h\bar{u}\bar{v})}{\partial y}=-gh\frac{\partial\eta}{\partial x}+(\nu-\nu_d)\triangle(h\bar{u})+\frac{\tau_{hx}}{\rho_0}-\frac{\tau_{z_bx}}{\rho_0}} \tag{2.36}$$

by symmetry, the equation of the y-component of the velocity is given by:

$$\boxed{\frac{\partial(h\bar{v})}{\partial t}+\frac{\partial(h\bar{u}\bar{v})}{\partial x}+\frac{\partial(h\bar{v}^2)}{\partial y}=-gh\frac{\partial\eta}{\partial y}+(\nu-\nu_d)\triangle(h\bar{u})+\frac{\tau_{hy}}{\rho_0}-\frac{\tau_{z_by}}{\rho_0}.} \tag{2.37}$$

Finally, the Shallow Water equation are the equations (2.24), (2.36) and (2.37), for more details see [6], [7] and [8].

Chapter 3

Water Quality Model

For the ecological part we use the water quality model (WQM) of the FVCOM [2] which is derived from theWater Quality Analysis Simulation Program (WASP). In this model, some organic and inorganic nutrients (nitrogen and phosphorus), the phytoplankton and the dissolved oxygen are represented.

3.1 Transport Equation

In order to simulate the propagation of cyanobacteria in the Lake Taihu it is necessary to simulate the transport and diffusion of this component in the lake whose domain is denoted Ω. To represent the advection and diffusion of a substance $\phi(t, x)$ with x in the domain Ω, we use the following equation:

$$\frac{\partial(\rho\phi)}{\partial t} + div(\rho\phi\vec{u}) - div(\Gamma grad(\phi)) = S_\phi \tag{3.1}$$

in which there is a source term S_ϕ that will include all the biological and chemical reactions, and some terms for the advection and diffusion through the fluid with the current $\vec{u} = (u, v, w)$ which are detailed in the sequel, and Γ is the diffusion coefficient.

3.1.1 Advection

It represents the fact that the substance is carried away by the currents: it is simply the transport of the substance. If it was the only process taken into account, the substance would travel at the same speed as the water and the area occupied by the substance would be constant. The advection term is given by:

$$div(\rho\phi\vec{u}) = \frac{\partial(\rho\phi u)}{\partial x} + \frac{\partial(\rho\phi v)}{\partial y} + \frac{\partial(\rho\phi w)}{\partial z}$$

3.1.2 Diffusion

If we deposit a drop of substance at a given point in a medium without flow and if we observe it a few time later, the initial drop will have been broadened. The molecules of the substance dissolved in the water will move from the points of greatest concentration to those of lower concentration: they will follow the gradient of concentrations. This process is called diffusion and occurs because of the continuous agitation of all the

molecules of the fluid. In a liquid the flow of mass by diffusion is governed by the first law of Fick; the second law of Fick includes also the time. The diffusion term is given by:

$$div(\Gamma grad(\phi)) = \frac{\partial}{\partial x}\left(\Gamma\frac{\partial \phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(\Gamma\frac{\partial \phi}{\partial y}\right) + \frac{\partial}{\partial z}\left(\Gamma\frac{\partial \phi}{\partial z}\right)$$

3.2 Biological processes

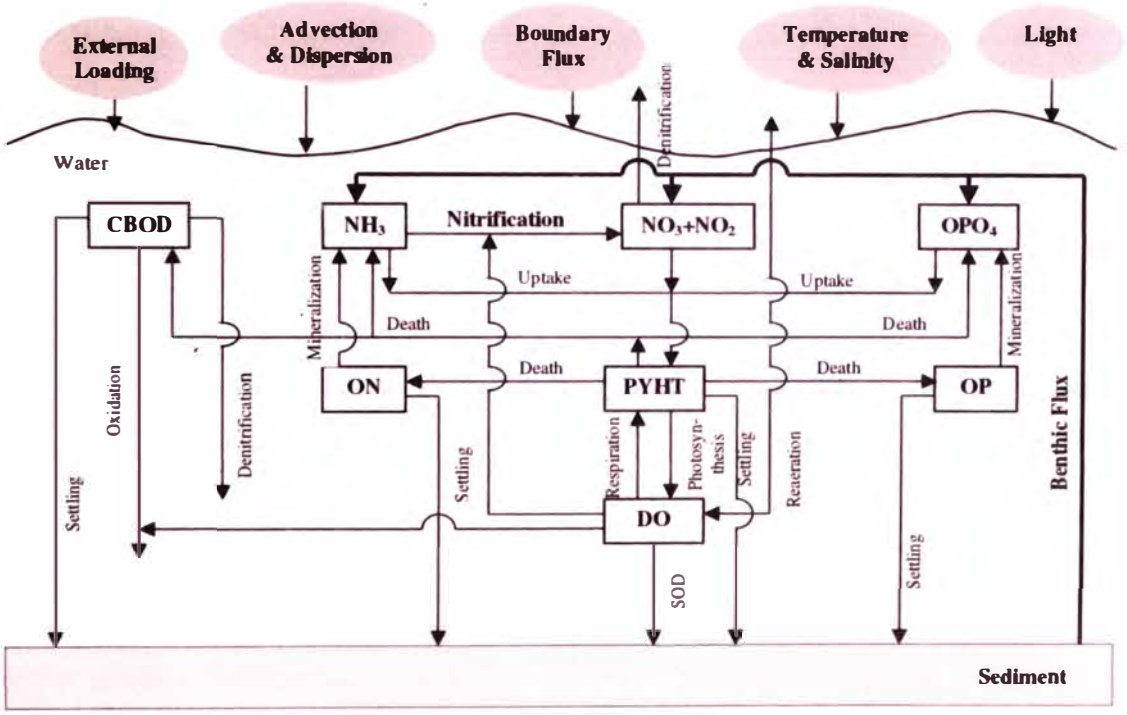


Figure 3.1. Scheme of the Water Quality Model of FVCOM (FVCOM-WQM[2]).

In the WQM model, there are 8 water quality variables:

1. Concentration dissolved oxygen (DO).
2. Concentration carbonaceous biochemical oxygen demand ($CBOD$).
3. Concentration phytoplankton as carbon ($PHYT$).
4. Concentration ammonium nitrogen (NH_4).
5. Concentration nitrate and nitrite nitrogen ($NO_3 + NO_2$).
6. Concentration ortho-phosphorus or inorganic phosphorus (OPO_4).
7. Concentration organic nitrogen (ON).
8. Concentration organic Phosphorus (OP).

The interaction between these variables are shown in Figure 3.1: it implies 11 biological processes.

- Death (or mortality) of phytoplankton.
- Passive respiration of phytoplankton;
- Uptake by phytoplankton of nutrients (or phytoplankton growth) NH_4, $NO_3 + NO_2$ and OPO_4;
- Remineralization of OP in OPO_4 and of ON in NH_4 (of detritus in nutrients);
- Nitrification;
- Denitrification;
- Settling of phytoplankton $PHYT$ and OP and ON (detritus);
- Benthic flux;
- Sediment oxygen demand;
- Oxidation;
- Reaeration.

The dynamic of each biological variable is expressed by the following partial differential equation:

$$\frac{\partial C}{\partial t}+\frac{\partial uC}{\partial x}+\frac{\partial vC}{\partial y}+\frac{\partial wC}{\partial z}-\frac{\partial}{\partial x}\left(A_h\frac{\partial C}{\partial x}\right)-\frac{\partial}{\partial y}\left(A_h\frac{\partial C}{\partial y}\right)-\frac{\partial}{\partial z}\left(K_h\frac{\partial C}{\partial z}\right)=S+W_0 \quad (3.2)$$

where C is the concentration of the water quality component; u, v and w are the water velocity components corresponding to the conventional Cartesian coordinate system (x, y, z); A_h and K_h are the coefficients of the horizontal viscosity and vertical eddy diffusion respectively; S is the function that represents the internal source or sink of the water quality component; W_0 is the external loading from point (river discharge) and non-point sources (ground water input and atmospheric deposition) of the water quality components.

3.3 Chlorophyll a

The Phytoplankton are microorganisms which involve Diatoms, Cyanobacterias, Dinoflagellates and other groups of algae. They are present in the freshwater ecosystems and using Chlorophyll (Chlorophyll a and Chlorophyll b) to generate energy through the photosynthesis. Chlorophyll a is a specific form of chlorophyll which is used to transform the sunlight into the chemical energy and gives the green color to the phytoplankton. For more details see [1] and [9].

$$C_{Chla} = \frac{PHYT}{a_{cchla}} \quad (3.3)$$

we usually use a proportional relationship between phytoplankton and chlorophyll a, equation (3.3) where a_{cchla} is the phytoplankton carbon to chlorophyll a ratio. There are two reasons for that: first we know that phytoplankton is the most important provider of chlorophyll a in the ecosystem; secondly, we also know that the phytoplankton contains some chlorophyll a (that is needed for the photosynthesis) but obviously, it is not only composed of chlorophyll a.

3.4 Equation for the Water Quality Components

The equation of FVCOM-WQM are described in details in [2], it's a modified version of the EPA Water Quality Analysis Simulation Program(WASP [10]). The state variables of the model are

Name	Definition	Unit
C_{do}	Concentration of dissolved oxygen	$mgO_2.L^{-1}$
C_{cbod}	Concentration of carbonaceous biochemical oxygen demand	$mgO_2.L^{-1}$
C_p	Concentration of phytoplankton	$mgC.L^{-1}$
C_{NH_4}	Concentration of ammonium nitrogen	$mgN.L^{-1}$
C_{NO_3}	Concentration of nitrate and nitrite nitrogen	$mgN.L^{-1}$
C_{ip}	Concentration of inorganic phosphorus	$mgP.L^{-1}$
C_{on}	Concentration of organic nitrogen	$mgN.L^{-1}$
C_{op}	Concentration of organic phosphorus	$mgP.L^{-1}$
C_{Chla}	Phytoplankton chlorophyll concentration	$mgChla.L^{-1}$
T	Water temperature in Celsius degrees	$°C$
T_K	Water temperature in Kelvin	K
S_ϕ	Water salinity	$mg\,salt.L^{-1}$

Table 3.1. The state variable

The reaction term S_ϕ of the equations of the 8 water quality variables are given here after. The parameters and some complementary equations are detailed in Appendix A and in next Section 3.5 respectively.

- **Dissolved oxygen** (DO):

$$
\begin{aligned}
S_{do} = & \underbrace{k_{r1}\theta_{r1}^{(T-T_r)}(C_s - C_{do})}_{\text{reaeration}} \underbrace{-k_{d1}\theta_{d1}^{(T-T_r)}\left(\frac{C_{do}}{K_{BOD}+C_{do}}\right)C_{cbod}}_{\text{oxidation}} \\
& \underbrace{-r_{oc}k_{r2}\theta_{r2}^{(T-T_r)}C_p}_{\substack{\text{phytoplankton}\\ \text{respiration}}} \underbrace{-2r_{on}k_{ni}\theta_{ni}^{(T-T_r)}\left(\frac{C_{do}}{K_{NITR}+C_{do}}\right)C_{NH_4}}_{\text{nitrification}} \\
& \underbrace{+G_p\left(r_{oc}+\frac{3}{2}r_{on}a_{nc}(1-P_{NH_4})\right)C_p}_{\substack{\text{phytoplankton}\\ \text{growth}}} \underbrace{-\frac{SOD}{100D_s}\theta_{SOD}^{(T-T_r)}}_{\text{sediment demand}} \underbrace{-k_{r3}}_{\substack{\text{bacterial}\\ \text{respiration}}}
\end{aligned} \tag{3.4}
$$

- **Carbonaceous biochemical oxygen demand** $(CBOD)$:

$$S_{cbod} = \underbrace{r_{oc}(k_{par}+k_{grs})\theta_{mr}^{(T-T_r)}C_p}_{\text{phytoplankton death}} \underbrace{-k_{d1}\theta_{d1}^{(T-T_r)}\left(\frac{C_{do}}{K_{BOD}+C_{do}}\right)C_{cbod}}_{\text{oxidation}}$$
$$\underbrace{-\frac{w_{3S}(1-f_{D2})}{D_s}C_{cbod}}_{\text{settling}} \underbrace{-\frac{5}{4}r_{on}k_{dn}\theta_{dn}^{(T-T_r)}\left(\frac{K_{NO_3}}{K_{NO_3}+C_{do}}\right)C_{NO_3}}_{\text{denitrification}} \quad (3.5)$$

- **Phytoplankton** $(PHYT)$:

$$S_p = \underbrace{G_pC_p}_{\text{phytoplankton groth}} \underbrace{-D_pC_p}_{\text{phytoplankton death and respiration}} \underbrace{-\frac{w_{2S}}{D_s}C_p}_{\text{settling}} \quad (3.6)$$

- **Ammonium nitrogen** (NH_4)

$$S_{NH_4} = \underbrace{a_{nc}D_p(1-f_{on})C_p}_{\text{phytoplankton death}} \underbrace{+k_{m1}\theta_{m1}^{(T-T_r)}\frac{C_p}{K_{mPc}+C_p}C_{on}}_{\text{mineralization}}$$
$$\underbrace{-a_{nc}G_pP_{NH_4}C_p}_{\text{phytoplankton growth}} \underbrace{-k_{ni}\theta_{ni}^{(T-T_r)}\frac{C_{do}}{K_{NITR}+C_{do}}C_{NH_4}}_{\text{nitrification}} \quad (3.7)$$

- **Nitrate and nitrite nitrogen** (NO_3):

$$S_{NO_3} = \underbrace{k_{ni}\theta_{ni}^{(T-T_r)}\frac{C_{do}}{K_{NITR}+C_{do}}C_{NH_4}}_{\text{nitrification}} \underbrace{-a_{nc}G_p(1-P_{NH_4})C_p}_{\text{phytoplankton growth}}$$
$$\underbrace{-k_{dn}\theta_{dn}^{(T-T_r)}\frac{K_{NO_3}}{K_{NO_3}+C_{do}}C_{NO_3}}_{\text{denitrification}} \underbrace{+\frac{B_2}{1000D_s}}_{\text{benthic flux}} \quad (3.8)$$

- **Organic nitrogen** (ON):

$$S_{on} = \underbrace{a_{nc}D_pf_{on}C_p}_{\text{phytoplankton death}} \underbrace{-k_{m1}\theta_{m1}^{(T-T_r)}\frac{C_p}{K_{mPc}+C_p}C_{on}}_{\text{mineralization}} \underbrace{-\frac{w_{7S}(1-f_{D7})}{D_s}C_{on}}_{\text{settling}} \quad (3.9)$$

- **Ortho-phosphorus** (OPO_4):

$$S_{ip} = \underbrace{a_{pc}D_p(1-f_{op})C_p}_{\text{phytoplankton death}} \underbrace{+k_{m2}\theta_{m2}^{(T-T_r)}\frac{C_p}{K_{mPc}+C_p}C_{op}}_{\text{mineralization}}$$
$$\underbrace{-a_{pc}G_pC_p}_{\text{phytoplankton growth}} \underbrace{+\frac{B_3}{1000D_s}}_{\text{benthic flux}} \quad (3.10)$$

- **Organic phosphorus** (OP)

$$S_{op} = \underbrace{a_{pc} D_P f_{op} C_p}_{\text{phytoplankton death}} \underbrace{-k_{m2}\theta_{m2}^{(T-T_r)} \frac{C_p}{K_{mPc}+C_p} C_{op}}_{\text{mineralization}} \underbrace{-\frac{w_8 s(1-f_{D8})}{D_s} C_{op}}_{\text{settling}} \quad (3.11)$$

3.5 Complementary equations

To complete the equations of the 8 water quality variables we need the following complementary equations:

- The average segment depth:

$$\bar{D} = D \times \frac{z_k + z_{k+1}}{2}$$

for a segment with depth between $z_k D$ and $z_{k+1} D$, where D is height of the water column.

