University of Hamburg
Department of Mathematics

Mathematical Modelling in Engineering: Theory, Numerics and Applications

Parameter Identification for the Viscous Shallow Water Equations

Master's Thesis
Marco Vinicio Capó Rangel
Student ID: 6562780

1st examiner:	Jun.-Prof. Dr. Winnifried Wollner
2nd examiner:	Prof. Dr. Jörn Behrens
Thesis handed in on:	30.10.2014

Contents

Acknowledgements

First of all, I would like to thank Prof. Winnifried Wollner for his guidance and continuous support during the last six months. His guidance regarding technical matters on the Shallow Water Equations, optimization, finite elements and about the deal.ii and DopeLib libraries were invaluable for this work.

Additionally, I would like to send my sincere thanks to Prof. Bruno Rubino for allowing me to be part of this master's program, as well as the other coordinators, Prof. Ingenuin Gasser and, specially, Prof. Astrid Benz, for always supporting us and advising us in different topics regarding the academia and life in Hamburg.

Finally, I want to express my gratitude to my parents and sisters, who always had faith in me and supported me besides the critical situation in my country, Venezuela. Last but not least I want to thank Gabriela for being my major inspiration, partner and support the during last two years, no words can describe my gratefulness, I am thankful to have you all as my family.

Marco

Introduction

It is well known that the quality of the initial conditions plays an important role in the analysis of atmospheric dynamics and weather forecast. This is a direct consequence of the nonlinearity of all these geophysical processes which imposes an important sensitivity to the initial conditions [1, 23].

Parameter Identification has become a relevant topic in the environmental science, since it combines effectively all the sources of information (e.g. Observations of the ocean) with models and numerical methods to obtain an optimal representation of the state of the ocean, atmosphere, etc [29, 42]. This technique can rely on variational methods [38], while others use Kalman filtering and its extensions [17, 28], statistical interpolation [36] or more empirical approaches such as the nudging method [22, 39].

In this project we present an application of this technique on the Shallow Water Equations (SWE) based on a Galerkin finite element approach. The Shallow Water Equations model the propagation of disturbances in water and other incompressible fluids. They can also be used in numerical weather prediction to study large-scale wave in the atmosphere and ocean [33].

In Chapter 1, we present a brief outlook of the viscous SWE, we refer to the deduction of the SWE as well as the different ways to model the viscosity term and its properties. Finally, we state the settings that we considered in our formulation.

In Chapter 2, we describe the strategy used to simulate the SWE. Further, in order to examine the properties of the simulation solver, we consider three relevant examples and we address the following questions: Which time scheme fits our forward problem better? What is the effect that the addition of a viscosity term to the transport equation has on the simulation of the phenomenon? In the case of a quasistationary process, how can we improve our solutions without changing our current framework?

In Chapter 3, we refer to the Parameter Identification problem, we examine the optimization problem settings as well as the solution process that we followed. At the end of this section we present different examples under many settings to corroborate the proper performance of our solver.

Chapter 1

Shallow Water Equations

The SWE are used to model the motion of a shallow layer of homogeneous incompressible fluid in a three-dimensional rotating sub-domain and, in particular, to simulate the vertical average dynamics of the fluid in terms of the horizontal velocity and depth variation [21].

In general, this ODE system is modeled by the three-dimensional incompressible Navier-Stokes-Coriolis system in $\mathbb{R}^3$ together with a (nonlinear) free moving surface boundary condition and the underlying assumption that the depth of the fluid is small compared to the wave length of the disturbance, i.e. $\frac{H}{L} << 1$, where H and L are the characteristics values for the vertical and horizontal length scales of motion [44].

1.1 Derivation of the SWE

In order to have a better understanding of the phenomena described by the SWE, we will first have a short overview on the deduction of the SWE from the Navier Stokes equations for incompressible flows and the "free surface conditions", which are explained in [41, 44].

For our notation, p represents the pressure, η is the vertical displacement of the free surface, $\mathbf{u} = (u, v, w)$ the three-dimensional velocity, $\mathbf{v} = (u, v)$ the two-dimensional velocity, ρ the density, g the acceleration due to gravity and $h_0(x, y)$ the bottom topography (see Figure 1.1).

The three dimensional Navier Stokes equations and the "free surface conditions" are given by:

$$\begin{aligned}
&\text{Momentum equation:} && \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \frac{1}{\rho} \nabla p + g \hat{z} = 0, && (1.1)\\
&\text{Continuity equation:} && \nabla \cdot \mathbf{u} = 0, && (1.2)\\
&\text{Free surface condition:} && p = 0, \ \frac{\partial \eta}{\partial t} + \mathbf{v} \cdot \nabla \eta = w \ \text{ on } z = \eta(x, y, t), && (1.3)\\
&\text{Bottom boundary condition:} && \mathbf{u} \cdot \nabla (z + h_0(x, y)) = 0, \ \text{on } z = -h_0(x, y). && (1.4)
\end{aligned}$$

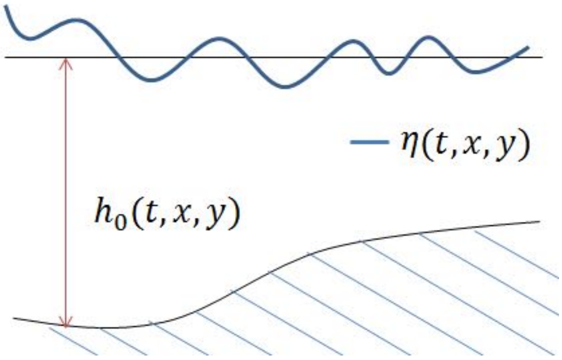

Figure 1.1: Definition of h_0 and η

First, we integrate the continuity equation vertically:

$$\begin{aligned}
0 &= \int_{-h_0}^{\eta} [\nabla \cdot \mathbf{u}] \, \mathrm{d}z \\
&= \int_{-h_0}^{\eta} \left[\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right] \, \mathrm{d}z \\
&\overset{\text{Leibniz}}{=} \frac{\partial}{\partial x} \int_{-h_0}^{\eta} u \, \mathrm{d}z - u\Big|_{z=\eta} \frac{\partial \eta}{\partial x} + u\Big|_{z=-h_0} \frac{\partial(-h_0)}{\partial x} \\
&\quad + \frac{\partial}{\partial y} \int_{-h_0}^{\eta} v \, \mathrm{d}z - v\Big|_{z=\eta} \frac{\partial \eta}{\partial y} + v\Big|_{z=-h_0} \frac{\partial(-h_0)}{\partial y} \\
&\quad + w\Big|_{z=\eta} - w\Big|_{z=-h_0}
\end{aligned} \tag{1.5}$$

Using (1.4), the last term in (1.5) is canceled. Hence, (1.5) can be rewritten as:

$$0 = \frac{\partial}{\partial x} \int_{-h_0}^{\eta} u \, \mathrm{d}z - u|_{z=\eta} \frac{\partial \eta}{\partial x} + \frac{\partial}{\partial y} \int_{-h_0}^{\eta} v \, \mathrm{d}z - v|_{z=\eta} \frac{\partial \eta}{\partial y} + w|_{z=\eta}. \tag{1.6}$$

Substituting (1.6) in the surface condition (1.3), one gets:

$$\boxed{\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x} \int_{-h_0}^{\eta} u \, \mathrm{d}z + \frac{\partial}{\partial y} \int_{-h_0}^{\eta} v \, \mathrm{d}z = 0.} \tag{1.7}$$

On the other hand, the last term of the momentum equations (1.1) is:

$$\frac{\partial}{\partial t} w + u \frac{\partial}{\partial x} w + v \frac{\partial}{\partial y} w + w \frac{\partial}{\partial z} w = -\frac{-1}{\rho} \frac{\partial}{\partial z} p - g,$$

or, in terms of the material derivative $\dfrac{\mathrm{d}w}{\mathrm{d}t}$:

$$\frac{\mathrm{d}w}{\mathrm{dt}} = \frac{-1}{\rho}\frac{\partial p}{\partial z} - g. \tag{1.8}$$

Using the "long wave approximation" the last equation can be used to determine a relation between the pressure and the acceleration due to gravity.

Long wave approximation: the wave length is much longer than the depth of the fluid, therefore we can assume that there are no vertical variations in (u, v) and neglecting the vertical acceleration, $\frac{\mathrm{d}w}{\mathrm{d}t} = 0$, in (1.8) (you can think of the SWE as a boundary layer [27]). We obtain:

$$\begin{aligned} \int_z^\eta \frac{\partial p}{\partial z}\,\mathrm{d}z &= -\int_z^\eta \rho g\,\mathrm{d}z, \\ p(x,y,\eta,t) - p(x,y,z,t) &= -\rho g(\eta(x,y,t) - z), \\ p(x,y,z,t) &= \rho g(\eta(x,y,t) - z), \end{aligned} \tag{1.9}$$

where we used the surface condition $p(x, y, \eta, t) = 0$. Using this expression for the hydrostatic pressure (1.9) and further assuming that there are no vertical variations in (u, v), the horizontal momentum equations of the shallow-water system are obtained as follows,

$$\begin{cases} \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} + v\dfrac{\partial u}{\partial y} + g\dfrac{\partial \eta}{\partial x} = 0, \\ \dfrac{\partial v}{\partial t} + u\dfrac{\partial v}{\partial x} + v\dfrac{\partial v}{\partial y} + g\dfrac{\partial \eta}{\partial y} = 0. \end{cases} \tag{1.10}$$

The conservation of mass given by (1.7) becomes:

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}[(\eta + h_0)u] + \frac{\partial}{\partial y}[(\eta + h_0)v] = 0. \tag{1.11}$$

Equations (1.10) and (1.11) are the shallow-water equations.

The non-viscous SWE system ((1.10),(1.11)) can be written in the following matrix form:

$$\frac{\partial}{\partial t}\begin{pmatrix} \eta \\ u \\ v \end{pmatrix} + \begin{bmatrix} u & \eta + h_0 & 0 \\ g & u & 0 \\ 0 & 0 & u \end{bmatrix} \frac{\partial}{\partial x}\begin{pmatrix} \eta \\ u \\ v \end{pmatrix} + \begin{bmatrix} v & 0 & \eta + h_0 \\ 0 & v & 0 \\ g & 0 & v \end{bmatrix} \frac{\partial}{\partial y}\begin{pmatrix} \eta \\ u \\ v \end{pmatrix} = -\begin{pmatrix} u\dfrac{\partial h}{\partial x} + v\dfrac{\partial h}{\partial y} \\ 0 \\ 0 \end{pmatrix}.$$

The eigenvalues of the first coefficient matrix are:

$$u, \quad u \pm \sqrt{g(\eta + h_0)}, \tag{1.12}$$

and those of the second coefficient matrix are:

$$v, \quad v \pm \sqrt{g(\eta + h_0)}. \tag{1.13}$$

Since the eigenvalues (1.12) and (1.13) are real and distinct, the non-viscous shallow-water equations are hyperbolic partial differential equations [41]. Therefore, as we can see in [24], the SWE admit discontinuous solutions. Such discontinuities are called "bore" and they approximate a breaking wave [41].

Additionally, if we consider the shallow-water equations in a rotating frame (the rotation axis is perpendicular to $x - y$ plane), the Coriolis term should be added to the momentum equation [41]. In that case, Vallis [46] proposes:

$$\begin{cases} \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} + v\dfrac{\partial u}{\partial y} - fv + g\dfrac{\partial \eta}{\partial x} & = & 0, \\ \dfrac{\partial v}{\partial t} + u\dfrac{\partial v}{\partial x} + v\dfrac{\partial v}{\partial y} + fu + g\dfrac{\partial \eta}{\partial y} & = & 0, \\ \dfrac{\partial \eta}{\partial t} + \dfrac{\partial}{\partial x}[(\eta + h_0)u] + \dfrac{\partial}{\partial y}[(\eta + h_0)v] & = & 0, \end{cases} \tag{1.14}$$

where f is the Coriolis parameter.

In terms of the two-dimensional velocity, $\mathbf{v} = (u, v)$, (1.14) can be rewritten as:

$$\begin{cases} \partial_t \mathbf{v} + (\mathbf{v} \cdot \nabla)\mathbf{v} + f\hat{k} \times \mathbf{v} + g\nabla\eta = 0, \\ \partial_t \eta + \nabla \cdot [\mathbf{v}(h_0 + \eta)] = 0. \end{cases} \tag{1.15}$$

1.1.1 Adding viscosity: Hyperbolic-Parabolic problems

Similarly, as we described in the previous section, Gerbeau and Marche [19, 32] derived the viscous SWE. Various models (or approximations) for the viscosity term are possible. In general, Hao [21] expressed the nonlinear viscous SWE as follows:

$$\begin{cases} \mathbf{h}_t + \operatorname{div}(\mathbf{hv}) = 0, \\ (\mathbf{hv})_t + \operatorname{div}(\mathbf{hv} \otimes \mathbf{v}) + g\mathbf{h}\nabla\mathbf{h} + f(\mathbf{hv})^{\perp} = \operatorname{div}\left(2\xi(\mathbf{h})D(\mathbf{v})\right) + \nabla(\lambda(\mathbf{h})\operatorname{div}\, \mathbf{v}), \\ \mathbf{h}(\mathbf{0}) = \mathbf{h_0}, \quad \mathbf{v}(0) = \mathbf{v}_0, \end{cases} \tag{1.16}$$

where $\mathbf{h}(x, y) = h_0(x, y) + \eta(x, y)$, $D(\mathbf{v}) = \dfrac{1}{2}(\nabla\mathbf{v} + (\nabla\mathbf{v})^T)$, $\xi \geq 0$ and λ are the dynamical viscosities satisfying $\lambda + \xi \geq 0$.

As seen in [21, 33], the SWE with viscosity fits into the class of hyperbolic-parabolic systems.

1.2 Settings for our formulation

For our analysis, as in [3, 31], we define the viscous SWE as:

$$\begin{cases} \partial_t \mathbf{v} + (\mathbf{v} \cdot \nabla)\mathbf{v} + f\hat{k} \times \mathbf{v} + g\nabla\eta - A_{visc}\Delta\mathbf{v} = 0 \quad \text{in } \Omega \times [0, T] \\ \partial_t \eta + \nabla \cdot [\mathbf{v}(h_0 + \eta)] = 0 \quad \text{in } \Omega \times [0, T] \end{cases} \tag{1.17}$$

This formulation is obtained after setting the dynamical viscosities as $\xi(\mathbf{h}) = A_{visc} \geq 0$ and $\lambda(\mathbf{h}) = -A_{visc}$ in (1.16), additionally taking $h_0(x, y)$ constant.

Moreover, for our model we also consider a supplementary Laplacian term, $\varepsilon \triangle \eta$ ($\varepsilon \geq 0$), in the transport equation, as described in [3]. The addition of this term changes the nature of our model. For a big enough $\varepsilon > 0$, the system becomes parabolic.

We know that a major problem associated with the finite element solution of the differential equations describing such transport type phenomena is that solutions, in general, fail to capture non-smooth behavior of the unknown variables. In particular, if the field variables change rapidly within the thin internal or boundary layers, different numerical schemes have shown inaccurate and unstable results (see Yazdani [37]). Since our modified model is parabolic, it does not admit discontinuous solutions. Hence, our finite element based solver is applicable under these settings.

The model that we will consider in our analysis is:

$$\begin{cases} \partial_t \mathbf{v} + (\mathbf{v} \cdot \nabla)\mathbf{v} + f\hat{k} \times \mathbf{v} + g\nabla\eta - A_{visc}\Delta\mathbf{v} = 0 \quad \text{in } \Omega \times [0, T], \\ \partial_t \eta + \nabla \cdot [\mathbf{v}(h_0 + \eta)] - \epsilon\Delta\eta = 0 \quad \text{in } \Omega \times [0, T], \\ \eta(0) = \eta_0, \quad \mathbf{v}(0) = \mathbf{v}_0. \end{cases} \tag{1.18}$$

We assume periodic boundary conditions on $\partial\Omega$, where $\Omega = [0, L_x] \times [0, L_y]$ and $[0, T]$ is the observed time interval. In the next section we specify the regularity conditions of the initial data $(\eta_0, \mathbf{v}_0)$.

Further, our functional space V is defined as:

$$V := \left\{ f = (\eta, \mathbf{v}) \in H^1(\Omega)^3 \text{ s.t. } f \text{ is periodic in } \partial\Omega \right\}. \tag{1.19}$$

Considering (1.18) one can get the associated weak formulation of the viscous SWE by multiplying with test functions $\psi = (\chi, \phi)^T \in V$and integrating over the domain Ω:

$$\begin{cases} (\partial_t \mathbf{v}, \phi) + (v \cdot \nabla\mathbf{v}, \phi) + (f\hat{k} \times \mathbf{v}, \phi) + (g\nabla\eta, \phi) - A_{visc}(\triangle\mathbf{v}, \phi) = 0, \ \forall \ \psi \in V, \\ (\partial_t \eta, \chi) + (\nabla \cdot (\mathbf{v}(h_0 + \eta)), \chi) - (\varepsilon\triangle\eta, \chi) = 0. \end{cases} \tag{1.20}$$

For the second order terms, using the boundary conditions, we get:

$$(\triangle\mathbf{v}, \phi) = \int_{\Omega} \phi \cdot \triangle\mathbf{v} \ \mathrm{d}\Omega \overset{\text{Green's Identity}}{=} \int_{\partial\Omega} \phi \frac{\partial\mathbf{v}}{\partial n} \ \mathrm{d}\Gamma - \int_{\Omega} \nabla\mathbf{v}\nabla\phi \ \mathrm{d}\Omega \overset{\text{by periodicity in } \partial\Omega}{=} - \int_{\Omega} \nabla\mathbf{v}\nabla\phi \ \mathrm{d}\Omega$$

Analogously, $(\triangle\eta, \chi) = -\int_{\Omega} \nabla\eta\nabla\chi \ \mathrm{d}\Omega$. Therefore, the weak formulation of the SWEs that we will consider is the following:

$$\begin{cases} (\partial_t \mathbf{v}, \phi) + (\mathbf{v} \cdot \nabla\mathbf{v}, \phi) + (f\hat{k} \times \mathbf{v}, \phi) + (g\nabla\eta, \phi) + A_{visc}(\nabla\mathbf{v}, \nabla\phi) = 0, \ \forall \ \psi \in V \\ (\partial_t \eta, \chi) + (\nabla \cdot (\mathbf{v}(h_0 + \eta)), \chi) + (\varepsilon\nabla\eta, \nabla\chi) = 0 \end{cases} \tag{1.21}$$

1.3 Existence of solution and well-posedness:

There are different papers where the existence and uniqueness of the SWE is approached under different settings and boundary conditions [43]. Bui [10] proved local existence and uniqueness of classical solutions to the Dirichlet problem for the unforced viscous SWE and fix $\xi(\mathbf{h}) = \mathbf{h}$ and $\lambda(\mathbf{h}) = 0$, using Lagrangian coordinates and Hölder space estimates, he assumed initial data $\mathbf{h_0} \in C^{1,\alpha}(\Omega)$ and $\mathbf{v}_0 \in C^{2,\alpha}(\Omega)$.

Other authors considered the solution to be spatially periodic, e.g. Kloeden [26], he proved global existence and uniqueness using Sobolev space estimates by following the energy method of Matsumura and Nishida [34], he considered the initial data $(\mathbf{h}_0, \mathbf{v}_0) \in H^4(\Omega)$. Both authors do not include the Coriolis force, however both state that the inclusion of this force will not alter the results of their respective theorems [43].

Sundbye [43] proves global existence of strong solutions for the unforced initial value problem and for the forced initial boundary value problem. Polynomial L^2 and L^∞ decay rates are established for the Dirichlet problem.

Cheng and Tadmor [12] considered the non-viscous SWE (1.15) and proved the long time existence of approximate periodic solutions for initial data $(\mathbf{h_0}, \mathbf{v}_0) \in H^m(\Omega)^3$ with $m > 5$.

Flori, Orenga and Peybernes in [18] proposed a free boundary shallow water model for which they gave an existence theorem. Such model is fairly similar to (1.17) without the Coriolis force and $g = 1$:

$$\begin{cases} \dfrac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v} - \mu \triangle \mathbf{v} + \nabla \mathbf{h} = 0 \text{ in } \Omega, \\ \dfrac{\partial \mathbf{h}}{\partial t} + \operatorname{div}(\mathbf{h}\mathbf{v}) = 0 \text{ in } \Omega, \end{cases} \tag{1.22}$$

with initial data $(\mathbf{h_0}, \mathbf{v}_0)$ satisfying:

$$\mathbf{h}_0 \log \mathbf{h_0} \in L^1(\Omega), \ \mathbf{h_0} \geq 0, \tag{1.23}$$

$$\mathbf{v}_0 \in H^{5/2}(\Omega), \tag{1.24}$$

and the small data condition:

$$M_0 = \frac{1}{2}\|\mathbf{v}_0\|^2_{L^2(\Omega)} + \int_\Omega \mathbf{h_0} \log \mathbf{h_0} + \frac{1}{e}\text{meas}(\Omega) + \frac{1}{2}\|\mathbf{v}_0\|_{H^2(\partial\Omega)^2}$$
$$< \beta \min\left(\left(\frac{2\mu}{C_{GN}}\right)^2 ; \left(\frac{2\alpha}{TC_{GN}}\right)^2\right), \tag{1.25}$$

$$\|\mathbf{v}_0\|_{L^2(\Omega)} < \min\left(2\frac{\mu}{C_{GN}}, 2\frac{\alpha}{C_{GN}T}\right), \tag{1.26}$$

where α and β are two positive numbers such that $\alpha + \beta = \dfrac{1}{2}$ and C_{GN} is the best constant satisfying Gagliardo-Niremberg inequality:

$$\|\mathbf{v}\|_{L^4(\Omega)} \leq C_{GN}\|\mathbf{v}\|_{L^2(\Omega)}\|\mathbf{v}\|_{H^1(\Omega)}. \tag{1.27}$$

Under the conditions (1.23), (1.24), (1.25), (1.26) and (1.27), Flori, Orenga and Peybernes [18] proved existence of solutions for problem (1.22). Additionally, they showed that $(\mathbf{h}, \mathbf{v})$ verify the following regularity conditions:

$$\mathbf{v} \in L^2(0,T; H^1(\Omega)) \cap L^\infty(0,T; L^2(\Omega)), \tag{1.28}$$

$$\mathbf{h} \log \mathbf{h} \in L^\infty(0,T; L^1(\Omega)),\ \mathbf{h} \geq 0, \tag{1.29}$$

$$\mathbf{h} \in L^2(\Omega). \tag{1.30}$$

Since our goal is to obtain the solution of the weak formulation of the viscous SWE, we pay special attention to the work of Bresh and Desjardins in [6]. They proved the global existence of weak solutions for arbitrarily large data for the viscous shallow water equation with periodic boundary conditions and additional drag friction and capillary terms:

$$\begin{cases} \partial_t \mathbf{h} + \operatorname{div}(\mathbf{h}\mathbf{v}) = 0, \\ \partial_t(\mathbf{h}\mathbf{v}) + \operatorname{div}(\mathbf{h}\mathbf{v} \otimes \mathbf{v}) + \dfrac{(\mathbf{h}\mathbf{v})^\perp}{\mathrm{Ro}} + r_0\mathbf{v} + r_1\mathbf{h}|\mathbf{v}|\mathbf{v} - \kappa\mathbf{h}\nabla\triangle\mathbf{h} + \\ \qquad + \dfrac{\mathbf{h}\nabla\mathbf{h}}{\mathrm{Fr}^2} - \nu\operatorname{div}(\mathbf{h}\nabla\mathbf{v}) = \mathbf{h}f, \end{cases} \tag{1.31}$$

where $\mathrm{Fr} > 0$ denotes the Froude number, $\mathrm{Ro} > 0$ the Rossby number and $\kappa \geq 0$ the capillary coefficient. System (1.31) is supplemented with initial conditions $(\mathbf{h}_0, m_0 = \mathbf{h}_0\mathbf{v}_0)$ satisfying:

$$\begin{array}{lll} \mathbf{h}_0 \in L^2(\Omega), & \dfrac{|m_0|^2}{\mathbf{h}_0} \in L^1(\Omega), & \sqrt{k}\nabla\mathbf{h}_0 \in (L^2(\Omega))^2, \\ \nabla\sqrt{\mathbf{h}_0} \in (L^2(\Omega))^2, & -r_0 \log_- \mathbf{h}_0 \in L^1(\Omega), & \end{array} \tag{1.32}$$

where $m_0 = 0$ on $\mathbf{h}_0^{-1}(\{0\})$ and $\log_- g = \log\min(g, 1)$.

Additionally, Bresh and Desjardins proved that the weak solution $(\mathbf{h}, \mathbf{v})$ satisfies the following regularity properties:

$$\begin{array}{ll} \nabla\sqrt{\mathbf{h}} \in L^\infty(0,T; (L^2(\Omega))^2), & \sqrt{\mathbf{h}}\mathbf{v} \in L^\infty(0,T; (L^2(\Omega))^2), \\ \sqrt{\mathbf{h}}\nabla\mathbf{v} \in L^2(0,T; (L^2(\Omega))^4), & \nabla\mathbf{h} \in L^2(0,T; (L^2(\Omega))^2), \\ \sqrt{r_0}\mathbf{v} \in L^2(0,T; (L^2(\Omega))^2), & r_1^{1/3}\mathbf{h}^{1/3}\mathbf{v} \in L^3(0,T; (L^3(\Omega))^2), \\ \sqrt{\kappa}\nabla^2\mathbf{h} \in L^2(0,T; (L^2(\Omega))^4). & \end{array} \tag{1.33}$$

In our case, we are not considering drag friction nor capillary terms. Therefore, we set $r_0 = r_1 = \kappa = 0$ and the initial conditions (1.32) are set as:

$$h_0 + \eta_0 \in L^2(\Omega), \quad \frac{|(h_0 + \eta_0) \cdot \mathbf{v}_0|^2}{h_0 + \eta_0} \in L^1(\Omega) \quad \text{and} \quad \nabla\sqrt{h_0 + \eta_0} \in (L^2(\Omega))^2. \tag{1.34}$$

From now on we will consider our initial data $(\eta_0, \mathbf{v}_0)$ of (1.18) satisfying (1.34). If we additionally assume $h_0(x, y)$ constant in the bounded domain Ω, then we can neglect $h_0(x, y)$ from the initial conditions (1.34).

Chapter 2

Simulation (Forward Problem)

In this section we analyze the strategy followed to solve the problem:

$$\begin{cases} (\partial_t \mathbf{v}, \phi) + (\mathbf{v}\cdot\nabla\mathbf{v}, \phi) + (f\hat{k}\times\mathbf{v}, \phi) + (g\nabla\eta, \phi) + A_{visc}(\nabla\mathbf{v}, \nabla\phi) = 0, \ \forall\ \psi \in V \\ (\partial_t\eta, \chi) + (\nabla\cdot(\mathbf{v}(h_0+\eta)), \chi) + (\varepsilon\nabla\eta, \nabla\chi) = 0, \end{cases} \tag{2.1}$$

given initial data $(\eta_0, \mathbf{v}_0)$ satisfying (1.34) and periodic boundary conditions on $\partial\Omega$.

Problem (2.1) can be rewritten in the form:

$$\begin{cases} (\partial_t\mathbf{u}, \psi) = a(\mathbf{u}, \psi), \forall\psi = (\chi, \phi)^t \in V \text{ on } [0,T]\times\Omega, \\ \text{where } \mathbf{u} = (\eta, \mathbf{v})^t \in [0,T]\times\Omega, \\ \text{and the initial condition } \mathbf{u}_0 = (\eta_0, \mathbf{v}_0)^t \in \Omega, \end{cases} \tag{2.2}$$

where $a(\mathbf{u}, \psi)$ is given by:

$$\begin{aligned} a(\mathbf{u}, \psi) \ &= \ -[(\mathbf{v}\cdot\nabla\mathbf{v}, \phi) + (f\hat{k}\times\mathbf{v}, \phi) + (g\nabla\eta, \phi) + A_{visc}(\nabla\mathbf{v}, \nabla\phi) + \\ &\quad +(\nabla\cdot(\mathbf{v}(h_0+\eta)), \chi) + (\varepsilon\nabla\eta, \nabla\chi)]. \end{aligned} \tag{2.3}$$

Since we are dealing with a nonstationary problem, we need to define a time stepping scheme for our numerical approximation. In this project, we consider three different time stepping schemes: the Backward Euler scheme (BE), Crank Nicolson Scheme (CN) and Fractional-step θ scheme (FS). These methods are based on finite differences and have different properties that can be useful for our simulation.

2.1 Time discretization: Short overview on the time stepping schemes

Suppose we have a partial differential equation of the form:

$$\begin{cases} (\partial_t\mathbf{u}, \psi) = a(\mathbf{u}, \psi), \text{ on } [0,T]\times\Omega \\ \mathbf{u}_0 = (\eta_0, \mathbf{v}_0)^t \in \Omega, \end{cases} \tag{2.4}$$

The time scheme produces a sequence $\{\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \ldots\}$ such that $\mathbf{u}_k \approx \mathbf{u}(t^k)$ for $0 = t^0 < t^1 < \cdots < t^k < \cdots < t^n = T$ and t_k is given by $t^k = k\cdot\mathrm{dt}$, $\forall k = \{0, \ldots, n\}$ (dt being the time step size).

2.1.1 Backward Euler Scheme (BE)

The Backward Euler Scheme is an implicit method which computes the approximations using:

$$\left(\frac{\mathbf{u}_{k+1}-\mathbf{u}_k}{\mathrm{dt}},\psi\right)=a(\mathbf{u}_{k+1},\psi),\ \forall k=\{0,\dots,n-1\}. \tag{2.5}$$

The Backward Euler method has order one, i.e. the error at a specific time t is $O(h)$. The region of absolute stability for the BE method includes the whole left half of the complex plane, therefore it is strongly A-stable, making it suitable for the solution of stiff equations. Moreover, the BE scheme is L-stable. However, this method tends to be very dissipative [11].

2.1.2 Crank Nicolson Scheme (CN)

The Crank Nicolson scheme is a second-order convergence in time method which is motivated on the trapezoidal rule. This method is a combination of the Forward Euler method at k and the Backward Euler method at $k+1$:

$$\left(\frac{\mathbf{u}_{k+1}-\mathbf{u}_k}{\mathrm{dt}},\psi\right)=\frac{1}{2}\left[a(\mathbf{u}_{k+1},\psi)+a(\mathbf{u}_k,\psi)\right],\ \forall k=\{0,\dots,n-1\}. \tag{2.6}$$

This method is $A-$stable and, in comparison to the BE scheme, has very little dissipation. However it may have instabilities caused by rough initial and/ or boundary data and spurious oscillations when large time steps or small spatial steps are needed; in those cases the less accurate BE scheme is preferred, which is both stable and immune to oscillations [15, 45].

2.1.3 Fractional-Step θ Scheme (FS)

This is one of the most attractive time discretization schemes for incompressible flows, as it combines several attributes like strong $A-$ stability, very little numerical dissipation and second order accuracy [35].

