INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids **31**: 1037–1059 (1999)

IMPROVEMENT AND GENERALIZATION OF A FINITE ELEMENT SHALLOW-WATER SOLVER TO MULTI-LAYER SYSTEMS

JORGE MACÍAS*, CARLOS PARES AND MANUEL J. CASTRO

Departamento de Análisis Matemático, Universidad de Málaga, Campus de Teatinos, 29080 Málaga, Spain

SUMMARY

This paper improves and generalizes to multi-layer systems the shallow-water solver presented in [Bermúdez *et al.*, *IMA J. Numer. Anal.*, **11**, 79–97 (1991)]. The model equations are discretized in time using the method of characteristics and the Euler implicit method. The space discretization is performed using the first-order Raviart–Thomas mixed finite element. A formulation of the non-linear equations to solve at each time step that takes into account regions without water is given, and numerical results are presented in which this situation takes place for the one-dimensional case. These non-linear problems are solved by a duality technique with an automatic choice of parameters that greatly improves the convergence of the algorithm. A preconditioner has been designed for solving the linear problems that appear at each iteration of the duality method, which significantly reduces the computational cost. This is illustrated with some numerical examples. Finally, an application of the multi-layer model to a realistic geometry of the Alboran Sea is presented, giving good results from a qualitative point of view. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: shallow-water equations; duality methods; FEM; preconditioners; Alboran Sea

1. INTRODUCTION

The goal of this paper is to improve the shallow-water solver presented by Bermúdez *et al.* in [1] and generalize it to a multi-layer system. This solver has been successfully used by these authors in the simulation of tidal currents in estuaries. Besides its good stability properties, we want to point out here its potential capability for dealing with regions without water appearing inside the computational domain. Nevertheless, in this case numerical difficulties arise that until now have been only solved in the one-dimensional case.

The equations are formulated in conservative form and the horizontal viscosity effects are neglected, so that the system of partial differential equations (PDEs) to solve is hyperbolic. The discretization in time is performed by using a method of characteristics [3] for the momentum equation and the implicit Euler scheme for the continuity equation. The first-order Raviart–Thomas mixed finite element is used to discretize in space.

Regions without water are introduced using a formulation of the boundary value problems appearing at each time step in the form of a variational inequality. This formulation could not be introduced with an explicit treatment of the continuity equation. This is the reason for the

* Correspondence to: Departamento de Análisis Matemático, Universidad de Málaga, Campus de Teatinos, 29080 Málaga, Spain. E-mail: grupo@anamat.cie.uma.es

CCC 0271–2091/99/231037–23$17.50

Received 12 March 1998
Revised 4 August 1998

choice of the Euler implicit method to discretize this equation, despite the damping effects introduced by this choice. At this first stage of our work, these side effects are not very important, as we are interested in steady states or solutions without sharp gradient regions. Nevertheless, the model can be used for simulating transient regimes and for capturing shock phenomena, if small enough time steps are taken. In this case, the additional cost due to the implicit treatment needs to be reasonable. Therefore, improving the numerical algorithm is the first goal of this work. Another further development in this direction will be to incorporate adaptative mesh techniques.

The non-linear problems appearing at each time iteration were solved in [1] by using a duality method proposed by Bermúdez and Moreno in [2]. A drawback of this method is the need for a manual choice of two parameters upon which the convergence strongly depends. In this paper, a variant of this method is proposed, in which the two parameters are replaced by two functions computed automatically and optimally. This modification dramatically improves the convergence rate of the algorithm. The convergence has also been improved by the use of an IC-type preconditioner, well adapted to the characteristics of the linear problems that appear. We illustrate this with some numerical examples.

The problem that gave rise to the questions addressed in this paper was the modelling of the dynamics of water masses in the Alboran Sea (the western most part of the Mediterranean). In this sea, two layers of water can be distinguished: the surface Atlantic water penetrating into the Mediterranean through the Strait of Gibraltar, and the deeper, denser Mediterranean water flowing into the Atlantic. The observation of this simplified picture shows that, if a bi-dimensional model is going to be used to simulate the flow in this region, it is necessary to consider, at least, a two-layer model. Therefore, the second goal of this work is to generalize the shallow-water solver to multi-layer systems.

It is interesting that the model to be used allows the presence of zones where the thickness of one of the layers becomes vanishingly thin. In the Strait of Gibraltar, when strong, tide-induced currents pass through, the flow shows very large oscillations at the interface. This can cause deep water upwell such that, in the zone of upwelling, the surface layer of Atlantic water becomes vanishingly thin. Therefore, we are dealing with a problem analogous to that of the emergence of land areas in a one-layer system.