- Dissolved oxygen saturation concentration:

$$\begin{aligned} \log(C_s) = & -139.34411 + (1.575701 \times 10^5) T_K^{-1} - (6.642308 \times 10^7) T_K^{-2} \\ & + (1.243800 \times 10^{10}) T_K^{-3} - (8.621949 \times 10^{11}) T_K^{-4} \\ & - 0.5535 \times S \times (0.031929 - 19.428 T_K^{-1} + 3867.3 T_K^{-2}) \end{aligned}$$

- Growth rate of phytoplankton

$$G_p = k_{gr}\theta_{gr}^{(T-T_r)} f_1(C_{NH_4}, C_{NO_3}, C_{ip}) f_2(I, f, \bar{D}, C_{Chla})$$

- Nutrient limitation factor of the phytoplankton growth rate

$$f_1(C_{NH_4}, C_{NO_3}, C_{ip}) = \min\left(\frac{C_{NH_4}+C_{NO_3}}{K_{mN}+C_{NH_4}+C_{NO_3}}, \frac{C_{ip}}{K_{mP}+C_{ip}}\right)$$

- Light limitation factor of the phytoplankton growth rate, with two formulation:

1. Di Toro:

$$f_2(I, f, \bar{D}, C_{Chla}) = \frac{ef}{K_e(C_{Chla})\bar{D}} \left(e^{-\frac{I_0(t)}{I_s} e^{-K_e(C_{Chla})\bar{D}}} - e^{-\frac{I_0(t)}{I_s}}\right)$$

2. Formulation of Smith:

$$f_2(I, f, \bar{D}, C_{Chla}) = \frac{e}{K_e(C_{Chla})\bar{D}} \left(e^{-\frac{I_0(t)}{I_s} e^{-K_e(C_{Chla})\bar{D}}} - e^{-\frac{I_0(t)}{I_s}}\right)$$

where:

$$t_R(t) = \mod(t, 24)\ ;\ C_{Chla} = \frac{C_p}{a_{cchla}}\ ;\ f = \frac{t_D - t_U}{24}$$

$$I_a(t) = 0.9\frac{I(t)}{f}\ ;\ I_s = \frac{k_{gr}\theta_{gr}^{(T-T_r)} a_{cchla} e}{\phi_{max} K_c f_u}$$

$$I_0(t) = \begin{cases} \frac{\pi}{2}\frac{I(t)}{f}\sin\left(\pi\frac{t_R(t)-t_U}{24f}\right) & \text{if } t_R(t) \in]t_U, t_D[\\ 0 & \text{elsewhere.} \end{cases}$$

$$a_{cchla} = \max\left(0.3\frac{\phi_{max} K_c f_u}{k_{gr}\theta_{gr}^{(T-T_r)} e} I_a(t)\frac{1 - e^{-K_e(C_{Chla})\bar{D}}}{K_e(C_{Chla})\bar{D}}, 20\right)$$

$$K_e(C_{Chla}) = K'_e + K_{eshd}(C_{Chla})$$

1. Di Toro : $K_{eshd}(C_{Chla}) = 0.0088 \times (1000 \times C_{Chla}) + 0.054 \times (1000 \times C_{Chla})^{0.67}$
2. Smith : $K_{eshd}(C_{Chla}) = K_c \times 1000 \times C_{Chla}$

- Death an respiration rate of phytoplankton

$$D_P = k_{r2}\theta_{r2}^{(T-T_r)} + (k_{par} + k_{grz})\theta_{mr}^{(T-T_r)}$$

- Ammonium preference form

$$P_{NH_4} = \frac{C_{NH_4} C_{NO_3}}{(K_{mN} + C_{NH_4})(K_{mN} + C_{NO_3})} + \frac{C_{NH_4} K_{mN}}{(C_{NH_4} + C_{NO_3} + 10^{-30})(K_{mN} + C_{NO_3})}$$

- Reaeration rate

$$k_{r1} = \max(k_{r1}^f, k_{r1}^w)\ ;\ k_{r1}^f = \theta_{r1}^{(T-T_r)} \times a\bar{v}^b\bar{D}^{-c}$$

$$(a, b, c) = \begin{cases} (5.349, 0.67, 1.85) & \text{if } \bar{D} < 0.6906m\ (Owen's formulation) \\ (5.049, 0.97, 1.67) & \text{if } \bar{D} > 0.6906m,\ d < 0\ (Churchill's formulation) \\ (3.93, 0.50, 1.50) & \text{if } \bar{D} > 0.6906m,\ d \geq 0\ (O'Connor - Dobbins's formulation) \end{cases}$$

$$d = \bar{D} - 4.411\bar{v}^{2.9135}\mathbb{1}_{\bar{v}\geq 0.518}.$$

$$k_{r1}^w = \frac{86400}{100\bar{D}} \begin{cases} \left(\frac{D_{O_w}}{\upsilon_w}\right)^{2/3}\left(\frac{\rho_a}{\rho_w}\right)^{1/2}\frac{k^{1/3}}{\Gamma_0}\sqrt{C_d}100W & \text{if } W < 6m/s \\ \left[\frac{1}{\left(\frac{D_{O_w}}{\upsilon_w}\right)^{2/3}\left(\frac{\rho_a}{\rho_w}\right)^{1/2}\frac{k^{1/3}}{\Gamma_u}\sqrt{C_d}100W} + \frac{1}{\left(\frac{D_{O_w}}{kz_0}\frac{\rho_a\upsilon_a}{\rho_w\upsilon_w}\sqrt{C_d}\right)^{1/2}\sqrt{100W}}\right]^{-1} & \text{if } 6 \leq W \leq 20m/s \\ \left(\frac{D_{O_w}}{kz_e}\frac{\rho_a\upsilon_a}{\rho_w\upsilon_w}\sqrt{C_d}\right)^{1/2}\sqrt{100W} & \text{if } W > 20m/s \end{cases}$$

where

$$\rho_a = 1.29 \cdot 10^{-3} - 4.0 \cdot 10^{-6} \times T_a$$

$$v_a = 1.33 \cdot 10^{-1} + 9.0 \cdot 10^{-4} \times T_a$$

$$v_w = 1.64 \cdot 10^{-2} - 2.4514 \cdot 10^{-4} \times T$$

$$D_{o_w} = 1.2 \cdot 10^{-5} + 4.58 \cdot 10^{-7} \times T$$

$$\Gamma_u = \begin{cases} \Gamma_0 \frac{\sqrt{C_d}100W}{U_{*c}} e^{1-\frac{\sqrt{C_d}100W}{U_{*c}}} & \text{if } \sqrt{C_d}100W \geq U_{*c} \\ \Gamma_0 & else \end{cases}$$

and z_0 is solution of

$$\frac{1}{z_e} + \frac{\lambda_1 \sqrt{C_d}100W}{v_a} e^{\frac{\sqrt{C_d}100W}{U_{*t}}} = \frac{1}{1000} e^{\frac{k}{\sqrt{C_d}}}$$

Chapter 4

Finite Volume Method

The Finite Volume Method (FVM) is a great alternative to other methods that enable to solve partial differential equations numerically (finite difference method or finite elements method for example). FVM is very efficient and often used to simulate fluid dynamics and the advection, diffusion and reaction of substances evolving in aquatic environment. In this work we want to simulate the PDE (3.1) and we choose the FVM for that. In the sequel, we will show how to discretize a reaction-advection-diffusion equation of the form (3.1) with the FVM.

4.1 Preliminary result

The next theorem is important to obtain the discretization of the equations with the FVM. For more details, we can refer to [2], [3], [4] and [11].

Theorem 4.1.1. Divergent Theorem *Suppose V is a subset of $\mathbb{R}^n$ which is compact and has a boundary S_b, also indicated with $\partial V = S_b$. If F is a continuously differentiable vector field defined on V including the neighborhood S_b, then we have:*

$$\int_V div(F)dV = \int_{S_b} \vec{n} \cdot F dA$$

Proof. Not given here. ■

4.2 Numerical scheme for the unstructured grid

The idea of the method is to integrate the equation over a control volume and to use the Divergence's theorem (Theorem 4.1.1) to simplify the equation. The control volume is defined by the mesh of the domain and the point at which we will compute the solution of the equation. If the mesh is a triangular mesh and if we want to compute the solution at the center of each triangle, then the control volume will be the area of the triangles (Figure 4.1(a)). If we want to compute the solution at each triangle vertex then the control volume will depend on how many triangles have this point as vertex (Figure 4.1(b)).

(a) Center (b) Vertex

Figure 4.1. Control volumen in a triangle mesh.

In the sequel, we choose to compute the solution at the center of the triangle and therefore consider as control volumes the areas of the triangles of the mesh. An example of triangle is given in Figure 4.2. P is the centroid of the control volume ΔV. For each triangle's side ∂A_i $(i = 1, 2, 3)$ of ΔV, we denote Δn_i the length of the triangle's side, c_i the center of the triangle's side ∂A_i, A_i the centroid of the second triangle which also has ∂A_i as one of his sides, $\vec{n}_i$ the normal vector to ∂A_i that is exterior to ΔV, $\vec{e}_{n_i}$ the vector of the ∂A_i direction and $\vec{e}_{\xi_i}$ the vector in the direction of $\overrightarrow{PA}$. We also denote $\Delta\xi_i$ the distance between P and A, $d'_{0,i}$ the distance between P and c_i and d'_i the distance between A and c_i (Figure 4.2).

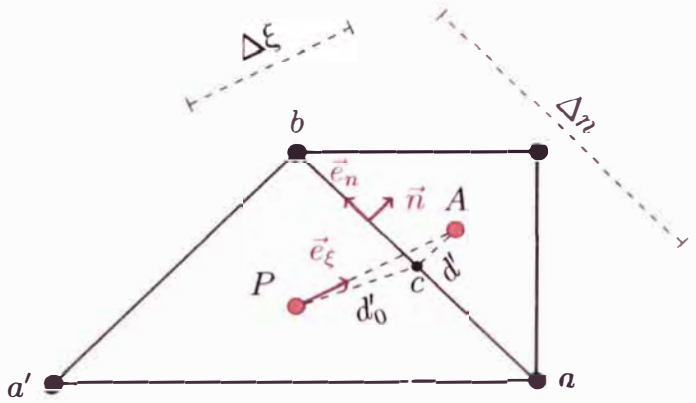


Figure 4.2. Elements of the triangle grid

Let us now integrate the transport equation (3.1) over the control volume ΔV which is the area of the triangle aba' (see Figure 4.2). We get:

$$\int_{\Delta V} \left[\frac{\partial \phi}{\partial t} + div(\phi \vec{u} - \Gamma grad(\phi)) \right] dV = \int_{\Delta V} S_\phi dV,$$

$$\int_{\Delta V} \frac{\partial \phi}{\partial t} dV + \int_{\Delta V} div(\phi \vec{u} - \Gamma grad(\phi)) dV = \int_{\Delta V} S_\phi dV.$$

Then, using the divergence's theorem, and assuming that the quantity ϕ and $\frac{\partial \phi}{dt}$ are constant in the control volume, we lobtain:

$$\left. \frac{\partial \phi}{\partial t} \right|_P \Delta V + \sum_{i=1}^{3} \int_{\partial A_i} \vec{n}_i \cdot (\phi \vec{u} - \Gamma grad(\phi)) dA = S_{\phi_P} \Delta V,$$

$$\Leftrightarrow \left. \frac{\partial \phi}{\partial t} \right|_P \Delta V + \sum_{i=1}^{3} \vec{n}_i \cdot [\phi_{c_i} \vec{u}_{c_i} - \Gamma grad(\phi)_{c_i}] \Delta n_i = S_{\phi_P} \Delta V,$$

Then, discretizing $\left.\frac{\partial \phi}{\partial t}\right|_P$ in $\frac{\phi_P - \phi_P^0}{\Delta t}$ using finite difference, we get:

$$\boxed{\frac{\Delta V}{\Delta t}\phi_P + \sum_{i=1}^{3}\left[(\vec{n}_i \cdot \vec{u}_{c_i})\phi_{c_i} - \Gamma \vec{n}_i \cdot grad(\phi)_{c_i}\right]\Delta n_i = \frac{\Delta V}{\Delta t}\phi_P^0 + S_{\phi_P}\Delta V,} \tag{4.1}$$

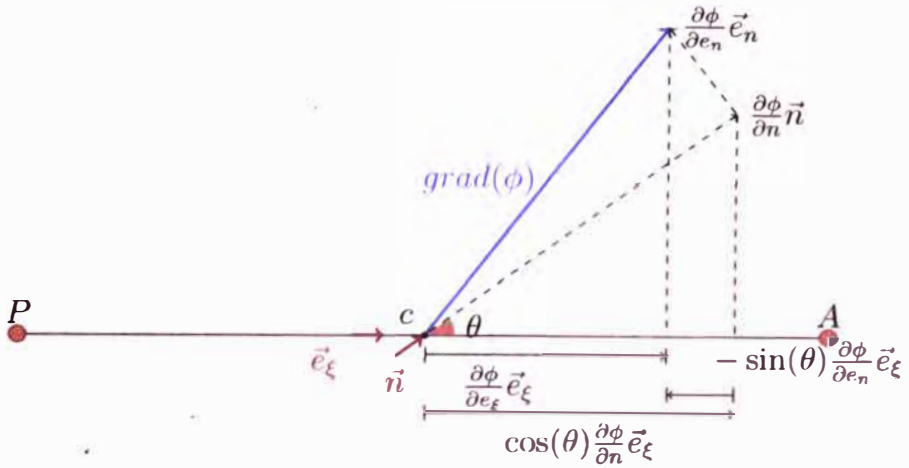


Figure 4.3. Inner product between gradient vector and normal vector.

We can then develop the advection and diffusion terms as follows:

- for the advection, we get:

$$\phi_c = \left(\frac{1}{d_0'\left(\frac{1}{d_0'} + \frac{1}{d'}\right)}\right)\phi_P + \left(\frac{1}{d'\left(\frac{1}{d_0'} + \frac{1}{d'}\right)}\right)\phi_A. \tag{4.2}$$

- for the diffusion, we know that:

$$grad(\phi) = \frac{\partial \phi}{\partial x}\vec{i} + \frac{\partial \phi}{\partial y}\vec{j} = \frac{\partial \phi}{\partial n}\vec{n} + \frac{\partial \phi}{\partial e_n}\vec{e}_n.$$

Using the Figure (4.3), we obtained the following expression:

$$\cos(\theta)\frac{\partial \phi}{\partial n} - \sin(\theta)\frac{\partial \phi}{\partial e_n} = \frac{\partial \phi}{\partial e_\xi},$$

$$\Leftrightarrow \frac{\partial \phi}{\partial n} = \frac{1}{\cos(\theta)}\frac{\partial \phi}{\partial e_\xi} + \tan(\theta)\frac{\partial \phi}{\partial e_n},$$

which leads to:

$$\vec{n} \cdot grad(\phi) = \frac{\partial \phi}{\partial n} = \frac{1}{\cos(\theta)}\frac{\partial \phi}{\partial e_\xi} + \tan(\theta)\frac{\partial \phi}{\partial e_n}.$$

For the $\cos(\theta_i)$ and $\tan(\theta_i)$, we can use the following expressions:

$$\cos(\theta) = \vec{n} \cdot \vec{e}_\xi \quad \text{and} \quad \tan(\theta) = -\left(\frac{\vec{e}_n \cdot \vec{e}_\xi}{\vec{n} \cdot \vec{e}_\xi}\right). \tag{4.3}$$

Finally, using the previous expressions and the finite difference approximations:

$$\frac{\partial \phi_{c_i}}{\partial e_{\xi_i}} = \frac{\phi_{A_i} - \phi_P}{\Delta \xi_i}; \frac{\partial \phi_{c_i}}{\partial e_{n_i}} = \frac{\phi_{b_i} - \phi_{a_i}}{\Delta n_i},$$

we obtain:

$$\vec{n} \cdot grad(\phi)_{c_i} = \frac{1}{\Delta \xi_i (\vec{n}_i \cdot \vec{e}_{\xi_i})} \phi_{A_i} - \frac{1}{\Delta \xi_i (\vec{n}_i \cdot \vec{e}_{\xi_i})} \phi_P - \frac{(\vec{e}_{n_i} \cdot \vec{e}_{\xi_i})}{\Delta n_i (\vec{n}_i \cdot \vec{e}_{\xi_i})} \phi_{b_i} + \frac{(\vec{e}_{n_i} \cdot \vec{e}_{\xi_i})}{\Delta n_i (\vec{n}_i \cdot \vec{e}_{\xi_i})} \phi_{a_i}$$

To compute ϕ_{a_i} and ϕ_{b_i} it is necessary to do an interpolation. The considered interpolation uses the values of the centers of all the triangles that share the triangle vertex a_i or b_i.

4.3 Conditions

To solve the equation (3.1) it is necessary to put some condition on the time and space boundary $\partial\Omega$. In the sequel we will present the Initial Condition(IC), Dirichlet Condition (DC) and Neumann Condition (NC) that can be used.

4.3.1 Definition

Initial Condition

The Initial condition for the equation is the values of ϕ in the internal domain at the beginning of the simulation (also denoted time $t = 0$):

$$\phi(0, x) = F(x) \quad , \quad x \in int(\Omega)$$

Dirichlet Condition

The Dirichlet condition imposes the values that the solution ϕ will take on the boundary:

$$\phi(t, x) = G(t, x) \quad , \quad (t, x) \in \mathbb{R}_0^+ \times \partial\Omega$$

Neumann Condition

The Neumann condition for the equation (3.1) is the values of the partial derivative of ϕ along of the some or all point on the boundary

$$\vec{n} \cdot \nabla(\phi)(x, y) = 0 \quad , \quad (x, y) \in \partial\Omega$$

4.3.2 Numerical implementation

It is important to explain how the boundary conditions are implemented and which conditions we will use because if we don't choose the correct condition we will have oscillations in the numerical solution.

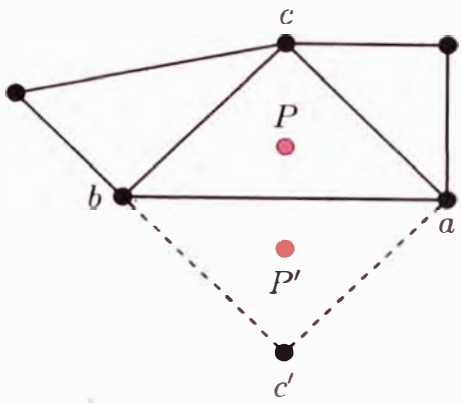


Figure 4.4. Edge $\overline{ab}$ is on the boundary of the mesh

In the Figure 4.4 we show how the boundary conditions are technically implemented in the FVM. For each triangle $\overline{abc}$ that has a side $\overline{ab}$ on the boundary we create a triangle $\overline{abc'}$ which is symmetric of $\overline{abc}$ with respect to $\overline{ab}$. The solution at point P is then computed using the auxiliary center P' of this new triangle.

We will now explain which boundary condition we should choose.