One iteration of the one step theta scheme in $t^k \to t^{k+1}$ is given by:

$$\left(\frac{\mathbf{u}_{k+1}-\mathbf{u}_k}{\mathrm{dt}},\psi\right)=\theta a(\mathbf{u}_{k+1},\psi)+(1-\theta)a(\mathbf{u}_k,\psi),\ \forall k=\{0,\dots,n-1\}. \tag{2.7}$$

The FS scheme is based on this one step theta scheme by combining three substeps

$$t^k \to t^{k+\alpha} \to t^{k+1-\alpha} \to t^{k+1},$$

using time step sizes $\alpha\cdot\mathrm{dt},(1-2\alpha)\mathrm{dt}$ and $\alpha\cdot\mathrm{dt}$. It can be shown that the resulting scheme is second order accurate if we choose $\alpha=1-\sqrt{\frac{1}{2}}\approx 0.293$. Each of the three substeps is a theta step using the three values θ, $1-\theta$ and θ. For every $\theta\in(1/2,1]$ strong $A-$stability of the combined scheme is given, therefore this scheme is well-suited for computing solutions with rough data [9].

The FS scheme between t^k and t^{k+1} can be written as:

Step 1: Compute an approximation to $\mathbf{u}_{k+\alpha}$

$$\left(\frac{\mathbf{u}_{k+\alpha}-\mathbf{u}_k}{\alpha\cdot\mathrm{dt}},\psi\right)=\theta a(\mathbf{u}_{k+\alpha},\psi)+(1-\theta)a(\mathbf{u}_k,\psi). \tag{2.8}$$

Step 2: Compute an approximation to $\mathbf{u}_{k+1-\alpha}$

$$\left(\frac{\mathbf{u}_{k+1-\alpha}-\mathbf{u}_{k+\alpha}}{(1-2\alpha)\cdot\mathrm{dt}},\psi\right)=(1-\theta)a(\mathbf{u}_{k+1-\alpha},\psi)+\theta(\mathbf{u}_{k+\alpha},\psi). \tag{2.9}$$

Step 3: Compute an approximation to $\mathbf{u}_{k+1}$

$$\left(\frac{\mathbf{u}_{k+1}-\mathbf{u}_{k+1-\alpha}}{\alpha\cdot\mathrm{dt}},\psi\right)=\theta a(\mathbf{u}_{k+1},\psi)+(1-\theta)a(\mathbf{u}_{k+1-\alpha},\psi). \tag{2.10}$$

2.2 Space discretization: Short overview Q1 finite element

After discretization in time, the space is treated, as usually, with a Galerkin finite element scheme, here based on the Q1 finite element.

Definition 2.2.1 *Following Ciarlet's definition of a finite element [13],* $(\mathcal{K},\mathcal{P},\mathcal{N})$ *is a finite element if:*

(i) $\mathcal{K}\subseteq\mathbb{R}^n$ *be a bounded closed set with nonempty interior and piecewise smooth boundary (the element domain).*

(ii) $\mathcal{P}$ *be a finite-dimensional space of functions on* $\mathcal{K}$ *(the space of shape functions).*

(iii) $\mathcal{N}=\{\mathcal{N}_1,\mathcal{N}_2,\ldots,\mathcal{N}_k\}$ *be a basis for* $\mathcal{P}'$ *(the set of nodal variables).*

Definition 2.2.2 *Let* $(\mathcal{K},\mathcal{P},\mathcal{N})$ *be a finite element. The basis* $\{\phi_1,\phi_2,\ldots,\phi_k\}$ *of* $\mathcal{P}$ *dual to* $\mathcal{N}$ *(i.e.* $\mathcal{N}_i(\phi_j)=\delta_{ij}$*) is called the nodal basis of* $\mathcal{P}$.

2.2.1 Q1 Finite Element

This finite element is defined as $\mathcal{K}$ being any rectangle, $\mathcal{P}=\mathcal{Q}_1=\mathrm{span}\{1,x,y,xy\}$ and $\mathcal{N}=\{\mathcal{N}_1,\mathcal{N}_2,\mathcal{N}_3,\mathcal{N}_4\}$, where $\mathcal{N}_i(v)=v(z_i)$ and z_1,z_2,z_3,z_4 are the corners of $\mathcal{K}$ (see Figure 2.1):

Figure 2.1: Q1 finite element

Following the definition (2.2.2), given any rectangle $\mathcal{K}$ with corners: $z_1 = (x_1, y_1), z_2 = (x_2, y_1), z_3 = (x_2, y_2), z_4 = (x_1, y_2)$, the nodal basis can be defined by:

$$\begin{aligned} \phi_1(x,y) &= \frac{(x-x_2)(y-y_2)}{(x_1-x_2)(y_1-y_2)}, \quad \phi_2(x,y) = \frac{(x-x_1)(y-y_2)}{(x_2-x_1)(y_1-y_2)}, \\ \phi_3(x,y) &= \frac{(x-x_1)(y-y_1)}{(x_2-x_1)(y_2-y_1)}, \quad \phi_4(x,y) = \frac{(x-x_2)(y-y_1)}{(x_1-x_2)(y_2-y_1)}. \end{aligned}$$

Definition 2.2.3 *A subdivision of a domain* Ω *is a finite collection of element domains* $\{\mathcal{K}_i\}_i$ *such that:*

1) $\operatorname{int} \mathcal{K}_i \cap \operatorname{int} \mathcal{K}_j = \emptyset$ *if* $i \neq j$ *and*

2) $\bigcup\limits_i \mathcal{K}_i = \bar{\Omega}$.

In our case Ω is a rectangular domain and the subdivision $\{\mathcal{K}_i\}_i$ is such that in each domain $\mathcal{K}_i$ we define a Q1 finite element, i.e. in each $\mathcal{K}_i$, we define the space discretization $V_h := (\mathcal{Q}_1)^3$. Hence, the basis of V_h, $(\{\psi_i\}_{i=1}^{N=12})$ in each subdivision is of the form:

$$\left\{ \begin{pmatrix} \phi_i \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \phi_i \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \phi_i \end{pmatrix} \right\}_{i=1}^{4}$$

The subdivision of the rectangular domain Ω can be easily implemented by using the Deal.II command:

```
domain.refine_global(level).
```

This command refines all cells in the domain *level* times, therefore the generated mesh will have $2^{2 \cdot level}$ cells (see Figure 2.2).

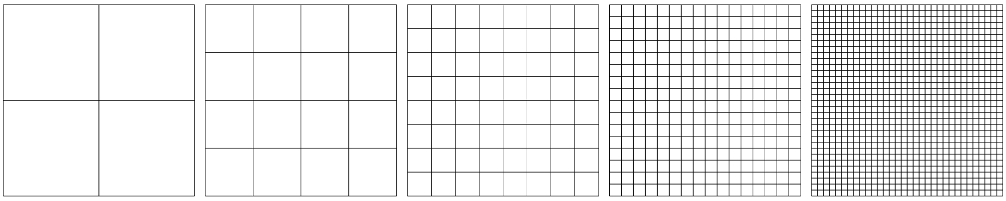

Figure 2.2: Mesh refinement for $level = 1, 2, 3, 4, 5$

2.3 Numerical Approximation of the SWE

Considering the time and space discretization described in Section 2.1 and Section 2.2, problem (2.2) can be seen as:

$$(\mathcal{P}) \begin{cases} \text{Given an initial data } \mathbf{u}_0 \text{ satisfying (1.34) :} \\ \text{Determine } \mathbf{u}_{k+1}\ \forall k = \{0, \dots, n-1\},\ \text{ satisfying} \\ (\mathbf{u}_{k+1} - \mathbf{u}_k, \psi) = \theta \cdot \mathrm{dt} \cdot a(\mathbf{u}_{k+1}, \psi) + (1-\theta) \cdot \mathrm{dt} \cdot a(\mathbf{u}_k, \psi), \forall \psi \in V_h \text{ and } \theta \in [0,1]. \end{cases}$$

For $\theta = 1$ gives the Backward Euler scheme, $\theta = 0$ Backward Forward scheme, $\theta = 1/2$ Crank Nicolson scheme and for any $\theta \in [0,1]$ a one step theta scheme (one of the three substeps of the Fractional step θ scheme).

Hence, problem $(\mathcal{P})$ is equivalent to finding $\mathbf{u} \in V_h$ such that

$$F(\mathbf{u}) = a^{\mathrm{dt}}(\mathbf{u}, \psi) - f(\psi) = 0,\ \forall \psi \in V_h,$$

with $\mathbf{u} = (\eta, \mathbf{v})^t = (\eta_{k+1}, \mathbf{v}_{k+1})^t = \mathbf{u}_{k+1}$ and $a^{\mathrm{dt}}(\mathbf{u}, \psi),\ f(\psi)$ given by:

$$\begin{aligned} a^{\mathrm{dt}}(\mathbf{u}, \psi) &= (\mathbf{u}, \psi) - \theta \cdot \mathrm{dt} \cdot a(\mathbf{u}, \psi) \\ f(\psi) &= (\mathbf{u}_k, \psi) + (1-\theta) \cdot \mathrm{dt} \cdot a(\mathbf{u}_k, \psi) \end{aligned} \tag{2.11}$$

To do so, we use the Newton method with line search, which is already implemented in DopeLib [20]:

```
Algorithm 2.3.1   Newton Method with line search
Given $\mathbf{u}_0 \in V_h$. Fix $k = 0$, $\rho \in (0,1)$, $\sigma \in (0,1/2)$.
While ($\|F(\mathbf{u}_k)\|_{L^2(\Omega)}$ is not small enough) do

    Step 1: Determine a direction $\mathbf{u}^+ \in V_h$ satisfying $F'_{\mathbf{u}}(\mathbf{u}_k)\mathbf{u}^+ = -F(\mathbf{u}_k)$, i.e.
            $a^{\mathrm{dt}\prime}_{\mathbf{u}}(\mathbf{u}_k, \psi)(\mathbf{u}^+) = -a^{\mathrm{dt}}(\mathbf{u}_k, \psi) + f(\psi), \forall \psi \in V_h$
            Since $\mathbf{u}^+ \in V_h$, there exist coefficients $\{c_j\}_{j=1}^N$ s.t. $\mathbf{u}^+ = \sum_{j=1}^N c_j \psi_j$.
            We determine $c = \{c_j\}_{j=1}^N$ solving the system $A^k c = b^k$,
            $A^k_{ij} = a^{\mathrm{dt}\prime}_{\mathbf{u}}(\mathbf{u}_k, \psi_i)(\psi_j), \quad b^k = -a^{\mathrm{dt}}(\mathbf{u}_k, \psi_i) + f(\psi_i)$.
```

```
Step 2:   Determine Step-Length:
          Set $l = 0$;   $\mathbf{u}_{k+1} = \mathbf{u}_k + \rho^l \mathbf{u}^+$;
          While $\left(\|F(\mathbf{u}_{k+1})\|_{L^2(\Omega)}$ is not small enough$\right)$ do
              $l = l + 1$;   $\mathbf{u}_{k+1} = \mathbf{u}_k - \rho^l \mathbf{u}^+$
          End While

$k = k + 1$

End While
```

2.4 Demonstrative Examples

Next we introduce four examples that will let us understand the properties of our solver. These results were obtained using a computer with an Intel Core i3-2310M 2.10 GHz Processor and 4.00 GB of RAM memory:

2.4.1 Lake at rest (Initial Test)

This is the most simple test that is implemented to verify a SWE solver [16]. It consists on setting $\eta_0(x, y) \equiv 0$ and $\mathbf{v}_0(x, y) \equiv (0, 0)^t$ in Ω. As in [3], we fix the parameters $\varepsilon = A_{visc} = 0.1$, $T = 1$, the coriolis force $f(x, y) = 4 \cdot 86164.1 \cdot \pi \cdot \sin(2\pi y)$ and the domain $\Omega = [0, 1]^2$.

In the following figure we can see the simulation of $\eta(t, x, y)$ for the BE scheme, CN scheme and FS scheme for different time steps and a spatial grid with 256 elements. Besides the time scheme, we obtained the following stationary solution:

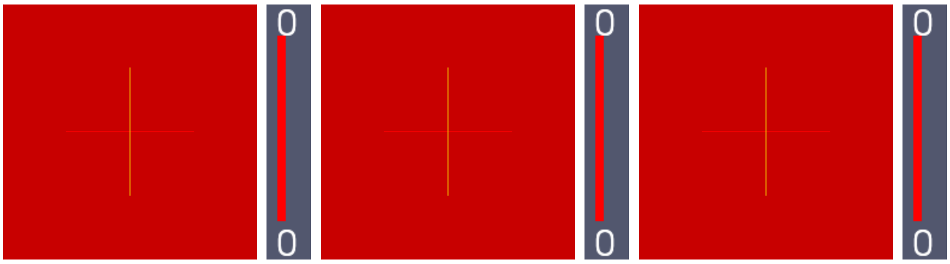

Figure 2.3: Stationary solution $\eta(t, x, y)$, $\mathbf{v}(t, x, y)$ for $t = [0, 1]$

This is, $\eta(t, x, y) \equiv 0$, $\mathbf{v}(t, x, y) \equiv (0, 0)^t$, which is the expected result [16].

2.4.2 Double Bump (Time Schemes)

In this example we consider a height fluctuation η_0 with two bumps (one positive and one negative). To do so, we take $\eta_0(x, y) = \sin(2\pi x) + \sin(2\pi y)$ and, additionally, $\mathbf{v}_0(x, y) \equiv (0, 0)^t$ in Ω. We fix the parameters $\varepsilon = 0$, $A_{visc} = 0.1$, $T = 1$, the Coriolis force $f(x, y) = 4 \cdot \pi \cdot \sin(2\pi y)$, $h_0(x, y) = 1000$, $g = 9.81$ and the domain $\Omega = [0, 1]^2$.

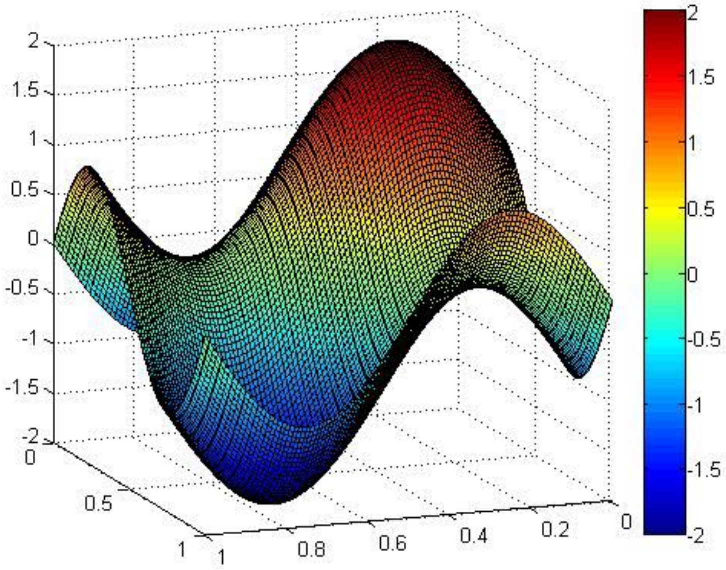

Figure 2.4: Initial data $\eta_0(x, y) = \sin(2\pi x) + \sin(2\pi y)$ in $\Omega = [0, 1]^2$

In the following plots we can see the simulation of $\eta(t, x, y)$ for the BE scheme, CN scheme and FS scheme for different number of time steps $= \{50, 500, 5000\}$ and a spatial grid with 256 elements.

- **Backward Euler**
    - Time steps = 50. CPU time = 3.64 sec.

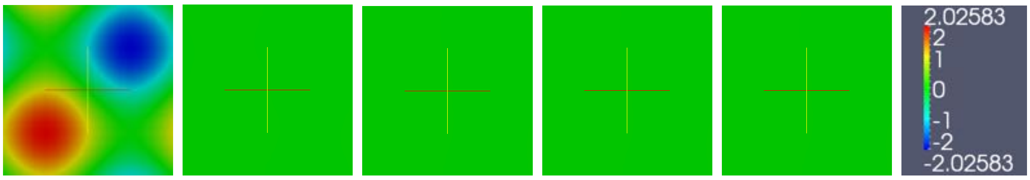

Figure 2.5: 50 time steps BE Simulation for $t = \{0, 0.10, 0.20, 0.50, 1\}$

- Time steps = 500. CPU time = 34.15 sec.

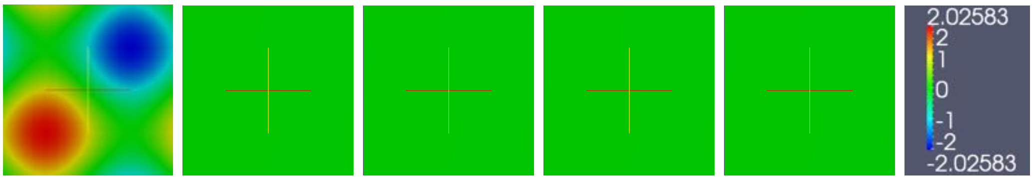

Figure 2.6: 500 time steps BE Simulation for $t = \{0, 0.10, 0.20, 0.50, 1\}$

- Time steps = 5000. CPU time = 216.10 sec.

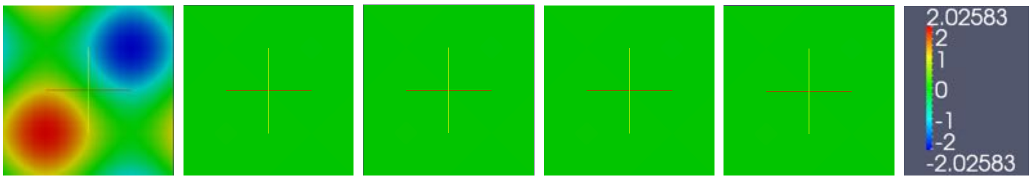

Figure 2.7: 5000 time steps BE Simulation for $t = \{0, 0.10, 0.20, 0.50, 1\}$

- **Crank Nicolson**

- Time steps = 50. CPU time = 4.70 sec.

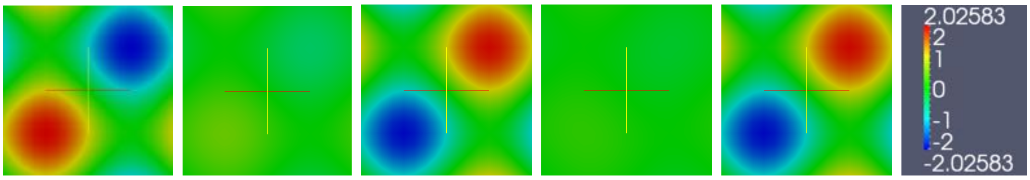

Figure 2.8: 50 time steps CN Simulation for $t = \{0, 0.10, 0.20, 0.50, 1\}$

- Time steps = 500. CPU time = 44.70 sec.

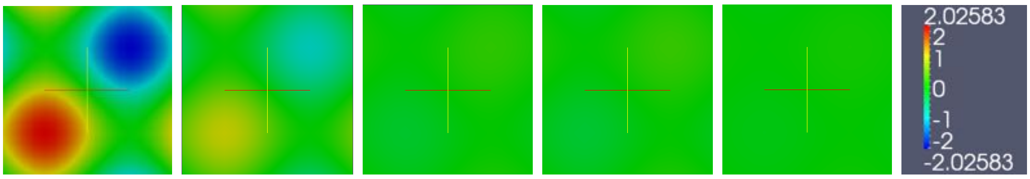

Figure 2.9: 500 time steps CN Simulation for $t = \{0, 0.10, 0.20, 0.50, 1\}$

- Time steps = 5000. CPU time = 396.71 sec.

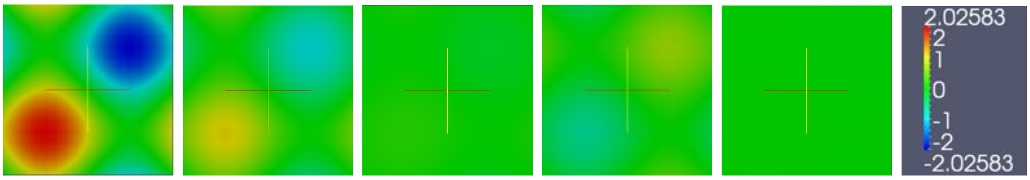

Figure 2.10: 5000 time steps CN Simulation for $t = \{0, 0.10, 0.20, 0.50, 1\}$

- **Fractional-Step** θ

○ Time steps = 50. CPU time = 15.40 sec.

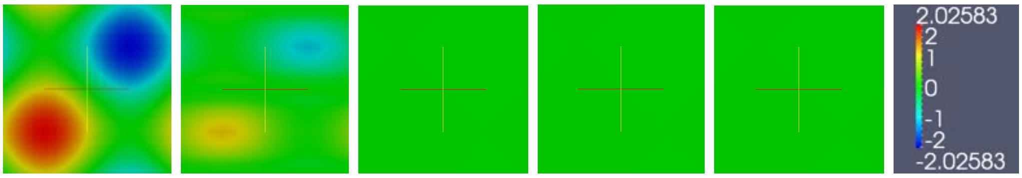

Figure 2.11: 50 time steps FS Simulation for $t = \{0, 0.10, 0.20, 0.50, 1\}$

○ Time steps = 500. CPU time = 163.63 sec.

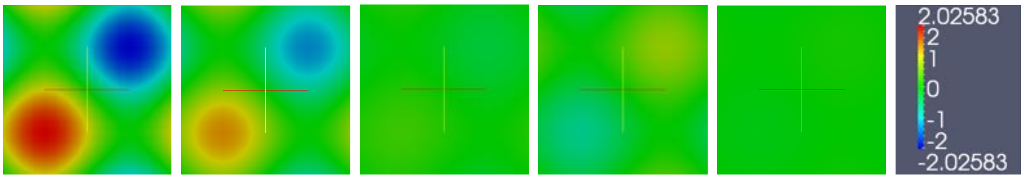

Figure 2.12: 500 time steps FS Simulation for $t = \{0, 0.10, 0.20, 0.50, 1\}$

○ Time steps = 5000. CPU time = 1159.16 sec.

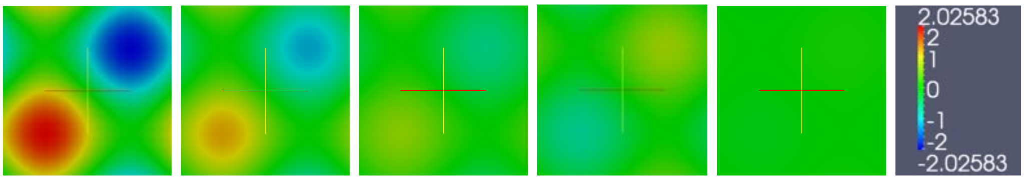

Figure 2.13: 5000 time steps FS Simulation for $t = \{0, 0.10, 0.20, 0.50, 1\}$

In the previous figures we observe that, for the different schemes and number of time steps, both bumps do not vanish "strictly" in time. As we will see in Figure 2.14, η(middle point of Ω) oscillates till converging to an equilibrium state as $t \rightarrow T$. In the next examples we will also verify the smoothness effect that the laplacian term ($\varepsilon \triangle \eta$, $\varepsilon > 0$) adds to such descent.

For the smallest number of time steps (50), in the CN scheme, the final height profile is not vanished at time T. However, as we increase the number of time steps, the CN scheme has a similar behavior as the FS scheme (more details will be provided later). In that sense, the CN scheme did not offer reliable results for small time steps (the description of the evolution of the model changes greatly as we increase the number of time steps). As we commented in the previous chapter, it is well known that this scheme can have instabilities for small number of time steps depending on the initial data [11, 45].

The BE scheme showed a very dissipative behavior which improved as we increased the number of time steps (see Figure 2.14). In general, the FS method is better suited for these computations. Given the oscillating behavior of this case, the number of time steps play an important role when determining the dynamics of the model and we observe that, even for small number of time steps (50), the FS scheme manages to anticipate such an oscillating behavior with convergence to an equilibrium state (which can not be achieved for big time steps in the case of BE and CN schemes, see figures 2.5, 2.8 and 2.14). Similar results were obtained by other authors in the simulation of incompressible flows [9, 35]. The behavior of the different time schemes for the height profile in this example is repeated for other values of $\eta_0(x, y)$.

For the reasons given above, we expect that $\eta(\text{middle point of } \Omega) \xrightarrow{t\to\infty} \int\limits_\Omega \eta_0(x, y)\mathrm{d}\Omega/|\Omega|$, in other words, that the bump(s) vanishes in time, in this example $\int\limits_\Omega \eta_0(x, y)\mathrm{d}\Omega = 0$. In the next figures we show the relation between *time* vs η (middle point of Ω) for the different time schemes:

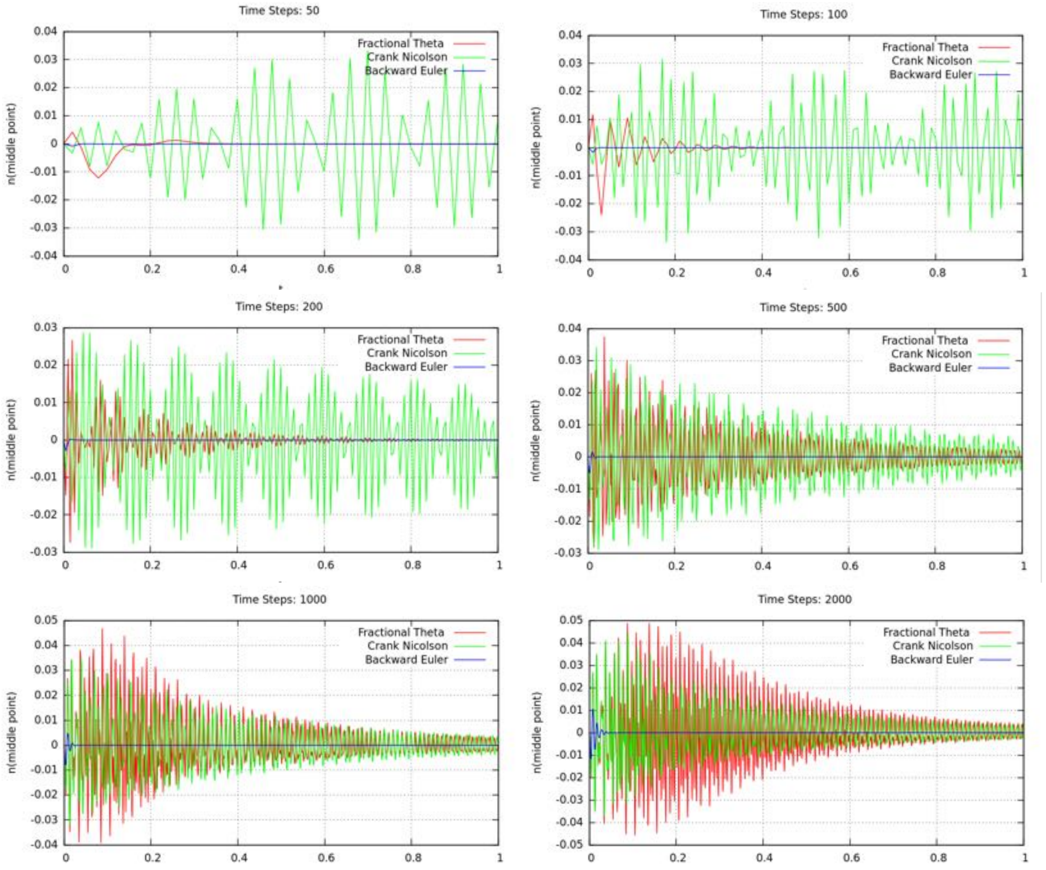

Figure 2.14: Middle point evaluation w.r.t. time at different time steps

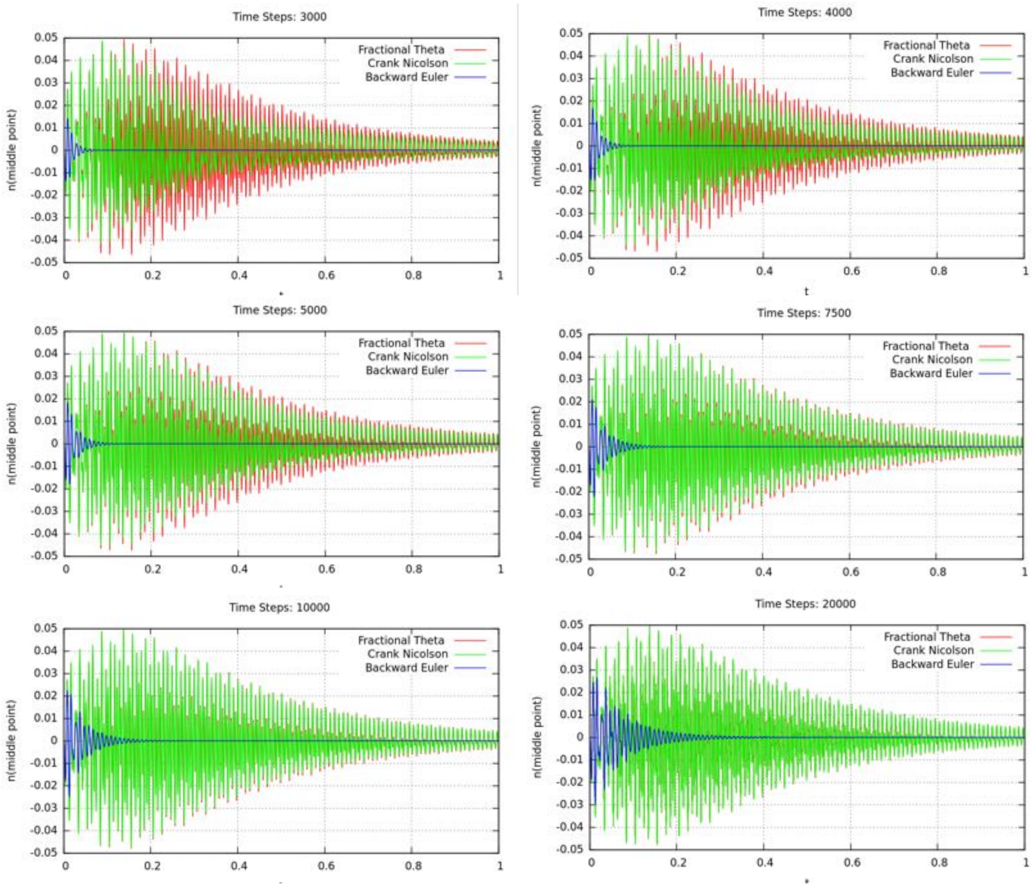

Figure 2.15: Middle point evaluation w.r.t. time at different time steps

In Figure 2.14 and Figure 2.15 is simpler to see that, for smaller number of time steps, the CN scheme may have instabilities and that the BE scheme tends to exhibit a dissipative behavior for the different time steps that we analyzed. One can also see that, as the time step decreases, the FS and CN schemes seems converge to the same solution, as we can observe in Figure 2.16.