We finally present an application of the multi-layer model to a realistic geometry of the Alboran Sea, giving good results from a qualitative point of view.

2. THE EQUATIONS

We first consider the one-layer case. In [1] the following formulation of the shallow water equations in flux form is considered:

$$\begin{cases} \dfrac{\partial \mathcal{Q}}{\partial t} + \nabla\cdot(\bar{u}\otimes\mathcal{Q}) + \dfrac{g}{2}\nabla(\eta^2 + 2\eta H) = g\eta\nabla H + F, \quad \text{in } \Omega\times(0,T), \\ \dfrac{\partial \eta}{\partial t} + \nabla\cdot\mathcal{Q} = 0, \quad \text{in } \Omega\times(0,T), \end{cases} \tag{1}$$

where η is the elevation above a reference level (the mean sea level, for example), H is the depth from the reference level to the bottom of the basin and $h = H + \eta$ is the thickness of the layer (see Figure 1). $\bar{u}$ is the average velocity through the depth, $\mathcal{Q} = h\bar{u}$ the flux, g the gravity,

and F external forces (Coriolis, wind and bottom effects). Here, $(0, T)$ is the time interval when the flow is studied and Ω represents the horizontal projection of the volume occupied by water, whose boundary will be represented by $\partial\Omega$.

For the sake of simplicity of the mathematical presentation, let $\mathcal{Q}\cdot n = 0$ at $\partial\Omega$ and $F = 0$. These two restrictions will not be imposed in the numerical results where other boundary conditions are allowed and Coriolis and wind effects are taken into account. In [1], only the case where $\eta + H > 0$ is considered, i.e. there is always water above each point of Ω at any time.

If regions without water are considered, then η may reach the value $-H$. To take this situation into account let us introduce the following notations:

$$\Omega_t^+ = \{x \in \Omega:\ \eta(x, t) > -H(x)\},$$

$$\Omega_t^0 = \{x \in \Omega:\ \eta(x, t) = -H(x)\},$$

$$\Sigma_t = \partial\Omega_t^0,$$

where $t \in (0, T)$ represents an arbitrary time level. Observe that Ω_t^+, Ω_t^0 and Σ_t are subsets of Ω. They represent respectively the region occupied by water at time t, the region without water at the same time level and the boundary between these two regions. The index t is necessary as these regions can change with time. As the evolution of the boundary between the regions with and without water depends on the motion of the fluid and on the geometry of the basin, it is an unknown of the problem, i.e. Σ_t has the character of a free boundary. For simplicity, we assume that it is defined by a regular curve with the equation $\phi(x, t) = 0$. At Σ_t the conditions must be $\eta = -H$ and $(\partial\phi/\partial t) + \nabla\phi \cdot u = 0$. This latter condition means that particles at the water–land boundary remain there.

In order to establish the set of equations in the general case, observe that in Ω_t^+, Equations (1) remain valid, while in Ω_t^0 they simplify to $\mathcal{Q} = 0$, $\eta = -H$, and let us consider the following notations:

$$Q^+ = \{(x, t) \in \Omega \times (0, T):\ x \in \Omega_t^+\},$$

$$Q^0 = \{(x, t) \in \Omega \times (0, T):\ x \in \Omega_t^0\},$$

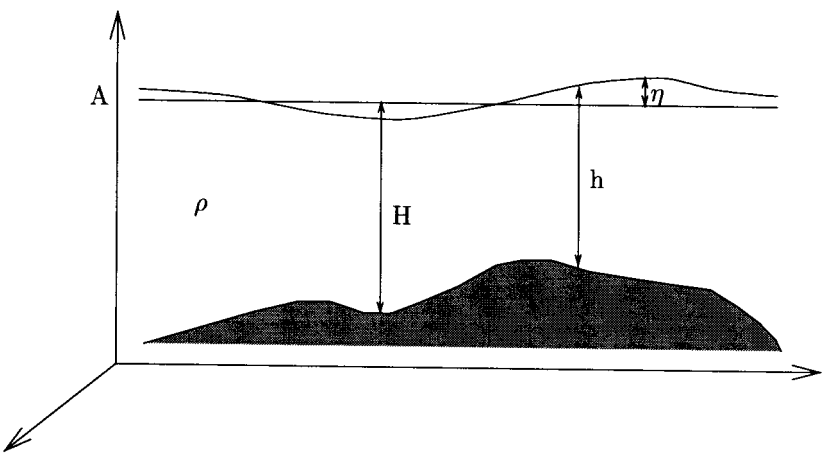

Figure 1. Notations.