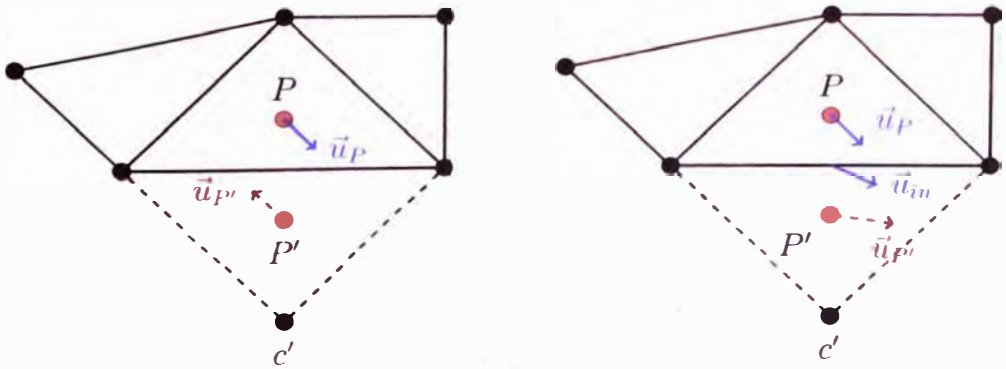
For the transport

We have the choice between the Dirichlet condition (DC) and the Neumann condition (NC) but both of them can not be put at the same point. If we have some source or sink on the boundary (some river for example) we will use the DC. If we have a natural wall we will use the NC. In both case we use the center P' to impose the condition on the boundary.

$$\phi_P = \text{input at the point } P' \tag{4.4}$$

$$\phi_P - \phi_{P'} = 0 \tag{4.5}$$

For the hydrodynamic



(a) With Neuman condition

(b) With Boundary condition

Figure 4.5. Treatment of velocity on the boundary.

For the velocity the treatment depend of the condition on the boundary for the hydrodynamics. If we use the NC which means that the normal velocity equals to zero $\left(\frac{\partial \vec{u}}{\partial \eta} = 0\right)$ on the boundary, then the velocity at the auxiliary triangle $\vec{u}_{P'}$ will be taken equal to the velocity of the triangle on the boundary $\vec{u}_P$ with opposite sign (see equation (4.6) and Figure 4.5(a)).

If we use the DC, it means that we have some source (like a river) on the boundary triangle with velocity $\vec{u}_{in}$. In this case the velocity at the auxiliary triangle $\vec{u}_{P'}$ will be the sum of the velocity of the boundary triangle $\vec{P}$ in the opposite sign and the velocity on the boundary $\vec{u}_{in}$ (see equation (4.7) and Figure 4.5(b)).

$$\vec{u}_{P'} = -\vec{u}_P. \tag{4.6}$$

$$\vec{u}_{P'} = -\vec{u}_P + 2\vec{u}_{in}. \tag{4.7}$$

The reason for use the equation (4.7) is because we approximate the velocity $\vec{u}_{in}$ as the sum of $\vec{u}_P$ and $\vec{u}_{P'}$, which are at the same distance of the triangle's edge.

Chapter 5

Simulation of the Coupled Model

In this chapter, we will work on the coupling of the hydro-ecological model composed of the Navier-Stokes equations (Chapter 2) and the Water Quality model (chapter 3) that is used to simulate the concentration of the cyanobacteria in the lake.

For the simulation of the hydro-ecological model we use the Finite Volume method (chapter 4). We implemented the method with Python3(see [12]) and saved the results in the formát of FVCOM software that is in netCDF (Network Common Data Form [13]) files. netCDF is a set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data as it is explained in [13]. To visualize the results we used the Paraview software which is an open source multiple-platform application in which we can do interactive scientific visualization in many dimensions (for more details see [14]).

We used the following test cases to simulate: (1) The current effect part and (2) hydro-ecological model, with different initial conditions:

1. **To test the current effect**
   We do not consider the source terms in the equation (3.1), in others words we impose $S = 0$ for all the simulation.

2. **To test the coupled model**
   To simulate the whole hydro-ecological model, we need to solve the equation (3.1) for the eight water quality components at the same time.

5.1 Simulations on a simplified square domain

The first simulations of the equation (3.1) were performed in a simplified square domain to verify if the discretized equation was correctly implemented. It is important to do this step because in this way we can identify the mistakes in the programme code and also in the discretized equation.

5.1.1 Computation of the velocity field

In order to solve the equation (3.1) we need as input the current over all the simplified square domain. For that, we considered a simplified current over the domain: the

velocity was defined on the square domain Ω by following equation:

$$\vec{u} = \left(-\frac{\partial \psi}{\partial y}, \frac{\partial \psi}{\partial x} \right) \tag{5.1}$$

where $\psi(x, y)$ satisfies the Laplace's equation:

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = 0 \qquad (x, y) \in \Omega \tag{5.2}$$

$$\psi(x, y) = 0 \qquad (x, y) \in \{0\} \times [0, 10] \cup [0, 4] \times \{0, 10\}, \tag{5.3}$$

$$\psi(x, y) = 1 \qquad (x, y) \in [6, 10] \times \{0, 10\} \cup \{10\} \times [0, 10], \tag{5.4}$$

$$\psi(x, y) = \sin(0.5(x - 5)\pi) + 0.5 \qquad (x, y) \in [4, 6] \times \{0, 10\}. \tag{5.5}$$

Consider the general continuity equation for a fluid with density ρ (see [4], Section 2.1.1 Mass conservation in three dimensions) which is define by the following equation:

$$\frac{\partial \rho}{\partial t} + div(\rho \vec{u}) = 0. \tag{5.6}$$

As our case is an incompressible fluid (i.e. Freshwater) the density ρ does not change with the time. In this context, the equation (5.6) becomes:

$$div(\vec{u}) = 0. \tag{5.7}$$

The velocity field $\vec{u} = (u_x, u_y)$ by his own definition (equation (5.1)) has the divergence equal to zero:

$$div(\vec{u}) = -\frac{\partial^2 \psi}{\partial x \partial y} + \frac{\partial^2 \psi}{\partial y \partial x} = 0.$$

As the function ψ has the conditions (5.3), (5.4) and (5.5) on the boundary. At the moment to compute the velocity field as we define in the equation (5.1), the conditions (5.3) and (5.4) give us that the velocity field does not get out of the domain, and the condition (2.21) give us that the velocity field coming into the domain when $y = 0$ and coming out to the domain when $y = 10$. Using the sine into the interval $[4, 5]$, the y-component of the velocity continuously increases until $x = 5$, and into the interval $[5, 6]$ the y-component of the velocity decrease until $x = 6$. This velocity simulates a river coming into the simple square when $y = 0$ and a river coming out when $y = 10$, in both case when x is between 4 and 6.

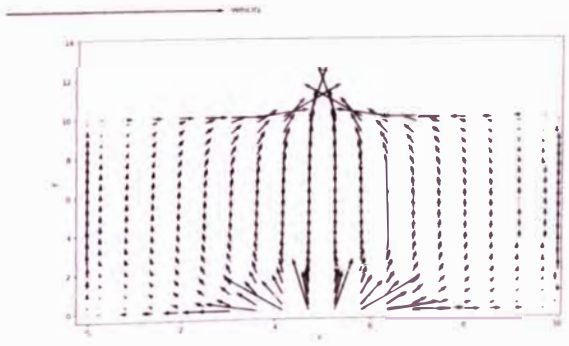
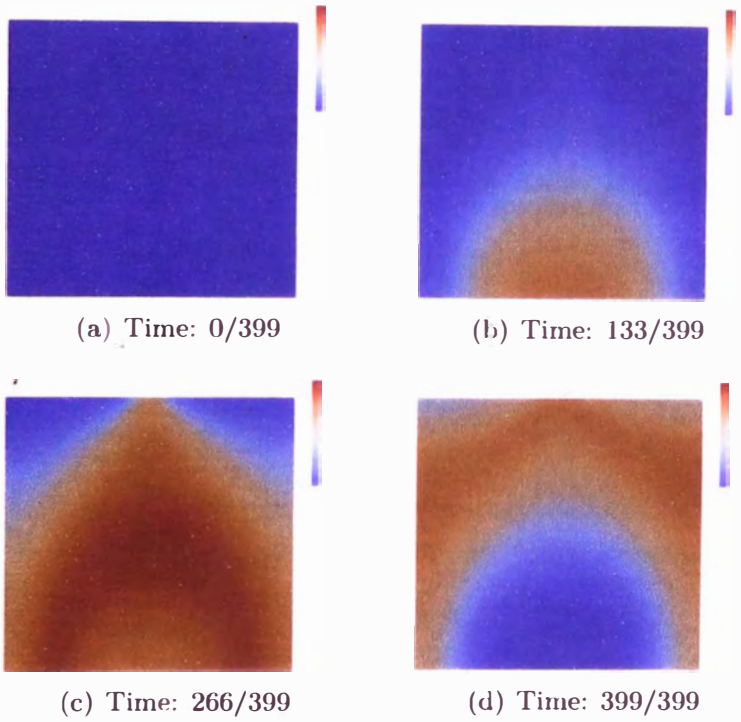


Figure 5.1. Velocity field on a square domain $\Omega = [0, 10]^2$.

5.1.2 Simulation of the advection-diffusion of an inert component

In this paragraph, we show the results obtained for the simulation of the advection-diffusion of an inert component (no reaction term). We used a constant initial condition on the square domain equal to zero and the velocity field is the one described in Section 5.1.1. The simulation is shown in Figure 5.2 with a time between $t_0 = 0$ until $t_L = 200$ and a partition with 400 points.



(a) Time: 0/399 (b) Time: 133/399

(c) Time: 266/399 (d) Time: 399/399

Figure 5.2. Simulation of the advection and diffusion of an inert component over the square domain $\Omega = [0, 10]^2$ with the velocity field shown in figure 5.1.

As we can see in Figure 5.2(a) the initial condition equal to zero in whole domain. For each time t, we imposed that the river enter concentration of the inner component into the square domain using the following expression for each time t between 0 and 200:

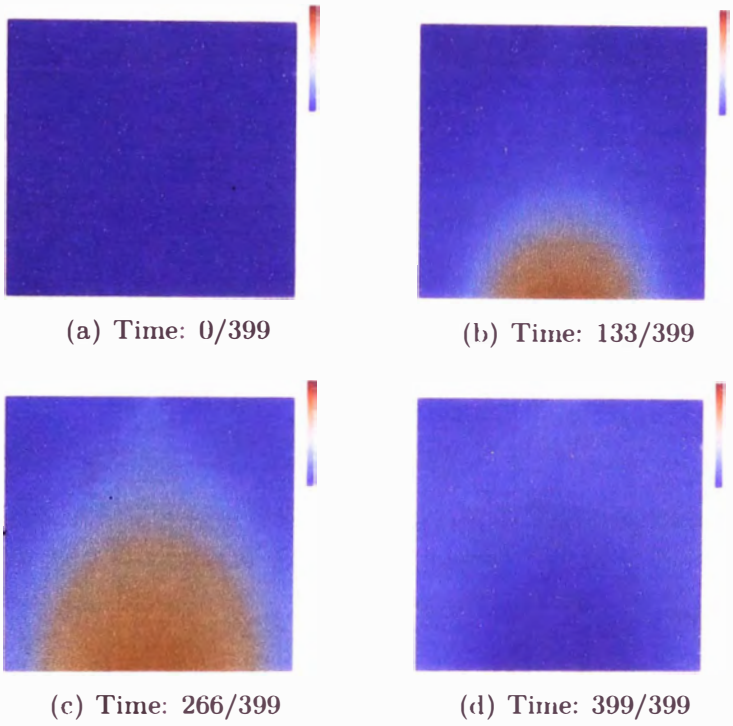
$$9.975 * \sin^2\left(\frac{t \cdot \pi}{t_L}\right), \tag{5.8}$$

this expression continuously increases up to $t = 100$ with its highest value equal to 9,975, and after continuously decrease until zero. In this context, in Figures 5.2(b) and 5.2(c) we can see how the concentration move with the current. Finally in Figure 5.2(d) the inner component's concentration leaves the square domain by the river.

5.1.3 Simulation of the WASP model coupled with the hydrodynamics

In this paragraph, we show the results obtained for the sumulation of the advection-diffusion and reaction of the water quality variables. We used a constant initial condi-

tion on the square domain equal to 0 for all the variables and a source on the boundary for the phytoplankton equal to equation (5.8) (the others are equal to zero). The velocity field is the one described in Section 5.1.1. The simulation is shown in Figure 5.3 with a time between $t_0 = 0$ until $t_L = 200$ and a partition with 400 points.



(a) Time: 0/399 (b) Time: 133/399

(c) Time: 266/399 (d) Time: 399/399

Figure 5.3. Simulation of the coupled model.

As we can see in Figure 5.3(a) the initial condition equal to the zero in whole domain. For each time t, we imposed that the river enter concentration of the phytoplankton component into the square domain using the expression for each time t between 0 and 200 similar to equation (5.8) with its highest value equal to 0.1476. For the rest of the water quality components we impose the conditions equal to zero. In this context, in Figures 5.3(b) and 5.3(c) we can see how the concentration move with the current. Finally in Figure 5.2(d) the phytoplankton's concentration leaves the square domain by the river.

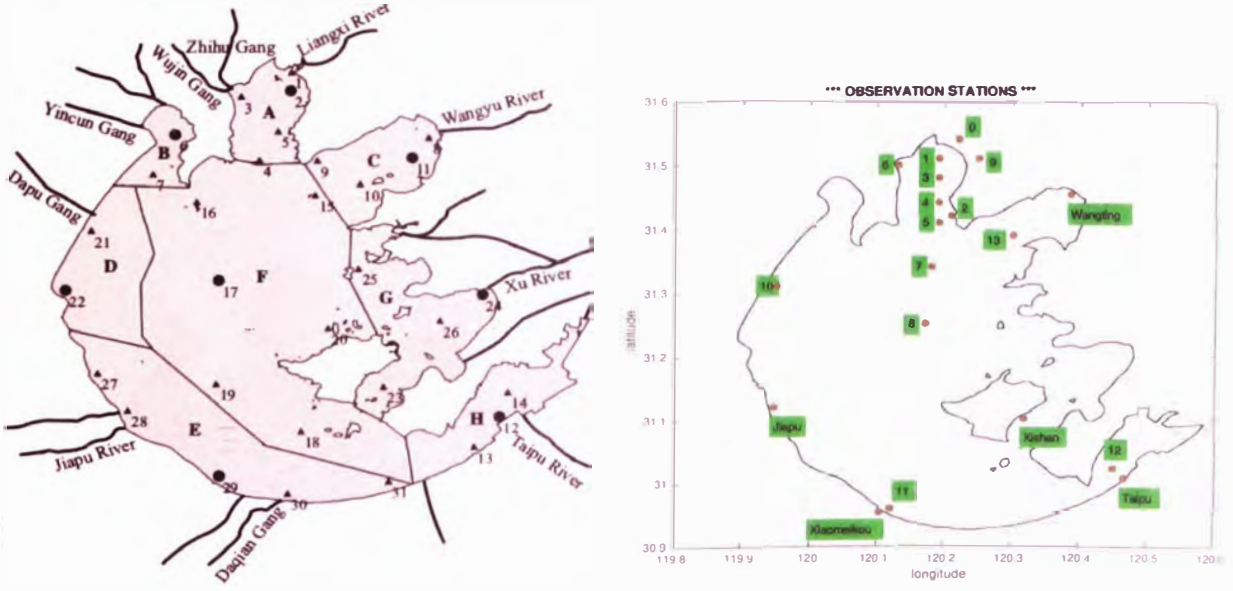
5.2 Simulation of the lake Taihu

In this section, we will present the results obtained by simulation of the coupled hydro-ecological model on the Lake Taihu.

Lake Tai or Lake Taihu is a large freshwater lake located in the Yangtze Delta plain in Wuxi, China (Figure 5.4(a)). The lake belongs to the Jiangsu province and the southern shore forms its border with the Zhejiang province. It has a surface area of $2338km^2$, which stretches from North to South over $68.5km$, and from West to East over $34km$ (mean value). Its mean depth is $1.9m$ and its maximal depth is $2.6m$ (varying a little bit depending on the season): Lake Taihu is therefore a shallow lake.

5.2.1 Available data of lake Taihu

For the computation of the initial and boundary conditions on the 2D hydro-ecological model, we will use the data measured between May 2001 and May 2002 at several observation stations located in the Lake Taihu. There are 14 stations (Figure 5.4(b)) and 17 rivers (Figure 5.4(a)) that are managed by:



(a) Taihu's rivers

(b) Taihu's observation station

Figure 5.4. Stations and Rivers in the Lake Taihu.

- the NIGLAS: Nanjing Institute of Geography and Limnology, Chinese Academy of Sciences (CAS),
- and the TLLER: Taihu Laboratory for Lake Ecosystem Research.

For each of these stations, we have some measurements of:

- physical variables such as water depth, temperature, secchi depth, SS, PH, conductivity,
- dissolved oxygen (DO) and COD,
- nitrogen cycle: NH_4, NO_2, NO_3, total nitrogen (TN), total dissolved nitrogen (TDN)
- phosphorus cycle: PO_4, total phosphorus (TP), dissolved total phosphorus (DTP)
- and other variables such as chlorophyli a ($Chla$), K^+, Na^+, Ca^{2+}, Mg^{2+}, F^-,Cl^-, SO_4^{2-} , SiO_4.

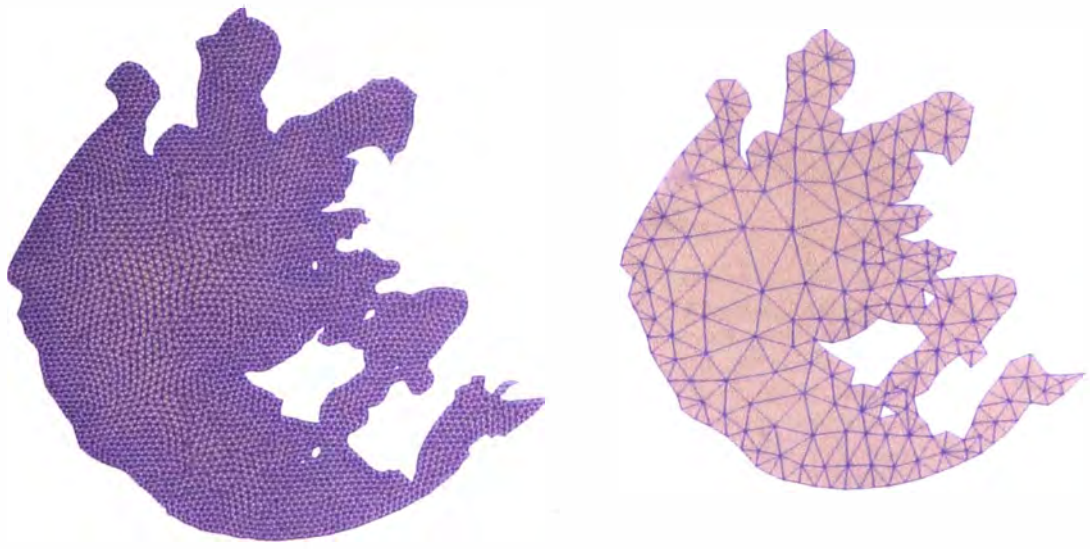
We also have access to other data sets:

- rivers data: inflow rates, outflow rates, nutrient concentration;
- meteorological data: wind, precipitation, light intensity;
- experimental data: for example, growth rate of phytoplankton.