One of the possible drawbacks of the FS scheme is associated to the computation time. As it is expected, when performing the simulation for a fixed number of time steps, the FS method has a higher CPU time. This is due to the fact that it requires three implicit sub-steps per time step. In the next figure we compare the solution achieved with 20000 time steps, $\eta^{20000,FS}_{\text{middle point }\Omega}$, with respect to the approximations obtained for the BE, CN, FS schemes with different time steps. Such comparison is made by calculating the relative errors:

$$\frac{\sum\limits_{i=0}^{20000}\left(\eta^{20000,FS}_{\text{middle point }\Omega}(t_i)-\eta^{ts,method}_{\text{middle point }\Omega}(t_i)\right)^2}{\sum\limits_{i=0}^{20000}\left(\eta^{20000,FS}_{\text{middle point }\Omega}(t_i)\right)^2},$$

where $ts=\{50,100,200,500,1000,2000,3000,4000,5000,7500,9000,20000\}$ and *method* is any of the three time schemes that we have been considering. The additional points needed to calculate the error for number of time steps smaller than 20000 are obtained via linear interpolation.

Time steps	BE	CN	FS
50	3.64	4.70	15.40
100	8.28	9.36	29.91
200	50.35	17.90	63.64
500	34.15	44.07	163.63
1000	62.56	88.25	354.96
2000	101.40	181.52	660.11
3000	162.14	274.40	867.73
4000	200.31	363.29	1041.91
5000	216.10	396.71	1159.16
7500	325.06	584.14	1622.51
9000	471.96	671.07	1787.18
10000	551.13	739.02	1987.40
20000	1029.03	1440.98	3848.23

Table 2.1: CPU time (sec) for different time steps

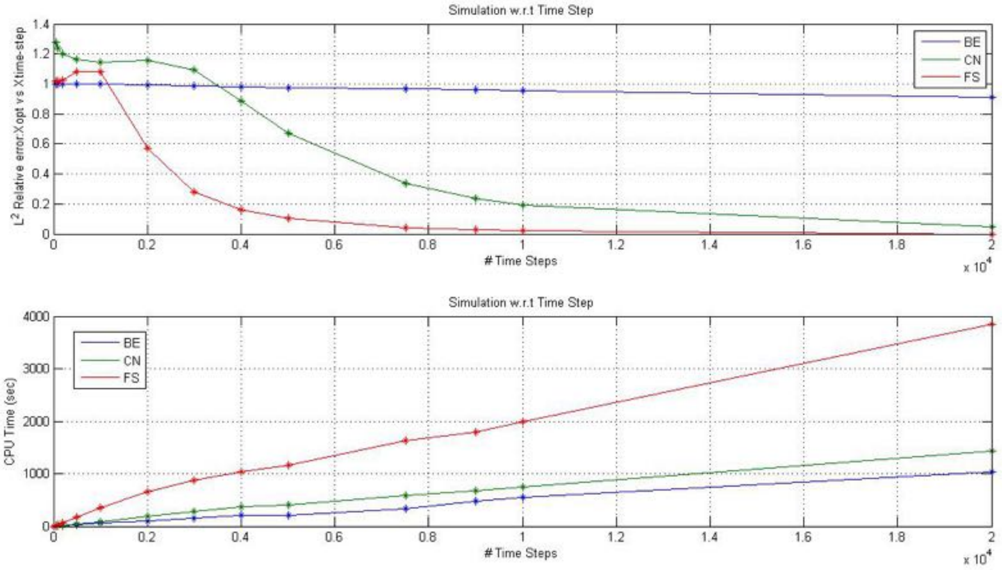

Figure 2.16: Comparison CPU time vs accuracy

As we previously commented, in this figure we observe the convergence of the FS and CN scheme to the same solution, while the BE scheme does not suffer a relevant variation as we decrease the time step. This pointwise comparison is specially relevant for a large number of time steps since, in this particular case, the dynamic of the model is oscillatory. Therefore, it is normal to expect that for small time steps we observe a larger error in the FS scheme, even when it predicts correctly the behavior of η(middle point of Ω) (see Figure 2.14).

Additionally, we observe in Table 2.1 that the FS scheme has a higher CPU time for a fixed number of time steps. However, we notice that in order to achieve “realistic” results the BE and CN schemes need a much greater time step and, therefore, a possible higher CPU time than the one required for the FS scheme simulation with a bigger time step. For example, as we can see in Figure 2.16 the FS scheme with 2000 time steps (CPU time = 660.11 sec) is closer to the optimal solution than the BE scheme with 20000 time steps (CPU time = 1029.03 sec). In that sense, the comparison of the CPU time at a fix number of time steps is not necessarily a relevant indicator.

For the initial settings of our model, we can not observe an evident impact of the Coriolis force over the dynamics of the model. In order to see this, we variate the Coriolis force $f(x, y)$ for a fixed 500 time steps FS scheme. We obtained the following results:

- $f(x, y) \equiv 0$

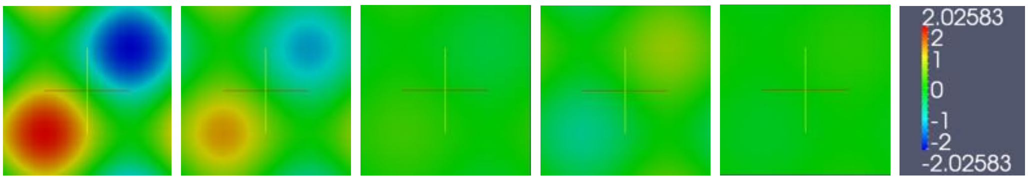

Figure 2.17: Simulation with $f(x, y) \equiv 0$ and $t = \{0, 0.10, 0.20, 0.50, 1\}$

- $f(x, y) = 4 \cdot \pi \sin(2\pi y)$

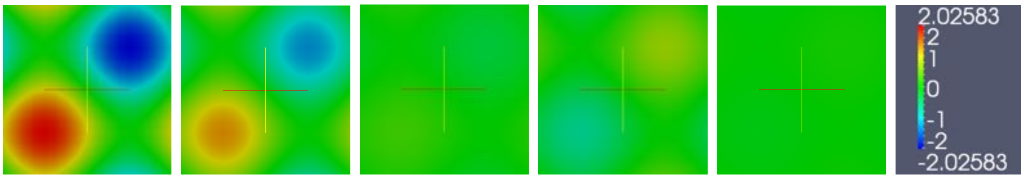

Figure 2.18: Simulation with $f(x, y) = 4 \cdot \pi \sin(2\pi y)$ and $t = \{0, 0.10, 0.20, 0.50, 1\}$

- $f(x, y) = 4 \cdot 861.641 \cdot \pi \sin(2\pi y)$

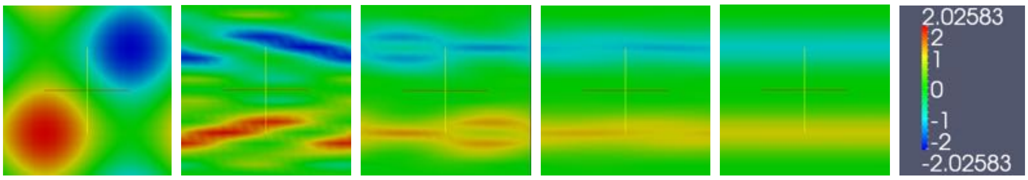

Figure 2.19: Simulation with $f(x, y) = 4 \cdot 861.641 \cdot \pi \sin(2\pi y)$ and $t = \{0, 0.10, 0.20, 0.50, 1\}$

- $f(x,y) = 4 \cdot 8616.41 \cdot \pi \sin(2\pi y)$

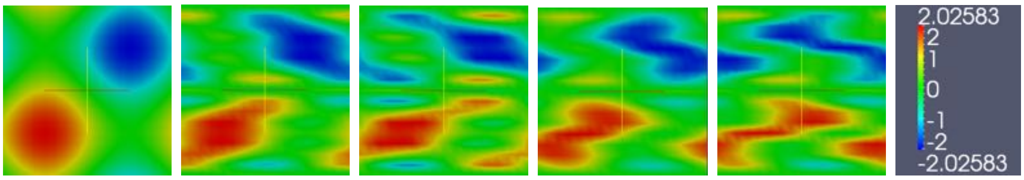

Figure 2.20: Simulation with $f(x,y) = 4{\cdot}8616.41{\cdot}\pi \sin(2\pi y)$ and $t = \{0, 0.10, 0.20, 0.50, 1\}$

- $f(x,y) = 4 \cdot 86164.1 \cdot \pi \sin(2\pi y)$

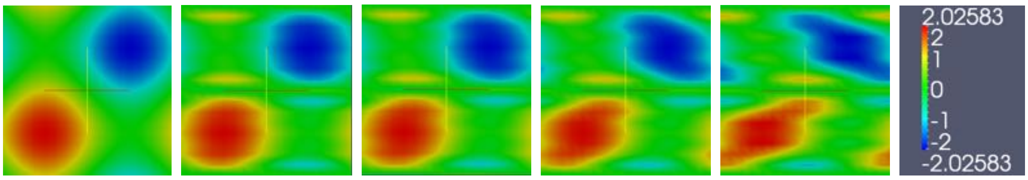

Figure 2.21: Simulation with $f(x,y) = 4{\cdot}86164.1{\cdot}\pi \sin(2\pi y)$ and $t = \{0, 0.10, 0.20, 0.50, 1\}$

Further, in Figure 2.22, we can see the evolution of $\|\eta(t)\|^2_{L^2(\Omega)}$ for all the Coriolis forces used before.

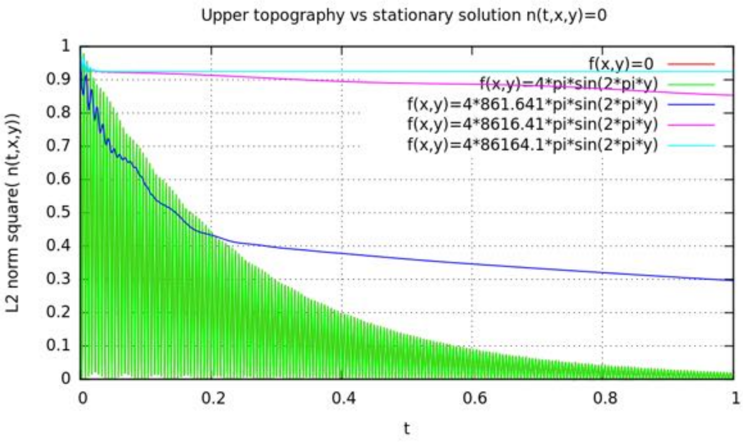

Figure 2.22: $\|\eta(t)\|^2_{L^2(\Omega)}$, $t \in [0,1]$ for all Coriolis forces used before

We see that, if we neglect or consider a small enough Coriolis force ($f(x,y) = 0$, $f(x,y) = 4 \cdot \pi \sin(2\pi y)$), the bumps tend too the equilibrium state as we increase the time, even when it does not vanish strictly in time. As we increase the Coriolis force, the bump takes longer to vanish or simply does not vanish at all. This is due to a higher presence of the inertial force, which, as we can see in Figure 2.19, 2.20 and 2.21, has a bigger effect on the dynamics of the wave rather than other forces (i.e. gravity). Such effect is represented in an apparent deflection in the dynamics of the model. However, as we will analyze in the upcoming chapter, if we add a large enough viscous term to the transport equation we can achieve a smooth solution with a decay effect due to the viscous laplacian term. In the next figure we take again $f(x,y) = 4 \cdot 86164.1 \cdot \pi \sin(2\pi y)$ and we add the viscous term with $\varepsilon = 0.1$:

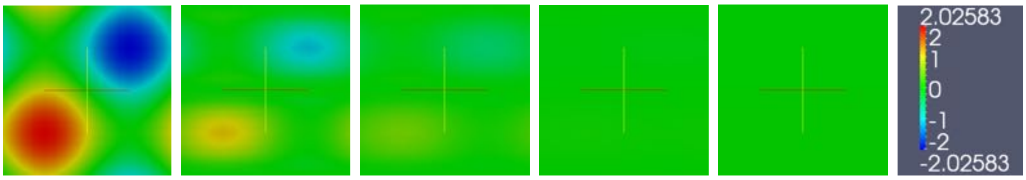

Figure 2.23: Simulation with $f(x,y) = 4 \cdot 86164.1 \cdot \pi \sin(y)$, $\varepsilon = 0.1$ and $t = \{0, 0.10, 0.20, 0.50, 1\}$

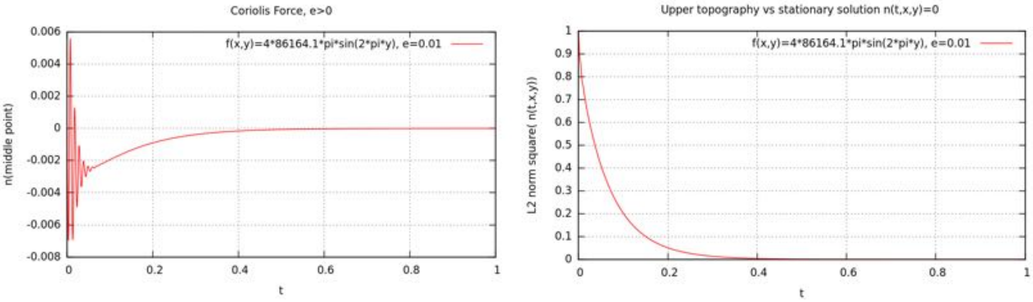

Figure 2.24: Midpoint evaluation and $\|\eta(t)\|^2_{L^2(\Omega)}$, $t \in [0,1]$ for $f(x,y) = 4 \cdot 86164.1 \cdot \pi \sin(y)$, $\varepsilon = 0.1$

As we will refer in the next example, the addition of the viscous term in the transport equation ($\varepsilon > 0$) generates a solution with a smooth decay to the stationary solution ($\eta(t,x,y) = 0$) as we observe in Figure 2.24 with respect to the non-smooth decay obtained for $f(x,y) = 0$ and $\varepsilon = 0$ (Figure 2.22).

2.4.3 Bump (Relation ε vs A_{visc})

The consideration of a supplementary viscosity term in a transport type equation is a common practice as wee can see in [5, 14]. Córdoba analyzed a $1D$ transport type equation and verified the formation of singularities in finite time for a generic family of initial data [14]. By adding the viscosity term described in Section 1.2, the finite time singularity is prevented and the solutions exist globally in time [8, 14].

In this example, we take one single bump given by $\eta_0(x,y) = \sin(\pi x)^2 \cdot \sin(\pi y)^2$. We fix $T = 2$. The Coriolis force $f(x,y) = 4 \cdot 86164.1 \cdot \pi \cdot \sin(2\pi y)$ and the domain $\Omega = [0,1]^2$. In this example we analyze the solution for different values of ε and A_{visc}, with the FS scheme (for the reasons given before), 200 time steps and a spatial grid with 256 elements.

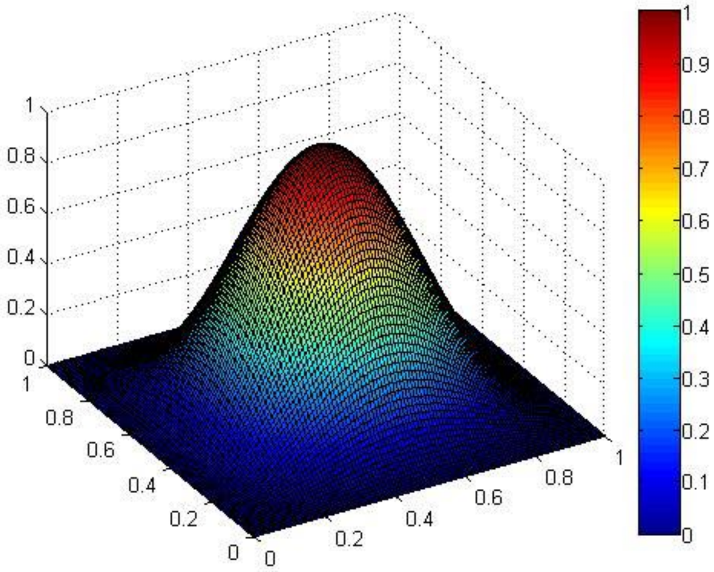

Figure 2.25: Initial data $\eta_0(x, y) = \sin(\pi x)^2 \cdot \sin(\pi y)^2$ in $\Omega = [0, 1]^2$

Next, we see the simulation for three cases: $\varepsilon = A_{visc} = 0$, $\varepsilon = 0$ and $A_{visc} = 0.1$ and $\varepsilon = A_{visc} = 0.1$.

- $\varepsilon = 0$, $A_{visc} = 0$

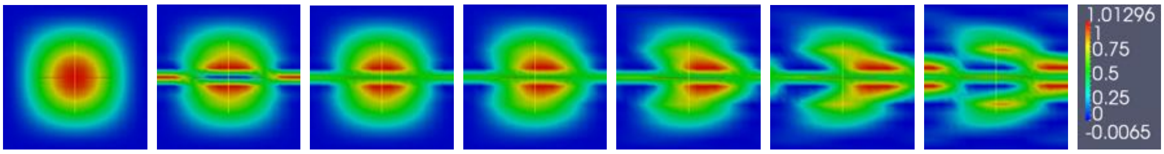

Figure 2.26: $\varepsilon = 0$, $A_{visc} = 0$. Simulation for $t = \{0, 0.01, 0.02, 0.05, 0.2, 0.5, 1, 2\}$

In the next case, we take $A_{visc} \neq 0$.

- $\varepsilon = 0$, $A_{visc} = 0.1$

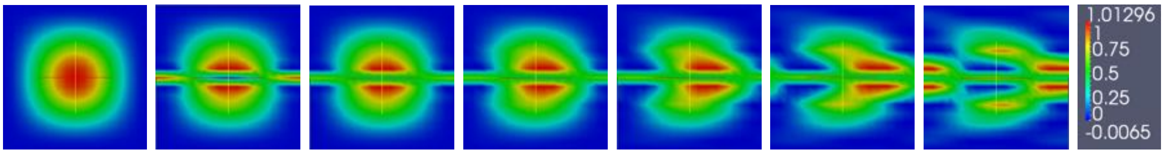

Figure 2.27: $\varepsilon = 0$, $A_{visc} = 0.1$. Simulation for $t = \{0, 0.01, 0.02, 0.05, 0.2, 0.5, 1, 2\}$

Now we consider, $\varepsilon \neq 0$.

- $\varepsilon = 0.1$, $A_{visc} = 0.1$

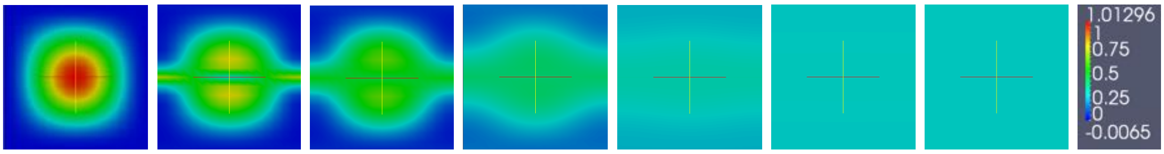

Figure 2.28: $\varepsilon = 0.1$, $A_{visc} = 0.1$. Simulation for $t = \{0, 0.01, 0.02, 0.05, 0.2, 0.5, 1, 2\}$

Adding the viscous term ($\varepsilon \neq 0$) has an important effect: it suppresses the "wave-breaking" phenomenon. This term generates a dispersion effect which causes the wave to spread and acts against the steeping effect of the nonlinearity [30]. Hence, we expect to obtain a smooth solution which, depending on the initial data, might approach a shock wave as $\varepsilon \to 0$. In the theory, this is known as the *vanishing viscosity effect* [5]. Further, in the case $\varepsilon \neq 0$, we observe that the middle point evaluation converges to $\int\limits_{\Omega} \eta_0(x, y)\mathrm{d}\Omega = 0.25$ in time:

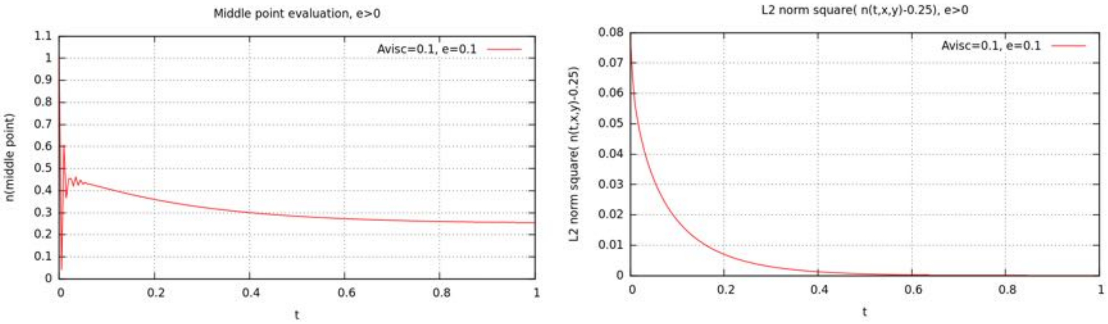

Figure 2.29: Middle point evaluation and $\|\eta(t) - 0.25\|^2_{L^2(\Omega)}$ for $\varepsilon = 0.1$ and $A_{visc} = 0.1$

As expected, we observe in Figure 2.29 a smooth decay of $\eta(t, x, y)$ to the stationary state in time. In the next plots we see the results for $\mathbf{v}_x$ and $\mathbf{v}_y$ for $\varepsilon = 0.1$ and $A_{visc} = 0.1$. We observe that the velocity in the x direction is of order $\mathcal{O}(10^{-3})$ and in the y direction is of order $\mathcal{O}(10^{-5})$. Therefore, the bump will vanish faster in the x direction (as seen in Figure 2.28).

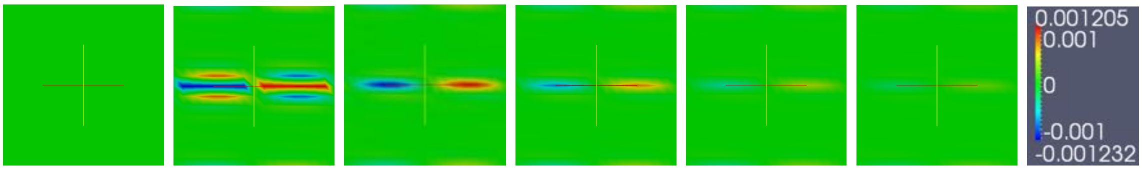

Figure 2.30: $\varepsilon = 0.1$, $A_{visc} = 0.1$. Simulation of $\mathbf{v}_x$ for $t = \{0, 0.02, 0.05, 0.1, 0.15, > 0.2\}$

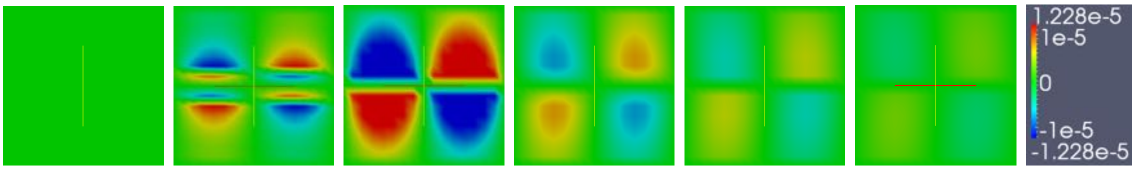

Figure 2.31: $\varepsilon = 0.1$, $A_{visc} = 0.1$. Simulation of $\mathbf{v}_y$ for $t = \{0, 0.02, 0.05, 0.1, 0.15, > 0.2\}$

As we increase ε, as expected, the bump decays faster:

- $\varepsilon = 0.1$, $A_{visc} = 0.1$

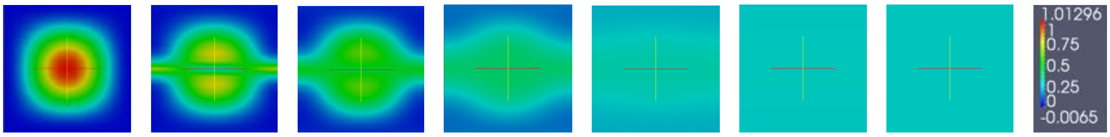

Figure 2.32: $\varepsilon = 0.1$, $A_{visc} = 0.1$. Simulation of η_0 for $t = \{0, 0.02, 0.05, 0.20, 0.50, 1, 2\}$

- $\varepsilon = 0.5$, $A_{visc} = 0.1$

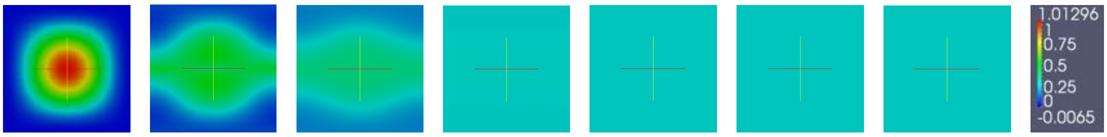

Figure 2.33: $\varepsilon = 0.5$, $A_{visc} = 0.1$. Simulation of η_0 for $t = \{0, 0.02, 0.05, 0.20, 0.50, 1, 2\}$

- $\varepsilon = 1$, $A_{visc} = 0.1$

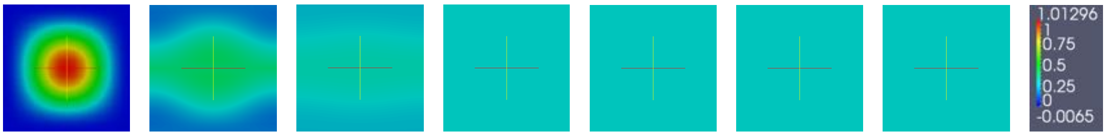

Figure 2.34: $\varepsilon = 1$, $A_{visc} = 0.1$. Simulation of η_0 for $t = \{0, 0.02, 0.05, 0.20, 0.50, 1, 2\}$

As $\varepsilon > 0$ increases, the second order term will have a bigger effect on the equation describing the dynamics of $\eta(t, x, y)$. This term has the same effect as the thermal diffusivity in the heat equation [48]. In other words, ε quantifies the rate at which the upper topography is "smoothed out" in time.

Finally, we analyze the effect of the space and time discretization over the viscous formulation that we proposed in this example. First, we consider the simulation for $f(x, y) = 4 \cdot 86164.1\pi \sin(2\pi y)$ and $\varepsilon = 0.1$, with different *level* of refinements = {3,4,5,6} and 200 time steps. As we will see in example 2.4.4, the refinement of the mesh in sectors for which the dynamics of the state variable changes rapidly, has an important effect in the outcome of the simulation. In the next figures we see the simulation for the different levels of refinement:

- $level = 3$

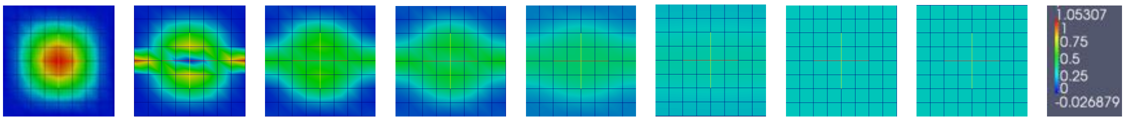

Figure 2.35: Simulation of η_0 for $level = 3$ and $t = \{0, 0.02, 0.05, 0.10, 0.20, 0.50, 1, 2\}$

- $level = 4$

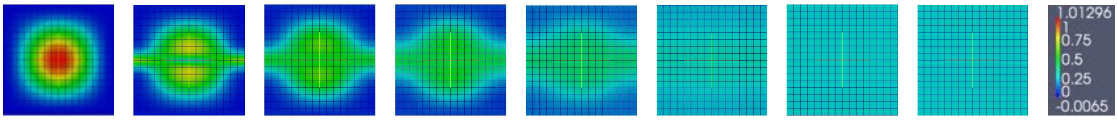

Figure 2.36: Simulation of η_0 for $level = 4$ and $t = \{0, 0.02, 0.05, 0.10, 0.20, 0.50, 1, 2\}$

- $level = 5$

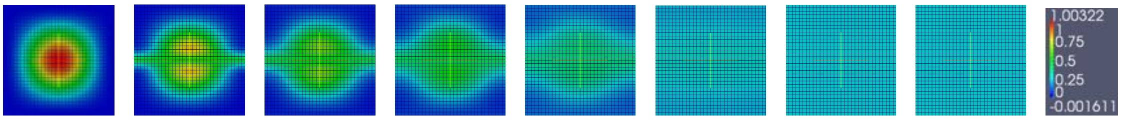

Figure 2.37: Simulation of η_0 for $level = 5$ and $t = \{0, 0.02, 0.05, 0.10, 0.20, 0.50, 1, 2\}$

- $level = 6$

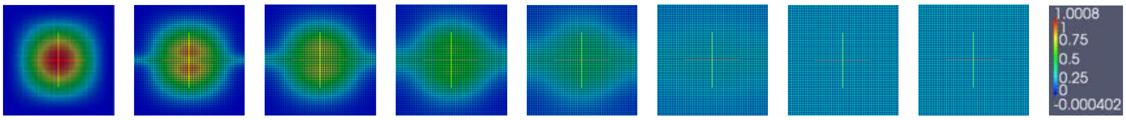

Figure 2.38: Simulation of η_0 for $level = 6$ and $t = \{0, 0.02, 0.05, 0.10, 0.20, 0.50, 1, 2\}$

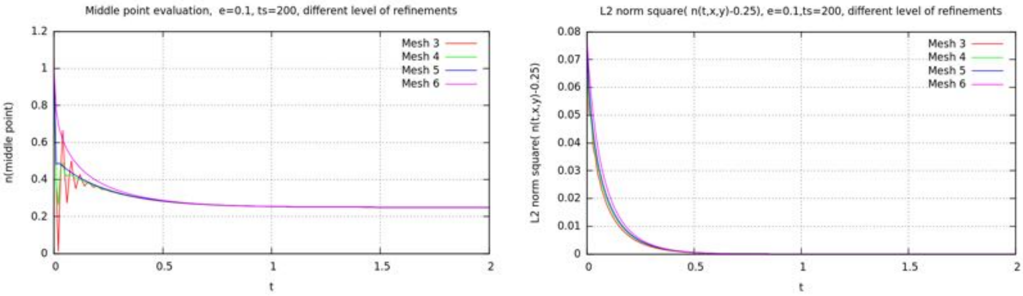

Figure 2.39: Middle point evaluation $\|\eta(t, x, y) - 0.25\|^2_{L^2(\Omega)}$ for different levels of refinement

In general, we observe the expected behavior for the different levels of refinement of $\eta(t, x, y)$. However, for higher level refinements we obtain a more accurate description of the dynamics of the bump. For example, for the mesh of *level* 3, 4 and $t = 0.02$ (Figure 2.35, 2.36), we notice that the middle point (where at time $t = 0$, the height fluctuation achieves the maximum height) vanishes drastically to zero for $t \to 0$, while, as we increase the mesh refinements, the sector around the middle point vanishes smoothly till achieving the equilibrium state (See Figure 2.38 and 2.39). After some $t < 0.50$, we obtain a similar result for the different levels of refinements in terms of the middle point evaluation of the height fluctuation and $\|\eta(t, x, y) - 0.25\|^2_{L^2(\Omega)}$.