$$\Sigma = \{(x, t) \in \Omega \times (0, T):\ x \in \Sigma_t\}.$$

Now, Q^+, Q^0 and Σ are subsets of the space–time domain $\Omega \times (0, T)$. They respectively represent the region occupied by water, the region without water and their boundary in the space–time domain. Therefore, '$x \in \Omega_t^+$' is equivalent to '$(x, t) \in Q^+$' and both formulae mean that there is water above the point x of Ω at time t. The introduction of these space–time regions allows us to specify the domain of the general equations

$$\begin{cases} \dfrac{\partial \mathscr{Q}}{\partial t} + \nabla \cdot (\bar{u} \otimes \mathscr{Q}) + \dfrac{g}{2} \nabla(\eta^2 + 2\eta H) = g\eta \nabla H \quad \text{in } Q^+, \\ \dfrac{\partial \eta}{\partial t} + \nabla \cdot \mathscr{Q} = 0 \quad \text{in } Q^+, \\ \mathscr{Q} = 0, \qquad \eta = -H \quad \text{in } Q^0, \\ \dfrac{\partial \phi}{\partial t} + \nabla \phi \cdot u = 0, \qquad \eta = -H \quad \text{at } \Sigma, \\ \mathscr{Q} \cdot n = 0 \quad \text{at } \partial\Omega \\ + \text{ initial conditions.} \end{cases} \tag{2}$$

3. TIME DISCRETIZATION

The discretization in time of the equations is performed by using a method of characteristics [3] for the convective term in the momentum equation and the implicit Euler method in the mass conservation equation. Given $\mathscr{Q}^n$ and η^n, the discretized equations are

$$\begin{cases} \dfrac{\mathscr{Q}^{n+1}}{\Delta t} + \dfrac{g}{2} \nabla((\eta^{n+1})^2 + 2\eta^{n+1} H) - g\eta^{n+1} \nabla H = \dfrac{J^n \mathscr{Q}^n[X^n]}{\Delta t} \quad \text{and} \\ \eta^{n+1} = \eta^n - \Delta t \nabla \cdot \mathscr{Q}^{n+1} \quad \text{in } \Omega^+_{t_{n+1}}, \\ \mathscr{Q}^{n+1} = 0, \qquad \eta^{n+1} = -H \quad \text{in } Q^0_{t_{n+1}}, \\ \eta^{n+1} = -H \quad \text{at } \Sigma_{t_{n+1}}, \\ \mathscr{Q}^{n+1} \cdot n = 0 \quad \text{at } \partial\Omega, \\ \mathscr{Q}^{n+1} = u^{n+1} h^{n+1}, \end{cases} \tag{3}$$

where the subscripts t_{n+1} indicate the corresponding domains in time $t = \Delta t(n+1)$ and $\Sigma_{t_{n+1}}$ is the water–land boundary at time $t = \Delta t(n+1)$. By $X^n(x)$ we denote the position in time t_n of the particle being at point x at time t_{n+1}, i.e. $X^n(x) = X(x, t_{n+1}; t_n)$, where $\tau \to X(x, t; \tau)$ is the trajectory of the particle at point x at time t, and, therefore, $X(x, t; \tau)$ is the solution of

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}\tau} X(x, t; \tau) = u(X(x, t; \tau), \tau), \\ X(x, t; t) = x. \end{cases}$$

Finally, $J^n(x) = J(x, t_{n+1}; t_n)$, where $J(x, t; \tau)$ represents the evolution of the element of volume, which is the solution of the following ordinary differential equation:

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}\tau} J(x, t; \tau) = \nabla \cdot u(X(x, t; \tau), \tau)\ J(x, t; \tau), \\ J(x, t; t) = 1. \end{cases}$$

The discretization is performed by taking into account the identity

$$\frac{\partial \mathscr{Q}}{\partial t} + \nabla \cdot (u \otimes \mathscr{Q}) = \frac{\mathrm{D}}{\mathrm{D}t} (J\mathscr{Q}),$$

where $\mathrm{D}/\mathrm{D}t$ denotes the total derivative, and using a finite difference scheme to discretize the derivative.

Observation 3.1
Note that X^n: $\Omega^+_{t_{n+1}} \to \Omega^+_{t_n}$ and X^n: $\Sigma_{t_{n+1}} \to \Sigma_{t_n}$.