5.2.2 Mesh

For the simulation it is important decide which mesh will be used how it was explained in Chapter 4. Here we used two triangle meshes:

- a first mesh composed of 3741 nodes which form 6949 triangles as in Figure 5.5(a).
- a second mesh composed of 285 nodes which form 410 triangles as in Figure 5.5(b).



(a) Mesh 1: 3741 nodes.

(b) Mesh 2: 285 nodes.

Figure 5.5. Triangle meshes of the lake Taihu.

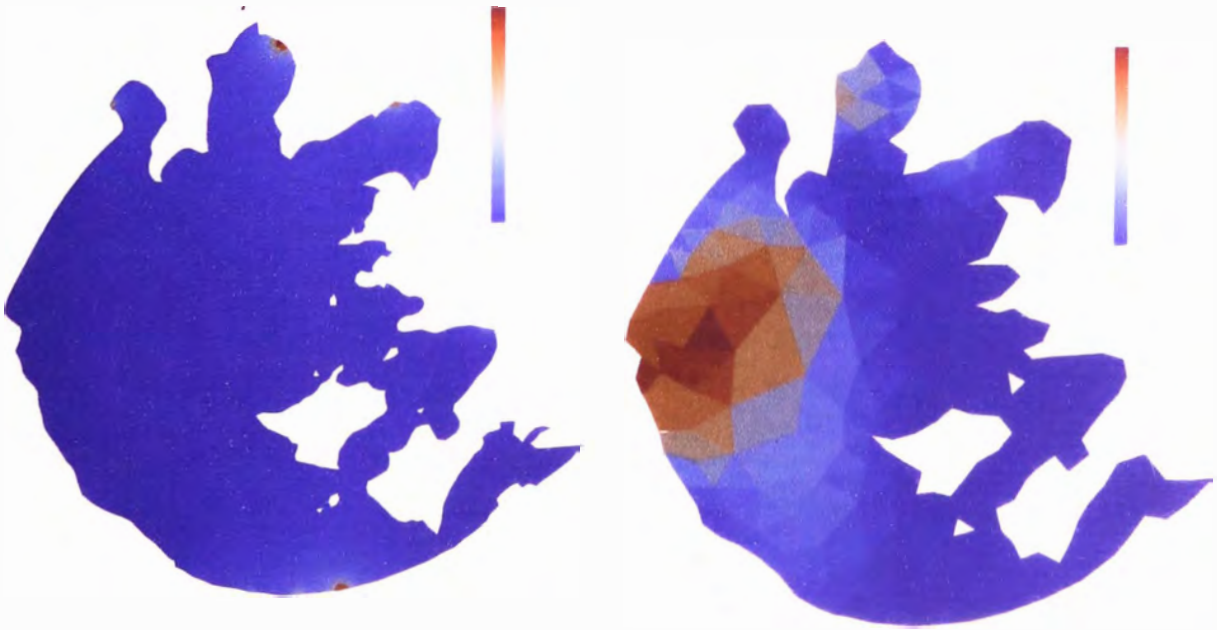
5.2.3 Time Period and boundary condition of the Simulation

For the simulation of the Lake Taihu, we used a time period of one year for the mesh with 285 node (Figure 5.5(b)), between may 2001 until may 2002 to be able to use the data described in Section 5.2.1. For the mesh with 3741 nodes (Figure 5.5(a)), we used a time period of two months between may 2001 until jun 2002.

For the boundary condition, we considered the Neumann Condition on the whole boundary (see Section 4.3): it means that for this first step, we neglected the rivers as we know that the current are mainly induced to the wind in the case of Lake Taihu.

5.2.4 Initial condition

For the initial and boundary condition, we used the information of the 14 stations and 17 rivers obtained by the NIGLAS (Section 5.2.1) to interpolate different initial conditions. The interpolation was made with a function of Scipy package, *scipy/interpolate/griddata* using the cubic method. The initial condition for the phytoplankton is shown on Figure 5.6.



(a) 3741 nodes.

(b) 285 nodes.

Figure 5.6. Initial condition for the phytoplankton.

5.2.5 Velocity

To solve the equation (3.1), we need to know the velocity over the period of time, under consideration. To compute the velocity we used the FVCOM software [2], without the rivers. The FVCOM solve the Navier-Stokes equations (2.1), (2.2) and (2.3). The velocity field obtained for the two meshes are shown in Figures (5.7) and (5.8).

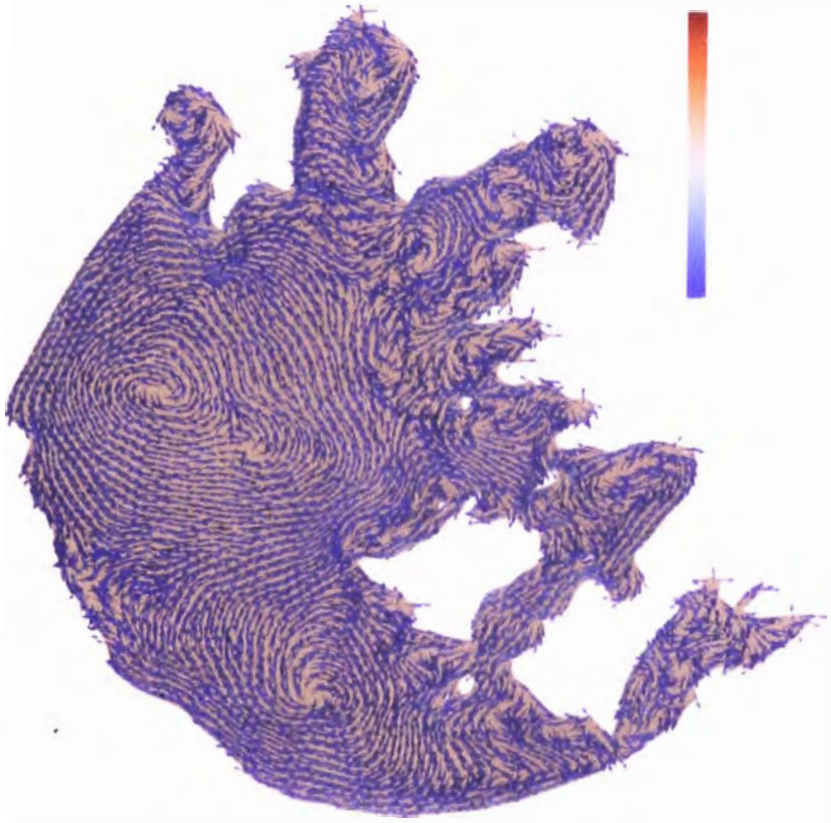


Figure 5.7. Velocity with 3741 nodes.

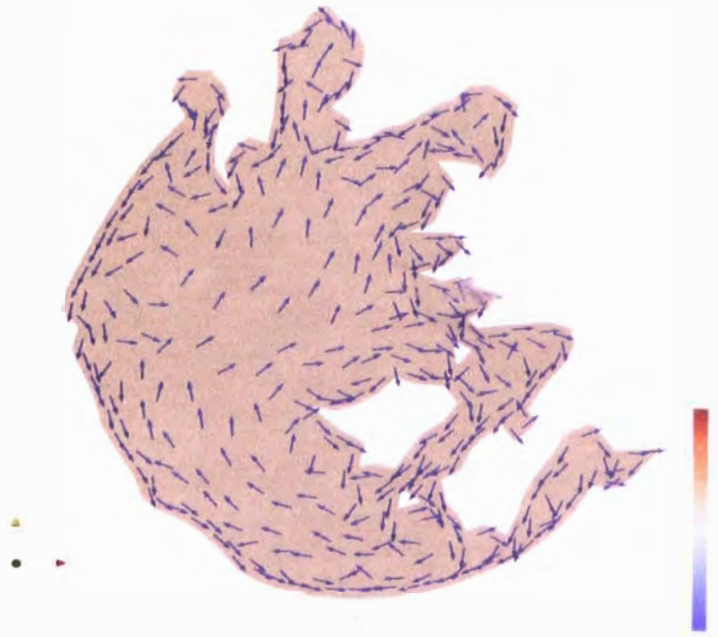


Figure 5.8. Velocity with 285 nodes.

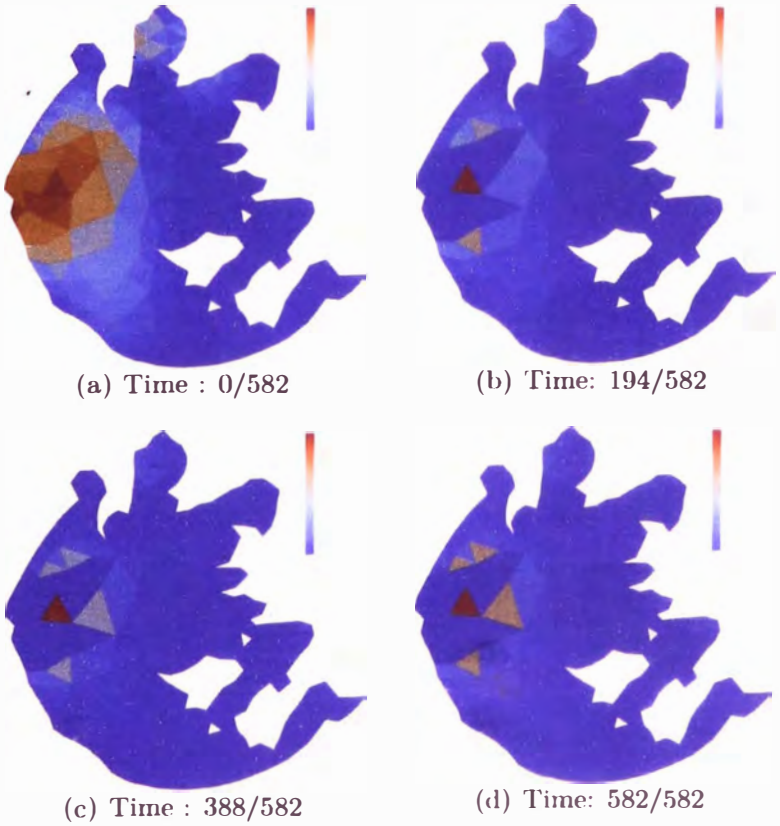
5.2.6 Results

As it was explained in Section 5.2.4 all the inputs necessary to simulate the test cases (current effect and coupled model) were generating before running the simulations. In Appendix B there is all the explanations necessary to generate the input and all the command lines to run all the scripts. With the script, I simulated the two test cases.

Current effect simulation

The simulation of the coupled model is shown on Figure 5.9, for this simulation we used the initial condition shown in Figure 5.6(b) and the velocity field shown in Figure 5.8.

The Time period used (as it was explained in Section 5.2.3) is one year between may 2001 and may 2002. The time partition has 582 points. The mesh used for the simulation is the small mesh with 285 nodes as shown in Figure 5.5(b).



(a) Time : 0/582 (b) Time: 194/582

(c) Time : 388/582 (d) Time: 582/582

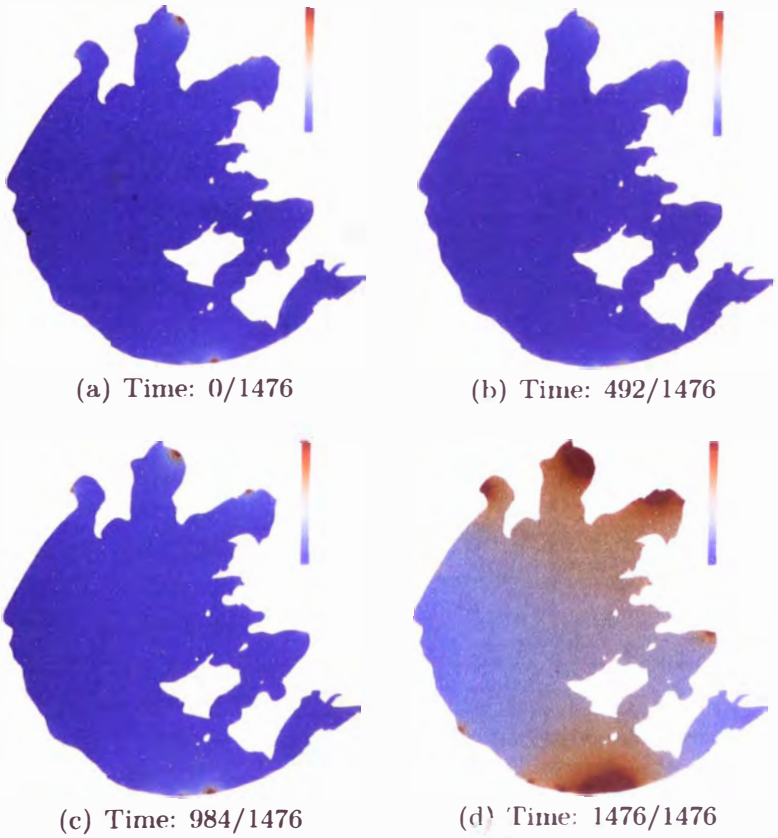
Figure 5.9. Phytoplankton concentration in the simulation to see the current effect using the mesh with 285 nodes.

In Figure 5.9(a), we see the initial condition for the simulation with the small mesh. In the Figure 5.9(b), 5.9(c) and 5.9(d) we can see how the current effect move the phytoplankton concentration around the lake.

Coupled Model simulation

The simulation of the coupled model is shown on Figure 5.10, for the simulation we used the initial condition shown in Figure 5.6(a) and the velocity field shown in Figure 5.7.

The Time period used for this simulation is two months: between may 2001 and jun 2001. The reason is that as we are using Python to simulate it take a long period of time. the time period we have information of the lake's station and the rivers to do the interpolation of the initial condition is between may 2001 and may 2002, as was explained in Section 5.2.1. The time partition The mesh used for the simulation is the small mesh with 3741 nodes as shown in Figure 5.5(a).



(a) Time: 0/1476 (b) Time: 492/1476

(c) Time: 984/1476 (d) Time: 1476/1476

Figure 5.10. Phytoplankton concentration in the simulation of the coupled model using the mesh with 3741 nodes.

For the water quality model simulation, in Figure 5.10(a) we see the initial condition considered for this simulation. In Figure 5.10(b) we see how the phytoplankton concentration is close to the boundary(concentration decrease). In Figure 5.10(c) we can see how the phytoplankton concentration growths again and in Figure 5.10(d) it growths a lot in the last step, which this means the phytoplankton form a big concentration in these areas.

Chapter 6

Sensitivity Analysis

The sensitivity analysis (SA) is the study of how the uncertainties on the outputs of a mathematical model can be explained by the uncertainties of some specific inputs. In others words as it is explained in the reference [16]: "we will able to answer questions of the type "which of the uncertain input factors is more important in determining the uncertainty in one of the input factors, which factor should we choose to reduce the most the variance of the output?", answering this question is important because a few important factors are identified, the modeler may choose to simplify the model structure by eliminating parts that appear to be irrelevant or be may decide to proceed with model limping and extract a simpler model from the complex one", for more details see [16].

6.1 Morris Method

The sensitivity method proposed by Morris in [17](1991) and modified by Campolongo et al in [18](2003) take in account two sensitivity measure for each factor as it is explained in [16]: "a measure μ that estimates the overall effect of the factor on the output, and a measure σ that, according to Morris, estimates the ensemble of the second and higher-order effects in which the factor is involved (including curvatures and interaction effects)". The measure μ is the average of the output variations at certainly number of inputs as we will explain in the following sections, in the case of Campolongo's measure consider the measure μ^* which is the average absolute of the output variations at certain number of inputs consider for the measure μ.

About the Morris method in the reference [17] explain that: "The guiding philosophy in this work is that a major role of a preliminary experiment is to determine, which inputs may be considered to have effects which are (a) negligible (b) linear and additive, (c) non-linear or (d) involved in interactions with other factors". In the reference [17], they also propose to visualize the results with a graphical plot where we can see both sensitivity measures (μ, σ). For the case of the Campolongo's measures, we consider a graphical plot of the couple (μ^*, σ).

6.1.1 Discretization of the space

Let denote k the number of parameters to be considered in the SA. Each parameter has his own definition interval $[a_i, b_i]$ $(i = 1 : k)$. For each input factor we consider a partition of size p of the definition interval, that is a set of p discrete values in the definition interval:

$$\left\{a_i, a_i + \frac{b_i - a_i}{(p-1)}, a_i + \frac{2(b_i - a_i)}{(p-1)}, \ldots, a_i + \frac{(p-2)(b_i - a_i)}{(p-1)}, b_i\right\}.$$

The region of experimentation Ω is defined as the product of the definition intervals of each parameter: it is therefore a $k-$dimensional hypercubic with $p-$level grid.

In the sequel, X will denote a $k-$dimensional vector with the model inputs X_i, $i = 1 : k$.