Refining the mesh, as seen in Figure 2.39, shows the smooth decay for $\eta_0(x, y) = \sin(\pi \cdot x) + \sin(\pi \cdot y)$ to the stationary state $\eta(t, x, y) = 0.25$. Further, for a fix mesh with 4 levels of refinement, we variate the number of time steps= $\{100, 200, 500, 1000, 5000\}$:

- Number of time steps = 100

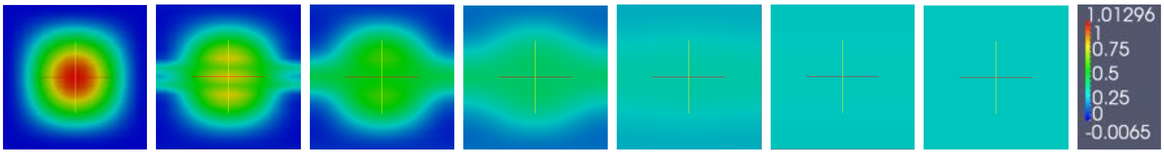

Figure 2.40: Simulation of η_0 for number of time steps = 100 and $t = \{0, 0.02, 0.05, 0.20, 0.50, 1, 2\}$

- Number of time steps = 200

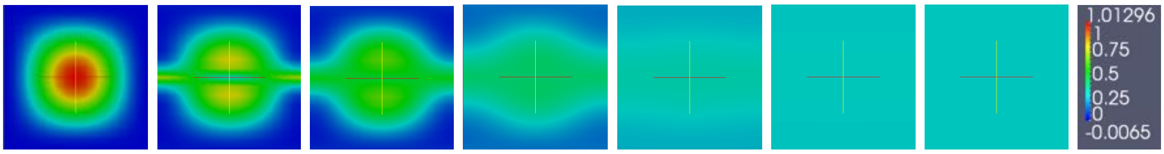

Figure 2.41: Simulation of η_0 for number of time steps = 200 and $t = \{0, 0.02, 0.05, 0.20, 0.50, 1, 2\}$

- Number of time steps = 500

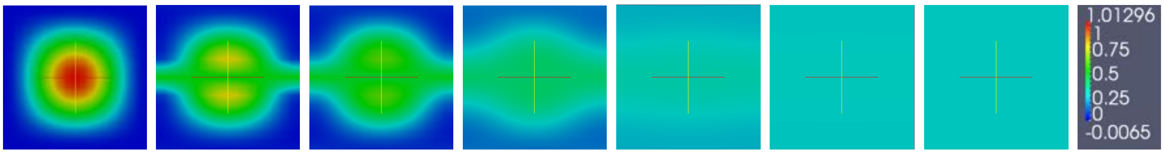

Figure 2.42: Simulation of η_0 for number of time steps = 500 and $t = \{0, 0.02, 0.05, 0.10, 0.20, 0.50, 1, 2\}$

- Number of time steps = 1000

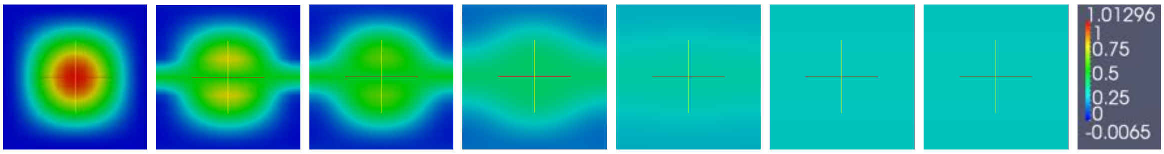

Figure 2.43: Simulation of η_0 for number of time steps = 1000 and $t = \{0, 0.02, 0.05, 0.10, 0.20, 0.50, 1, 2\}$

- Number of time steps = 5000

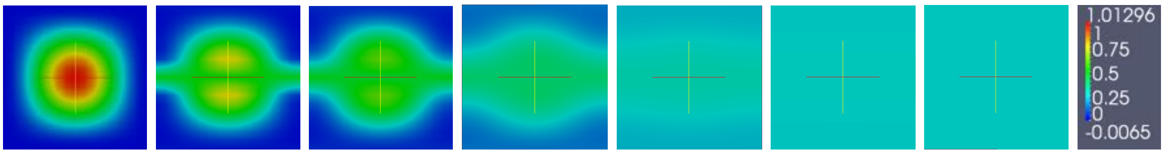

Figure 2.44: Simulation of η_0 for number of time steps = 5000 and $t = \{0, 0.02, 0.05, 0.10, 0.20, 0.50, 1, 2\}$

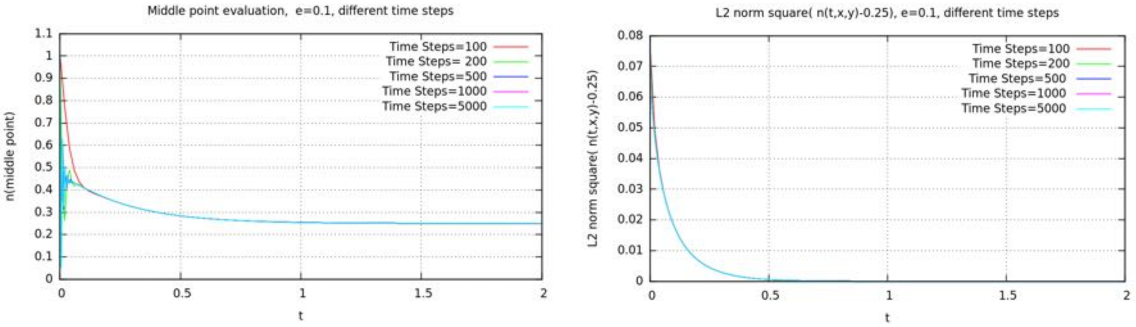

Figure 2.45: Middle point evaluation $\|\eta(t, x, y) - 0.25\|^2_{L^2(\Omega)}$ for different number of time steps

In Figures 2.45,2.40, 2.41,2.42,2.43,2.44, we observe that the patterns shown for 200, 500, 1000 and 5000 time steps are fairly similar in both $\|\eta(t, x, y) - 0.25\|^2_{L^2(\Omega)}$ and the middle point evaluation. Considering these results, it seems that for a fix level of refinements, the variation of the number of time steps does not introduce a relevant change in the dynamics description of the problem; on the other hand, increasing the level of refinement introduced greater modifications in this example, for which we can still observe the vanishing effect. To see this, we consider the variation of the time steps for a spatial mesh with 6 levels of refinement:

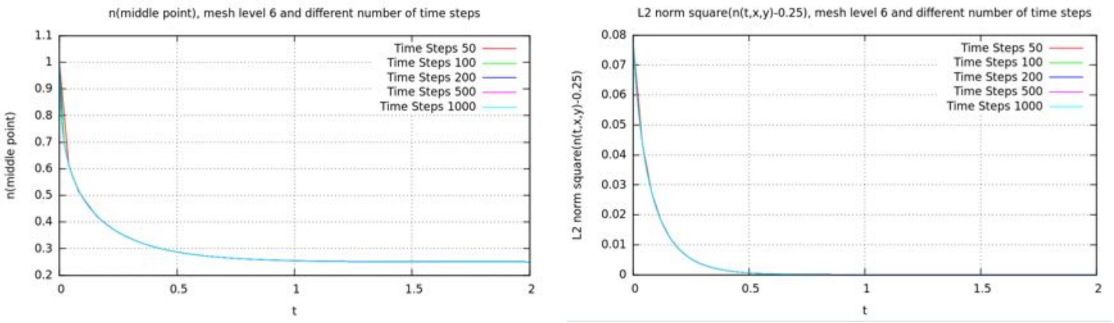

Figure 2.46: Simulation of η_0 for $level = 6$ and $t = \{0, 0.02, 0.05, 0.10, 0.20, 0.50, 1, 2\}$

In this case, the variation of the number of time steps is superfluous. The dynamics of this model, for a large ε, differs from the oscillatory behavior that we had in the last example (for which the selection of the time step was relevant). For this kind of problems, the refinement of the space discretization give us a more accurate description of the bump's vanishing at every point of the domain Ω.

2.4.4 Advection of a quasistationary vortex (Mesh Modification)

The quasi-stationary vortex is advected in 2D with flat bottom topography. $\Omega = [0, 100]^2$. The following example is taken from S. Vater [47] and Beckers [3], the vortex is defined in polar coordinates, where the tangential velocity $v_\theta(r)$ is given by

$$v_\theta(r) = \begin{cases} v_{max} \dfrac{s \cdot r}{r_m^2 - r^2} \cdot \sqrt{2a \exp\left(\dfrac{a}{r^2 - r_m^2}\right)} & 0 \leq r < r_m, \\ 0 & \text{otherwise}, \end{cases} \tag{2.12}$$

as function of the radius

$$r = \sqrt{(x-50)^2 + (y-50)^2},$$

which is defined as the distance from the central point Ω. The quantity v_{max} is the maximum tangential velocity within the vortex, r_m is the radius of the vortex, and a a smoothness scaling factor. In the test case at hand, these parameters are set to $v_{max} = 0.01$, $r_m = 25$ and $a = 2000$. Furthermore, the scaling factor is:

$$s = \frac{|r_{vm}^2 - r_m^2|}{r_{vm}\sqrt{2a \exp\left(a/(r_{vm}^2 - r_m^2)\right)}}, \tag{2.13}$$

where:

$$r_{vm} = \frac{1}{2}\sqrt{-2a + 2\sqrt{a^2 + 4r_m^4}},$$

is the radius at which v_{max} is attained.

The initial height field can be computed to:

$$\eta_0(x, y) = \begin{cases} h_{bg} - \dfrac{v_{max}^2 s^2}{g} \exp\left(\dfrac{a}{r^2 - r_m^2}\right) & \text{for } 0 \leq r < r_m, \\ h_{bg}, & otherwise. \end{cases} \tag{2.14}$$

The vortex is advected in x-direction with $(u_{bg}, v_{bg}) = (0.0025, 0)$. The initial velocity field is:

$$u_0(x, y) = u_{bg} - v_\theta(r) \sin\theta \quad \text{and} \quad v_0(x, y) = v_{bg} + v_\theta(r) \cos\theta,$$

with $\theta = \arctan\left(\dfrac{y-50}{x-50}\right)$.

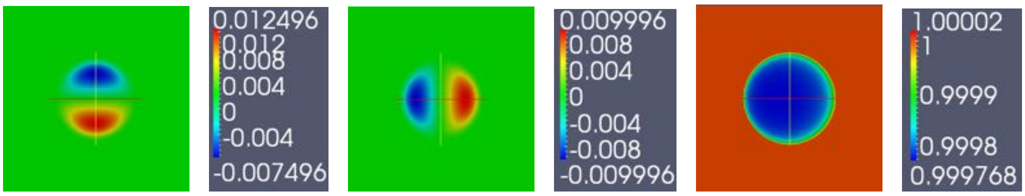

Figure 2.47: Initial values u_0, v_0, η_0. Example 2.4.4

Since the dynamics of this example takes place in a sector of the whole domain Ω (around the middle point), it could be useful to modify the mesh around the area of interest:

A modification to the mesh

In this case, we are interested in refining the mesh in a neighborhood around the middle point of the domain Ω.

Given a mesh $\{(x_i, y_i)\}_{i=1}^n$ of a rectangular domain Ω with corners $\{(0,0),\ (0,b),\ (a,0),\ (a,b)\}$, a slope $m > 1$ and $a_1, a_2 \in \mathbb{R}$ such that

$$a_1 < \min\left(\frac{a}{2}, \frac{b}{2}\right) \leq \max\left(\frac{a}{2}, \frac{b}{2}\right) < a_2.$$

The new mesh is given by $\{f(x_i), f(y_i)\}_{i=1}^n$ where $f : [z_1, z_2] \to [z_1, z_2]$ is a strictly increasing mapping. In this example, $z_1 = 0$, $z_2 = a$ (if the modification is in the x direction) and $z_2 = b$ (if it is in the y direction). $f(x_i)$ is given by:

$$f(x_i) = \begin{cases} mx_i,\ 0 \leq x_i \leq a_1, \\ ma_1 + m^*(x_i - a_1),\ a_1 < x_i \leq a_2, \\ a - m(a - a_2) + m(x - a_2),\ a_2 < x_i \leq a \quad (b \text{ for } y_i), \end{cases}$$

where $m^* = \dfrac{a - m(a - a_2 + a_1)}{a_2 - a_1}$. In order to guarantee that f is strictly increasing we need to impose the condition $m \leq \dfrac{a}{a - a_2 + a_1}$. Analogously, $f(y_i)$ is defined. The implementation of this mapping can be found in the main file (main.cc).

An example of the output of the mesh modification can be seen in Figure 2.48. In this case, $a = b = 100$, $a_1 = 40$, $a_2 = 60$ and $m = 1.2$.

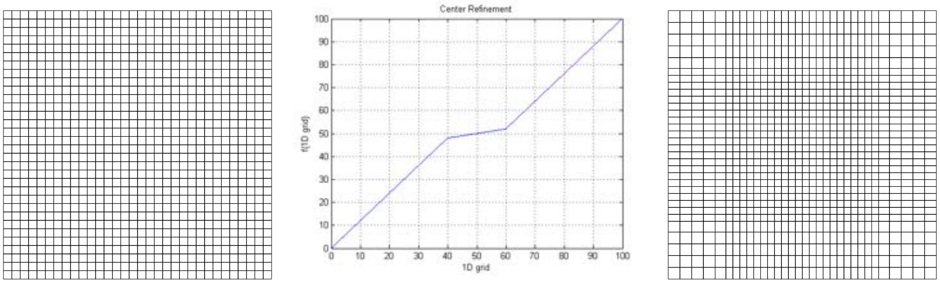

Figure 2.48: Mesh modification example

Alternatively, we can also achieve a higher level of refinement in small sectors of Ω by taking $m^* = \frac{1}{2^s}$ (s being the number of refinements that we wish to increase). Since $f(a_2)-f(a_1) = m^*(a_2-a_1)$, the length of the refined sector will increase as a_2-a_1 is bigger. Naturally, as we increase the length of the sector, in the outsides of it we will get a much less accurate approximation. In particular, if we want to modify a mesh *level k* so that we obtain a mesh with $k+s$ level of refinements and symmetric with respect to the middle point of Ω, the length of the outer elements of the mesh is $L_k = \frac{a}{2^{s+1}}\left((2^s-1)\frac{1}{2^{k-1}}\right)$. If we want to increase one level of refinement, i.e. $s=1$, the length of the outer element is $L_k = \frac{a}{4}\left(1+\frac{1}{2^{k-1}}\right) \geq \frac{a}{4}$. This is, if we gather all the elements of the mesh around the middle point of Ω ($a_1 \to 0$, $a_2 \to a$), then the outer element will have a length of at least one quarter of the length of the whole domain. Therefore, the approximation outside the central sector can be unsatisfactory if the dynamics of the phenomena in this area are complex.

Further, in this example we want to make a refinement around one sector of the domain Ω. Following the same reasoning we could modify the mesh in order to refine it around M different sectors, given a set of slopes $\{m_i^*\}_{i=1}^{M-1}$ and limit points $\{a_1^i, a_2^i\}_{i=1}^{M}$.

For example, we want to modify "Mesh *level* 3", so that we can achieve a centered sector with a refinement of the same length as "Mesh *level* 4":

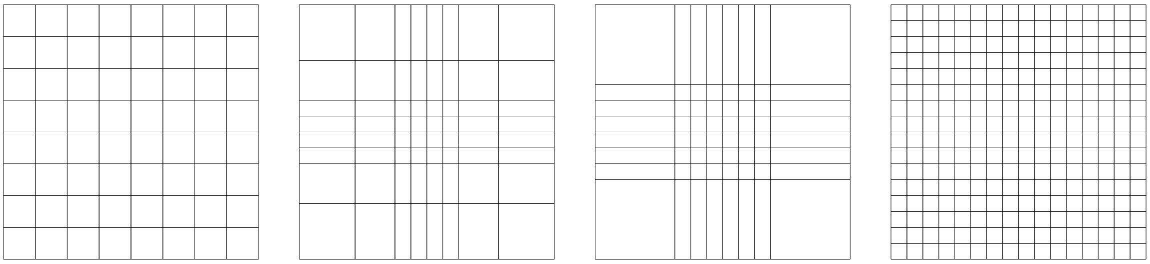

Figure 2.49: Mesh 3, Mesh $3^{20,80}$, Mesh $3^{10,90}$, Mesh 4

For the quasistationary vortex example, we will perform the simulation for $T = 0.05$ (50 time steps), $\varepsilon = A_{visc} = 0$, $f(x,y) \equiv 0$ and $g = 1$ as in [47]. We will compare the results obtained for a space discretization of $level = 7$ with respect to $level = 6$ and $level = 5$.

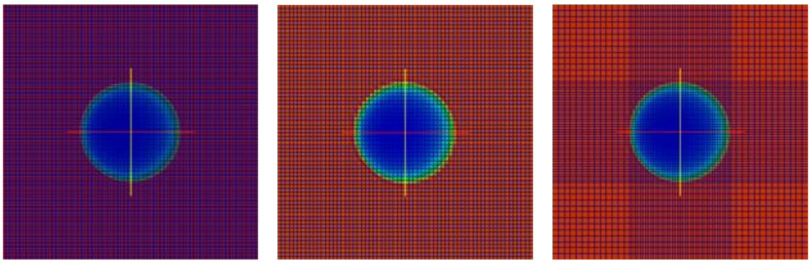

Figure 2.50: Initial height for: Mesh level 7, Mesh level 6, Mesh level 6 Modified ($a_1 = 10$, $a_2 = 90$, $m = 1.20$)

- $Level = 7,\ CPU\ time\ = 1149.91\ sec.$

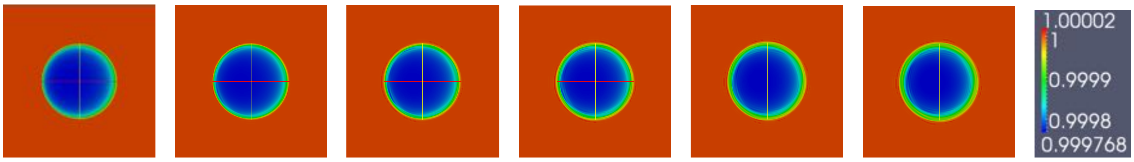

Figure 2.51: Simulation of the quasistationary vortex for mesh level 7 and $t = \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$

- $Level = 6,\ CPU\ time\ = 215.20\ sec.$

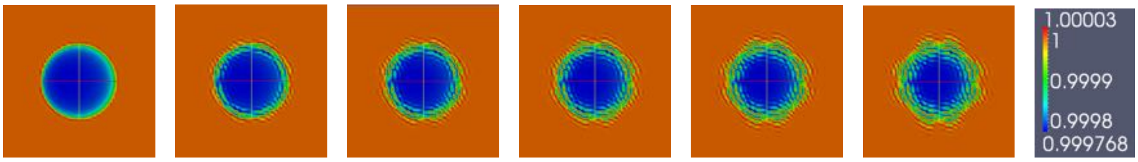

Figure 2.52: Simulation of the quasistationary vortex for mesh level 6 and $t = \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$

- $Level = 6$. Modified mesh ($a_1 = 10$, $a_2 = 90$, $m = 1.20$), CPU time = 264.87 sec

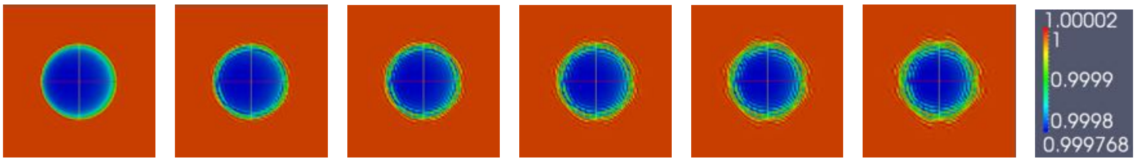

Figure 2.53: Simulation of the quasistationary vortex for mesh level 6 modified and $t = \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$

We notice some small displacements around the vortex respect to the solution obtained for $level = 7$. Since the bottom surface is almost flat with a shallow vortex in the center, for less refined meshes we obtain bigger perturbations around the vortex. Additionally, we observe that the reduction of one level of refinement implies a considerable reduction of CPU time. For example, *level 6* (Modified case) has a CPU time = 264.87 sec, which represents a 23% of the CPU time for *level 7*. This situation is more evident as we keep reducing the level of the refinement:

- *Level* = 7.

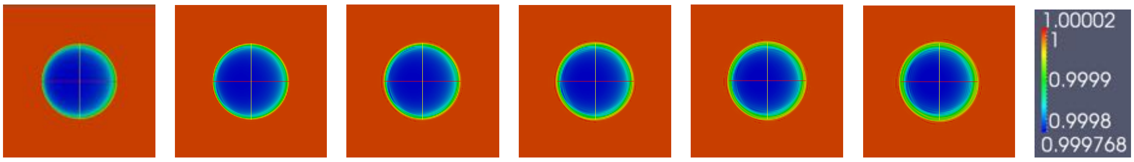

Figure 2.54: Simulation of the quasistationary vortex for mesh level 7 and $t = \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$

- *Level* = 5.

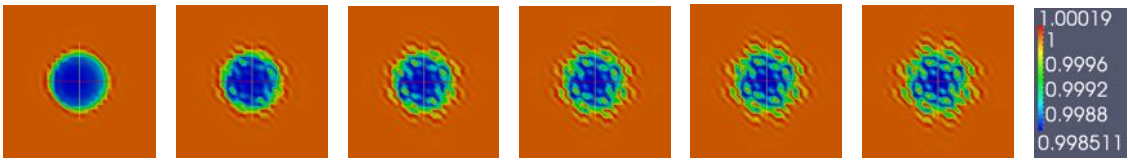

Figure 2.55: Simulation of the quasistationary vortex for mesh level 5 and $t = \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$

- *Level* = 5. Modified mesh ($a_1 = 10$, $a_2 = 90$, $m = 1.20$):

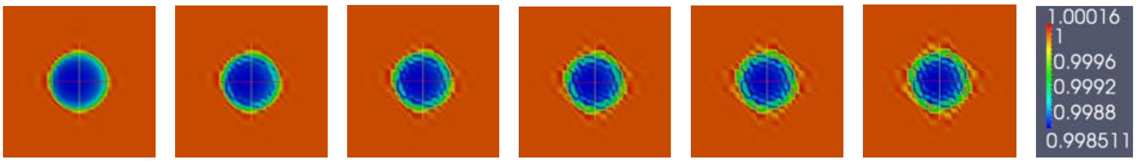

Figure 2.56: Simulation of the quasistationary vortex for mesh level 5 modified and $t = \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$

Having a good understanding of the phenomena's evolution given some initial setting allows us to perform modifications to the mesh such that it is adapted in critical areas of the domain Ω. In this example, we adapted the mesh in a neighborhood of the middle point, where the dynamics of the model takes place, specially in the thin ring-shaped sector (in green color in Figure 2.54), which serves as a "boundary" between the vortex area and the flat surface where the dynamic of the phenomena is almost nonexistent. On the other hand, the mapping that we proposed allows us to obtain more accurate results by keeping the same amount of degrees of freedom in the forward model as well as keeping the conforming nature of our finite element framework.

Chapter 3

Parameter Identification

In our analysis we consider a cost functional of the form:

$$J(u,q) = \frac{1}{2}\left[c_0\|\mathbf{v}(T) - \mathbf{v}_d\|^2 + c_1\|\eta(T) - \eta_d\|^2\right] + \frac{\alpha}{2}\|q - Q^d\|^2. \tag{3.1}$$

We are interested in measuring the velocity and height profile at final time T with respect to an expected profile $u_d = (\eta_d, \mathbf{v}_d)^t$. $c_0, c_1 \in \{0,1\}$ allows us to consider which profile we want to match in the optimization problem. $q \in Q = V$ is the control, in our case $q = u_0 = (\eta_0, \mathbf{v}_0)^t$.

The last term of the cost functional, with $\alpha > 0$ is used as a regularization term which guarantees the well-posedness of the optimization problem. $Q^d \in Q$ is usually set as 0. However, it can also be taken as $Q^d = (Q^\eta, Q^v)^t$, where Q^v and Q^η are the initial velocity and height profile that we suspect generating $u_d = (\eta_d, \mathbf{v}_d)^t$ at final time. Q^d is supposed to satisfy the regularity conditions specified in (1.34).

Consequently the optimization problem that we want to solve is:

$$\begin{cases} \min\limits_{q\in Q} J(u,q) = c_0\frac{1}{2}\left[\|\mathbf{v}(T) - \mathbf{v}^d\|^2 + c_1\|\eta(T) - \eta^d\|^2\right] + \frac{\alpha}{2}\|q - Q^d\|^2, \\ s.t.\ a^*(q,u,\psi) = (\partial_t u, \psi) - a(u,\psi) + (u(0) - q, \psi(0)) = 0,\ \forall\ \psi \in V \text{ and } t \in [0,T]. \\ \text{ with periodic boundary conditions on } \partial\Omega. \end{cases} \tag{3.2}$$

By fixing the initial data $u_0 = q$ the last term in (3.2) disappears. With this fixed initial data we can solve the forward problem (simulation). In our case, we take the initial control $q = 0$.

Since $q = u_0$, we have that $u = u(q)$. Hence, we can instead consider the reduced optimization problem:

$$\min_{q\in Q} j(q) \Longleftrightarrow \min_{q\in Q} J(q, u(q)) \text{ s.t. } a^*(q,u(q))(\psi) = 0,\ \forall\psi \in V. \tag{3.3}$$

Assuming the existence and sufficient regularity of u, the necessary optimal conditions of the first and second order are: $j'(q)(\delta q) = 0,\ j''(q)(\delta q, \delta q) \geq 0,\ \forall \delta q \in Q$.

To solve the reduced optimization problem, we use the inexact Newton method with Armijo type line-seach, which is already implemented in DopeLib [20].

Algorithm 3.0.1 Inexact Newton method for the reduced formulation

```
Choose $q_0 \in Q$, $\rho \in (0,1)$, $\sigma \in (0, \frac{1}{2})$ and let $k = 0$.
Choose $0 < \mathrm{TOL} < 1$ and $k_{max} \in \mathbb{N}$
Calculate $u(q_0)$ and $z(q_0)$
Calculate $d_0$ satisfying $\nabla^2 j(q_0) d_0 = -\nabla j(q_0)$ (Use a CG method (inexact). Needs $\delta u$ and $\delta z$ in each iteration.)
Set $\delta_0 = \|\nabla j(q_0)\|_Q$
while $\delta_k > \mathrm{TOL}$ and $k < k_{max}$ do
    Determine Step-Length: Set $l = 0$
    $q_{k+1} = q_k - \rho^l d_k$
    $r = (\nabla j(q_k), d_k)_Q$.
    while $j(q_{k+1}) > j(q_k) + \sigma \rho^l r$ do (Armijo type line-search)
        $l = l + 1$
        $q_{k+1} = q_k - \rho^l d_k$
    end while
    Calculate $u(q_{k+1})$ and $z(q_{k+1})$
    Calculate $d_{k+1}$ satisfying $\nabla^2 j(q_{k+1}) d_{k+1} = -\nabla j(q_{k+1})$ (Use a CG methods (inexact). Needs $\delta u$ and $\delta z$ in each iteration.)
    Set $\delta_{k+1} = \|\nabla j(q_{k+1})\|_Q$
    $k = k + 1$
end while
```

The difficulty in the Newton formulation rises in the calculation of $\nabla^2 j(q_{k+1})$ and $\nabla j(q_{k+1})$. Additionally, we can obtain the gradient and the hessian of $j(q)$ using the Lagrange formalism.

We define the Lagrangian (using the weak form of $a^*(q,u)(z)$):

$$\begin{aligned}
\mathcal{L}(q,u,z) &= J(q,u) - a^*(q,u)(z) \\
&= J(q,u) - \Big[\int_0^T \big((\partial_t \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v} + f\hat{k} \times \mathbf{v} + g\nabla\eta, z^v) + A_{visc}(\nabla \mathbf{v}, \nabla z^v) \\
&\quad + (\partial_t \eta + \nabla \cdot [\mathbf{v}(h_0 + \eta)], z^\eta) + \varepsilon(\nabla\eta, \nabla z^\eta)\big)\mathrm{d}t\Big] + (q - u(0), z(0))
\end{aligned}$$

The adjoint state $z = (z^\eta, z^v)$ is the Lagrange multiplier for the equality constraint $a^*(q,u)(\psi) = 0, \ \forall \psi \in V$.

Solution Process

- **Step 1:** Calculation of $j'(q)\delta q$:

  First, notice that $j'(q)$ is the Riesz representation for the gradient $\nabla j(q)$, i.e.

  $$(\nabla j(q), \delta q)_Q = j'(q)\delta q, \quad \forall \delta q \in Q.$$

  Since $j(q) = \mathcal{L}(x)$ with $x = (q, u, z)$, we know that

  $$j'(q)\delta q = \mathcal{L}'_q(x)\delta q + \mathcal{L}'_u(x)u'(q)\delta q + \mathcal{L}'_z(x)\frac{\mathrm{d}}{\mathrm{d}q}z \cdot \delta q.$$

  Due to the choice $u = u(q)$, it holds $\mathcal{L}'_z(x)\chi = 0,\ \forall \chi \in V$. We define $z = z(q)$ such that $\mathcal{L}'_u(x)\phi = 0,\ \forall \phi \in V$ (**Adjoint Equation**).