4. VARIATIONAL FORMULATION

Problem (3) is an obstacle-type problem (cf. [4]). In the case studied here, the sea bottom plays the role of the obstacle. As in many other cases, problem (3) can be formulated as a minimization problem

$$\frac{1}{\Delta t} \int_\Omega \mathscr{Q}^{n+1}(z - \mathscr{Q}^{n+1})\, \mathrm{d}x + g\Delta t \int_\Omega \nabla \cdot \mathscr{Q}^{n+1} \nabla H (z - \mathscr{Q}^{n+1})\, \mathrm{d}x$$
$$+ \frac{g}{2\Delta t} (\Psi(\eta^n - \Delta t \nabla \cdot z) - \Psi(\eta^n - \Delta t \nabla \cdot \mathscr{Q}^{n+1})) \geq \langle \tilde{L}^n, z - \mathscr{Q}^{n+1} \rangle, \quad \forall z \in V, \tag{4}$$

where V is the functional space $V = \{v \in (L^2(\Omega))^2:\ \nabla \cdot v \in L^3(\Omega);\ v \cdot n = 0 \text{ at } \partial\Omega\}$, $\langle\ .\ ,\ .\ \rangle$ denotes the duality between V and its dual V', $\tilde{L}^n = (1/\Delta t) J^n \mathscr{Q}^n[X^n] + g\eta^n \nabla H$ and given $v \in L^3(\Omega)$, $\Psi(v) = \int_\Omega \psi(x, v(x))\, \mathrm{d}x$ with

$$\psi_x(\eta) = \psi(x, \eta) = \begin{cases} \dfrac{1}{3}\eta^3 + \eta^2 H(x) & \text{if } \eta + H \geq 0, \\ +\infty & \text{otherwise.} \end{cases}$$

This formulation, which enables taking into account regions without water, can be interpreted as the search for the minimum of an energy functional, where the regions with negative thickness of water are penalized with infinite energy. In this sense, if the functional space V is thought of as the set of admissibles (or finite energy) fluxes, the introduction of the functional

Ψ allows us to eliminate the elements of this space leading to a negative thickness for the water layer.

It has been proved in [1] that if $\eta^n \in L^3(\Omega)$, $J^n \mathcal{Q}^n[X^n] \in (L^2(\Omega))^2$ and $H \in C^1(\Omega)$, problem (4) has a unique solution of $\mathcal{Q}^{n+1} \in V$, $\eta^{n+1} \in L^3(\Omega)$.

Using the subdifferential of $\psi_x(\eta)$, $G_x(\eta)$ (see [5]), problem (4) can be reformulated as a variational problem:

Find $\mathcal{Q}^{n+1} \in V$, $\theta^{n+1} \in L^{3/2}(\Omega)$ such that:

$$\begin{aligned}
&\frac{1}{\Delta t} \int_\Omega \mathcal{Q}^{n+1} \cdot z \, \mathrm{d}x - \frac{g}{2} \int_\Omega \theta^{n+1} \nabla \cdot z \, \mathrm{d}x - g \int_\Omega (\eta^n - \Delta t \nabla \cdot \mathcal{Q}^{n+1}) \nabla H \cdot z \, \mathrm{d}x \\
&= \frac{1}{\Delta t} \int_{\Omega^+_{t^{n+1}}} J^n \mathcal{Q}^n[X^n] \cdot z \, \mathrm{d}x, \quad \forall z \in V, \qquad (5) \\
&\theta^{n+1} \in G_x(\eta^n - \Delta t \nabla \cdot \mathcal{Q}^{n+1}), \\
&\mathcal{Q}^{n+1} = u^{n+1}(\eta^{n+1} + H).
\end{aligned}$$

Observe that, given $x \in \Omega$, G_x is the multi-variate operator defined by

$$G_x(\eta) = \partial_x \psi(\eta) = \begin{cases} \eta^2 + 2\eta H(x) & \text{if } \eta + H > 0 \\ (-\infty, \ -H^2] & \text{if } \eta + H = 0. \end{cases} \qquad (6)$$

In the regions of zero thickness, an interval is obtained due to the 'infinite jump' of the ψ_x operator. It can be proved that problems (3) and (5) are formally equivalent (see [6]).