6.1.2 Random Trajectories

To perform a sensitivity analysis of a model, we need to run the model several times with different input values. The input values have to be chosen randomly, but we also want to explore the space Ω correctly. Morris proposed a method to generate the input values. For that, we first have to choose a number of trajectories that is denoted r in the sequel. As it is explained in the book [16], for each of the r trajectories, we will generate a set of $k+1$ input parameter values vector $\{X^j\}$ with $j = 1, \ldots, k+1$ in the region of experimentation Ω. To generate this set, we start by randomly selecting a 'basis' vector X^* in Ω. The first input values vector, X^1, is obtained by increasing one component of X^* by one level of the grid. The second sampling point is generated from X^1: it differs from X^1 only by its i^{th} component that has been either increased or decreased by one level of the grid. We choose the index i randomly in the set $\{1, 2, \ldots, k\}$ different to the index chosen to get the first input values. Mathematically, it is written $X^2 = (X_1^1, \ldots, X_{i-1}^1, X_i^1 \pm \Delta_i, X_{i+1}^1, \ldots, X_k^1) = X^1 \pm e_i\Delta_i$ with $\Delta_i = \frac{b_i - a_i}{p-1}$ and e_i the k-dimensional vector with zeros values except its i^{th} component that is equal to 1. The following input values vector are generated with the recurrence equation, $X^{j+1} = X^j \pm e_{i(j)}\Delta_{i(j)}$ (Figure 6.1). This method enables to generate, for each of the r trajectories, a set of $(k+1)$ input values vectors $X^1, X^2, \ldots, X^{k+1}$, with the key property that two consecutive input values vectors differ from each other by only one component. Furthermore any component of the 'basis vector' X^* has been selected at least once to be increased by one level of the grid; it will enable to calculate one elementary effect for each input factor as it will be explained in the sequel.

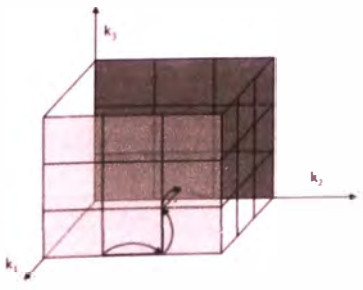


Figure 6.1. Random trajectories with k = 3, we can see how the trajectory changes only in one component in each step.(Figure of [16]).

Note that the vector X^* is used to generate the other sampling points but it is not one of them and the model is never evaluated at X^*. And that a sampling point X^{l+1}, with l in $\{1, \ldots, k\}$, may be different from X^l in one component because one of its components has been increased or decreased. The succession of sampling points $\{X^1, X^2, \ldots, X^{k+1}\}$ defines what is called a trajectory in the input space.

6.1.3 Elementary effect

Once we have generated the r trajectories, we can simulate the model for each of the input values vectors X^j $(j = 1 : k+1)$ of the r trajectories. The obtained outputs will then be used to compute what is called "elementary effect" in the sequel.

The elementary effect is the variation of the output between two consecutive elements of a random trajectory. The elementary effect for the i^{th} input is defined as it is explained in the article [17]: consider the value Δ which is a predetermined multiple of $1/(p-1)$ and the value $\tilde{\Delta}_i$, which is the real variation of the scale Δ in the parameters i^{th} values space (i.e. $\tilde{\Delta}_i = (b_i - a_i)\Delta$). For each element $X = (x_1, x_2, \ldots, x_k)$ of a random trajectory, the elementary effect of the i^{th} input factor of X is defined as:

$$d_i(X) = \frac{y(x_1, x_2, \ldots, x_i + \tilde{\Delta}, \ldots, x_k) - y(X)}{\Delta},$$

where $y(X)$ is the output of the model obtained with input values equal to x. Note that to compute this elementary effect, the input values vector $(x_1, x_2, \ldots, x_k) \in \Omega$ has to be such that the transformed input values vector.

By using the method introduced in Section 6.1.2 to generate the input values vectors, we know that for any input factor $i = 1 : k$ and for any trajectory $j = 1 : r$, there exists one index number $m(j)$ such that $X_j^{m(j)+1} = X_j^{m(j)} + e_i\tilde{\Delta}_i$. And we have:

$$d_i(X_j^{m(i)}) = \frac{y(X_j^{m(i)+1}) - y(X_j^{m(i)})}{\Delta},$$

Remember that each trajectory has $k+1$ elements; so we have $r \times k$ elementary effects.
Remark: In the reference [16], the authors use $\tilde{\Delta}_i = \Delta$; it is because to illustrate the method, they assume that the all input values have a definition interval equal to $[0, 1]$ (i.e. $a_i = 0$ and $b_i = 1$).

We will use r trajectories to compute the sensitivity measure for the i^{th} input of the k parameters. The sensitivity measure is definite by:

$$\mu_i = \frac{1}{r}\sum_{j=1}^{r} d_i(X_j^m), \tag{6.1}$$

$$\mu_i^* = \frac{1}{r}\sum_{j=1}^{r} |d_i(X_j^m)|, \tag{6.2}$$

$$\sigma_i = \sqrt{\frac{1}{r}\sum_{j=1}^{r}\left(d_i(X_j^m) - \mu_i\right)^2}, \tag{6.3}$$

where X_j^m is the m^{th} input values vector of the j^{th} trajectory. Remember each trajectory has $k+1$ elements, in that context, we will have $r * k$ elementary effects.

In this work, we considered the sensitivity measure suggested by Campolongo because with our model the elementary effect can take positive and negative values. So, when we compute the average, some terms may cancel each other and the sensitivity analysis will not give us any information.

In [17], a practical choice for the parameters p and Δ is given: they propose to choose a value of p that is even and to take Δ equal to $\frac{p}{2(p-1)}$. With this choice, although the design sampling strategy does not guarantee an equal-probability for the elementary effects of each trajectory, at least a certain symmetric treatment of inputs is ensured.

6.2 Consideration for the WASP model

The Morris method has been developed for scalar real values models. In the case of the WASP model, we have a model with 8 outputs where each output is a function of the time. To apply the Morris method in the case of the WASP model we propose to adapt the method. We will apply the Morris method for each water quality variable, but as each output is a function of time, to compute the elementary effect we use the 1-norm for continuous functions instead of the absolute value. Then the elementary effect is defined as follows:

$$d_i(x) = \frac{||y(x_1, x_2, \ldots, x_i + \tilde{\Delta}, \ldots, x_k) - y(x)||}{\Delta}$$

with $i = 1, \ldots, k$ and where the norm $||\cdot||$ is defined by

$$||y|| = \int_0^T |y(s)| ds.$$

The sensitivity measures μ^* and σ are then given by the same expression.

Finally, we will have 8 SA results, one for each water quality components.

6.3 Implementation

For the implementation of the sensitivity analysis method, I used the scripts developed by Julien Diot during his master (internship made during the 2^{nd} year of Montpellier SupAgro engineer school in 2016 under the supervision of Céline Casenave in the unit MISTEA, INRA). I modified and improved the script to make the sensitivity analysis of the WASP model.

For the implementation, it was necessary to use Python scientific packages like Numpy, Scipy, Matplotlib, Bokeh, and SALib. The data that we used for the simulation are stored in a simple format .txt files; this files contained data of the years 2001 and 2002 of the lake Taihu for each of the 19 stations as we describe in Section 5.2.1.

6.3.1 For WQM

For the sensitivity analysis, we consider a simple case without the advection-diffusion effect, the external loading and the non-point sources:

$$\frac{dC}{dt} = S_\phi \tag{6.4}$$

where C is the concentration of one water quality component and S_ϕ the respective reaction term as described in Section 3.4. To obtain a solution of this ordinary differential equation we considered the initial condition given by the observation data at one station in 2001. The values are given in Table 6.1 for each component.

Component	Initial Condition	Unit
C_{do}	9.975	mgO2/L
C_{cbod}	4.52	mgC/L
C_p	0.011815*12.5	mgC/L
C_{NH4}	0.0935	mgN/L
C_{NO3}	0.4405	mgN/L
C_{on}	1.781	mgN/L
C_{ip}	0.0015	mgP/L
C_{op}	0.068	mgP/L
C_{chl}	0.011815	mgC/L

Table 6.1. Initial condition for the simulation at station 8

The simulation of equation (6.4) with the initial condition given in Table 6.1 leads to the results shown in Figure 6.2. When we compare the simulation results with the real data we see a big difference with the reality (Figure 6.3). To improve the model, we need to calibrate it by modifying the input parameters of the model. As there is a lot of parameters to identify, we will first use the sensitivity analysis to identify the most important parameters for each equation of the water quality components.

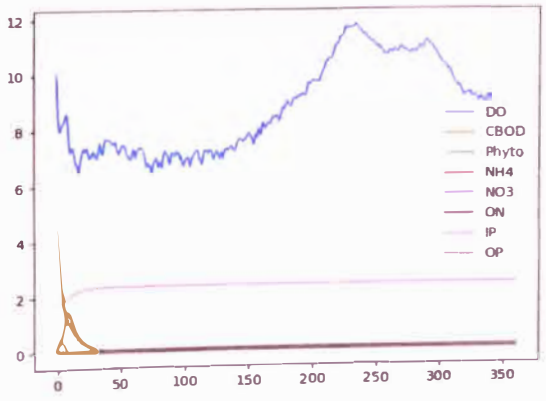


Figure 6.2. Water quality simulation

We perform the sensitivity analysis on a simple model without hydrodynamics because it is less complex and computationally expensive than the coupled hydro-ecological model that takes into account all the biological processes with the effects of

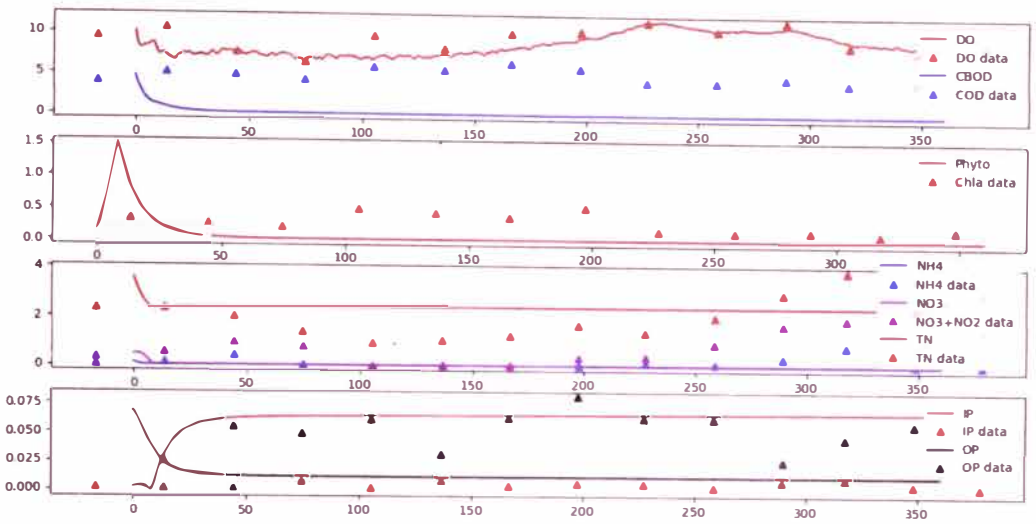


Figure 6.3. Compare with the real data

the diffusion, advection and the hydrodynamics. The code developed is organized in several folders and files (see appendix B for more details):

- The Data folder contains all the measured data available at each station of the lake Taihu during the years 2001 and 2002 as explained in Section 5.2.1.
- The *fonctions.py* script contains the equation of the water quality model as described in Section 3.4, the complementary equation of Section 3.5 and the functions necessary to obtain the data of the Data folder and make them usable in the simulations. The script *parametres.py* contains the information about the parameter values involved in the equations (see Appendix A).
- The *Modele_WASP_lac_Tai_v4.py* script is the main script of the water quality model implementation. It uses the functions of the script *fonctions.py*. This script performs the import of the Data folder, the integration of the ordinary differential equation with the *ode*() function of the Scipy package for one station and the plot of the real data and the results using the Matplotlib packages.

ODE Method

In the first version developed by Julien Diot, the script was using the *odeint*() function of the scientific packages Scipy for the integration, but when the scripts were running, the *odeint*() function returned a warning: *"ODEintWarning: Excess work done on this call(perhaps wrong Dfun type). Run with full_output=1 to get quantitative information."*. This problem comes from the fact that the equations we wanted to simulate can have strong variations in a short period of the time (it is a problem that is "stiff") and the function can't do correctly the computation. To solve this problem, we replaced the *odeint*() function by the *ode*() function of Scipy which has more options for the integration but whose computational cost can be more expensive than the one of the *odein*() function.

6.3.2 SA of WQM

For the SA of the WQM model, we considered 40 parameters which are given in Appendix A.4 with their respective lower and upper bounds (a_i and b_i) that are necessary for the SA. We adapted the WQM implemented in Section 6.3.1 to obtain the output of each water quality component. In python, there exists the SALib package to do the SA, but the SA method that is implemented in this package only works with real values function. In our case, as explained before, we needed to apply the SA for the WQM that has 8 water quality variable outputs, each of them is a function of the time. The files for the SA implementation and all the files necessary for the simulation of the WQM are explained in details in Appendix B.3. Here we will explain only a few important files used for the simulation:

- The *parameters.py* script contains the number parameters and the definition interval for each parameter. This is necessary to obtain the random trajectories for the Morris method.
- The *morris_methode_v4.py* script generates 2 files: one for each sensitivy measure that are stored in a folder. This script takes the information about the parameters in *parameters.py*, and then uses the sample function of the packages SAlib to generate a random trajectories set which will be used to compute the output of the model with the $Model\_WASP()$ function. Finally, the script computes the elementary effect for each output and each input factor and compute sensitivity measures mu^* and σ which are stored in the corresponding files.
- The *interactive_plotting.py* script enables to plot the sensitivity analysis results for the Morris method. This script needs as input the name of the folder in which the results are stored: then it imports the results and make an interactive plotting using the Bokeh packages.

SAMPLE Method

As explained before, an important step of the SA is the generation of the random trajectories that can be performed as explained in Section 6.1.2. For the numerical implementation, we used a function of the SAlib package which is optimized.

Verification of the results

In the literature, the authors usually consider between 4 and 10 random trajectories to do the SA. This number depends on the inputs number, the computational model cost and the choice of the level number. During my internship, I studied the effect of the trajectories number on the result of the SA. I will now explain the script that I have developed to do that:

- The *morris_methode_v4_analysis_box.py* script uses the *morris_methode_v4.py* to do the SA with all the even number of trajectories between 2 until 100. Moreover, it generates a folder with all the values of the SA measures.

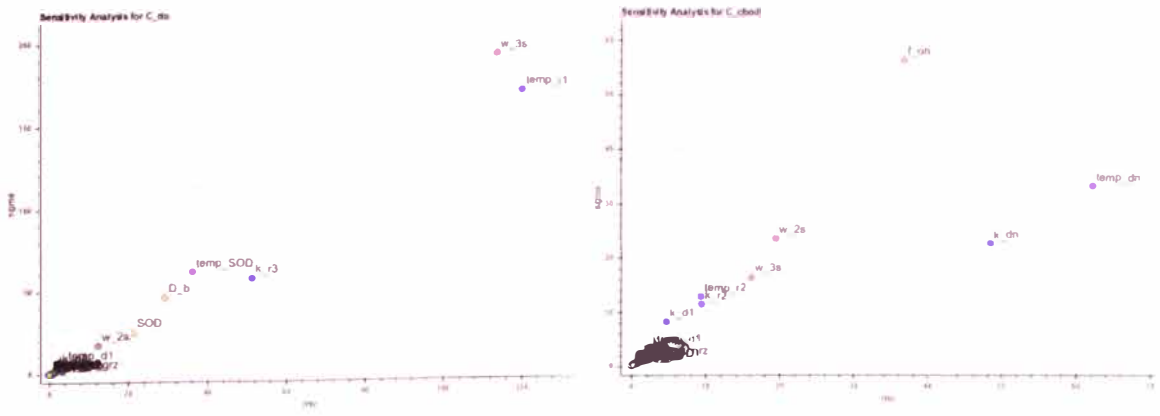
- The *interactive_plotting_analysis_box.py* script uses the *interactive_plotting.py* script as template to plot the results of the *morris_methode_v4_analysis_box.py* with the Bokeh packages. It has three options to plot the SA results:
  - **Trajectories**:
    A plot of μ and σ value versus the number of trajectories for each water quality components and each parameter.
  - **Box**:
    A box-plot of the SA measure versus the parameters for each water-quality variable.
  - **Mean**:
    A plot of the mean value of μ_i versus the mean value of σ_i for each water quality component.

6.4 Results

We performed the SA for the equation (6.2) with $r = 4$ trajectories and compare the results with the mean values of the SA results obtained with an even number of trajectories between 2 and 100. For all the SA performed, the method used is the Morris Method that is explained in the section 6.1. For all the simulations we used the initial condition given in Table 6.1 and the SA is applied over the 40 parameters given in Table A.8.

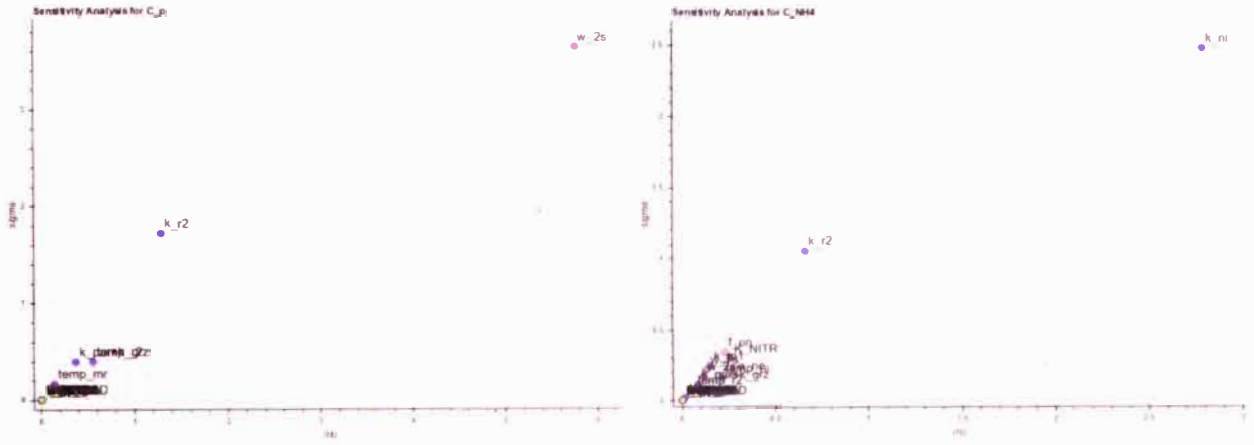
Sensitivity analysis using $r = 4$ trajectories

The SA results obtained with $r = 4$ trajectories are shown in Figures 6.4, 6.5, 6.6 and 6.7; it gives us the order of importance among all the 40 parameters through a plot of σ versus μ for each water quality variable.