  Therefore,

  $$j'(q)\delta q = \mathcal{L}'_q(x)\delta q = (z(0) + \alpha(q - Q^d), \delta q). \tag{3.4}$$

Taking $\phi(t) = (\tilde{\eta}(t), \tilde{\mathbf{v}}(t))$ periodic in Ω, we also assume $z(t) = (z^\eta(t), z^v(t))^t$ periodic in Ω, the adjoint equation can be written as:

$$\begin{aligned}
\mathcal{L}'_u(x)\phi &= \mathcal{L}'_{\mathbf{v}}(x)\tilde{v} + \mathcal{L}'_\eta(x)\tilde{\eta} = 0, \quad \forall \phi \in V. \\
&= c_0(\mathbf{v}(T) - \mathbf{v}_d, \tilde{\mathbf{v}}(T)) + c_1(\eta(T) - \eta_d, \tilde{\eta}(T)) - \qquad (3.5) \\
&\quad -[\int\limits_0^T (\partial_t\tilde{\mathbf{v}} + \tilde{\mathbf{v}} \cdot \nabla\mathbf{v} + \mathbf{v} \cdot \nabla\tilde{\mathbf{v}} + f\hat{k} \times \tilde{\mathbf{v}} + g\nabla\tilde{\eta}, z^v) + A_{visc}(\nabla\tilde{\mathbf{v}}, \nabla z^v) \\
&\quad + (\partial_t\tilde{\eta} + \nabla \cdot [\tilde{v}(h_0 + \eta)] + \nabla \cdot (\mathbf{v}\tilde{\eta}), z^\eta) + \varepsilon(\nabla\tilde{\eta}, \nabla z^\eta))\,\mathrm{d}t] - (\tilde{u}(0), z(0)) \\
&= 0.
\end{aligned}$$

By integration by parts (on the terms $\partial_t\tilde{v}$ and $\partial_t\tilde{\eta}$) equation (3.5) can be rewritten as:

$$\begin{aligned}
\mathcal{L}'_u(x)\phi &= c_0(\mathbf{v}(T) - \mathbf{v}_d, \tilde{\mathbf{v}}(T)) + c_1(\eta(T) - \eta_d, \tilde{\eta}(T)) + \int\limits_0^T \big[(\partial_t z^v, \tilde{\mathbf{v}}) + (\partial_t z^\eta, \tilde{\eta}) - \\
&\quad -(\tilde{\mathbf{v}} \cdot \nabla\mathbf{v} + \mathbf{v} \cdot \nabla\tilde{\mathbf{v}} + f\hat{k} \times \tilde{\mathbf{v}} + g\nabla\tilde{\eta}, z^v) - A_{visc}(\nabla\tilde{\mathbf{v}}, \nabla z^v) - (\nabla \cdot [\tilde{v}(h_0 + \eta)] + \\
&\quad +\nabla \cdot (\mathbf{v}\tilde{\eta}), z^\eta) - \varepsilon(\nabla\tilde{\eta}, \nabla z^\eta)]\mathrm{d}t - (\tilde{u}(0), z(0)) - (\tilde{\mathbf{v}}(T), z^v(T)) + (\tilde{\mathbf{v}}(0), z^v(0)) - \\
&\quad -(\tilde{\eta}(T), z^\eta(T)) + (\tilde{\eta}(0), z^\eta(0)) = 0
\end{aligned}$$

We can see that the terms at $t = 0$ disappear. In order to get the adjoint equation we consider the different cases:

1) If we assume $\tilde{u}(T) = (\tilde{\eta}(T), \tilde{v}(T))^t = (0)^3$, the adjoint equation implies the solution of the weak problem:

   $$\begin{aligned}
   &(\partial_t z^v, \tilde{\mathbf{v}}) + (\partial_t z^\eta, \tilde{\eta}) - (\tilde{\mathbf{v}} \cdot \nabla\mathbf{v} + \mathbf{v} \cdot \nabla\tilde{\mathbf{v}} + f\hat{k} \times \tilde{\mathbf{v}} + g\nabla\tilde{\eta}, z^v) - A_{visc}(\nabla\tilde{\mathbf{v}}, \nabla z^v) \\
   &-(\nabla \cdot [\tilde{v}(h_0 + \eta)] + \nabla \cdot (\mathbf{v}\tilde{\eta}), z^\eta) - \varepsilon(\nabla\tilde{\eta}, \nabla z^\eta) = 0, \quad \forall \tilde{u} = (\tilde{\eta}, \tilde{\mathbf{v}}) \in V
   \end{aligned} \tag{3.6}$$

2) $\tilde{\mathbf{v}}(T)$ with compact support on Ω and $\tilde{\eta}(T) \equiv 0$ on Ω, then:

$$(c_0(\mathbf{v}(T) - \mathbf{v}_d) - z^v(T), \tilde{\mathbf{v}}(T)) = 0$$

which gives the initial values of the adjoint variable z^v:

$$z^v(T) = c_0(\mathbf{v}(T) - \mathbf{v}_d) \text{ in } \Omega \tag{3.7}$$

3) $\tilde{\eta}(T)$ with compact support on Ω, and $\tilde{\mathbf{v}}(T) \equiv 0$ on Ω, then:

$$(c_1(\eta(T) - \eta_d) - z^\eta(T), \tilde{\eta}(T)) = 0.$$

Hence, the initial value of the adjoint variable z^η is:

$$z^\eta(T) = c_1(\eta(T) - \eta_d) \text{ in } \Omega \tag{3.8}$$

Therefore, collecting the equations (3.6), (3.7) and (3.8), the weak system for the adjoint variable $z = (z^\eta, z^v)^t$ is

$$\begin{cases} (\partial_t z^v, \tilde{\mathbf{v}}) + (\partial_t z^\eta, \tilde{\eta}) - (\tilde{\mathbf{v}} \cdot \nabla \mathbf{v} + \mathbf{v} \cdot \nabla \tilde{\mathbf{v}} + f\hat{k} \times \tilde{\mathbf{v}} + g\nabla \tilde{\eta}, z^v) - A_{visc}(\nabla \tilde{\mathbf{v}}, \nabla z^v) \\ \qquad -(\nabla \cdot [\tilde{v}(h_0 + \eta)] + \nabla \cdot (\mathbf{v}\tilde{\eta}), z^\eta) - \varepsilon(\nabla \tilde{\eta}, \nabla z^\eta) = 0, \quad \forall \tilde{u} = (\tilde{\eta}, \tilde{\mathbf{v}}) \in V \\ z^v(T) = c_0(\mathbf{v}(T) - \mathbf{v}_d) \text{ in } \Omega \\ z^\eta(T) = c_1(\eta(T) - \eta_d) \text{ in } \Omega \\ z = (z^\eta, z^{\mathbf{v}})^t \text{ periodic in } \Gamma \end{cases} \tag{3.9}$$

As one can see, the adjoint equation needs to be solved backward in time. Therefore we need to introduce a reasonable dual time-stepping scheme (e.g. BE, CN, FS Schemes).

In order to solve the system (3.14) we use a similar strategy as in the simulation. Since we assume $z \in V_h$, there exist coefficients $\{c_1, \ldots, c_N\}$ such that $z = \sum_{j=1}^{N} c_j \psi_j$, where N is the number of degrees of freedom in the element and $\{\psi_j = (\chi_j, \phi_j)^t\}_{j=1}^N$ is the basis of V_h.

Considering a time discretization of problem (3.14) as shown in section 2.3, and taking $\psi = (\chi, \phi)^t = (\tilde{\eta}, \tilde{v})^t$, we obtain

$$\begin{cases} \left(\dfrac{(z^v)^{n+1} - (z^v)^n}{\text{dt}}, \phi \right) + \left(\dfrac{(z^\eta)^{n+1} - (z^\eta)^n}{\text{dt}}, \chi \right) - \theta[(\phi \cdot \nabla \mathbf{v}^{n+1} + \mathbf{v}^{n+1} \cdot \nabla \phi, (z^v)^{n+1}) + \\ +(f\hat{k} \times \phi, (z^v)^{n+1}) + (g\nabla \chi, (z^v)^{n+1}) + A_{visc}(\nabla \phi, \nabla (z^v)^{n+1}) + (\nabla \cdot (\phi(h_0 + \eta^{n+1})) + \\ +\nabla \cdot (\mathbf{v}^{n+1}(\chi)), (z^v)^{n+1}) + (\varepsilon \nabla \chi, \nabla (z^v)^{n+1})] = (\theta - 1)[(\phi \cdot \nabla \mathbf{v}^n + \mathbf{v}^n \cdot \nabla \phi, (z^v)^n) \\ +(f\hat{k} \times \phi, (z^v)^n) + (g\nabla \chi, (z^v)^n) + A_{visc}(\nabla \phi, \nabla (z^v)^n) + (\nabla \cdot (\phi(h_0 + \eta^n)) \\ +\nabla \cdot (\mathbf{v}^n(\chi)), (z^v)^n) + (\varepsilon \nabla \chi, \nabla (z^v)^n)]. \end{cases}$$

As we said before, the adjoint equation is solved backwards since we know the value of z at final time T (this value can be found in ElementValue_U at localfunctional.h).

Given the information of $u = (\eta, \mathbf{v})^t$ and $z = (z^\eta, z^v)^t$ at time step $n+1$, we want to approximate the value of z at time n. To do so, we solve a system of the form:

$$A^{n+1}c^n = b^{n+1}$$

where $A^{n+1} = K + \mathrm{dt}(1-\theta)B^{n+1}$. K is a $N \times N$ matrix given by $K_{ij} = (\psi_j, \psi_i)$ (this matrix can be found in ElementTimeMatrix at localpde.h). On the other hand, B^{n+1} is defined:

$$\begin{aligned} B_{ij}^{n+1} &= (\phi_i \cdot \nabla \mathbf{v}^n + \mathbf{v}^n \cdot \nabla \phi_i, \phi_j) + (f\hat{k} \times \phi_i, \phi_j) + (g\nabla\chi_i, \phi_j) + A_{visc}(\nabla\phi_i, \nabla\chi_j) + \\ &\quad + (\nabla \cdot (\phi_i(h_0 + \eta^n)) + \nabla \cdot (\mathbf{v}^n(\chi_i)), \chi_j) + (\varepsilon\nabla\chi_i, \nabla\chi_j) \end{aligned} \tag{3.10}$$

B^t is prescribed in ElementMatrix at localpde.h. Finally, the right hand side b^{n+1} is

$$\begin{aligned} b_i^{n+1} &= (z_v^{n+1}, \phi_i) + (z_\eta^{n+1}, \chi_i) - \Delta t\theta[(\phi \cdot \nabla \mathbf{v}^{n+1} + \mathbf{v}^{n+1} \cdot \nabla\phi_i, (z^v)^{n+1}) + (f\hat{k} \times \phi_i, (z^v)^{n+1}) + \\ &\quad + (g\nabla\chi_i, (z^v)^{n+1}) + A_{visc}(\nabla\phi_i, \nabla(z^v)^{n+1}) + (\nabla \cdot (\phi_i(h_0 + \eta^{n+1})) + \\ &\quad + \nabla \cdot (\mathbf{v}^{n+1}(\chi_i)), (z^v)^{n+1}) + (\varepsilon\nabla\chi_i, \nabla(z^v)^{n+1})] \end{aligned}$$

The first term of b_i^{n+1} is implemented in ElementEquation_U and the second term in ElementTimeEquation_U at Localpde.h.

On the other hand, the **Gradient Equation** $\mathcal{L}'_q(x)\tilde{q} = 0, \quad \forall \tilde{q} \in Q$, implies:

$$\begin{aligned} \mathcal{L}'_q(x)\tilde{q} &= J'_q(q,u)\tilde{q} - a'_q(q,u)(z;\tilde{q}) \\ &= \alpha(q - Q^d, \tilde{q}) + (\tilde{q}, z(0)) \\ &= (z(0) + \alpha(q - Q^d), \tilde{q}), \end{aligned}$$

From the *fundamental lemma in the calculus of variations*, the gradient equation implies that the optimal control q^* satisfies:

$$q^* = \frac{-z(0) + \alpha Q^d}{\alpha}. \tag{3.11}$$

Regarding the **State Equation** $\mathcal{L}'_z(x)(\chi) = 0$, $\forall \chi \in V$, we know that due to the choice $u = u(q)$ and the linearity of $a^*(q,u)(z)$ w.r.t. z, the state equation is satisfied.

- **Step 2:** Calculation of $j''(q)(\delta r, \delta q)$:

  As we know, $j''(q)$ is the Riesz representation for the hessian $\nabla^2 j(q)$, i.e.

  $$(\nabla^2 j(q)\delta r, \delta q)_Q = j''(q)(\delta r, \delta q), \quad \forall \delta q, \delta r \in Q.$$

To calculate the second derivative we need to solve additional equations. Let $\delta q \in Q$ be a given direction. We search $\delta u \in V$ solving the **Tangent Equation**:

$$\mathcal{L}''_{qz}(x)(\delta q, \psi) + \mathcal{L}''_{uz}(x)(\delta u, \psi) = 0, \ \forall \psi \in V.$$

We know that $\mathcal{L}''_{qz}(x)(\delta q, \psi) = (\delta q, \psi(0))$.

$$\begin{aligned}\mathcal{L}''_{uz}(x)(\delta u^v, \psi) &= -\int_0^T \Big((\partial_t \delta u^v + \delta u^v \cdot \nabla \mathbf{v} + \mathbf{v} \cdot \nabla \delta u^v + f\hat{k} \times \delta u^v + g\nabla \delta u^\eta, \phi\Big) + \\ &\quad +(A_{visc}\nabla \delta u^v, \nabla \phi) + (\partial_t \delta u^\eta + \nabla \cdot [\delta u^v(h_0 + \eta)] + \nabla \cdot (\mathbf{v}\delta u^\eta), \chi) + \\ &\quad (\varepsilon \cdot \nabla \delta u^\eta, \nabla \chi))\mathrm{d}t - (\delta u(0), \psi(0))\end{aligned}$$

Following the same reasoning that we used to deduce the adjoint equation, the tangent equation can be written as:

$$\begin{cases} (\partial_t \delta u^v + \delta u^v \cdot \nabla \mathbf{v} + \mathbf{v} \cdot \nabla \delta u^v + f\hat{k} \times \delta u^v + g\nabla \delta u^\eta, \phi) + (A_{visc}\nabla \delta u^v, \nabla \phi) + \\ (\partial_t \delta u^\eta + \nabla \cdot [\delta u^v(h_0 + \eta)] + \nabla \cdot (\mathbf{v}\delta u^\eta), \chi) + (\varepsilon \cdot \nabla \delta u^\eta, \nabla \chi) = 0 \text{ in } \Omega^T \\ \delta u(0) = (\delta u^\eta(0), \delta u^v(0))^T = \delta q \text{ in } \Omega \\ \delta u = (\delta u^\eta, \delta u^v)^t \text{ periodic in } \Gamma. \end{cases} \tag{3.12}$$

After discretizing (3.12), as we did in the adjoint equation, the information of this model can be found in ElementMatrix, ElementTimeMatrix, ElementEquation_UT and ElementTimeEquation_UT in localpde.h and Init_ElementRhs_QT in localfunctional.h.

Further, to find $\delta z \in V$ we solve the **Dual for Hessian equation:**

$$\mathcal{L}''_{qu}(x)(\delta q, \psi) + \mathcal{L}''_{uu}(x)(\delta u, \psi) + \mathcal{L}''_{zu}(x)(\delta z, \psi) = 0, \forall \psi \in V.$$

In this example, $\mathcal{L}''_{qu}(x)(\delta q, \psi) = 0, \forall \psi \in V$.

$$\begin{aligned}\mathcal{L}''_{uu}(x)(\delta u, \psi) &= c_0(\phi(T), \delta u^v(T)) + c_1(\chi(T), \delta u^\eta(T)) - \int_0^T [(\delta u^v \cdot \nabla \phi + \phi \cdot \nabla \delta u^v, z^v) + \\ &\quad +(\nabla \cdot (\delta u^v \cdot \chi) + \nabla \cdot (\phi \cdot \delta u^\eta), z^\eta)]\mathrm{d}t\end{aligned}$$

Finally, we determine

$$\begin{aligned}\mathcal{L}''_{zu}(x)(\delta z, \psi) = -\int_0^T [(\partial_t \phi + \phi \cdot \nabla \mathbf{v} + \mathbf{v} \cdot \nabla \phi + f\hat{k} \times \phi + g\nabla \chi - A_{visc}\Delta \phi, \delta z^v) + \\ +(\partial_t \chi + \nabla \cdot [\phi(h_0 + \eta)] + \nabla \cdot [\mathbf{v} \cdot \chi] - \varepsilon \Delta \chi, \delta z^\eta)]\mathrm{d}t - (\psi(0), \delta z(0))\end{aligned} \tag{3.13}$$

Equivalently, as in the adjoint equation, we get the system for $\delta z = (\delta z^\eta, \delta z^v)^t \in V$:

$$\begin{cases} (\partial_t \delta z^v, \phi) + (\partial_t \delta z^\eta, \chi) - (\phi \cdot \nabla \mathbf{v} + \mathbf{v} \cdot \nabla \phi + f\hat{k} \times \phi + g\nabla\chi, \delta z^v) - A_{visc}(\nabla\phi, \nabla\delta z^v) \\ \qquad -(\nabla \cdot [\tilde{v}(h_0 + \eta)] + \nabla \cdot (\mathbf{v}\chi), \delta z^\eta) - \varepsilon(\nabla\chi, \nabla\delta z^\eta) - [((\delta u^v \nabla)\phi + \\ \qquad +(\phi\nabla)\delta u^v, z^v) + (\nabla \cdot (\delta u^v \cdot \chi) + \nabla \cdot (\phi \cdot \delta u^\eta), z^\eta)] = 0, \quad \forall \psi = (\chi, \phi) \in V \\ \delta z^v(T) = c_0 \delta u^v(T) \text{ in } \Omega \\ \delta z^\eta(T) = c_1 \delta u^\eta(T) \text{ in } \Omega \\ \delta z = (\delta z^\eta, \delta z^v)^t \text{ periodic in } \Gamma. \end{cases}$$

The elements of the discretized dual for the Hessian Equation can be found in ElementValue_UU in localfunctional.h and ElementMatrix, ElementTimeMatrix, Init_ElementRhs_QTT, ElementEquation_UT and ElementEquation_UTT in localpde.h.

Considering this choice of directions δq, δu, δz and for $\delta r \in Q$, the second derivative of j can be expressed as:

$$j''(q)(\delta q, \delta r) = \mathcal{L}''_{qq}(x)(\delta q, \delta r) + \mathcal{L}''_{uq}(x)(\delta u, \delta r) + \mathcal{L}''_{zq}(x)(\delta z, \delta r). \tag{3.14}$$

We know that $\mathcal{L}''_{qq}(x)(\delta q, \delta r) = \alpha(\delta r, \delta q)$, $\mathcal{L}''_{uq}(x)(\delta u, \delta r) = 0$ and $\mathcal{L}''_{zq}(x)(\delta z, \delta r) = (\delta r, \delta z(0))$. Hence,

$$j''(q)(\delta q, \delta r) = \alpha(\delta r, \delta q) + (\delta r, \delta z(0)). \tag{3.15}$$

Using these terms, we can calculate the Newton direction δq as the solution of the problem:

$$j''(q)(\delta q, \chi) = -j'(q)(\chi), \quad \forall \chi \in Q \tag{3.16}$$

as finding $\delta q \in Q$ such that

$$\alpha(\delta q, \chi) + (\delta q, \delta z(0)) = -(\delta z(0) + \alpha(q - Q^d), \chi), \ \forall \chi \in Q. \tag{3.17}$$

Using this direction we can compute the iteration of the inexact Newton method proposed at the beginning of the chapter.

3.1 Examples

In this section, we will analyze the results obtained for the Parameter Identification problem for different settings.

In order to solve the forward and parameter identification problem, we set the following parameters at dope.prm:

Newton Solver Parameters	
Parameter	**Value**
line_maxiter	500
linesearch_rho	0.5
nonlinear_global_tol	1.e-10
nonlinear_maxiter	500
nonlinear_rho	0.1
nonlinear_tol	1.e-10

Table 3.1: Newton Solver Parameters

Reduced Newton Algorithm Parameters	
Parameter	**Value**
line_maxiter	500
linear_global_tol	1.e-7
linear_tol	1.e-7
linesearch_c	0.05
linesearch_rho	0.9
nonlinear_global_tol	1.e-7
nonlinear_maxiter	500
nonlinear_tol	1.e-7

Table 3.2: Reduced Newton Algorithm Parameters

3.1.1 Bump with different $\alpha = \{0.01, 1, 10\}$

The objective of this example is to compare the optimal control q^* for $Q^d(x,y) = (Q^\eta(x,y), Q^{\mathrm{v}}(x,y))$ defined as $Q^{\mathrm{v}}(x,y) = (0,0)$ and $Q^\eta(x,y) = \sin(\pi x)^2 \cdot \sin(\pi y)^2$. Additionally, we set $u^d(x,y) = (0,0,0.25)$, $\alpha = \{0.01, 1, 10\}$, $c_0 = 0$, $c_1 = 1$ (we just want to match at final time T the upper topography field), $T = 0.40$ (discretized in 20 time steps), $\varepsilon = A_{visc} = 0.1$, $h_0(x,y) = 1000$, the Coriolis force $f(x,y) = 4 \cdot 86164.1 \cdot \pi \sin(2\pi y)$ and the domain $\Omega = [0,1]^2$ discretized in a 256 elements mesh as in the Example 2.4.3.

In other words, we want to understand the behavior of the initial velocity field so that given an initial topography $Q^\eta(x,y) = \sin(\pi x)^2 \cdot \sin(\pi y)^2$, we obtain a flat topography $\eta(T,x,y) = 0.25$ at final time $T = 0.40$.

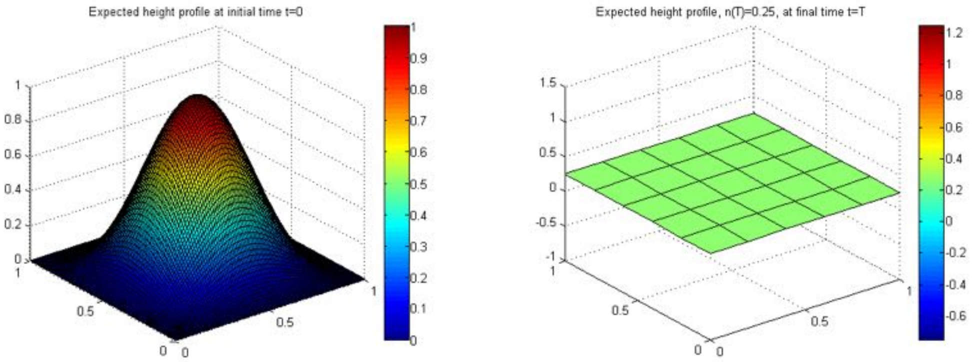

Figure 3.1: Expected initial and final upper topography. Example 3.1.1.

The Newton method for the reduced problem converged in 6 iterations for $\alpha = 0.01$ and $\alpha = 1$ and in 5 iterations for $\alpha = 10$ for the parameters set in Table 3.1 and Table 3.2. In the next tables we can see the results for the method in each iteration:

$\alpha = 0.01$, CPU time = 1551.67 sec.				
Iteration	**Cost Functional**	**Residual(rel)**	**LinearIters**	**Line search**
0	0.032	1	-	-
1	3.90e-04	1.10e-02	2	0
2	1.36e-04	5.46e-04	27	0
3	1.35e-04	6.17e-06	68	0
4	1.35e-04	6.57e-07	48	0
5	1.35e-04	4.05e-07	14	0
6	1.35e-04	3.70e-07	6	0

Table 3.3: Newton Iterations for $\alpha = 0.01$. Example 3.1.1.

$\alpha = 1$, CPU time = 285.37 sec.				
Iteration	**Cost Functional**	**Residual(rel)**	**LinearIters**	**Line search**
0	0.1	1	-	-
1	2.20e-02	3.76e-01	2	0
2	1.23e-03	6.53e-02	3	0
3	6.28e-04	4.85e-03	4	0
4	6.25e-04	2.30e-04	4	0
5	6.25e-04	1.92e-06	6	0
6	6.25e-04	9.77e-08	4	0

Table 3.4: Newton Iterations for $\alpha = 1$. Example 3.1.1.

$\alpha = 10$. CPU time = 184.37 sec.				
Iteration	**Cost Functional**	**Residual(rel)**	**LinearIters**	**Line search**
0	0.73	1	-	-
1	7.15e-03	9.40e-02	2	0
2	7.19e-04	8.86e-03	2	0
3	6.62e-04	8.35e-04	2	0
4	6.62e-04	3.74e-06	3	0
5	6.62e-04	1.68e-08	3	0

Table 3.5: Newton Iterations for $\alpha = 10$. Example 3.1.1.

As we decrease α the problem is "harder" to solve, as we can verify in the CPU time and number of linear iterations required to solve the optimization problem. As we will see in the following subsections for smaller values of α we will obtain a solution that will match better $u(T)$ to u_d. However the associated inverse problem tends to be unstable and, therefore, is more difficult to achieve a satisfactory result [25].

In the following plots we can see the optimal controls for the different values of α:

- Optimal control for $\alpha = 0.01$

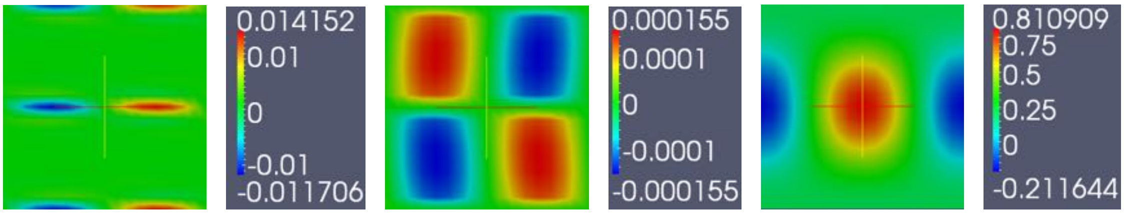

Figure 3.2: Optimal control $(u_0, v_0), \eta_0$ for $\alpha = 0.01$. Example 3.1.1.

- Optimal control for $\alpha = 1$

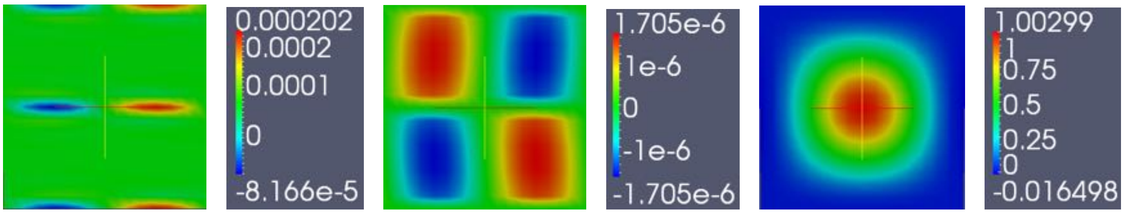

Figure 3.3: Optimal control $(u_0, v_0), \eta_0$ for $\alpha = 1$. Example 3.1.1.

- Optimal control for $\alpha = 10$

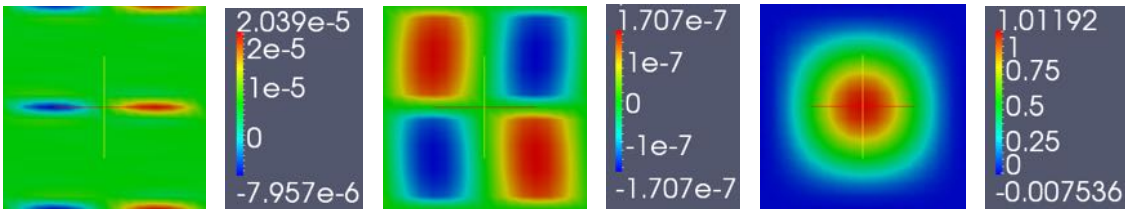

Figure 3.4: Optimal control $(u_0, v_0), \eta_0$ for $\alpha = 10$. Example 3.1.1.

As we increase $\alpha > 1$ our priority in the optimization problem is to match q as much as possible to Q^d (as we can see in Table 3.6). The optimal controls for $\alpha = 1$ and $\alpha = 10$ are fairly similar. However, the order of the velocities for $\alpha = 10$ are smaller, which is to expect since $Q^{\mathrm{v}} = (0, 0)$. For the different values of α, the velocity in the x-direction is larger than the velocity in the y-direction. Therefore, we expect the bump to dissipate "faster" in the x-direction (as seen in Section 2.4.3).

α	$\|q^* - Q^d\|^2_{L^2(\Omega)}$
0.01	$2.0378 \cdot 10^{-2}$
1	$5.17806 \cdot 10^{-5}$
10	$3.73543 \cdot 10^{-6}$

Table 3.6: α vs $\|q^* - Q^d\|^2_{L^2(\Omega)}$. Example 3.1.1

For $\alpha = 0.01$ we give more importance to matching $u(T)$ with u^d in the optimization problem. In this case, we obtained a different type of initial bump with a smaller height and two negative bumps in the boundary (due to the periodicity on $\partial\Omega$).

Considering the optimal control that we obtained, the bumps exhibit the next behavior in time:

- $\alpha = 0.01$

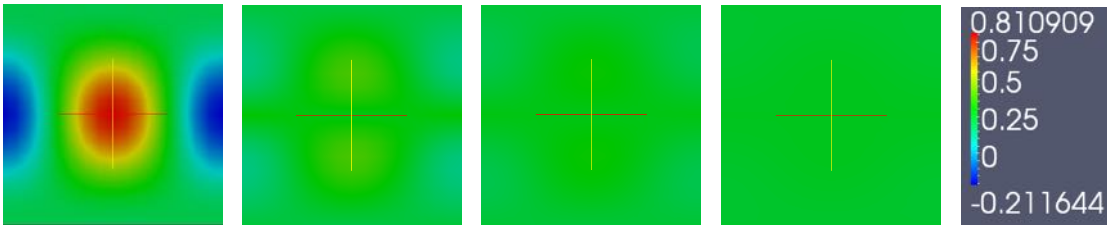

Figure 3.5: Evolution of $\eta(t, x, y)$ for $t = \{0, 0.10, 0.20, 0.40\}$ and $\alpha = 0.01$. Example 3.1.1.

- $\alpha = 1$

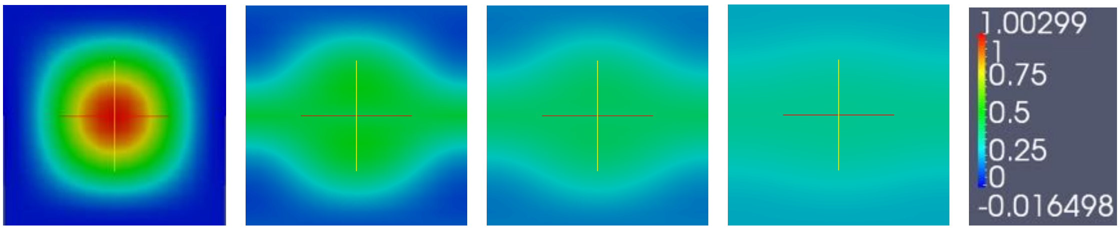

Figure 3.6: Evolution of $\eta(t, x, y)$ for $t = \{0, 0.10, 0.20, 0.40\}$ and $\alpha = 1$. Example 3.1.1.