5. NUMERICAL RESOLUTION

The problems to be solved at each time iteration are as follows:

Find $(\mathcal{Q}, \theta)$ such that $\forall z \in V$

$$\begin{aligned}
&\frac{1}{\Delta t} \int_\Omega \mathcal{Q} \cdot z \, \mathrm{d}x - \frac{g}{2} \int_\Omega \theta \nabla \cdot z \, \mathrm{d}x - g \int_\Omega (\eta_0 - \Delta t \nabla \cdot \mathcal{Q}) \nabla H \cdot z \, \mathrm{d}x = \langle F, z \rangle, \qquad (7) \\
&\theta(x) \in G_x(\eta_0 - \Delta t \nabla \cdot \mathcal{Q}) \quad \text{a.e. } x \in \Omega,
\end{aligned}$$

where η_0 would be η^n, which is assumed to be known, and $\langle F, z \rangle$ is the term given by the method of characteristics. Once solved, the solution at time $\Delta t(n+1)$ is given by $(\mathcal{Q}, \eta = \eta_0 - \Delta t \nabla \cdot \mathcal{Q})$.

- A *first difficulty* comes from the fact that the bilinear form appearing,

$$a(\mathcal{Q}, z) = \frac{1}{\Delta t} \int_\Omega \mathcal{Q} \cdot z \, \mathrm{d}x + g \Delta t \int_\Omega (\nabla \cdot \mathcal{Q})(\nabla H \cdot z) \, \mathrm{d}x,$$

is not elliptic in V.

To overcome this problem, we introduce the operator $G_x^w(\eta) = G_x(\eta) - w\eta = (G_x - wI)(\eta)$, where $w > 0$, to be chosen. Let $p = \theta - w(\eta_0 - \Delta t \nabla \cdot \mathcal{Q})$, where θ is the solution of (7). We have that $p(x) \in G_x^w(\eta_0 - \Delta t \nabla \cdot \mathcal{Q})$ and

$$\frac{g}{2}\int_\Omega p\nabla\cdot z\,\mathrm{d}x=\frac{g}{2}\int_\Omega \theta\nabla\cdot z\,\mathrm{d}x-\frac{gw}{2}\int_\Omega(\eta_0-\Delta t\nabla\cdot\mathcal{Q})\nabla\cdot z\,\mathrm{d}x,$$

from whence Equation (7) is written as

$$\frac{1}{\Delta t}\int_\Omega \mathcal{Q}\cdot z\,\mathrm{d}x-\frac{g}{2}\int_\Omega p\nabla\cdot z\,\mathrm{d}x-\frac{gw}{2}\int_\Omega(\eta_0-\Delta t\nabla\cdot\mathcal{Q})\nabla\cdot z\,\mathrm{d}x-g\int_\Omega(\eta_0-\Delta t\nabla\cdot\mathcal{Q})\nabla H\cdot z\,\mathrm{d}x$$
$$=\langle F,z\rangle.$$

Upon rearranging the terms, $(\mathcal{Q},p)$ is seen to satisfy

$$\frac{1}{\Delta t}\int_\Omega \mathcal{Q}\cdot z\,\mathrm{d}x+\frac{\Delta t\,gw}{2}\int_\Omega\nabla\cdot\mathcal{Q}\nabla\cdot z\,\mathrm{d}x+\Delta t\,g\int_\Omega\nabla\cdot\mathcal{Q}(\nabla H\cdot z)\,\mathrm{d}x$$
$$=\frac{g}{2}\int_\Omega p\nabla\cdot z\,\mathrm{d}x+g\int_\Omega\eta_0(\nabla H\cdot z)\,\mathrm{d}x+\frac{gw}{2}\int_\Omega\eta_0(\nabla\cdot z)\,\mathrm{d}x+\langle F,z\rangle,\quad p\in G_x^w(\eta_0-\Delta t\nabla\cdot\mathcal{Q}).$$

The bilinear form to be considered is now

$$a(\mathcal{Q},z)=\frac{1}{\Delta t}\int_\Omega \mathcal{Q}\cdot z\,\mathrm{d}x+\frac{\Delta t\,gw}{2}\int_\Omega\nabla\cdot\mathcal{Q}\nabla\cdot z\,\mathrm{d}x+\Delta t\,g\int_\Omega\nabla\cdot\mathcal{Q}(\nabla H\cdot z)\,\mathrm{d}x,$$

which is elliptic for sufficiently large w and bounded ∇H (see [6]). Note that the introduction of such a parameter as w is simply a mathematical artifice to obtain an equivalent formulation, but with an elliptic bilinear form.