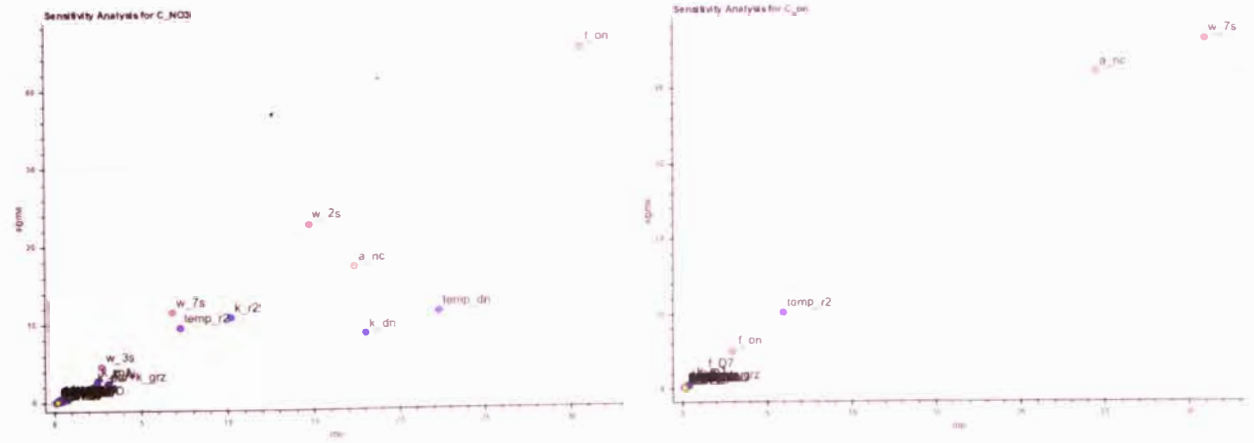
(a) C_{do} (b) C_{cbod}

Figure 6.4. SA of the C_{do} and C_{cbod} components



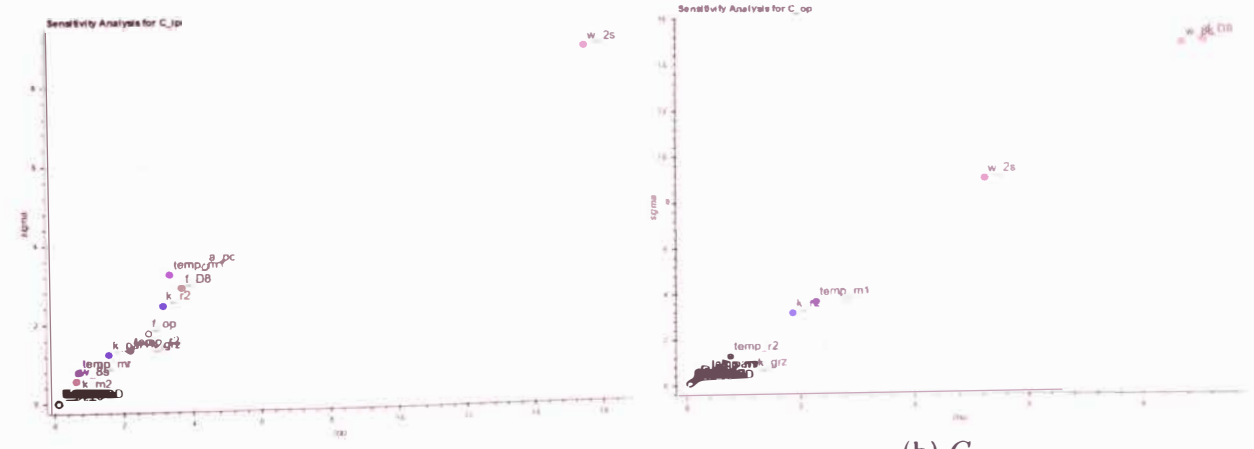
(a) C_p (b) C_{NH4}

Figure 6.5. SA of the C_p and C_{NH4} components



(a) C_{NO3} (b) C_{on}

Figure 6.6. SA of the C_{NO3} and C_{on} components



(a) C_{ip} (b) C_{op}

Figure 6.7. SA of the C_{ip} and C_{op} components

Parameters	C_{do}	C_{cbod}	C_p	C_{NH4}	C_{NO3}	C_{ip}	C_{on}	C_{op}
k_d1	9	8						
k_r2	10	7	2	2	6	5	9	5
k_r3	3							
k_ni				1	10			
k_dn	11	3			5			
k_gr								
$k\_par + k\_grz$			4	10	12	8	7	7
k_m1		11		7	11		6	
k_m2								
$temp\_r1$	2				13			
$temp\_d1$	8	9						
$temp\_ni$				6				
$temp\_r2$		6	3	11	8	7	3	6
$temp\_dn$		1			4			
$temp\_gr$								
$temp\_mr$			5			9	10	8
$temp\_m1$						3		4
$temp\_m2$								
$temp\_SOD$	4							
K_BOD								
K_NITR				3				
K_NO3								
K_mN								
K_mP								
K_mPc								
w_2s	7	4	1	9	2	1	8	3
w_3s	1	5			9			
w_7s				8	7		1	
w_8s						10		2
f_D3								
f_D7							5	
f_D8						4		1
f_on		2		4	1		4	
f_op						6		10
a_nc		10		5	3		2	
a_pc						2		9
D_b	5							
I_s								
K_e_prim								
SOD	6							

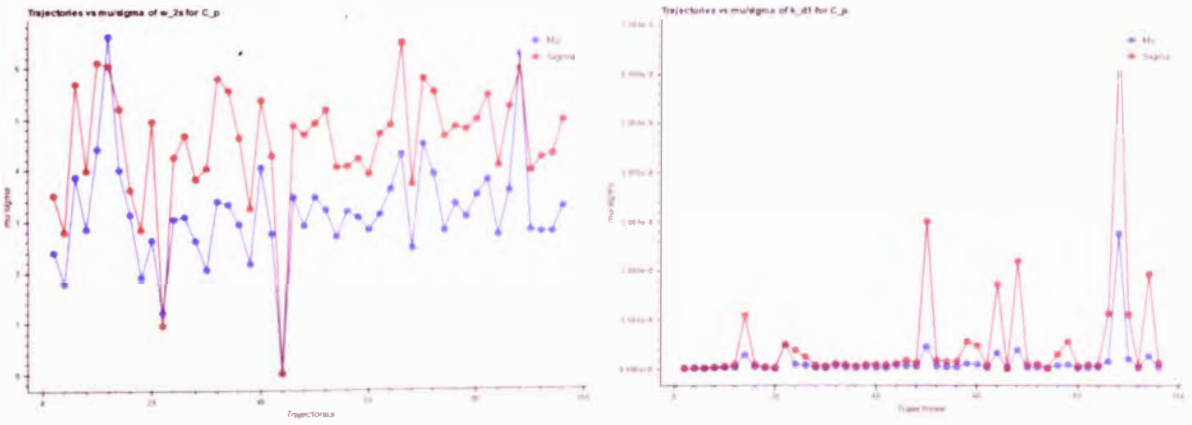
Table 6.2. Order of importance of the input parameters using the SA with $r = 4$ trajectories

In Table 6.2, the most important parameters that have μ and σ values bigger than one are given for each water quality component. In the case of C_p we obtained that the most important parameters are (in this order) w_2s, k_r2, $temp\_r2$, $k\_par + k_grz$

and *temp_mr*. For the other parameters, the corresponding μ and σ values are near zero, which means that if we change a little the parameter's value we will obtain a similar output value for C_p. But if we change a little bit any of the five important parameters listed above, the output value of C_p will change significantly. In the case of C_{cbod}, it is more complicated to identify the most important parameters because the points in the plot of σ versus μ are scattered (Figure 6.4(b)). For example for *f_on* and *temp_dn*, both parameters are important but *f_on* is more nonlinear or has more interactions than *temp_dn*, and the output value is more sensitivity to *temp_dn* than to*f_on*. In the same way, we can identify the parameters that are more important for all or almost all the water quality components like *k_r2*, *k_par* + *k_grz*, *temp_r2*, *w_2s*, *f_on*, *f_op* and *k_m1*.

Impact of the number of trajectories in the SA results

In the Figures 6.8(a) and 6.8(b) we see the values for μ (blue) and σ (red) obtained for the parameters *w_2s* and *k_d1* and for the output variable C_p for different number of trajectories (even number between 2 and 100).



(a) Parameter: *w_2s* (b) Parameter: *k_d1*

Figure 6.8. Impact of the number of trajectories for C_p

In Figures 6.9 and 6.10 we see the mean values of all SA measures that have been obtained with even numbers of trajectories between 2 and 100 of trajectories, for the output variable C_p and the parameters w_{2s} and k_{d1}. For each parameter we use a box plot where it is possible to identify the maximum, minimum and mean values and also the interquartile range. For example, in the Figure 6.9 it is easy to identify the five important parameters for μ and in Figure 6.10 we identify the same five important parameters for the σ.

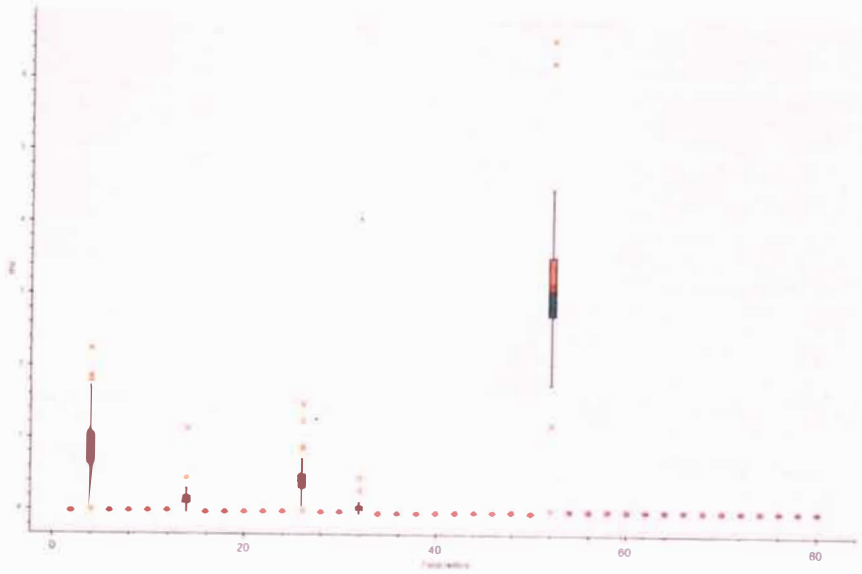


Figure 6.9. Box plotting of the μ of the impact of the number of trajectories for C_p.

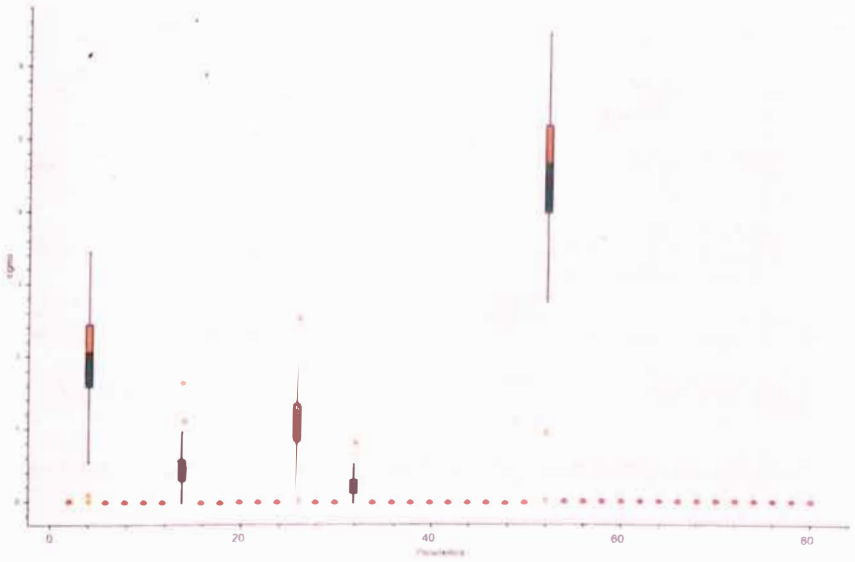
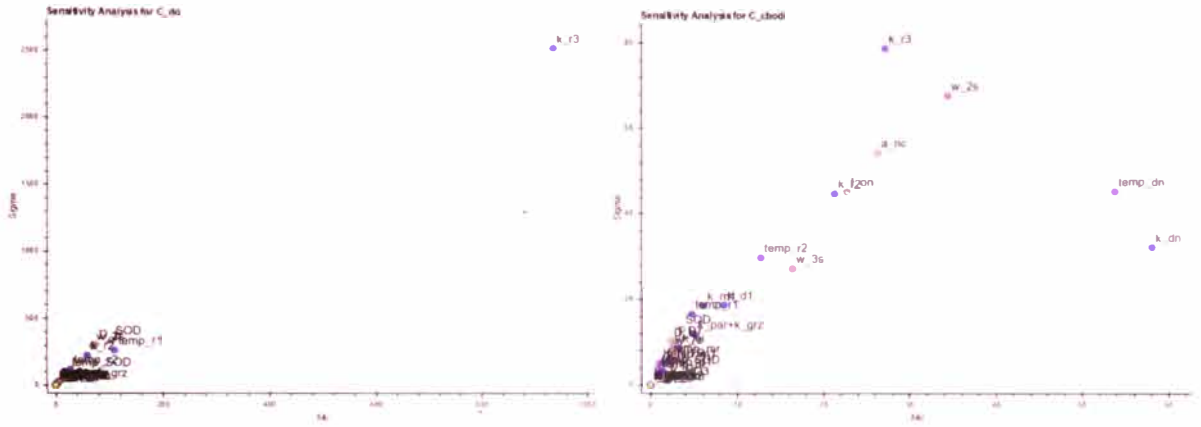


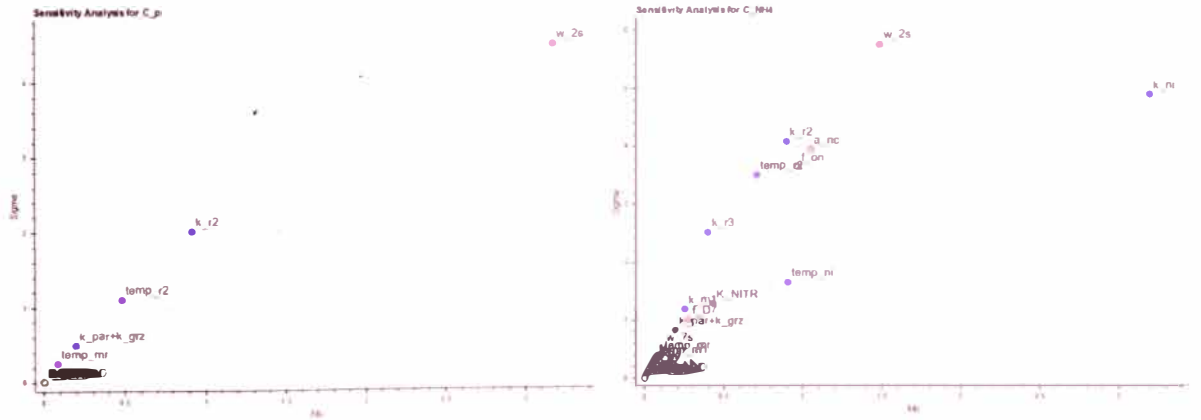
Figure 6.10. Box plotting of the σ of the impact of the number of trajectories for C_p.

In Figures 6.11, 6.12, 6.13 and 6.14, we see the plots of the mean value of σ versus the mean value of μ for each water quality variable and each parameter, the mean value being computed over the set of SA performed with different number of trajectories r. These plots give us the order of importance among all the 40 parameters for each water quality component.



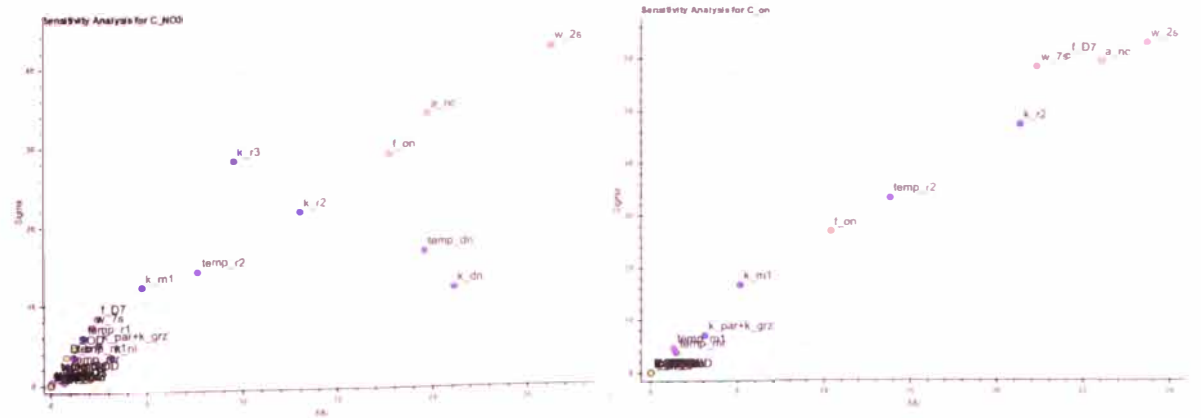
(a) For C_{do} (b) For C_{cbod}

Figure 6.11. mean's value of the SA of the SA analysis using the trajectories even number between 2 and 100.