- $\alpha = 10$

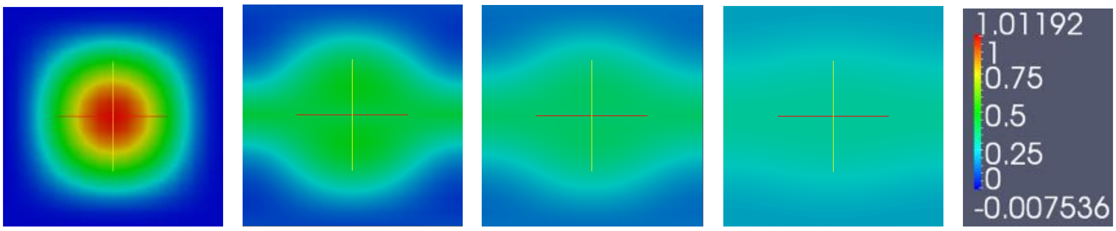

Figure 3.7: Evolution of $\eta(t, x, y)$ for $t = \{0, 0.10, 0.20, 0.40\}$ and $\alpha = 10$. Example 3.1.1

As expected, the case $\alpha = 0.01$ matches better $\eta(T, x, y)$ with $Q^{\eta}(x, y) = 0.25$ respect to the other values of α, as we can see in the next figures:

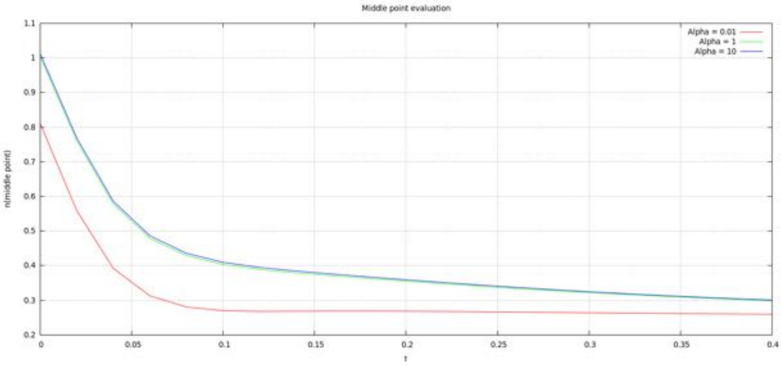

Figure 3.8: Evaluation of $\eta(t, x, y)$ for (x, y) middle point of Ω and $t \in [0, 1]$

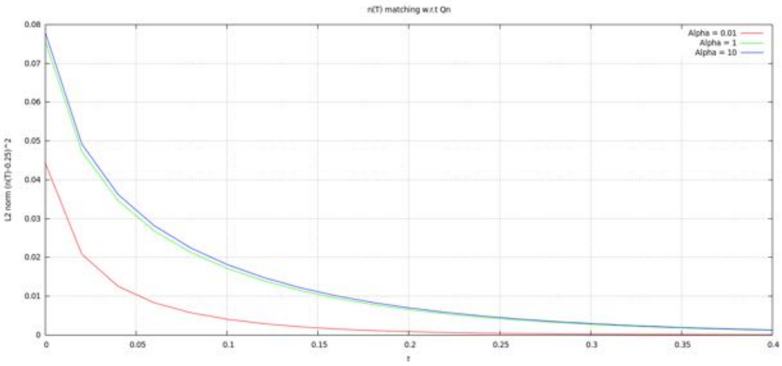

Figure 3.9: $\|\eta(t,x,y)-0.25\|^2_{L^2(\Omega)}$ for $t\in[0,1]$

For the optimal control for $\alpha = 0.01$, we obtain $\|\eta(T,x,y)-0.25\|^2_{L^2(\Omega)} = 6.71\cdot 10^{-5}$, while for α= 1, α= 10, $\|\eta(T,x,y)-0.25\|^2_{L^2(\Omega)}$ achieved bigger values, $1.198\cdot 10^{-3}$ and $1.286\cdot 10^{-3}$ respectively.

In order to have a better understanding of the obtained optimal control q^* with respect to the expected profile Q^d, we examine the adjoint variable at time $t=0$. From (3.11) we know that $z(0)=\alpha(Q^d-q^*)$, i.e. $z(0)$ represents the difference of the expected profile and the optimal control multiplied by α:

- Adjoint at $t=0$ for $\alpha=0.01$

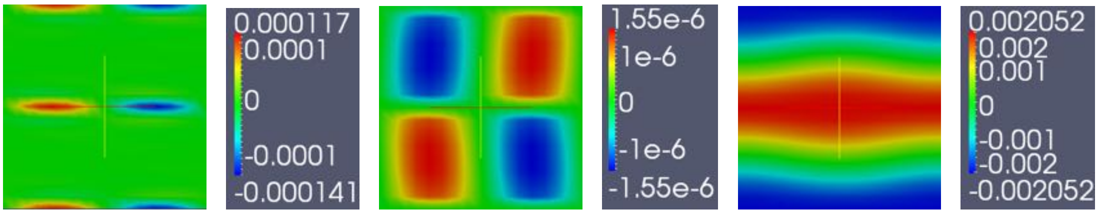

Figure 3.10: Adjoint at $t=0$ for $\alpha=0.01$. Example 3.1.1

- Adjoint at $t=0$ for $\alpha=1$

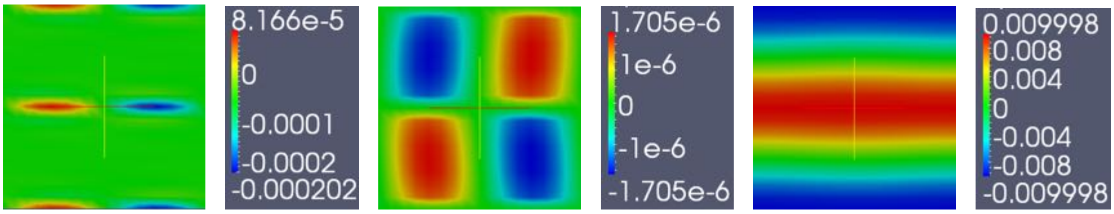

Figure 3.11: Adjoint at $t=0$ for $\alpha=1$. Example 3.1.1.

- Adjoint at $t = 0$ for $\alpha = 10$

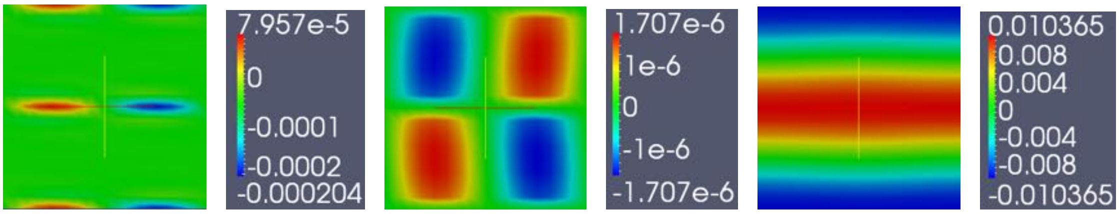

Figure 3.12: Adjoint at $t = 0$ for $\alpha = 10$. Example 3.1.1.

If we set $Q^{\mathbf{v}} \equiv 0^2$ then in $z^{\mathbf{v}}(0)$ we obtain the same pattern of the velocity field (with the opposite sign) that we got for $\alpha = \{0.01, 1, 10\}$. On the other hand, $z^{\eta}(0)$ indicates that the main variation between the optimal initial bump and Q^{η} occurs in the middle of the mesh in the y-direction. In the case of $\alpha = 0.01$, such difference is of order $\mathcal{O}(10^{-1})$, while for the case $\alpha = 1$ and $\alpha = 10$ the difference is of order $\mathcal{O}(10^{-3})$. ($\alpha = \{1, 10\}$ matches better the initial data)

Moreover,the evolution of $z^{\mathbf{v}}(t)$ and $z^{\eta}(t)$ can be observed in the next plots for $\alpha = \{0.01, 1, 10\}$:

- **Evolution of the adjoint variable** $z^{\mathbf{v}_x}(t)$
  - Evolution of the adjoint variable $z^{\mathbf{v}_x}(t)$ for $\alpha = 0.01$:

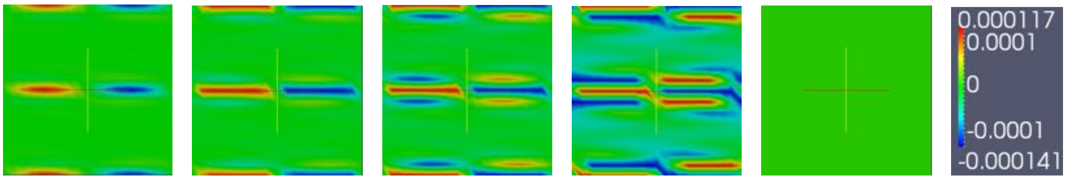

Figure 3.13: Evolution of $z^{\mathbf{v}_x}(t)$ for $t = \{0, 0.10, 0.20, 0.30, 0.40\}$ and $\alpha = 0.01$

  - Evolution of the adjoint variable $z^{\mathbf{v}_x}(t)$ for $\alpha = 1$:

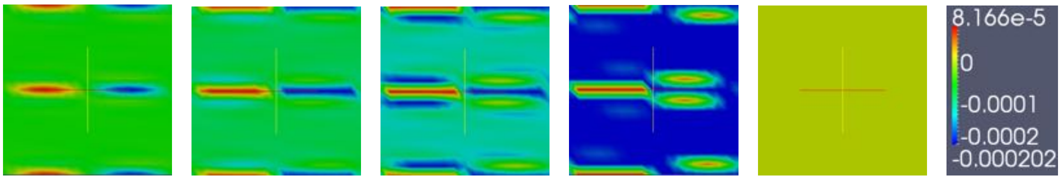

Figure 3.14: Evolution of $z^{\mathbf{v}_x}(t)$ for $t = \{0, 0.10, 0.20, 0.30, 0.40\}$ and $\alpha = 1$

  - Evolution of the adjoint variable $z^{\mathbf{v}_x}(t)$ for $\alpha = 10$:

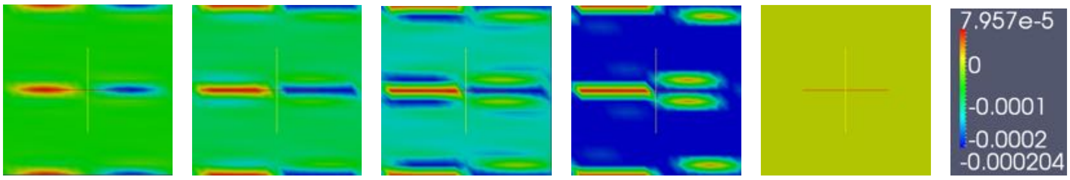

Figure 3.15: Evolution of $z^{\mathbf{v}_x}(t)$ for $t = \{0, 0.10, 0.20, 0.40\}$ and $\alpha = 10$

- **Evolution of the adjoint variable $z^{\mathbf{v}_y}(t)$:**
  - Evolution of the adjoint variable $z^{\mathbf{v}_y}(t)$ for $\alpha = 0.01$:

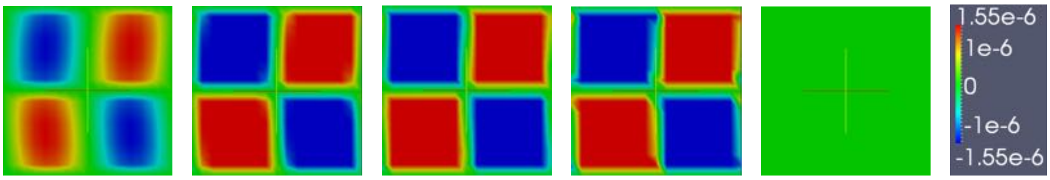

Figure 3.16: Evolution of $z^{\mathbf{v}_y}(t)$ for $t = \{0, 0.10, 0.20, 0.30, 0.40\}$ and $\alpha = 0.01$

  - Evolution of the adjoint variable $z^{\mathbf{v}_y}(t)$ for $\alpha = 1$:

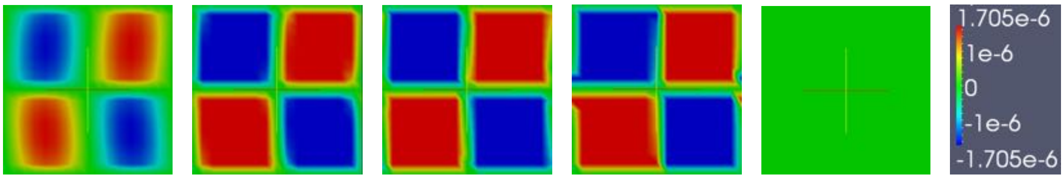

Figure 3.17: Evolution of $z^{\mathbf{v}_y}(t)$ for $t = \{0, 0.10, 0.20, 0.30, 0.40\}$ and $\alpha = 1$

  - Evolution of the adjoint variable $z^{\mathbf{v}_y}(t)$ for $\alpha = 10$:

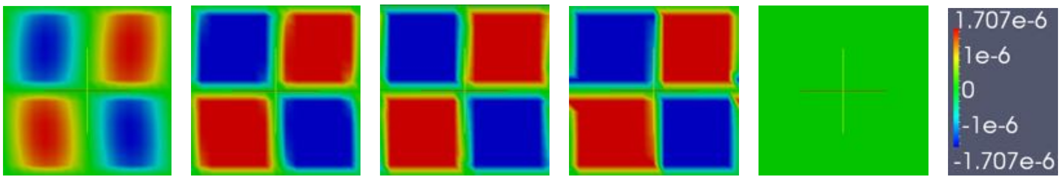

Figure 3.18: Evolution of $z^{\mathbf{v}_y}(t)$ for $t = \{0, 0.10, 0.20, 0.40\}$ and $\alpha = 10$

- **Evolution of the adjoint variable $z^{\eta}(t)$:**
    - Evolution of the adjoint variable $z^{\eta}(t)$ for $\alpha = 0.01$:

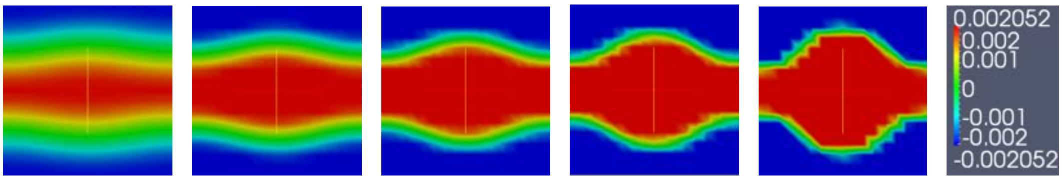

Figure 3.19: Evolution of $z^{\eta}(t)$ for $t = \{0, 0.10, 0.20, 0.30, 0.40\}$ and $\alpha = 0.01$

- Evolution of the adjoint variable $z^{\eta}(t)$ for $\alpha = 1$:

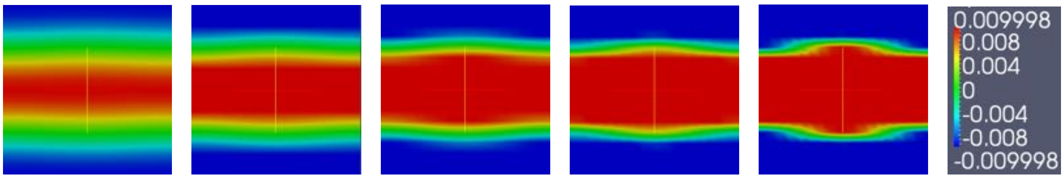

Figure 3.20: Evolution of $z^{\eta}(t)$ for $t = \{0, 0.10, 0.20, 0.30, 0.40\}$ and $\alpha = 1$

- Evolution of the adjoint variable $z^{\eta}(t)$ for $\alpha = 10$:

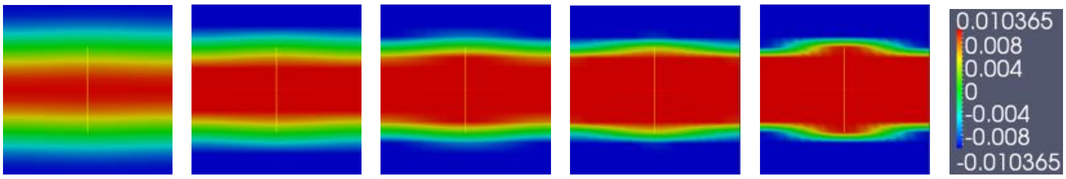

Figure 3.21: Evolution of $z^{\eta}(t)$ for $t = \{0, 0.10, 0.20, 0.40\}$ and $\alpha = 10$

As we can see in the figures 3.13, 3.14, 3.15, 3.16, 3.17 and 3.18, z^{v} vanishes for the different values of α at final time T. This is due to the choice $c_0 = 0$ and the definition of the initial condition in the adjoint equation, $z^{\mathrm{v}}(T) = c_0(\mathbf{v}(T) - \mathbf{v}_d) = 0^2 \in \Omega$. On the other hand, the initial condition $z^{\eta}(T) = \eta(T) - \eta_d \in \Omega$ is the difference between the height obtained at final time T and the expected height profile $\eta^d = 0.25$. We notice that in this case, $z^{\eta}(T)$ has a similar pattern as $z^{\eta}(0)$, as we can see in the figures 3.19, 3.20 and 3.21. However, for $\alpha = 0.01$ and $\alpha = 1$ such difference is of order $\mathcal{O}(10^{-3})$ and for $\alpha = 10$ is $\mathcal{O}(10^{-2})$ (for smaller values of α, $\eta(T)$ matches better with η^d).

3.1.2 Advection of a quasistationary vortex for $\alpha = \{0.01, 1, 10\}$

The objective of this example is to compare the optimal control q^* with $Q^d(x,y) = (Q^{\eta}(x,y), Q^{\mathrm{v}}(x,y))$ defined as $Q^{\mathrm{v}}(x,y) = (0,0)$ and $Q^{\eta}(x,y) = \eta^{(47,50)}(x,y)$, where

$\eta^{(47,50)}(x,y)$ is the vortex defined in Section 2.4.4 centered in the point $(47, 50)$. Additionally, we set $u^d(x,y) = (0, 0, \eta^{(50,50)}(x,y))$, $\alpha = \{0.01, 1, 10\}$, $c_0 = 0$, $c_1 = 1$, $T = 0.05$ (discretized in 50 time steps), $\varepsilon = A_{visc} = 0$, the Coriolis force $f(x,y) = 0$ and the domain $\Omega = [0, 100]^2$ discretized in a 256 elements mesh (modified mesh with parameters $m = 1.60$, $a_1 = 10$ and $a_2 = 90$).

As in the previous example, we want to understand the behavior of the initial velocity field so that given an initial topography $Q^\eta(x,y) = \eta^{(47,50)}(x,y)$, we obtain a final topography $\eta(T,x,y) = \eta^{(50,50)}(x,y)$ at final time $T = 0.05$.

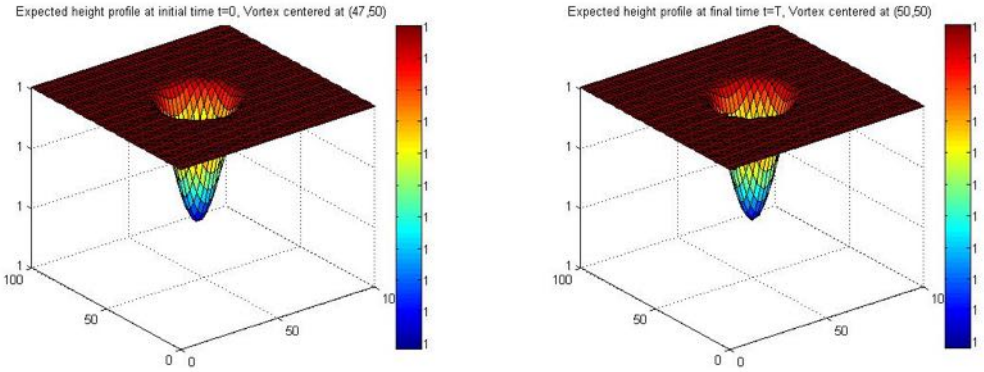

Figure 3.22: Expected initial and final topography. Example 3.1.2.

The Newton method for the reduced problem converged in 10 iterations for $\alpha = 0.01$, 8 iterations for $\alpha = 1$ and 5 iterations for $\alpha = 10$. In the next tables we can see the results for the method in each iteration:

$\alpha = 0.01$, CPU time = 24061.13sec.				
Iteration	**Cost Functional**	**Residual(rel)**	**LinearIters**	**Line search**
0	5e+3	1	-	-
1	6.03e-01	1.52e-01	2	0
2	8.96e-02	5.46e-02	3	0
3	9.57e-03	1.27e-02	4	0
4	1.13e-03	3.84e-03	6	0
5	1.82e-04	1.20e-03	6	0
6	4.18e-05	3.89e-04	8	0
7	1.36e-05	7.52e-05	18	0
8	1.76e-07	6.45e-06	214	0
9	2.20e-10	1.74e-07	-1	0
10	9.33e-11	1.46e-08	-1	0

Table 3.7: Newton Iterations for $\alpha = 0.01$. Example 3.1.2.

$\alpha = 1$, , CPU time = 7862.27sec.				
Iteration	**Cost Functional**	**Residual(rel)**	**LinearIters**	**Line search**
0	1e+4	1	-	-
1	4.05e-02	1.98e-02	2	0
2	6.07e-03	7.18e-03	3	0
3	6.50e-04	1.67e-03	4	0
4	7.61e-05	5.07e-04	6	0
5	1.15e-05	1.58e-04	6	0
6	4.94e-07	2.27e-05	14	0
7	1.53e-08	1.48e-06	32	0
8	4.72e-11	2.52e-08	320	0

Table 3.8: Newton Iterations for $\alpha = 1$. Example 3.1.2.

$\alpha = 10$, CPU time = 1149.43sec.				
Iteration	**Cost Functional**	**Residual(rel)**	**LinearIters**	**Line search**
0	5.5e+4	1	-	-
1	5.82e-05	1.35e-04	2	0
2	9.32e-06	5.13e-05	3	0
3	4.94e-07	1.12e-05	5	0
4	2.01e-08	1.11e-06	6	0
5	3.07e-10	3.88e-08	34	0

Table 3.9: Newton Iterations for $\alpha = 10$. Example 3.1.2.

As in example 3.1.1, we notice that for smaller values of α the optimization problem is "harder" to solve, the CPU time for $\alpha = 1$ and 10 is 32.67% and 4.77% of the CPU time for $\alpha = 0.01$. In the following plots we can see the optimal controls for the different values of α:

- Optimal control for $\alpha = 0.01$

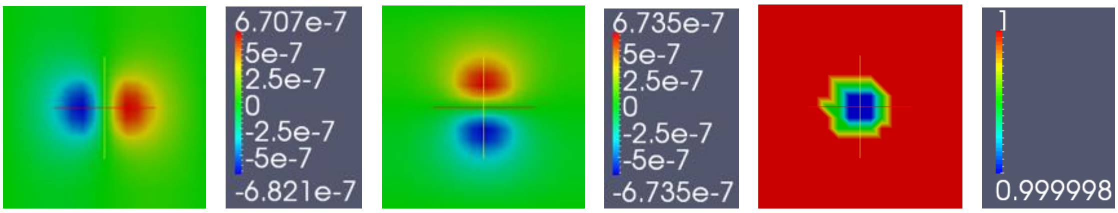

Figure 3.23: Optimal control $(u_0, v_0), \eta_0$ for $\alpha = 0.01$. Example 3.1.2.

- Optimal control for $\alpha = 1$

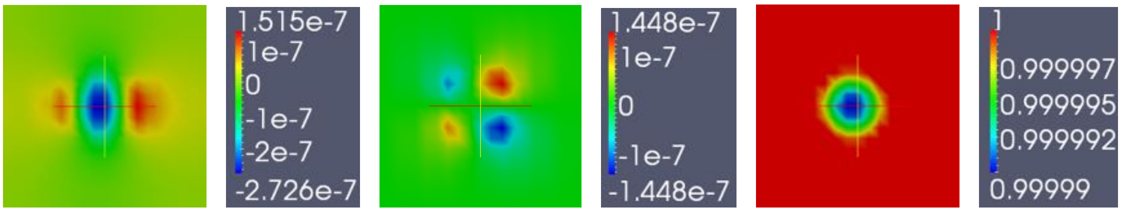

Figure 3.24: Optimal control $(u_0, v_0), \eta_0$ for $\alpha = 1$. Example 3.1.2.

- Optimal control for $\alpha = 10$

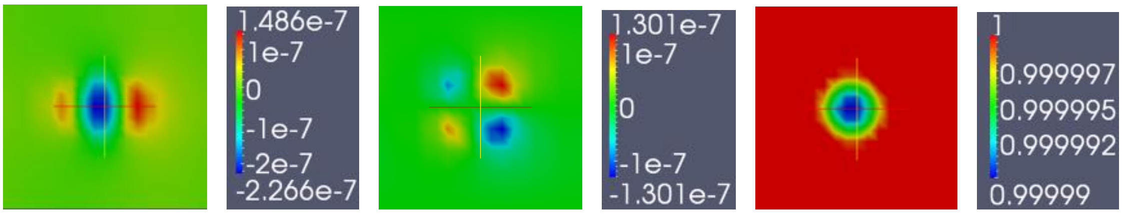

Figure 3.25: Optimal control $(u_0, v_0), \eta_0$ for $\alpha = 10$. Example 3.1.2

As in the previous example, we see that one of the main difference in the optimal control for $\alpha = \{0.01, 1, 10\}$ occurs in the optimal $\eta_0(x, y)$ for $\alpha = 0.01$. In this case, the optimal solution differs from the initial bump centered at $(47, 50)$ (see Table 3.10).

α	$\|q^* - Q^d\|^2_{L^2(\Omega)}$
0.01	$1.62 \cdot 10^{-8}$
1	$3.996 \cdot 10^{-11}$
10	$2.86 \cdot 10^{-11}$

Table 3.10: α vs $\|q^* - Q^d\|^2_{L^2(\Omega)}$. Example 3.1.2

For $\alpha = \{1, 10\}$, we observe small disturbances (of order $\mathcal{O}(10^{-7})$) for the velocity field in both directions. For $\alpha = 0.01$, we do not see such disturbances. However, in this case the initial "vortex" does not seem to be centered at $(47, 50)$ (see Figure 3.23 and Table 3.10), the center is closer to $(50, 50)$ and therefore the optimal initial velocity is also centered in this sector (see Figure 3.23).

Considering the optimal control that we obtained, the initial vortexes exhibit the following behavior in time:

- $\alpha = 0.01$

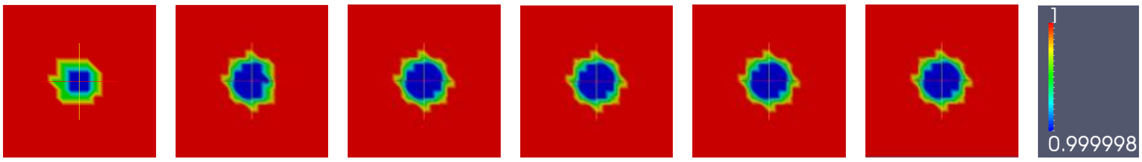

Figure 3.26: Evolution of $\eta(t,x,y)$ for $t = \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$ and $\alpha = 0.01$

- $\alpha = 1$

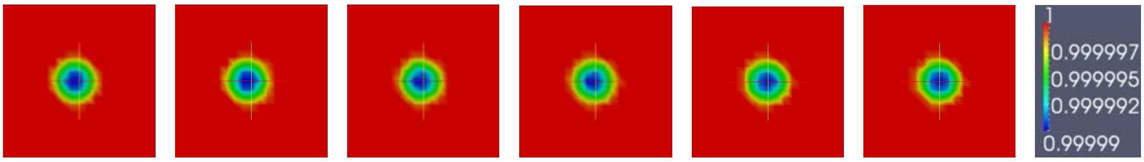

Figure 3.27: Evolution of $\eta(t,x,y)$ for $t = \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$ and $\alpha = 1$

- $\alpha = 10$

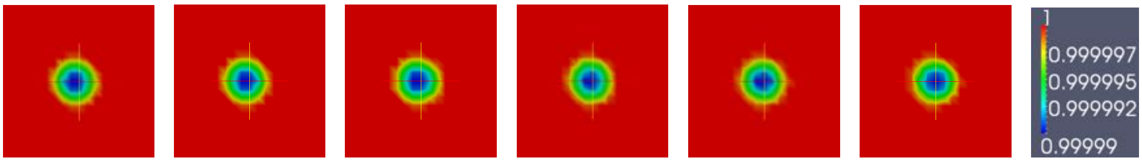

Figure 3.28: Evolution of $\eta(t,x,y)$ for $t = \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$ and $\alpha = 10$

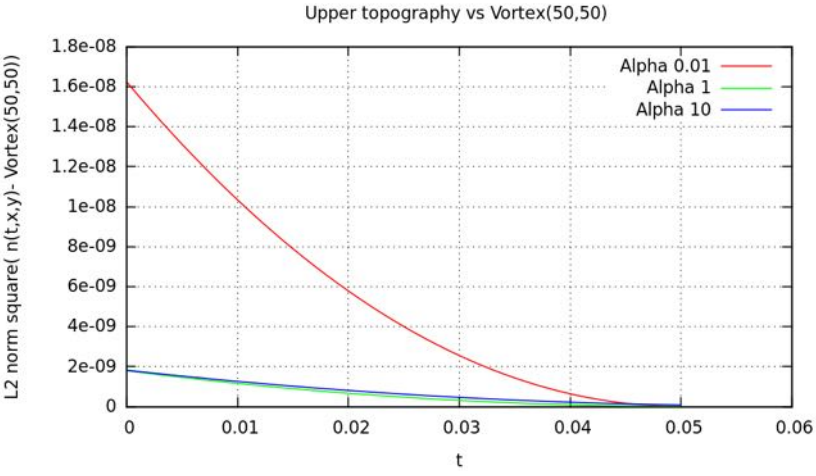

Figure 3.29: $\|\eta(t) - \eta^{(50,50)}\|^2_{L^2(\Omega)}$ for $\alpha = \{0.01, 1, 10\}$

α	$\|\eta(t) - \eta^{(50,50)}\|^2_{L^2(\Omega)}$
0.01	$1.71 \cdot 10^{-11}$
1	$1.86 \cdot 10^{-11}$
10	$9.02 \cdot 10^{-11}$

Table 3.11: α vs $\|\eta(t) - \eta^{(50,50)}\|^2_{L^2(\Omega)}$. Example 3.1.2

As expected, the solution for $\alpha = 0.01$ adjusts better $\eta(T)$ to $\eta^{(50,50)}$. Additionally, the optimal control for $\alpha = 10$ fits better the optimal control q^* with respect to the expected profile Q^d (see Table 3.10). For this problem, it seems appropriate to fix $\alpha = \{1, 10\}$, since $\alpha = 0.01$ does not improve significantly the matching of $\eta(T)$ vs $\eta^{(50,50)}$ (see Table 3.11), at the same time having an unsatisfactory performance when matching the optimal control q^* with Q^d.