Written in this form, the problem to be solved is:

Find $(\mathcal{Q},p)$ such that

$$a(\mathcal{Q},z)=\frac{g}{2}\int_\Omega p\nabla\cdot z\,\mathrm{d}x+\langle L,z\rangle,\quad\forall z\in V,\tag{9}$$
$$p\in G_x^w(\eta_0-\Delta t\nabla\cdot\mathcal{Q}),$$

where

$$\langle L,z\rangle=\langle F,z\rangle+g\int_\Omega\eta_0(\nabla H\cdot z)\,\mathrm{d}x+\frac{gw}{2}\int_\Omega\eta_0(\nabla\cdot z)\,\mathrm{d}x.$$

For the numerical solution of this problem, we may think of using a fixed point algorithm:

1. choose p^0,
2. solve $a(\mathcal{Q}^n,z)=\frac{g}{2}\int_\Omega p^n\nabla\cdot z\,\mathrm{d}x+\langle L,z\rangle$,
3. $p^{n+1}=G_x^w(\eta_0-\Delta t\nabla\cdot\mathcal{Q}^n)$.

Nevertheless, using such an algorithm is not possible as

1. G_x^w *is multi-valued*, and
2. G_x^w *is not Lipschitzian*, therefore, the convergence of the former algorithm cannot be assured.

• To overcome these *new difficulties*, following [2] we introduce the Yosida approximation of the operator G_x^w (see Appendix A), defined for values of λ and w, verifying $0<\lambda w<1$:

$$G^w_{x,\lambda}(\eta) = \begin{cases} \dfrac{2\lambda\eta + 1 - \lambda w + 2\lambda H - \sqrt{(1-\lambda w + 2\lambda H)^2 + 4\lambda\eta}}{2\lambda^2} & \text{if } \eta \geq -(1-\lambda w)H - \lambda H^2, \\ \dfrac{\eta + H}{\lambda} & \text{otherwise.} \end{cases}$$

The following property holds [2]:

$$p \in G^w_x(\eta) \Leftrightarrow p = G^w_{x,\lambda}(\eta + \lambda p) \quad \text{for } 0 < \lambda w < 1. \tag{10}$$

Besides, if $0 < \lambda w \leq \frac{1}{2}$, $G^w_{x,\lambda}$ is Lipschitzian with constant $1/\lambda$, i.e.

$$\left|G^w_{x,\lambda}(\eta_1) - G^w_{x,\lambda}(\eta_2)\right| \leq \frac{1}{\lambda}\left|\eta_1 - \eta_2\right|, \quad \forall \eta_1, \eta_2 \in \mathbb{R}. \tag{11}$$

Using property (10) the problem to be solved can be written as

$$\begin{cases} a(\mathcal{Q}, z) = \dfrac{g}{2}\displaystyle\int_\Omega p \nabla\cdot z \, \mathrm{d}x + \langle L, z \rangle, \\ p = G^w_{x,\lambda}(\eta_0 - \Delta t \nabla\cdot \mathcal{Q} + \lambda p), \end{cases} \tag{12}$$

which suggests the numerical scheme,

Algorithm 1

$$\begin{aligned} &\text{Choose } p^0, \\ &a(\mathcal{Q}^n, z) = \frac{g}{2}\int_\Omega p^n \nabla\cdot z \, \mathrm{d}x + \langle L, z \rangle, \\ &p^{n+1} = G^w_{x,\lambda}(\eta_0 - \Delta t \nabla\cdot \mathcal{Q}^n + \lambda p^n), \end{aligned} \tag{13}$$

taking λ and w (fixed) such that $\lambda w \leq \frac{1}{2}$ for the operator $G^w_{x,\lambda}$ to be Lipschitzian.

The convergence of $\mathcal{Q}^n$ to $\mathcal{Q}$ is assured at least in norm L^2 if $\lambda w = \frac{1}{2}$.

A disadvantage of the methodology followed in solving the problem (2) comes from the need of an *a priori* choice of the parameters λ and w (taken as fixed in the original works of Bermúdez *et al.* [1,2]). This drawback has been optimally solved for the problem treated in this study. For further details on the calculation of these parameters and on the convergence of the algorithm in more general problems see [7]. A second difficulty arises when regions without water appear, i.e. when $\eta = -H$, somewhere in the domain. In that case, the problem is how to compute the velocity. In what follows, we try to clarify these two ideas.