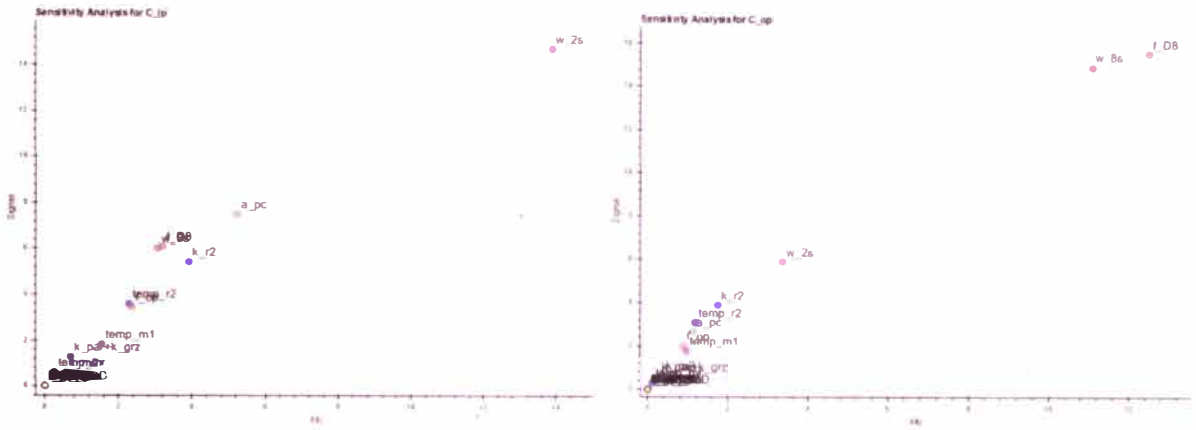
(a) For C_p (b) For C_{NH4}

Figure 6.12. mean's value of the SA of the SA analysis using the trajectories even number between 2 and 100.



(a) For C_{NO3} (b) For C_{on}

Figure 6.13. mean's value of the SA of the SA analysis using the trajectories even number between 2 and 100.



(a) For C_{ip} (b) For C_{op}

Figure 6.14. mean's value of the SA of the SA analysis using the trajectories even number between 2 and 100.

Parameters	C_{do}	C_{cbod}	C_p	C_{NH4}	C_{NO3}	C_{ip}	C_{on}	C_{op}
k_d1								
k_r2	6	7	2	4	5	3	5	4
k_r3	1	1		7	4			
k_ni				1				
k_dn		5			7			
k_gr								
$k\_par + k\_grz$			4	12		9	9	10
k_m1	9			10	9		8	
k_m2						10		9
$temp\_r1$	3							
$temp\_d1$								
$temp\_ni$				8				
$temp\_r2$	7	8	3	6	8	6	6	5
$temp\_dn$		3			6			
$temp\_gr$								
$temp\_mr$			5			11	11	11
$temp\_m1$						8	10	8
$temp\_m2$								
$temp\_SOD$	8							
K_BOD								
K_NITR				9				
K_NO3								
K_mN								
K_mP								
K_mPc								
w_2s	4	2	1	2	1	1	1	3
w_3s		9						
w_7s	10				11		4	
w_8s						5		2
f_D3								

f_D7				11	10		3	
f_D8						4		1
f_on		6		5	3		7	
f_op						7		7
a_nc		4		3	2		2	
a_pc						2		6
D_b	5							
I_s								
K_e_prim								
SOD	2							

Table 6.4. Order of importance using the mean values of the SA analysis.

In Table 6.4, we can see the order of the most importance parameters for each water quality component, order that has been obtained with the mean values of the Sa results. In the case of C_p for example, the most important parameters are w_2s, k_r2, $temp\_r2$, $k\_par + k\_grz$ and $temp\_mr$.

6.5 Discussion

When looking at the results of the study of the impact of the number of trajectories on the sensitivity analysis results that are given in Figures 6.8(a) and 6.8(b), we see that the values are most of the time similar but that there can be some peaks for certain number of trajectories. When plotting the box graph for μ in Figure 6.9 and σ in Figure 6.10, which include the maximum, minimum value and the average of the values for each parameter, we can clearly identify the most important parameters for each water quality component.

Then, when comparing for each water quality component the results through Tables 6.2 and 6.4, we see that the most important parameters are similar except for one variable and that for parameters with very close values the order of importance can be permuted between them. In the case of C_p the most important parameters are the same and in the same order: w_2s, k_r2, $temp\_r2$, $k\_par + k\_grz$ and $temp\_mr$. In the case of C_{NH4} we have almost the same parameters for both table except the difference in the parameters k_r3 and f_D7. In the case of C_{CNO3} the three first most important parameters f_on, w_2s and a_nc are the same but in different order for both tables. But in the SA it does not consider the follow parameters as the first ten most important: $temp\_dn$, w_3s, k_ni, $k\_par + k\_grz$ and $temp\_r1$, but the first SA using just only four trajectories takes in account these parameters.

Using the error defined by the following expression:

$$error = \frac{|values - mean_{values}|}{|mean_{values}|},$$

we can see if the values obtained with both analysis are similar. The error calculated for the five most important parameters for C_p are given in Table 6.5 for μ and in Table

6.6 for σ. In both cases, even the error values are high we still have the same order of the parameters, which means that the sensitivity analysis performed with $r = 4$ trajectories is sufficient for C_p.

Parameters	μ	μ_{mean}	$error$
k_r2	1.272260e+00	9.165082e-01	3.935783e-01
$k\_par + k\_grz$	3.587056e-01	1.943630e-01	8.455446e-01
$temp\_r2$	5.436709e-01	4.793158e-01	1.342645e-01
$temp\_mr$	1.301971e-01	8.153164e-02	5.968904e-01
w_2s	5.750091e+00	3.169842e+00	8.139992e-01

Table 6.5. μ's error between the result of the SA with 4 trajectories and the mean values.

Parameters	σ	σ_{mean}	$error$
k_r2	1.715766e+00	2.010826e+00	1.467357e-01
$k\_par + k\_grz$	3.921153e-01	4.906442e-01	2.186515e-02
$temp\_r2$	3.932476e-01	1.095945e+00	6.411794e-01
$temp\_mr$	1.668047e-01	2.429921e-01	3.135385e-01
w_2s	3.661535e+00	4.506207e+00	1.874463e-01

Table 6.6. σ's error between the result of the SA with 4 trajectories and the mean values.

Chapter 7

Conclusions and Recommendations

The first topic, the computational time is so long for the simulation using Python, the recommendation about this problem it is to change to one compiled programming languages like C or Fortran. Python code is not well-adapted for such a real case, that is why we will use a code such as FVCOM written in C and Fortran that is optimized and parallelized for future simulation. Even when we try to work with a small mesh, the result are fastest, we loss information about how the water quality components move with the current effect. In this context it is not recommendable do finer the mesh for the Lake Taihu, because the mesh is very small and the lake very tall as we see in Figure 5.5(b).

The $CBOD$ component takes negative values for the simulation. The same problem with $CBOD$, $NH4$, $NO3$, ON and OP components when we use the rivers to compute the velocity which is used for the simulation. For the moments the model does not work using the velocity consider the rivers to generate the currents with the FVCOM.

The coupling of the model was successful without consider the rivers on the boundary neither the sediment and benthic flux. Then, it is necessary to do better the model consider in 3D shallow water model (using layers) with the settling and benthic flux at the water quality model which are the connection between the hydrodynamics and ecological part.

About the Sensitivity Analysis, it was possibly find the most important parameters for the coupled model: k_r2, $k\_par + k\_grz$, $temp\_r2$, w_2s, f_on, f_op and k_m1 which are the parameters more relevant for each water quality component as we can see in Table 6.2. Even, it was possible do the analysis to show the independent of trajectories number with the sensitivity analysis measure (Table 6.4).

With this information, we are able to do the calibration to improve the model for the real case (the Lake Taihu). Even, it possible to do the sensitivity analysis with a set of parameters to improve the knowledge of which parameters are the most important to the model.

The next step for this work, it is add the condition of the rivers on the boundary

an important part of the boundary condition and maybe improve the complementary equations (Water Quality Model). After, it is make the calibration of the model for the particular case of the Lake Taihu and also add the sediment and the benthic flux to improve the model considering the 3D shallow water model.

Appendix A

Parameters

All the follow parameters values are in the follow bibliography: [1], [8], [10], [19], [20], [21] and [22].

A.1 Constant parameters of the model

Name	Description	Unit	FVCOM notation
a_{cchla}	Phytoplankton carbon to chlorophyll a ratio	$mgC \cdot (mgChla)^{-1}$	80
a_{nc}	Phytoplankton nitrogen-carbon ratio	$mgN \cdot (mgC)^{-1}$	RATIO_NC
a_{pc}	Phytoplankton phosphorus-carbon ratio	$mgP \cdot (mgC)^{-1}$	RATIO_PC
D	Depth (or more precisely height) of the water column (> 0)	m	DEP
D_s	Depth (or more precisely height) of the current model segment (> 0)	m	D(I)
D_b	Depth of benthic layer (> 0)	m	0.7 in FVCOM
f_{D3}	Fraction of CBOD which is dissolved	-	FD2
f_{D7}	Fraction of organic nitrogen which is dissolved	-	FD6
f_{D8}	Fraction of organic phosphorus which is dissolved	-	FD8
f_{on}	Fraction of dead and respired phytoplankton recycled to organic nitrogen pool	-	F_ONN(I)
f_{op}	Fraction of dead and respired phytoplankton recycled to organic phosphorus pool	-	FOP or F_OPP(I)
f_u	a unit conversion factor	mole photons$\cdot m^{-2} \cdot ly^{-1}$	
k_{d1}	CBOD deoxygenation rate	d^{-1}	K_DEOX
k_{dn}	Denitrification rate	d^{-1}	K_DENI

k_{gr}	Phytoplankton optimum growth rate at the reference temperature T r and optimum light/nutrients	d^{-1}	K_GROW
k_{m1}	Organic nitrogen mineralization rate	d^{-1}	K_MINE1
k_{m2}	Organic phosphorus mineralization rate	d^{-1}	K_MINE2
k_{ni}	Nitrification rate	d^{-1}	K_NITRR(I,K)
$k_{par}+$ k_{grz}	Phytoplankton basal loss rate	d^{-1}	K_MORT
k_{r2}	Phytoplankton endogenous respiration rate at the reference temperature T_r	d^{-1}	K_RESP
k_{r3}	Bacterial respiration rate	$mgO_2 \cdot d^{-1}$	K_RESP1 * 32 * 24 * 1.0E-3
K_{BOL}	Half-saturation concentration for oxygen limitation of CBOD oxidation	$mgO_2 \cdot L^{-1}$	KBOD
K_{mN}	Half-saturation concentration for nitrogen uptake	$mgN \cdot L^{-1}$	KMN*1.0E-3
K_{mP}	Half-saturation concentration for phosphorus uptake	$mgP \cdot L^{-1}$	KMP*1.0E-3
K_{mPc}	Half-saturation concentration for phytoplankton limitation of mineralization	$mgC \cdot L^{-1}$	KMPC
K_{NIT}	Half-saturation concentration for oxygen limitation of nitrification	$mgO_2 \cdot L^{-1}$	KNITR
K_{NO_3}	Half-saturation concentration for oxygen limitation of denitrification	$mgO_2 \cdot L^{-1}$	KNO3
K_c	the extinction coefficient per unit of chlorophyll a	$m^2 \cdot (mgChla)^{-1}$	
K'_e	Non algal light attenuation	m^{-1}	not used
r^1_{sed}	NH_4	-	RSED1(I)
r^2_{sed}	$NO_3 + NO_2$	-	RSED2(I)
r^3_{sed}	OPO_4	-	RSED3(I)
SOD	Sediment oxygen demand rate	$mgO_2 \cdot m^{-2} \cdot d^{-1}$	SODD(I,K)
T_r	reference temperature	$°C$	
w_{2S}	Settling velocity of PHYT	$m.d^{-1}$	WSS3
w_{3S}	Settling velocity of CBOD	$m.d^{-1}$	WSS2
w_{7S}	Settling velocity of particulate organic nitrogen	$m.d^{-1}$	WSS3 ($\omega_{7S} = \omega_{2S}$)
w_{8S}	Settling velocity of particulate organic phosphorus	$m.d^{-1}$	WSS3 ($\omega_{8S} = \omega_{2S}$)
θ_{d1}	Temperature adjustment factor for deoxygenation rate	-	TEMP_DEOX
θ_{dn}	Temperature adjustment factor for denitrification rate	-	TEMP_DENI

θ_{gr}	Temperature adjustment factor for phytoplankton growth rate	-	TEMP_GROW

Table A.2. Constant parameters of the model

A.2 Universal constants

Name	Description	Value
e	the base of natural logarithm	2.718[−]
r_{oc}	Ratio of dioxygen to carbon molar mass	$32/12 mgO_2 \cdot (mgC)^{-1}$
r_{on}	Ratio of dioxygen to nitrogen molar mass	$32/14 mgO_2 \cdot (mgN)^{-1}$
m_C	molar mass of the carbon	$12g \cdot mol^{-1}$
m_{O_2}	molar mass of dioxygen (O_2)	$32g \cdot mol^{-1}$
m_N	molar mass of the nitrogen	$14g \cdot mol^{-1}$
ρ_ω	density of water	$1g \cdot cm^{-3}$
κ	von Karman's coefficient	0.4[−]

Table A.4. Universal constants

A.3 External inputs of the model

Name	Description	Unit	FVCOM notation
I	Daily average incident solar radiation	$ly \cdot d^{-1}$	SOLAR_A
t_U	sunrise hour in the day (between 0 and 24)	h	TIME_u
T_D	sunset hour in the day (between 0 and 24)	h	time_d
W	time-varying wind speed at 10m above surface	$m \cdot s^{-1}$	
$\bar{v}$	average water velocity	$m \cdot s^{-1}$	
T_a	air temperature	$°C$	
ρ_a	density of air: either measured or given expressed as a function of T_a	$g \cdot cm^{-3}$	
ν_a	viscosity of air: either measured or given expressed as a function of T_a	$cm^2 \cdot s^{-1}$	

Table A.6. External inputs of the model

A.4 Parameter values for the sensitivity analysis

Names	min	max	init
k_d1	0.02	0.3	0.2
k_r2	0.02	0.6	0.1
k_r3	0.0199	0.201	0
k_ni	0.05	0.6	0.2
k_dn	0.09	0.16	0.11
k_gr	1	3	1.28
$k\_par + k\_grz$	0.01	0.1	0.04
k_m1	0.003	0.14	0.075
k_m2	0.18	0.22	0.2
$temp\_r1$	1.02	1.08	1.028
$temp\_d1$	1.02	1.08	1.047
$temp\_ni$	1.02	1.08	1.06
$temp\_r2$	1.01	1.2	1.05
$temp\_dn$	1.02	1.09	1.06
$temp\_gr$	1.01	1.2	1.066
$temp\_mr$	1	1.08	1.02
$temp\_m1$	1.02	1.09	1.06
$temp\_m2$	1.02	1.08	1.08
$temp\_SOD$	1.02	1.08	1.07
K_BOD	0.499	0.501	0.5
K_NITR	0.1	2	0.7
K_NO3	0.099	0.101	0.1
K_mN	0.02	0.23	0.025
K_mP	0.0009	0.052	0.002
K_mPc	0.099	0.101	1.0
w_2s	0.7	18	0
w_3s	0.04	800	0
w_7s	0.04	800	0
w_8s	0.04	800	400
f_D3	0.499	0.501	0.5
f_D7	0.99	1	1.0
f_D8	0.99	1	1.0
f_on	0.5	1	0.7
f_op	0.2	0.7	0.5
a_nc	0.05	0.48	0.25
a_pc	0.01	0.047	0.025
D_b	0.1	0.5	0.25
I_s	200	500	300
K_e_prim	0.13	10	1.5
SOD	0	4	1

Table A.8. Parameters and definition interval of the parameters

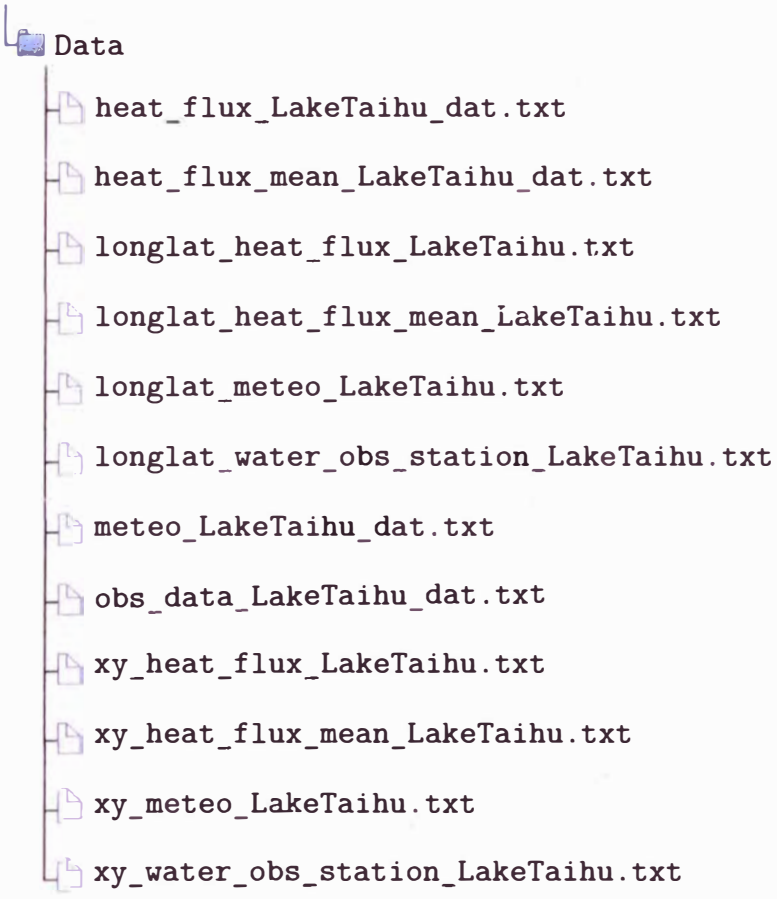
Appendix B

Implementation

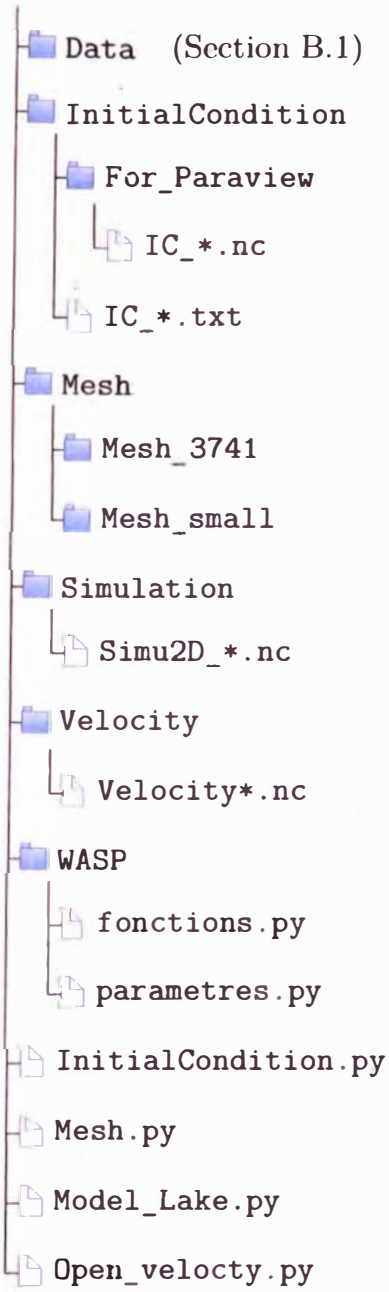
All the implementation was performed in the programming language Python3 and all the command lines presented hereafter are for suitable for running the scripts in Ubuntu.