The evolution of $z^{\mathbf{v}}(t)$ and $z^{\eta}(t)$ can be observed in the next plots for $\alpha = \{0.01, 1, 10\}$:

- **Evolution of the adjoint variable $z^{\mathbf{v}_x}(t)$:**
  - Evolution of the adjoint variable $z^{\mathbf{v}_x}(t)$ for $\alpha = 0.01$:

Figure 3.30: Evolution of $z^{\mathbf{v}_x}(t)$ for $t = \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$ and $\alpha = 0.01$

  - Evolution of the adjoint variable $z^{\mathbf{v}_x}(t)$ for $\alpha = 1$:

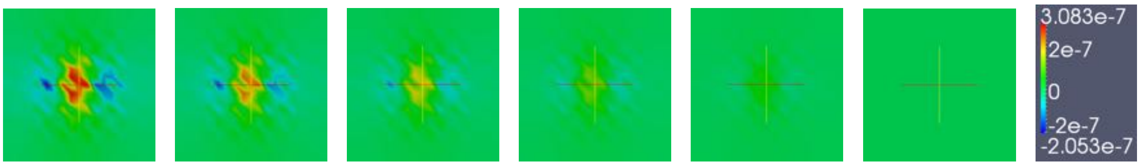

Figure 3.31: Evolution of $z^{\mathbf{v}_x}(t)$ for $t = \{0, 0.10, 0.20, 0.30, 0.40, 0.50\}$ and $\alpha = 1$

  - Evolution of the adjoint variable $z^{\mathbf{v}_x}(t)$ for $\alpha = 10$:

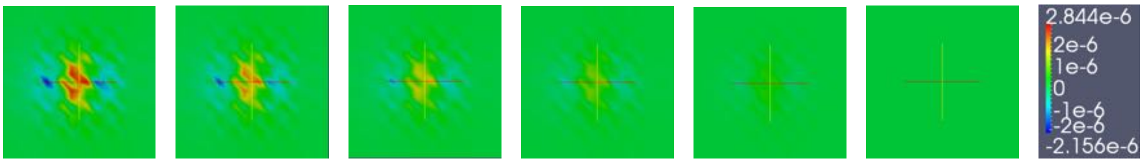

Figure 3.32: Evolution of $z^{\mathbf{v}_x}(t)$ for $t = \{0, 0.10, 0.20, 0.40, 0.50\}$ and $\alpha = 10$

- **Evolution of the adjoint variable $z^{\mathbf{v}_y}(t)$:**
  - Evolution of the adjoint variable $z^{\mathbf{v}_y}(t)$ for $\alpha = 0.01$:

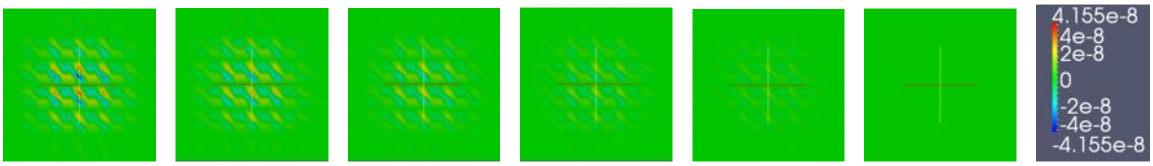

Figure 3.33: Evolution of $z^{\mathbf{v}_y}(t)$ for $t = \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$ and $\alpha = 0.01$

- Evolution of the adjoint variable $z^{\mathbf{v}_y}(t)$ for $\alpha = 1$:

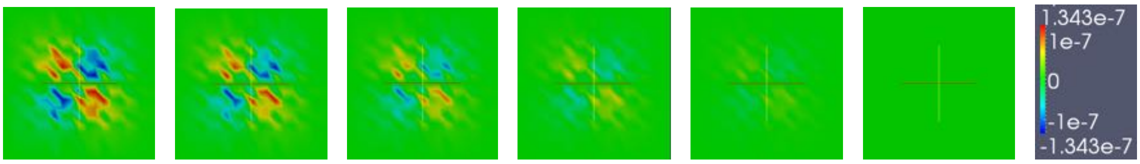

Figure 3.34: Evolution of $z^{\mathbf{v}_y}(t)$ for $t = \{0, 0.10, 0.20, 0.30, 0.40, 0.50\}$ and $\alpha = 1$

- Evolution of the adjoint variable $z^{\mathbf{v}_y}(t)$ for $\alpha = 10$:

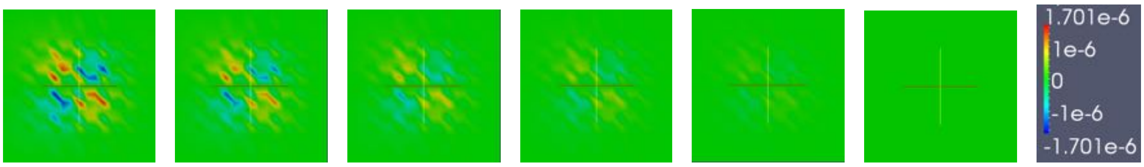

Figure 3.35: Evolution of $z^{\mathbf{v}_y}(t)$ for $t = \{0, 0.10, 0.20, 0.40, 0.50\}$ and $\alpha = 10$

○ **Evolution of the adjoint variable $z^{\eta}(t)$:**

- Evolution of the adjoint variable $z^{\eta}(t)$ for $\alpha = 0.01$:

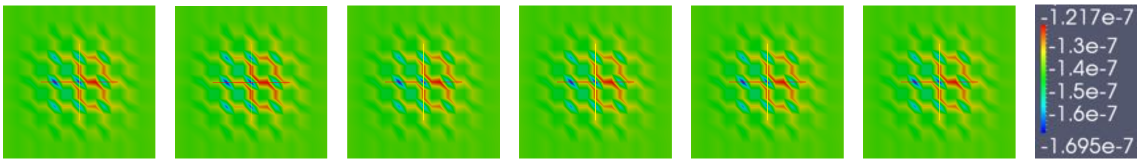

Figure 3.36: Evolution of $z^{\eta}(t)$ for $t = \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$ and $\alpha = 0.01$

- Evolution of the adjoint variable $z^{\eta}(t)$ for $\alpha = 1$:

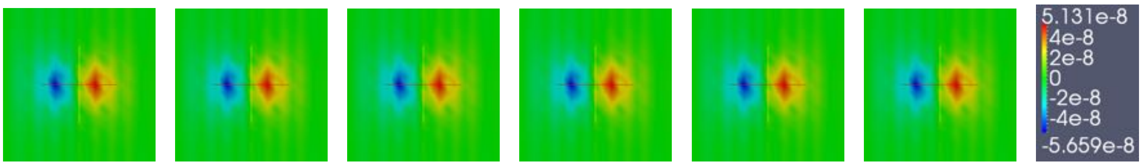

Figure 3.37: Evolution of $z^{\eta}(t)$ for $t = \{0, 0.10, 0.20, 0.30, 0.40, 0.50\}$ and $\alpha = 1$

- Evolution of the adjoint variable $z^{\eta}(t)$ for $\alpha = 10$:

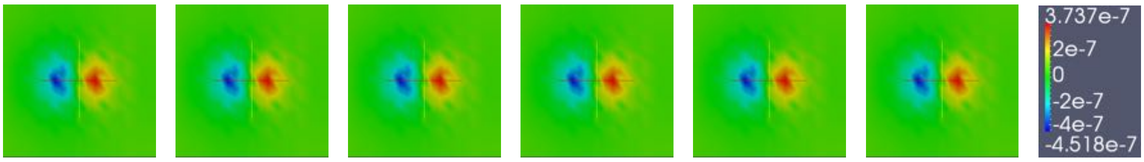

Figure 3.38: Evolution of $z^{\eta}(t)$ for $t = \{0, 0.10, 0.20, 0.40, 0.50\}$ and $\alpha = 10$

As in the previous case, since $z(0) = \alpha(Q^d - q^*)$ and we set $Q^{\mathbf{v}} = 0^2$, we expect $z^{\mathbf{v}}(0) = (z^{\mathbf{v}_x}(0), z^{\mathbf{v}_y}(0))$ to have the same pattern as the optimal initial velocity (multiplied by $-\alpha$). Again, we observe that $z^{\mathbf{v}}(t)$ vanishes in time till achieving $z^{\mathbf{v}}(T) = 0 \in \Omega$, this is due to the choice of $c_0 = 0$. Furthermore, for the different values of α, we observe a quasistationary behavior of $z^{\eta}(t)$ for $t \in [0, T]$, this means that both differences $z(0) = \alpha(Q^d - q^*)$ and $z^{\eta}(T) = \eta(T) - \eta^d$, vary in two quasi symmetric areas respect to $l_x = x - 50$. For the initial time (after dividing by α) such difference is of order $\mathcal{O}(10^{-8})$ for both $\alpha = \{1, 10\}$ and $\mathcal{O}(10^{-5})$ for $\alpha = 0.01$. (The initial data is adjusted better for bigger values of α).

3.1.3 Advection of a quasistationary vortex for different levels of refinement: $level = \{3, 4, 5\}$

This example has the same settings as Example 3.1.2. However, in this case we fix $\alpha = 1$ and we variate the level of refinements of the mesh: *level 3, level 4, level 5*, in order to verify the solution that we obtained in the last example as well as the possible spatial convergence of the optimal control q^*.

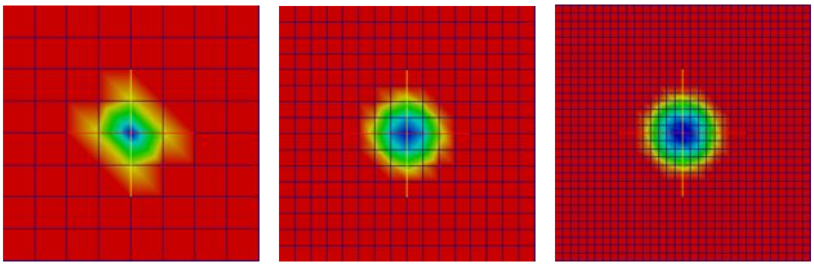

Figure 3.39: Q^d for a mesh of $level = 3, 4, 5$ of refinement

The Newton method for the reduced problem converged in 6 iterations for $level = 3$, 8 iterations for $level = 4$ and 10 iterations for $level = 5$. In the next tables we can see the results for the method in each iteration:

$level = 3$, CPU time = 659.06 sec				
Iteration	**Cost Functional**	**Residual(rel)**	**LinearIters**	**Line search**
0	1e+4	1	-	-
1	3.36e-03	3.25e-03	2	0
2	5.57e-04	1.19e-03	2	0
3	2.68e-05	2.33e-04	3	0
4	9.04e-07	3.68e-05	5	0
5	9.61e-09	1.93e-06	8	0
6	6.68e-10	3.64e-08	106	0

Table 3.12: Newton Iterations for $level = 3$. Example 3.1.3.

$level = 4$. CPU time = 10080.71 sec.				
Iteration	**Cost Functional**	**Residual(rel)**	**LinearIters**	**Line search**
0	1e+4	1	-	-
1	2.99e-02	1.52e-02	2	0
2	4.28e-03	5.46e-03	3	0
3	4.19e-04	1.29e-03	4	0
4	1.14e-04	6.20e-04	4	0
5	1.21e-05	1.65e-04	6	0
6	4.96e-07	2.31e-05	12	0
7	1.12e-08	1.39e-06	26	0
8	6.89e-11	2.27e-08	240	0

Table 3.13: Newton Iterations for $level = 4$. Example 3.1.3.

$level = 5$. CPU time = 86760.82 sec.				
Iteration	**Cost Functional**	**Residual(rel)**	**LinearIters**	**Line search**
0	1e+4	1	-	-
1	7.80e-02	3.36e-02	2	0
2	1.73e-02	1.40e-02	3	0
3	4.29e-03	5.30e-03	4	0
4	7.88e-04	2.00e-03	6	0
5	1.89e-4	8.37e-04	6	0
6	3.54e-05	3.12e-04	8	0
7	2.63e-06	6.27e-05	14	0
8	1.28e-07	6.98e-06	32	0
9	7.38e-10	2.60e-07	304	0
10	2.93e-11	2.34e-08	-1	0

Table 3.14: Newton Iterations for $level = 5$. Example 3.1.3.

In the following plots we can see the optimal controls for the different values of *level*:

- Optimal control for $level = 3$

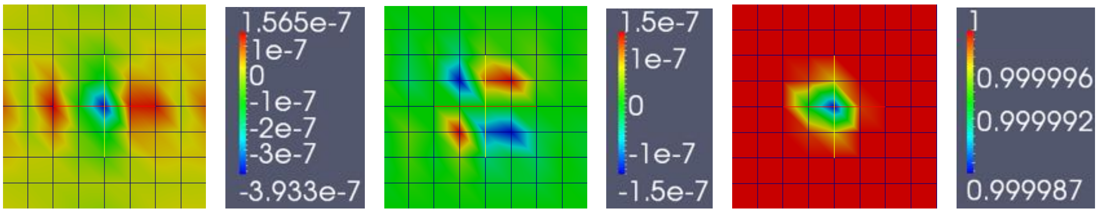

Figure 3.40: Optimal control $(u_0, v_0), \eta_0$ for $level = 3$. Example 3.1.3.

- Optimal control for $level = 4$

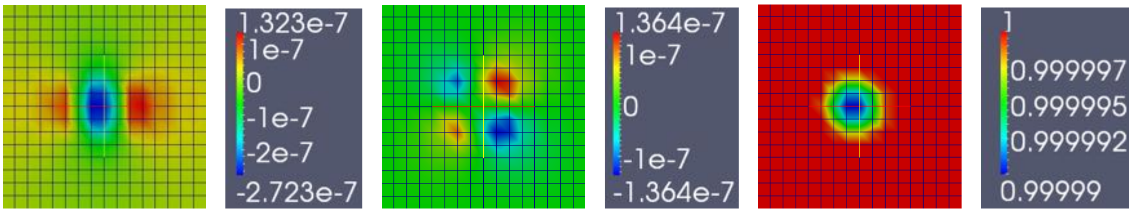

Figure 3.41: Optimal control $(u_0, v_0), \eta_0$ for $level = 4$. Example 3.1.3.

- Optimal control for $level = 5$

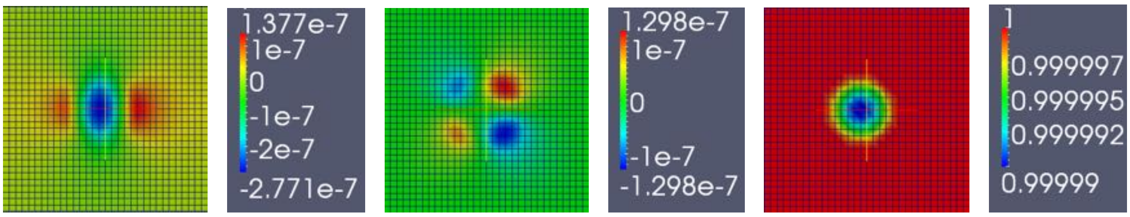

Figure 3.42: Optimal control $(u_0, v_0), \eta_0$ for $level = 5$. Example 3.1.3.

For the different levels of refinement we observe the same behavior (and possible convergence) of the optimal control q^* to the pattern shown in Figure 3.42. For *level* 3 the CPU time = 659.06 sec represents 0.07% of the CPU time for $level = 5$ and for *level* 4 we had 10080.71 sec (11.61% of the CPU time for $level = 5$). For the different levels of refinement (specially for $level = \{4, 5\}$) the optimal initial height matches $\eta^{(47,50)}$ (see Table 3.15):

Level	$\|q^* - Q^d\|^2_{L^2(\Omega)}$
3	$8.933 \cdot 10^{-10}$
4	$5.722 \cdot 10^{-11}$
5	$1.953 \cdot 10^{-11}$

Table 3.15: *level* vs $\|q^* - Q^d\|^2_{L^2(\Omega)}$. Example 3.1.3

Considering the optimal controls that we obtained, the initial vortexes exhibit the following behavior in time:

- $level = 3$

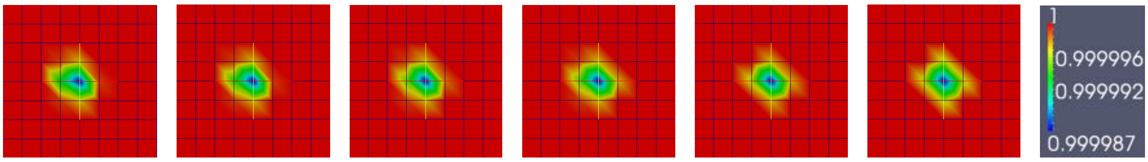

Figure 3.43: Evolution of $\eta(t, x, y)$ for $t = \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$ and $level = 3$

- $level = 4$

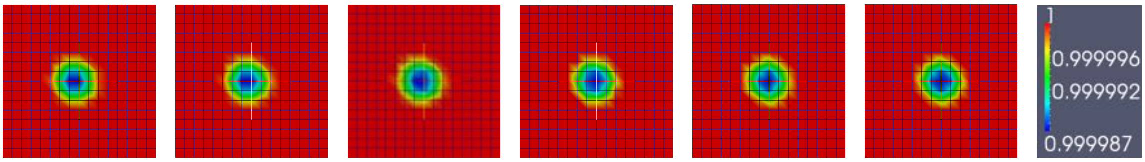

Figure 3.44: Evolution of $\eta(t,x,y)$ for $t = \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$ and $level = 4$

- $level = 5$

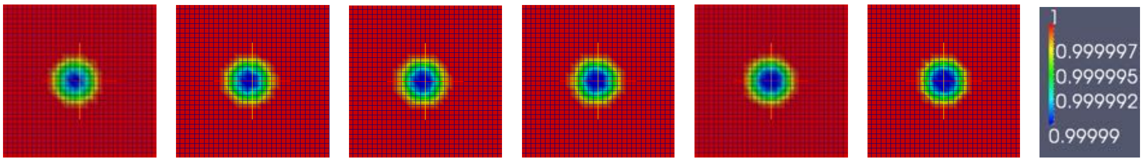

Figure 3.45: Evolution of $\eta(t,x,y)$ for $t = \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$ and $level = 5$

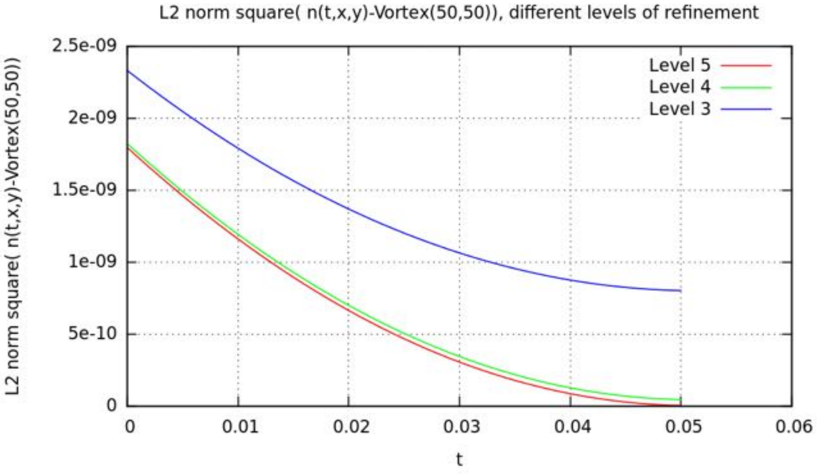

Figure 3.46: $\|\eta(t,x,y) - \eta^{(50,50)}\|^2_{L^2(\Omega)}$ for different levels of refinement

As we verified in the last example, the achieved optimal control allows a vortex of the form $\eta^{(47,50)}$ to approximate $\eta^{(50,50)}$ at time $T = 0.05$. As we reduce the mesh, we eliminate the "noise" in the boundary of the vortex as we can see in Figures 3.44 and 3.45. At the same time, the refinement of the mesh allows us to observe better the matching of $\eta(t,x,y)$ with the $\eta^{(50,50)}$. Besides the level of refinement, we observe a smooth decay to the expected height profile (See Table 3.16 , Figure 3.46).

Level	$\|\eta(T)-\eta^d\|^2_{L^2(\Omega)}$
3	$8.023\cdot 10^{-10}$
4	$4.465\cdot 10^{-11}$
5	$3.438\cdot 10^{-12}$

Table 3.16: *level* vs $\|\eta(T)-\eta^d\|^2_{L^2(\Omega)}$. Example 3.1.3

Moreover,the evolution of $z^{\mathbf{v}}(t)$ and $z^{\eta}(t)$ can be observed in the next plots for $\alpha = \{0.01, 1, 10\}$:

- ○ **Evolution of the adjoint variable $z^{\mathbf{v}_x}(t)$:**
  - Evolution of the adjoint variable $z^{\mathbf{v}_x}(t)$ for *level* $= 3$:

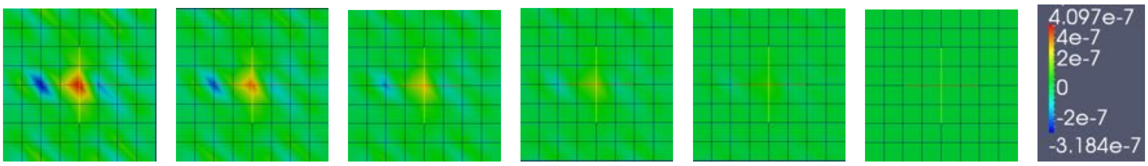

Figure 3.47: Evolution of $z^{\mathbf{v}_x}(t)$ for $t = \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$ and *level* $= 3$

  - Evolution of the adjoint variable $z^{\mathbf{v}_x}(t)$ for *level* $= 4$:

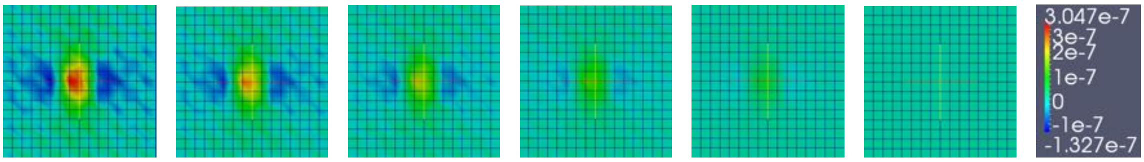

Figure 3.48: Evolution of $z^{\mathbf{v}_x}(t)$ for $t = \{0, 0.10, 0.20, 0.30, 0.40, 0.50\}$ and *level* $= 4$

  - Evolution of the adjoint variable $z^{\mathbf{v}_x}(t)$ for *level* $= 5$:

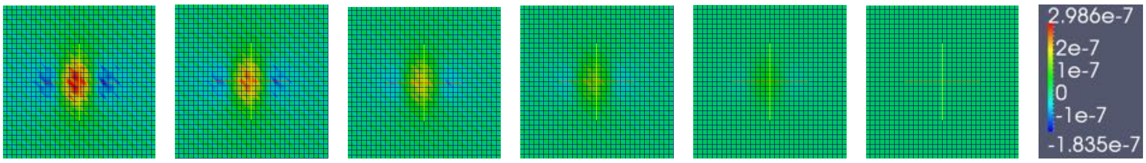

Figure 3.49: Evolution of $z^{\mathbf{v}_x}(t)$ for $t = \{0, 0.10, 0.20, 0.40, 0.50\}$ and *level* $= 5$

- ○ **Evolution of the adjoint variable $z^{\mathbf{v}_y}(t)$:**
  - Evolution of the adjoint variable $z^{\mathbf{v}_y}(t)$ for *level* $= 3$:

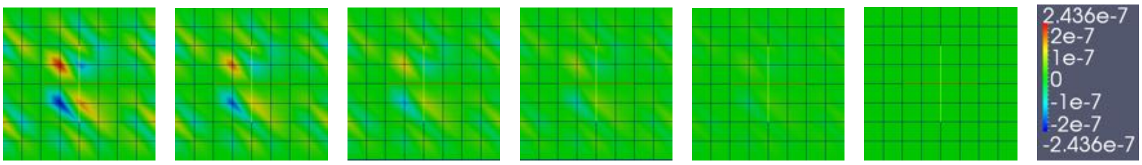

Figure 3.50: Evolution of $z^{\mathbf{v}_y}(t)$ for $t = \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$ and $level = 3$

- Evolution of the adjoint variable $z^{\mathbf{v}_y}(t)$ for $level = 4$:

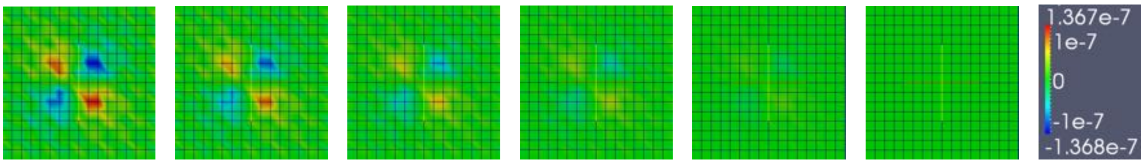

Figure 3.51: Evolution of $z^{\mathbf{v}_y}(t)$ for $t = \{0, 0.10, 0.20, 0.30, 0.40, 0.50\}$ and $level = 4$

- Evolution of the adjoint variable $z^{\mathbf{v}_y}(t)$ for $level = 5$:

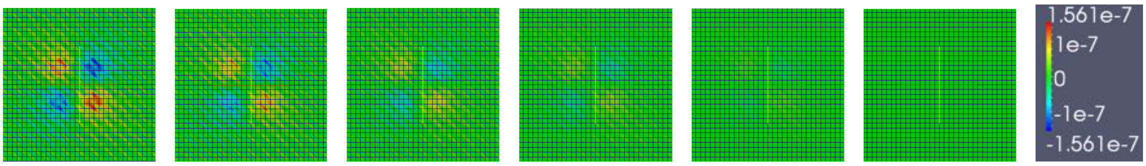

Figure 3.52: Evolution of $z^{\mathbf{v}_y}(t)$ for $t = \{0, 0.10, 0.20, 0.40, 0.50\}$ and $level = 5$

- **Evolution of the adjoint variable $z^{\eta}(t)$:**
  - Evolution of the adjoint variable $z^{\eta}(t)$ for $level = 3$:

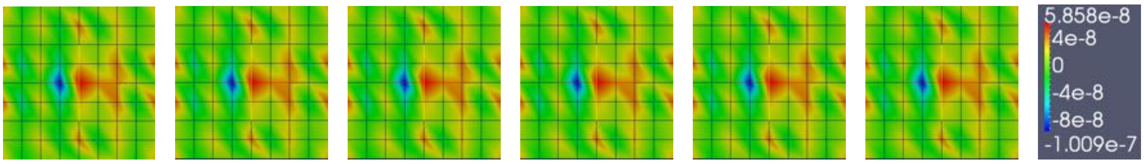

Figure 3.53: Evolution of $z^{\eta}(t)$ for $t = \{0, 0.01, 0.02, 0.03, 0.04, 0.05\}$ and $level = 3$

- Evolution of the adjoint variable $z^{\eta}(t)$ for $level = 4$:

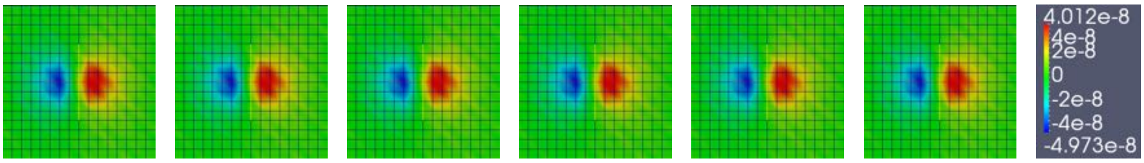

Figure 3.54: Evolution of $z^{\eta}(t)$ for $t = \{0, 0.10, 0.20, 0.30, 0.40, 0.50\}$ and $level = 4$

- Evolution of the adjoint variable $z^{\eta}(t)$ for $level = 5$:

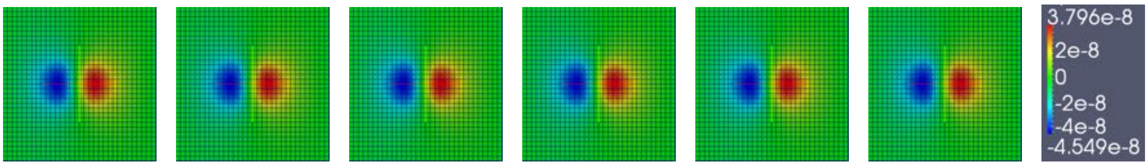

Figure 3.55: Evolution of $z^{\eta}(t)$ for $t = \{0, 0.10, 0.20, 0.40, 0.50\}$ and $level = 5$

We can see that for the different levels of refinement we obtained the same pattern for $z^{\mathbf{v}}(t)$. However, for $level = 3$ the error of such approximation is of order $\mathcal{O}(10^{-5})$, while for $level = 4$ the order is $\mathcal{O}(10^{-7})$ and for $level = 5$ of $\mathcal{O}(10^{-8})$. On the other hand, $z^{\eta}(t)$ for $level = 3$ has a non-symmetric pattern with oscillations outside central area, which is not supposed to happen since the dynamic in this sector is nonexistent. The refinement of this mesh shows the possible convergence of the control q to the pattern obtained in *level* 5.

3.1.4 Double bump with different final time $T = \{0.50, 0.75, 1\}$

We want to compare the optimal control q^* for $Q^d(x,y) = (Q^{\eta}(x,y), Q^{\mathbf{v}}(x,y))$ defined as $Q^{\mathbf{v}}(x,y) = (0,0)$ and $Q^{\eta}(x,y) = \sin(2\pi x) + \sin(2\pi y)$. Additionally, we set $u^d(x,y) = (0,0,0)$, $\alpha = 1$, $c_0 = 1$, $c_1 = 1$, $\varepsilon = A_{visc} = 0.1$, the Coriolis force $f(x,y) = 4 \cdot 86164.1 \cdot \pi \sin(y)$ and the domain $\Omega = [0,1]^2$ discretized in a 256 elements mesh.

We want to determine the behavior of the initial velocity field so that given an initial topography $Q^{\eta}(x,y) = \sin(2\pi x) + \sin(2\pi y)$, we obtain the topography $\eta(T,x,y) = 0$ at final time $T = \{0.50, 0.75, 1\}$. For $T = 0.50$ we take 100 time steps, for $T = 0.75$ we take 150 time steps and for $T = 1$ we take 200 time steps.

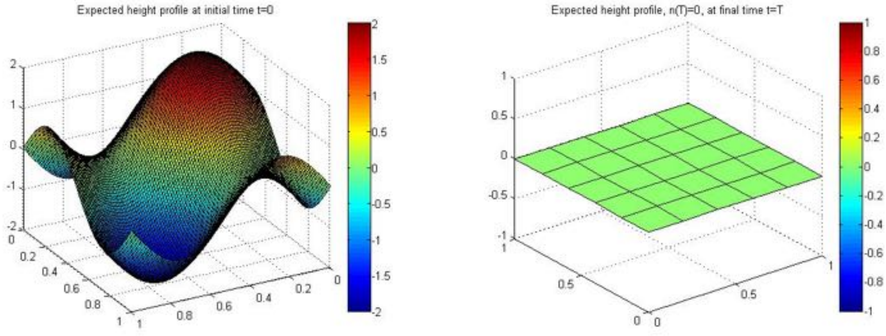

Figure 3.56: Expected initial and final topography. Example 3.1.4

The Newton method for the reduced problem converged in 3 iterations for $T = 0.50$ and $T = 0.75$ and in 2 iterations for $T = 1$. In the next tables we can see the results for the method in each iteration:

$T = 0.50$, CPU time = 95.44 sec				
Iteration	**Cost Functional**	**Residual(rel)**	**LinearIters**	**Line search**
0	0.5	1	-	-
1	5.10e-04	1.83e-03	2	0
2	5.09e-04	3.37e-06	2	0
3	5.09e-04	2.36e-08	3	0

Table 3.17: Newton Iterations for $T = 0.50$. Example 3.1.4.