5.1. Calculation of λ, w

The performance of the method strongly depends on a good choice of the parameters (λ, w) and, in general, this choice it is not easy to perform as λ and w have no relationship with the physics of the problem. Let us consider problem (12): if $\eta = \eta_0 - \Delta t \nabla\cdot \mathcal{Q}$ were known *a priori*, $p(x)$ would be the solution of the equation $p(x) = G^w_{x,\lambda}(\eta(x) + \lambda p(x))$. In that case, a fixed-point algorithm can be used for computing p: $p^{n+1}(x) = G^w_{x,\lambda}(\eta(x) + \lambda p^n(x))$. The

optimal λ should be that making the iteration function have a minimal slope at the fixed point $p(x)$. Here, the iteration function is $g(q) = G^w_{x,\lambda}(\eta(x) + \lambda q)$. Let us study two cases: the case when $\eta > -H$ everywhere and the case for which regions without water appear.

Case 1. $\eta > -H$ everywhere.
Assume $\eta > -H$ in all the domain. In this case, for G^w_x being univalent, the fixed point can be computed explicitly since

$$p(x) = G^w_{x,\lambda}(\eta(x) + \lambda p(x)) \Leftrightarrow p(x) = G^w_x(\eta(x)) = \eta(x)^2 + 2\eta(x)H(x) - w\eta(x).$$

Therefore, the objective is to determine the parameter λ such that $|g'(p)| = |g'(\eta(x)^2 + 2\eta(x)H(x) - w\eta(x))|$ is minimal, with $w = 1/2\lambda$. The most favourable case would be to determine λ such that

$$g'(\eta(x)^2 + 2\eta(x)H(x) - w\eta(x)) = 0.$$

Operation results in the optimal λ_x being given by $1/[4(H(x) + \eta(x))]$ (see [6]).

If a fixed-point iteration is to be performed:

$$q^{n+1} = G^w_{x,\lambda}(\eta + \lambda q^n), \tag{14}$$

it would be optimal to do so with the form

$$q^{n+1}(x) = G^{w_x}_{x,\lambda_x}(\eta(x) + \lambda_x q^n(x)), \tag{15}$$

as second-order convergence would be achieved. Nevertheless, η is unknown. In fact, we can consider Algorithm 1 as an approximation of (15), where η is approximated by $\eta \approx \eta_0 - \Delta t \nabla \cdot \mathscr{Q}^{n+1}$ at each iteration. The calculation of λ also requires us to know η. For a small Δt, a reasonable approximation is $\eta \approx \eta_0$.

Therefore, the following algorithm is proposed:

1. Take $\lambda_x = 1/[4(\eta_0(x) + H(x))], \quad \forall x \in \Omega$.
2. Take $w_x = 1/2\lambda_x$.
3. $p_0 = \eta_0^2 + 2\eta_0 H - w_x \eta_0 = -\eta_0^2$.
4. For $n = 0, 1, \ldots$

$$\begin{cases} a(\mathscr{Q}^n, z) = \dfrac{g}{2} \displaystyle\int_\Omega p^n \nabla \cdot z \, \mathrm{d}x + \langle L, z \rangle, \\ p^{n+1} = G^{w_x}_{x,\lambda_x}(\eta_0 - \Delta t \nabla \cdot \mathscr{Q}^n + \lambda_x p^n), \end{cases}$$

where the bilinear operator is now given by

$$a(\mathscr{Q}, z) = \frac{1}{\Delta t} \int_\Omega \mathscr{Q} \cdot z \, \mathrm{d}x + \frac{\Delta t \, g}{2} \int_\Omega w_x \nabla \cdot \mathscr{Q} \nabla \cdot z \, \mathrm{d}x + \Delta t \, g \int_\Omega \nabla \cdot \mathscr{Q}(\nabla H \cdot z) \, \mathrm{d}x, \tag{16}$$

and

$$\langle L, z \rangle = \langle F, z \rangle + g \int_\Omega \eta_0 (\nabla H \cdot z) \, \mathrm{d}x + \frac{g}{2} \int_\Omega w_x \eta_0 (\nabla \cdot z) \, \mathrm{d}x.$$

An advantage of computing λ and w in this way is that it avoids the *a priori* choice of these parameters. A second benefit is the saving in computer time as shown by the results in Section 9, where some numerical examples are presented (see Tables I and II). The optimal choice of

Table I. Performance of Algorithms 1 and 2. Test 1: channel coarse mesh

Algorithm	k	CPU time	Linear	Duality	Number non-zeros
1	0	11 015	30–35	75–80	5395
1	40	4112	10–12	75–80	16 712
2	0	1208	25–28	7–10	5395
2	20	726	11–14	7–10	10 573
2	40	690	9–11	7–10	13 704
2	50	631	8–9	7–10	16 789
2	60	a93	6–7	7–10	19 767
2	70	570	4–5	7–10	23 217
2	90	522	3–4	7–10	30 642

Linear, number of iterations required for solving the linear problems; duality, idem for the duality method.