B.1 Data

All the data discrete in Section 5.2.1 are organized in the files into the folder *Data*.



B.2 Coupled Model



Let us now describe some important functions used in the code to understand how to simulate the WQM:

- **fonctions.Ecosystem(t, x, fT_w, fT_a, fuW, fvW, ftU, ftD, fI, fS)**
  Return an array of length 8 in which each component represents the variation of one water quality component over a short variation of time. This function is called in the integration method that numerically solves the model, that is in the ode() function.

input :
- **t** (array) : time.
- **x** (array) : water quality components.

- **fT_w** (function): interpolated water temperature function.
- **fT_a** (function) : interpolated air temperature function.
- **fuW** (function) : interpolated wind rate function in the x-coordinate.
- **fvW** (function) : interpolated wind rate function in the y-coordinate.
- **ftU** (function) : interpolated sunrise hour function in the day.
- **ftD** (function) : interpolated sunset hour function in the day.
- **fI** (function) : interpolate daily average incident solar radiation function.
- **fS** (function) : interpolate water salinity function.

output :

- **dx** (numpy.array) : variation of water quality components.

Then, we implemented the FVM function that solves the transport equation in some domain with some triangle mesh, initial condition, boundary condition and a given velocity field.

- **FVM(Ver, Tri, aTri, Tad, aTad, aAad, aCad, aCen, P_t, Bou, IC, BC, NC, U, K, WQM, fT_w, fT_a, fuW, fvW, ftU, ftD, fI, fS)**
  Return the solution of the equation (3.1) solved with Finite Volume Method using unstructurade grid

input :

- **Ver** (array) : Mesh nodes.
- **Tri** (array) : Mesh triangles.
- **aTri** (array) : Auxiliary Mesh triangles.
- **Tad** (array) : Triangles close for each triangles
- **aTad** (array) : Triangles close for each triangles even the auxiliary triangles.
- **aAad** (array) : Edges close for each triangle even the auxiliary edges.
- **aCad** (array) : Centers close for each triangle even the auxiliary centers.
- **aCen** (array) : Center's triangle in 2D.
- **P_t** (array) : Partition of the time.
- **Bou** (array) : Boundary nodes.
- **IC** (array) : Initial condition.
- **BC** (array) : Boundary condition.
- **NC** (array) : Neumann condition.
- **U** (array) : Velocity.
- **K** (array) : Diffusion's coefficient.
- **WQM** (array) : Option to consider the ecological part.
- **fT_w** (function): interpolated water temperature function.
- **fT_a** (function) : interpolated air temperature function.
- **fuW** (function) : interpolated wind rate function in the x-coordinate.
- **fvW** (function) : interpolated wind rate function in the y-coordinate.
- **ftU** (function) : interpolated sunrise hour function in the day.

- **ftD** (function) : interpolated sunset hour function in the day.
- **fI** (function) : interpolate daily average incident solar radiation function.
- **fS** (function) : interpolate water salinity function.

output :

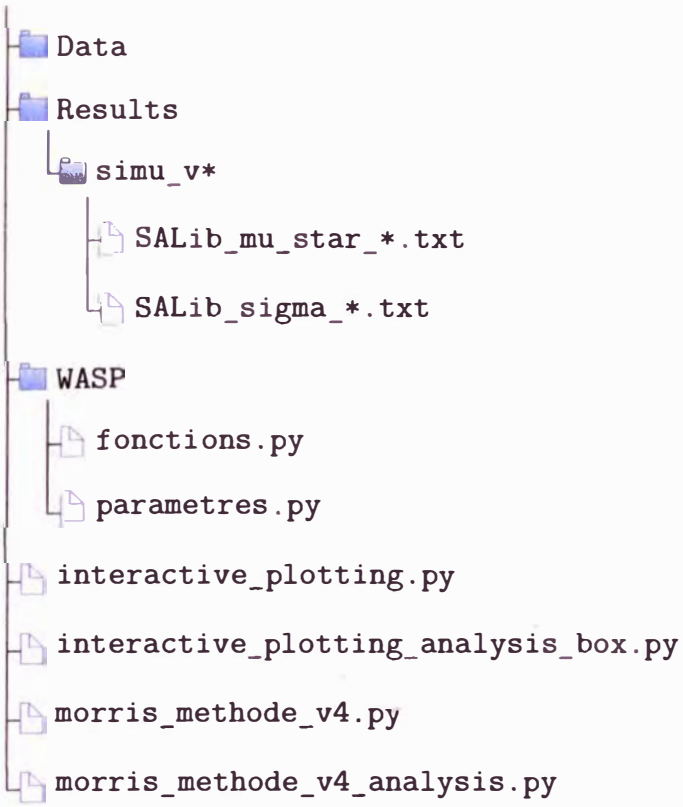
- **phi** (numpy.array) : variation of water quality components.

- **save(t, Ver, Tri, Cen, u, phi, NAME)**
  Generate a netCDF file with the simulation result of the coupled model. Save the results with the format compatible with FVCOM to be able to visualize the result with Paraview.

input :

- **t** (array) : Time.
- **Ver** (array) : The mesh nodes.
- **Tri** (array) : The mesh triangles.
- **Cen** (array) : The mesh center.
- **u** (numpy.array) : Velocity.
- **phi** (numpy.array) : Result of the water quality components.
- **Name** (string) : Name of the netCDF file.

B.3 Sensitivity Analysis



Now, we explain some important functions used in the code to understand how is implemented the sensitivity analysis:

- **morris_methode_v4.Model_WASP(X, dt=1, timeT=364)**
  Return a 2D-array where each component represents the output of the water quality model for a parameter's values set. This function is like the implementation of Section 6.3.1 for a parameter's values set.

input :
- **X** (array) : Parameters for the sensitivity analysis.
- **dt** (float) : Time between 2 points of the time partition in days (default 1).
- **timeT** (int): Duration of the modelling in days (default 364).

output :
- **Y** (array) : Solution of the WQM for X in a period of time for all water quality components.

- **interactive_plotting.all_parameters()**
  Return a dictionary with the the parameters names, parameters number and their definition intervals.

output :
- **problem** (dict) : Information about the parameters names, definition intervals and number of parameters.

- **interactive_plotting.all_names(folder)**
  Return the μ data and σ data that are extracted from the $name_mu$ and $name_sigma$ file.

input :
- **folder** (str) : File's name.

output :
- **name_mu** (array) : Data of the mu's result.
- **name_si** (array) : Data of the si's result.

- **interactive_plotting.all_data(name_mu, name_si)**
  Return the μ data and σ data which they are importing of the *name_mu* and *name_sigma* file.

input :
- **name_mu** (str) : Name of the file which contained the data of mu.
- **name_si** (str) : Name of the file which contained the data of sigma.

output :
- **mu** (array) : Data of mu.
- **si** (array) : Data of sigma.

- **SALib.sample.morris.sample(problem, N, num_levels, grid_jump, optimal_trajectories=None, local_optimization=True)**
  Returns a NumPy matrix containing the model inputs required for Method of Morris. The resulting matrix has (G+1)*T rows and D columns, where

D is the number of parameters, G is the number of groups (if no groups are selected, the number of parameters). T is the number of trajectories N, or *optimal_trajectories* if selected.

input :

- **problem** (dict) : The problem definition.
- **N** (int) : The number of trajectories to generate.
- **num_levels** (int) : The number of grid levels.
- **grid_jump** (int) : The grid jump size.
- **optimal_trajectories** (int) : The number of optimal trajectories to sample (between 2 and N).
- **local_optimization** (bool) : Flag whether to use local optimization according to Ruano et al. (2012) Speeds up the process tremendously for bigger N and num_levels. If set to False brute force method is used, unless gurobipy is available (default=True).

output :

- **sample** (array) :Returns a numpy.ndarray containing the model inputs required for Method of Morris. The resulting matrix has (G/D+1)*N/T rows and D columns, where D is the number of parameters.

B.4 Command line

To execute the model script in Ubuntu it is necessary to generate some input files that will have to be stored in the folders described in Appendices B.1, B.2, B.3.

Mesh

To generate the mesh files, we use the script Mesh.py in which we just have to modify the file's name with the nodes coordinates (*name_n*) and the triangles vertex (*name_t*).

```
...
name_n = 'LakeTaihu_cor.dat'
name_t = 'LakeTaihu_elements.txt'
...
```

Listing B.1. Script Mesh.py

```
$ python3 mesh.py
```

Initial Condition

For the generate the initial condition files are important to modify the files and folder names: (1) The folder mesh with 3741 nodes which is used as reference to identify the position of the rivers. (2) The rivers info file which include the rivers position (it means the node's number for each river in the mesh with 3741 nodes), (3) The rivers and stations values which content the values for each water quality component

in each rivers and station respectively, and (4) The folder mesh which we will be used in the simulation (for example the same mesh with 3741 nodes). This script generate an interpolation for each water quality component using the mesh which will be used in the simulation and the rivers and stations values (for example *'Rivers.txt'* and *'Stations.txt'*).

```
...
folder_m = './Mesh/Mesh_3741/'
...
folder_m_s = './Mesh/Mesh_small/'
...
name_IC_r = 'Rivers_zeros.txt'
name_IC_s = 'Stations_N3_p.txt'
...
name_save = 'IC_small_N3_p'
...
```

Listing B.2. Script InitialCondition.py

```
$ python3 InitialCondition.py
```

Velocity

For the velocity we use the script Open_velocity.py that extracts the velocity field from the file *name_file* in which the velocity values generated with FVCOM are stored and saves the values in the file with the name *name_save*.

```
...
name_file = '..\LakeTaihu_simu_20012002_small_mesh_0001.nc'
...
name_save = 'Velocity_small.nc'
...
```

Listing B.3. Script Open_velocity.py

```
$ python3 Open_velocity.py
```

Main

To execute the script *Model_Lake.py*, we have to modify the input files necessary to start the simulation like initial condition, mesh, etc.

```
07 # Time start
08 start = '01/05/2001'
09
10 ## File's names
11 # Mesh folder in the folder ./Mesh
12 name_folder = 'Mesh_small'
```

```
3
14 # Velocity name in the folder ./Velocity
15 name_velocity = 'Velocity_small.nc'
16
 7 # Initial Condition name in the folder ./InitialCondition
18 name_IC = 'IC_small_RealData.txt'
19
   # Data
 1 name_rivers = 'riv_info'                    # File of the info rivers
 2 name_flux = 'rivers_LakeTaihu_dat.txt'      # File of the rivers
23 fileT_w = 'obs_data_LakeTaihu_dat.txt'      # temperature de l'eau
24 fileT_a  = 'meteo_LakeTaihu_dat.txt'        # temperature de l'air
 5 fileWind= 'meteo_LakeTaihu_dat.txt'        # vitesse du vent
 6  fileI  = 'heat_flux_mean_LakeTaihu_dat.txt'    # radiation solaire en
      W/m^2
27  fileI2  = 'heat_flux_LakeTaihu_dat.txt'
 8
   # station number
30 nostation = 8
31
32 # Name for save the result
   NAME = 'Sim2D_1Y_W_RealData'

35 ## Option of the simulation
   # Rivers / Boundary Condition
 7 RIVERS = False

 9 # Ecological part
40 WQM = True
```

Listing B.4. Script Model_Lake.py

Finally, the command line to execute the main script of the coupled model and sensitivity analysis are given here-after:

```
$ python3 Model_Lake.py
```

Listing B.5. Execute the Coupled Model

```
$ python3 morris_methode_v4.py
```

Listing B.6. Execute the Sensitivity Analysis

Bibliography

[1] Boqiang Qin. *Lake Taihu, China: Dynamics and Environmental Change (Monographiae Biologicae)*, volume volume 87 of Monographiae Biologicae. springer edition, 2008.

[2] C. Chen, G. Cowles, and R. C. Beardsley. *An unstructured grid, finite-volume coastal ocean model: FVCOM User Manual.* 2004.

[3] Changsheng Chen, Hedong Liu, and Robert C. Beardsley. An unstructured grid, finite-volume, three-dimensional, primitive equations ocean model: Application to coastal ocean and estuaries. *Journal of Atmospheric and Oceanic Technology*, 20(1):159–186, 2003.

[4] W. Malalasekra H. Versteeg. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method (2nd Edition).* Prentice Hall, 2nd edition, 2007.

[5] Benoît Perthame Jean-Frédéric Gerbeau. Derivation of viscous saint-venant system for laminar shallow water; Numerical validation. *[Research Report] RR-4084, INRIA.*, 2000.

[6] Yu-E Shi. Résolution numérique des équations de Saint-Venant par la technique de projection en utilisant une méthode des volumes finis dans un maillage non structuré. Sciences de la Terre. Université de Caen, 2006. Français. <tel-00130539v2>.

[7] C. B. Vreugdenhil. *Numerical Methods for Shallow-Water Flow.* Water Science and Technology Library 13. Springer Netherlands, 1 edition, 1994.

[8] Zhen-Gang Ji. *Hydrodynamics and Water Quality: Modeling Rivers, Lakes, and Estuaries.* 2007.

[9] Song Toan Pham Phu. Research on the correlation between chlorophyll-a and organic matter bod, cod, phosphorus, and total nitrogen in stagnant lake basins. pages 177–191, 02 2014.

[10] Robert B. Ambrose, Tim A. Wool, and James L. Martin. *The water quality analysis simulation program, WASP5. Part A: model documentation.* 1993.

[11] Sandip Mazumder. *Numerical Methods for Partial Differential Equations: Finite Difference and Finite Volume Methods.* Elsevier AP, Academic Press, 2016.

[12] Python Software Foundation (PSF). Python. `"https://www.python.org/"`, 2001.

[13] Unidata University Corporation for Atmospheric Research. Network common data form (netcdf). "https://www.unidata.ucar.edu/software/netcdf/", 1993.

[14] Kitware Inc. National Technology and Engineering Solutions of Sandia, LLC (NTESS). Paraview. "https://www.paraview.org/", 2005.

[15] Lawrence C. Evans. *Partial Differential Equations: Second Edition.* Graduate Studies in Mathematics. American Mathematical Society, 2 edition, 2010.

[16] Francesca Campolongo Marco Ratto A. Saltelli, Stefano Tarantola. *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models.* Wiley, 1 edition, 2004.

[17] Max D. Morris. Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33:161–174, 1991.

[18] F. Campolongo, J. Cariboni, and A. Saltelli. An effective screening design for sensitivity analysis of large models. *ELSEVIER*, 22:1509–1518, 2007.

[19] Galen Bernard Kaufman. *Application of the Water Quality Analysis Simulation Program (WASP) to Evaluate Dissolved Nitrogen Concentrations in the Altamaha River Estuary, Georgia.* 2003.

[20] Jian Li, Danxun Li, and Xingkui Wang. Three-dimensional unstructured-mesh eutrophication model and its application to the xiangxi river, china. *Journal of Environmental Sciences*, 24(9):1569 – 1578, 2012.

[21] Weiping Hu, Sven E. Jørgensen, and Fabing Zhang. A vertical-compressed three-dimensional ecological model in lake taihu, china. *Ecological Modelling*, 190(3):367 – 398, 2006.

[22] Lianyuan Zheng, Changsheng Chen, and Frank Y. Zhang. Development of water quality model in the satilla river estuary, georgia. *Ecological Modelling*, 178(3):457 – 482, 2004.