$T = 0.75$, CPU time = 144.46 sec				
Iteration	**Cost Functional**	**Residual(rel)**	**LinearIters**	**Line search**
0	0.5	1	-	-
1	3.78e-05	8.40e-05	2	0
2	3.78e-05	2.18e-07	2	0
3	3.78e-05	3.32e-08	2	0

Table 3.18: Newton Iterations for $T = 0.75$. Example 3.1.4.

$T = 1$. CPU time = 510.57 sec				
Iteration	**Cost Functional**	**Residual(rel)**	**LinearIters**	**Line search**
0	0.5	1	-	-
1	1.80e-05	3.74e-06	2	0
2	1.80e-05	1.26e-08	3	0

Table 3.19: Newton Iterations for $T = 1$. Example 3.1.4.

In the following plots we can see the optimal controls for the different values of T:

- Optimal control for $T = 0.50$

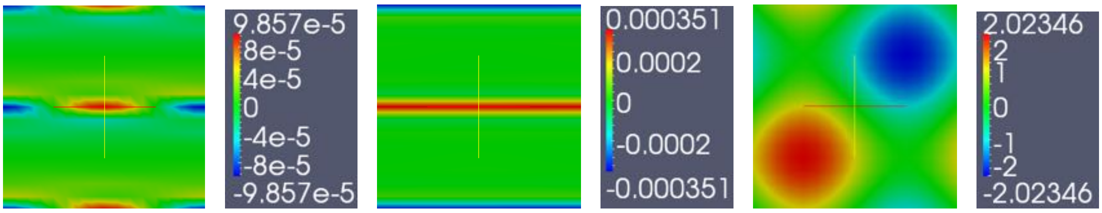

Figure 3.57: Optimal control $(u_0, v_0), \eta_0$ for $T = 0.50$. Example 3.1.4.

- Optimal control for $T = 0.75$

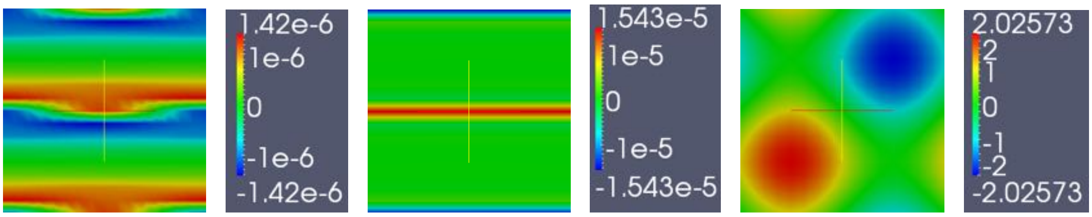

Figure 3.58: Optimal control $(u_0, v_0), \eta_0$ for $T = 0.75$. Example 3.1.4.

- Optimal control for $T = 1$

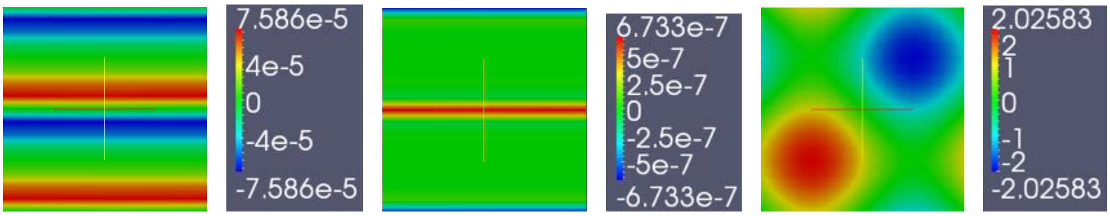

Figure 3.59: Optimal control $(u_0, v_0), \eta_0$ for $T = 1$. Example 3.1.4.

In this example, the main difference in the optimal control for the different values of T is in the velocity field. We can observe that for $T = 0.50$, the velocity field in the x direction keeps the same pattern as the one that we discussed in Chapter 2. The velocity in this direction seems to have a slightly bigger norm than for $T = 0.75$ and $T = 1$. Additionally, the velocity in the y direction keeps the same pattern for all the cases. However, for $T = 0.5$, it is of order $\mathcal{O}(10^{-3})$, while for $T = 0.75$ it is order $\mathcal{O}(10^{-5})$ and for $T = 1$ is $\mathcal{O}(10^{-7})$. This might be due to fact that, theoretically, the bump in the first case is supposed to vanish at a shorter time.

Considering the optimal control that we obtained, the initial bumps exhibit the following behavior in time:

- $T = 0.50$

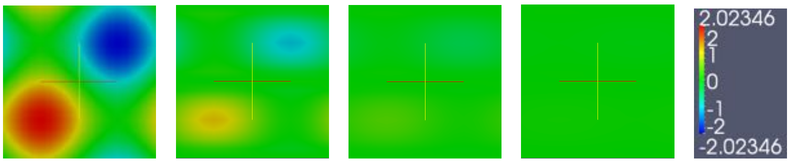

Figure 3.60: Evolution of $\eta(t, x, y)$ for $t = \{0, 0.10, 0.25, 0.50\}$ and $T = 0.50$.

- $T = 0.75$

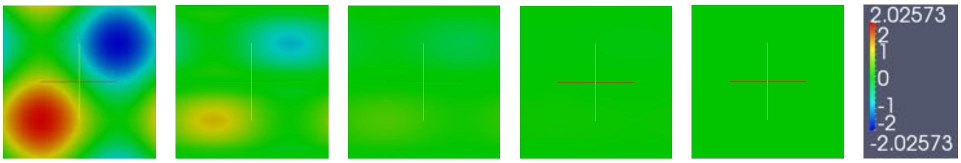

Figure 3.61: Evolution of $\eta(t,x,y)$ for $t = \{0, 0.10, 0.25, 0.50, 0.75\}$ and $T = 0.75$.

- $T = 1$

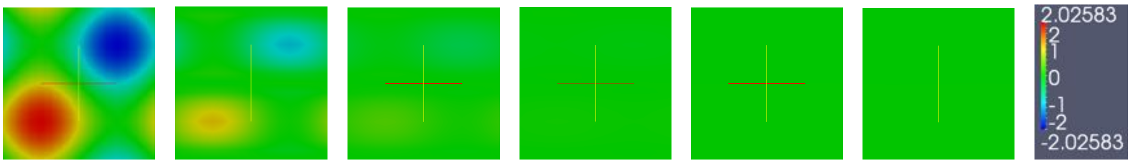

Figure 3.62: Evolution of $\eta(t,x,y)$ for $t = \{0, 0.10, 0.25, 0.50, 0.75, 1\}$ and $T = 1$

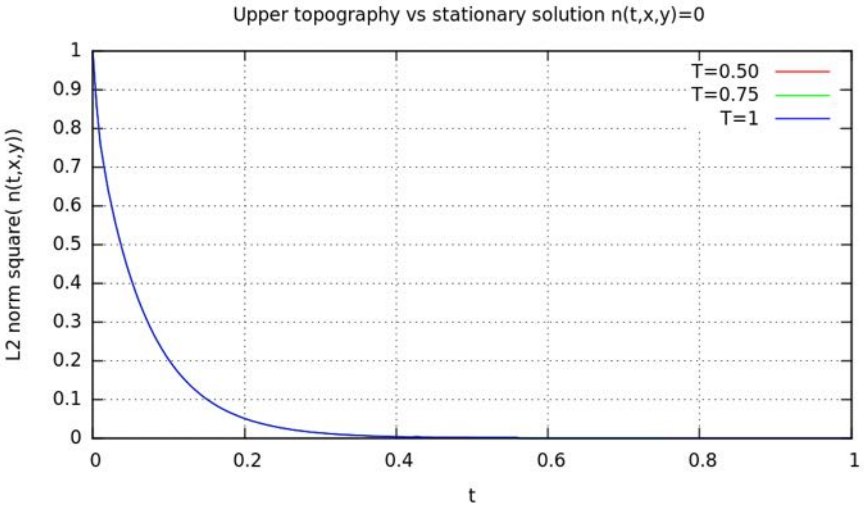

Figure 3.63: $\|\eta(t)\|^2_{L^2(\Omega)}$ for $T = \{0.50, 0.75, 1\}$

T	$\|q^* - Q^d\|^2_{L^2(\Omega)}$
0.5	$3.607 \cdot 10^{-5}$
0.75	$3.427 \cdot 10^{-5}$
1	$3.427 \cdot 10^{-5}$

Table 3.20: T vs $\|q^* - Q^d\|^2_{L^2(\Omega)}$. Example 3.1.4

T	$\|\eta^T(t)\|^2_{L^2(\Omega)}$, $t = 0.50$
0.50	$9.813 \cdot 10^{-4}$
0.75	$9.848 \cdot 10^{-4}$
1	$9.849 \cdot 10^{-4}$

T	$\|\eta^T(t)\|^2_{L^2(\Omega)}$, $t = 0.75$
0.50	–
0.75	$4.1391 \cdot 10^{-5}$
1	$4.1398 \cdot 10^{-5}$

T	$\|\eta^T(t)\|^2_{L^2(\Omega)}$, $t = 1$
0.50	–
0.75	–
1	$1.7959 \cdot 10^{-6}$

Table 3.21: T vs $\|\eta^T(t)\|^2_{L^2(\Omega)}$, $t = \{0.5, 0.75, 1\}$. Example 3.1.4

Besides the value of $T = \{0.50, 0.75, 1\}$, it seems that, in the three cases, the bump approaches the flat solution for $t < 0.50$. However at time $t = 0.50$, the case $T = 0.50$, as expected, slightly matches better $\eta(t)$ to the equilibrium state $u^\eta = 0$ ($\|\eta^T(t)\|^2_{L^2(\Omega)}$, with $t = 0.50$ is smaller for $T = 0.50$, see Table 3.21). As we can see in Table 3.21 the same situation occurs for $t = 0.75$.

The evolution of $z^{\mathbf{v}}(t)$ and $z^\eta(t)$ can be observed in the next plots for $t = \{0.5, 0.75, 1\}$:

- **Evolution of the adjoint variable $z^{\mathbf{v}_x}(t)$:**
  - Evolution of the adjoint variable $z^{\mathbf{v}_x}(t)$ for $T = 0.50$:

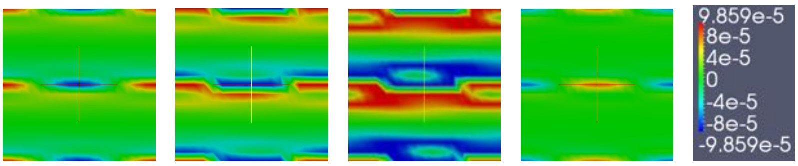

Figure 3.64: Evolution of $z^{\mathbf{v}_x}(t)$ for $t = \{0, 0.10, 0.25, 0.50\}$ and $T = 0.50$

  - Evolution of the adjoint variable $z^{\mathbf{v}_x}(t)$ for $T = 0.75$:

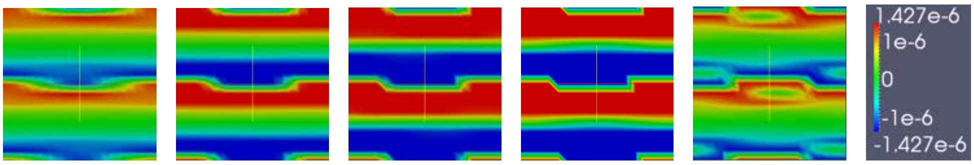

Figure 3.65: Evolution of $z^{\mathbf{v}_x}(t)$ for $t = \{0, 0.10, 0.25, 0.50, 0.75\}$ and $T = 0.75$

  - Evolution of the adjoint variable $z^{\mathbf{v}_x}(t)$ for $T = 1$:

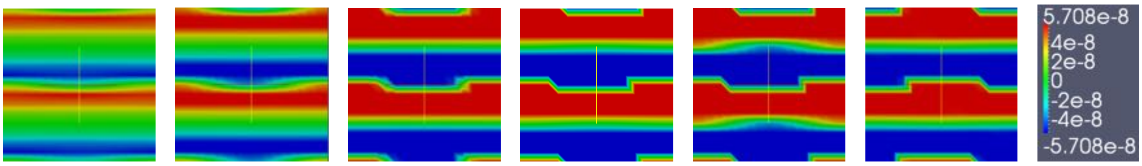

Figure 3.66: Evolution of $z^{\mathbf{v}_x}(t)$ for $t = \{0, 0.10, 0.25, 0.50, 0.75, 1\}$ and $T = 1$

- **Evolution of the adjoint variable $z^{\mathrm{v}_y}(t)$:**
  - Evolution of the adjoint variable $z^{\mathrm{v}_y}(t)$ for $T = 0.50$:

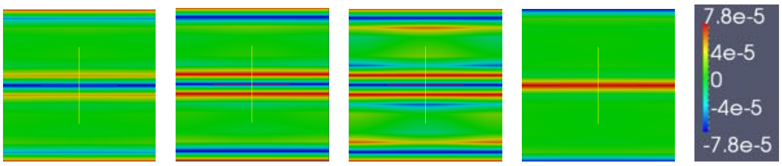

Figure 3.67: Evolution of $z^{\mathrm{v}_y}(t)$ for $t = \{0, 0.10, 0.25, 0.50\}$ and $T = 0.50$

  - Evolution of the adjoint variable $z^{\mathrm{v}_y}(t)$ for $T = 0.75$:

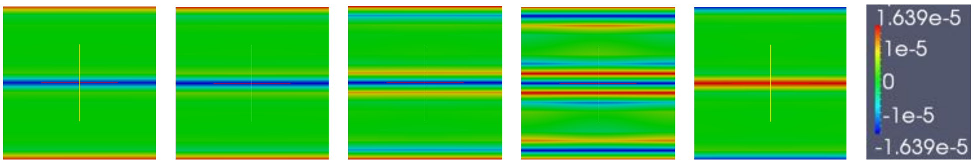

Figure 3.68: Evolution of $z^{\mathrm{v}_y}(t)$ for $t = \{0, 0.10, 0.25, 0.50, 0.75\}$ and $T = 0.75$

  - Evolution of the adjoint variable $z^{\mathrm{v}_y}(t)$ for $T = 1$:

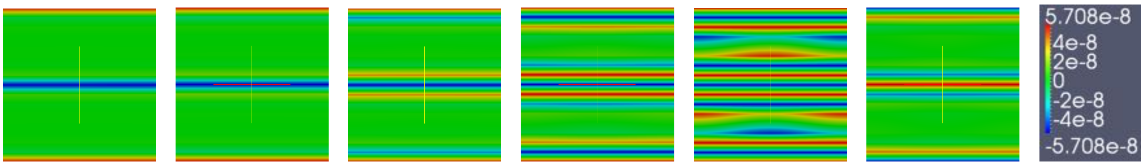

Figure 3.69: Evolution of $z^{\mathrm{v}_y}(t)$ for $t = \{0, 0.10, 0.25, 0.50, 0.75, 1\}$ and $T = 1$

- **Evolution of the adjoint variable $z^{\eta(t)}$:**
  - Evolution of the adjoint variable $z^{\eta}(t)$ for $T = 0.50$:

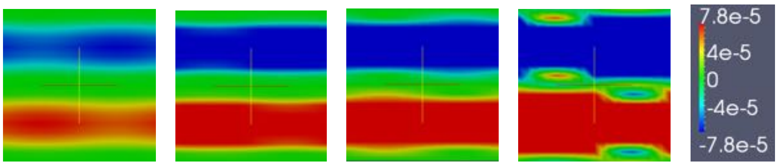

Figure 3.70: Evolution of $z^{\eta}(t)$ for $t = \{0, 0.10, 0.25, 0.50\}$ and $T = 0.50$

  - Evolution of the adjoint variable $z^{\eta}(t)$ for $T = 0.75$:

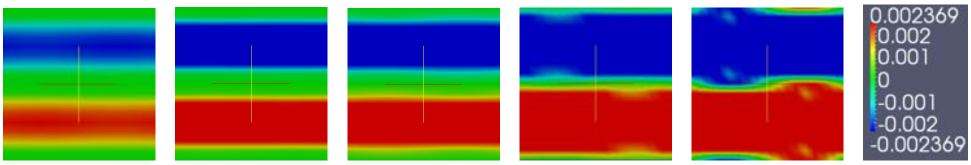

Figure 3.71: Evolution of $z^{\eta}(t)$ for $t = \{0, 0.10, 0.25, 0.50, 0.75\}$ and $T = 0.75$

- Evolution of the adjoint variable $z^{\eta}(t)$ for $T = 1$:

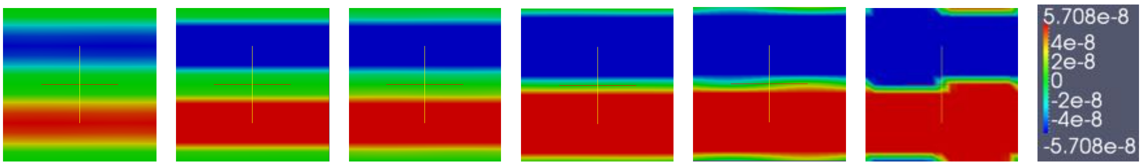

Figure 3.72: Evolution of $z^{\eta}(t)$ for $t = \{0, 0.10, 0.25, 0.50, 0.75, 1\}$ and $T = 1$

As in the previous case, since $z(0) = Q^d - q^*$ and $Q^{\mathbf{v}} = 0^2$, we expect $z^{\mathbf{v}}(0) = (z^{\mathbf{v}_x}(0), z^{\mathbf{v}_y}(0))$ to have the same pattern as the optimal initial velocity (with opposite sign), for this reason we observe that for $z^{\mathbf{v}}$ the case $T = 0.50$ has a bigger order as we commented in the beginning of the example. Additionally, $z^{\mathbf{v}}(t)$ does not vanish in time because we set $c_0 = 1$. Further, we also notice that for different values of T, $z^{\mathbf{v}}(T)$ has a similar pattern as $z^{\mathbf{v}}(0)$ but with opposite sign, remember that $z^{\mathbf{v}}(0) = \mathbf{v}(T) - \mathbf{v}^d$, while $z(0) = Q^d - q^*$ (The sign of the achieved result respect to Q^d at time zero is the opposite of u^d respect to the optimal solution at final time)

Finally, for $z^{\eta}(t)$ the case $T = 1$ has the smallest order ($\mathcal{O}(10^{-8})$) respect to the order $\mathcal{O}(10^{-5})$ of $T = 0.50$ and $\mathcal{O}(10^{-3})$ of $T = 0.75$. This means that, in the last case, we reproduce very accurately the expected initial height (Double Bump), while for the other cases we obtain an approximation of the double bump but with a smaller maximum height (or depth).

Conclusions

We have been able to successfully implement the forward simulation and parameter identification problem for the viscous SWE by the means of DopeLib and the deal.II Finite Element Library in C++. To verify the correctness of the implementation, we performed the simulation on different well known examples as well as different settings of the formulation. The results that we obtained were the expected ones [3, 9, 16, 43].

In our simulation, we verified that the FS time scheme offers more realistic results for small time steps. To do so, we considered different examples, in this report we show the "Double Bump" case (example 2.4.2), in which we perform the simulation for a fix mesh (256 elements) and variated the number of time steps for different time stepping schemes (BE, CN and FS). After different tests and calculations, we noticed that the BE scheme exhibited a dissipative behavior even for large number of time steps, furthermore the CN scheme seemed to converge to the same solution as the FS scheme for small time steps ($dt < 10^{-3}$). However for larger time steps ($dt > 10^{-2}$), the CN scheme showed instabilities and failed to converge to the expected solution, while the FS scheme for the different time steps that we analyzed predicted correctly the behavior of the state variables. Therefore, this scheme was used in the parameter identification section.

Additionally, we corroborated the importance of modifying the transport equation in our ODE system. As described in Section 1.2, we added, as in [3], a viscous term which changes the nature of our original formulation and allows the correct solution of the forward problem, also adding the expected descent effect in time [43, 48]. Moreover, in the case of a single bump (example 2.4.3), we performed both time and space discretization. We observed that the modification in the space discretization had a bigger effect on the outcome of the simulation, specially in the centered sector of the domain, where the Coriolis effect is smaller than in the other sectors of the bump. At the end of this section, we considered a mesh of 6 levels of refinement and tested different number of time step ranging from 50 to 1000. The simulation for these cases converged to the same solution.

On the other hand, we implemented in deal.II a mapping that modifies the element sizes of our mesh. Hence, the system that we consider remains with the same amount of degrees of freedom by redistributing the sizes of the elements in a way that the mesh adapts better in the areas where the dynamics of the process are more complex. Additionally, our finite element framework keeps its conforming nature.

In terms of the parameter identification, we proposed a cost functional that allows us to consider which profile we want to match in the optimization problem (velocity field or the upper topography of the fluid), as well as the expected behavior of such fields.

The optimization problem is solved using the reduced formulation of the problem by the means of the Newton method with Armijo type line search which is already implemented in DopeLib [20]. Additionally, we used the lagrangian formulation to calculate the gradient and hessian of the reduced cost functional. We performed different simulations for various settings and we obtain coherent results, showing numerical convergence to the initial behaviors that we set for the simulations in the second chapter. Further, we variated some parameters in the formulation, e.g. α. $\alpha = 1$ seems to be the ideal value for the settings that we considered ($Q^d \neq 0^3$), with this value the optimal control and the value of the state variable at final time T matches with the expected profiles. As we verified in example 3.1.2. (Advection of a quasistationary vortex), for values of $\alpha < 1$ it may happen that the state variable at final time does not improve drastically respect to bigger values of α, for which the initial data is much better adjusted, also, as we observed in Example 3.1.1 and 3.1.2, for small values of α, it is harder to achieve convergence due to the instability of the inverse problem.

Further steps in our work would be to make a deeper analysis of the adjoint variable and its evolution in time. We would also like to verify that the optimal control q^* satisfies the regularity assumptions showed in (1.34) [6], in this sense, it could be useful to implement other viscosity terms besides the one proposed in section 1.2, for which there are more theoretical results in terms of existence and uniqueness of solution as well as the regularities conditions, as we can see in [21, 43].

List of Figures

List of Tables

Bibliography

[1] D. Auroux, “Several data assimilation methods for geophysical problem”, Indian Journal of Pure and Applied Mathematics 37, 1(2006) 41-58.

[2] W. Bangerth, R. Hartmann, G. Kanschat, “Differential Equations Analysis Library”. (2010)

[3] S. Beckers, “Definition of parameters of the shallow water equations”. Personal conversation.

[4] C. Bernardi and O. Pironneau, “On the shallow water equations at low Reynolds number”, Comm. Partial Differential Equations 16 (1991), 59−104.

[5] S. Bianchini and A. Bressan, “Vanishing viscosity solutions to nonlinear hyperbolic systems”, Annals of Mathematics 161 (2005), 223−342.

[6] D. Bresch, B. Desjardins, “Existence of global weak solutions for a 2D viscous shallow water equations and convergence to the quasi- geostrophic model”, Comm. Math. Phys. 238 (12) (2003) 211223.

[7] D. Bresch, B. Desjardins, G. Métivier, “Recent mathematical results and open problems around shallow water equations”, J. Math. Fluid Mech (2006).

[8] A. Bressan, “Hyperbolic Systems of Conservation Laws. The One Dimensional Cauchy Problem”. Oxford University Press (2000).

[9] M. O. Bristeau, R. Glowinski, J. Periaux, “Numerical methods for the Navier-Stokes equations”, Comput. Phys. Rep. 6 (1987), 73−187.

[10] A. T. Bui, “Existence and uniqueness of a classical solution of an initial boundary value problem of the theory of shallow waters”, SIAM J. Math. Anal. 12 (1981), 229−241.

[11] J.C. Butcher, “Numerical Methods for Ordinary Differential Equations”, New York: John Wiley & Sons (2003), 57−71.

[12] B. Cheng, E. Tadmor, “Long-Time existence of smooth solutions for the rapidly rotating Shallow Water and Euler Equations ” SIAM J. MATH. ANAL. Vol. 39, No. 5, pp. 1668 – 1685 (2008).

[13] P. G. Ciarlet. "The Finite Element Method for Elliptic Problems, volume 40 of Classics in Applied Mathematics". Society for Industrial and Applied Mathematics (SIAM) (2002).

[14] A. Córdoba, D. Córdoba, M.A. Fontelos, "Formation of singularities for a transport equation with nonlocal velocity", Ann. of Math. (2) 162 (3) (2005). 1377 – 1389.

[15] J. Crank, P. Nicolson. "A practical method for numerical evaluation of solutions of partial differential equations of the heat conduction type". Proc. Camb. Phil. Soc. 43 (1). 50−67 (1947).

[16] O. Delestre, C. Lucas, P.A. Ksinant, F. Darboux, C. Laguerre, T.N.T. Vo, F. James, "SWASHES: a compilation of Shallow Water Analytic Solution for Hydraulic and Environmental Studies, (2014).

[17] G.Evensen, "Data Assimilation: The Ensemble Kalman Filter, Springer-Verlag, Berlin Heidelberg (2007).

[18] F. Flori, P. Orenga, M. Peybernes. "An existence result for a free boundary shallow water model using a Lagrangian scheme", ANN Inst Henri Poincare-ANAL. 01/2008; 25(4):725 – 741 (2008).

[19] J.F. Gerbeau, B. Perthame, "Derivation of viscous Saint-Venant system for laminar shallow water; Numerical validation", Discrete Contin. Dyn. Syst. Ser. (2001), 89−102.

[20] C. Goll, T. Wick, W. Wollner, "DOpElib: Differential Equations and Optimization Environment" (2013).

[21] C. Hao, L.Hsiao, H.L.Li, "Cauchy problem for viscous rotating shallow water equations", J. Diff. Eqns., 247 (2009). 3234−3257.

[22] Y. Ishikawa, T. Awaji, K. Akitomo and B. Qiu, "Successive correction of the mean sea surface height by the simultaneous assimilation of drifting buoy and altimetric data", J. Phys.Oceanogr., 26 (1996), 2381 – 2397.

[23] M.Jardak, I.M.Navon, M.Zupanski, "Ensemble Data Assimilation for the shallow water equation", *Journal of Geophysical Research*, DOI:10.1029 (2010).

[24] J. Kevorkian, "Partial Differential Equations: Analytical Solutions Techniques" , Chapman & Hall, New York (1990). Chapter 5.3.

[25] S. Kindermann, "Inverse Problems and Regularization", Industrial Mathematics Institute, University of Linz. Lecture Notes (2010).

[26] P. Kloeden, "Global existence of classical solutions in the dissipative shallow water equations", SIAM J. Math. Anal. 16 (1985), 301−315

[27] C. Kühbacher, "Shallow Water Derivation and Applications", Technische Universität Dortmund. Lecture Notes (2009).

[28] L. Kuznetsov, K. Ide, C.T. Jones, “A method for assimilation of Lagrangian data”, Mon. Wea. Rev., 131 (2003), 2247 – 2260.

[29] W. Lahoz, P.Schneider, “Data assimilation: making sense of Earth Observation”. Frontiers in environmental science. fenvs-02-00016 (2014).

[30] M. Landajuela, “Burgers Equation”. BCAM Internship- Summer 2011 (2011).

[31] P.L. Lions, “Mathematical Topics in Fluid Mechanics. Volume 2: Compressible Models”, Oxford Science Publications (1998), 236−254.

[32] F. Marche, “Derivation of a new two-dimensional viscous shallow water model with varying topography, bottom friction and capillary effects”, Eur. J. Mech. B Fluids 26 (2007), 49−63.

[33] C. Mascia, “A dive into shallow water”. Riv. Mat. Univ. Parma, Vol. 1, No.1 (2010). 77−149.

[34] A. Matsumura and T. Nishida, “The initial value problem for the equations of motion-viscous and heat-conductive gases”, J. Math. Kyoto Uni.20, No. 1 (1980), 67−104.

[35] D. Meidner, T. Richter, “A Posteriori Error Estimation for the Fractional Step Theta Discretization of the Incompressible Navier-Stokes Equations” (2014).

[36] A. Molcard, L. I. Piterbarg, A. Griffa, T. Ozgókmenö, A. Mariano, “Assimilation of drifter observations for the reconstruction of the Eulerian circulation field”, J. Geophys. Res., 108 (2003), 3056.

[37] V. Nassehi, A. Yazdani, “Finite element solution of multi-scale transport problems using the least squares based bubble function enrichment”, CoRR abs/1101.1826 (2011).

[38] I. M. Navon, “Practical and theoretical aspects of adjoint parameter estimation and identifiability in meteorology and oceanography”, Dyn. Atmos. Oceans, 27 (1997), 55 – 79.

[39] C. Paniconi, M. Marrocu, M. Putti and M. Verbunt, “Newtonian nudging for a Richards equation-based distributed hydrological model”, Adv. Water. Resour., 26 (2003), 161 – 178.

[40] J. Park. “Numerical Simulation of Wave Propagation using the Shallow Water Equations”. Harvey Mudd College (2007).

[41] H. Segur, H. Yamamoto, “Lecture 8: The Shallow-Water Equations” (2009).

[42] I.Strub, J. Percelay. O.P. Tossavainen, A. Bayen, “Comparison of two data assimilation algorithms for shallow water flows”, Amer. Ins. of Math. Sc. Vol 4.N 2. (2009). 409 – 430.

[43] L. Sundbye, “Global Existence of Solutions for the Viscous Shallow Water Equations”, Ph.D. Thesis, University of Colorado, Program in Applied Mathematics, Boulder, CO (1994).

[44] A.H. Techet, “2.016 Hydrodynamics Notes”, MIT open course ware (2005). Reading 7.

[45] J.W. Thomas. “Numerical Partial Differential Equations: Finite Difference Methods” Texts in Applied Mathematics 22. Berlin, New York: Springer-Verlag (1995). Chapter 3.

[46] G. Vallis, “Atmospheric and oceanic fluid dynamics”. Cambridge University Press (2006), $123 - 162$.

[47] S. Vater. “A multigrid-based multiscale numerical scheme for shallow water flows at low froude number”. Berlin : Freie Universität Berlin (2013). Chapter 4.

[48] B.K. Venkanna. “Fundamentals of Heat and Mass Transfer”. New Delhi: PHI Learning. p. 38 (2011).

Erklärung

Die vorliegende Arbeit habe ich selbständig verfasst und keine anderen als die angegebenen Hilfsmittel - insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen - benutzt.

Die Arbeit habe ich vorher nicht in einem anderen Prüfungsverfahren eingereicht. Die eingereichte schriftliche Fassung entspricht genau der auf dem elektronischen Speichermedium.

Hamburg, 30.10.2014