λ and w also avoid/mitigate some problems of convergence that may arise with the original method when vanishing thickness of water appeared.

A disadvantage we mention is that the ellipticity of the bilinear form may be lost (if $\eta + H \ll 1 \Rightarrow \lambda_x \gg 1 \Rightarrow w_x \ll 1$) or the conditioning of the matrix may deteriorate (if $\eta + H \gg 1 \Rightarrow \lambda_x \ll 1 \Rightarrow w_x \gg 1$). In practice, to avoid this, w_x is chosen between two bounding values: $w_{\min} \leq w_x \leq w_{\max}$ (see [6] for the calculation of these values). Also, the possibility exists of recalculating λ_x and is given by

$$\lambda_x = \frac{1}{4(\eta_0(x) - \Delta t \nabla \cdot \mathscr{Q}^n + H(x))},$$

when necessary.

Case 2. $\eta + H = 0$ in some parts of the domain.
If $x \in \Omega$ is such that $\eta(x) + H(x) = 0$ then the equation $p(x) = G^w_{x,\lambda}(\eta(x) + \lambda p(x))$ has an infinite number of solutions: $p \in (-\infty, -H^2 + wH]$.

In effect, $\eta + \lambda p = -H + \lambda p \leq -H - \lambda H^2 + \lambda w H = -(1 - \lambda w)H - \lambda H^2$ and therefore, $G^w_{x,\lambda}(\eta + \lambda p) = (\eta + \lambda p + H)/\lambda = p$.

Besides, if $p \in (-\infty, -H^2 + wH)$

$$g'(p) = \lambda \frac{\mathrm{d}G^w_{x,\lambda}}{\mathrm{d}\eta}(\eta + \lambda p) = \lambda \frac{1}{\lambda} = 1.$$

Table II. Performance of Algorithms 1 and 2. Test 2: channel refined mesh

Algorithm	k	CPU time	Linear	Duality	Number non-zeros
1	0	303 731	145–150	75–80	24 390
2	0	25 155	145–150	7–10	24 390
2	20	23 790	110–115	7–10	62 080
2	40	3921	13–15	7–10	126 715
2	50	3393	10–11	7–10	156 473
2	60	3010	7–8	7–10	187 564
2	70	2529	4–5	7–10	223 146
2	90	2421	2–3	7–10	288 853

Linear, number of iterations required for solving the linear problems; duality, idem for the duality method.

In the limit case, $p = -H^2 + wH$, we have $g'_-(p) = 1$, $g'_+(p) = -1$. This situation occurs for any λ and, as consequence, no optimal λ can be chosen.

The following algorithm suggests itself for taking into account the remarks above on an optimal computation of λ and w:

Algorithm 2

1. Choose a maximal value for λ_x avoiding degeneracy of the bilinear form: $\lambda_{\max}$.
2. Known η_0, compute

$$\lambda_x = \begin{cases} \dfrac{1}{4(H+\eta_0)} & \text{if } \eta_0 > -H + \dfrac{1}{4\lambda_{\max}}, \\ \lambda_{\max} & \text{otherwise.} \end{cases}$$

3. Take $w_x = 1/2\lambda_x$.
4. Choose

$$p^0 = \begin{cases} \eta_2^0 + 2\eta_0 H - w_x \eta_0 & \text{if } \eta_0 > -H, \\ -H^2 + w_x H & \text{if } \eta_0 = -H. \end{cases}$$

5. For $n = 0, 1, \ldots$

$$\begin{cases} a(\mathcal{Q}^n, z) = \dfrac{g}{2} \displaystyle\int_\Omega p^n \nabla \cdot z \, \mathrm{d}x + \langle L, z \rangle, \\ p^{n+1} = G^{w_x}_{x,\lambda_x}(\eta_0 - \Delta t \nabla \cdot \mathcal{Q}^n + \lambda_x p^n). \end{cases}$$

5.2. Calculation of u

Another numerical (and theoretical) difficulty arises in the computation of the velocity, u. The formula $u = \mathcal{Q}/h$ cannot be used for obtaining u at Σ_t if regions without water (Ω_t^0) appear in the domain. Indeed, we noted that at Σ_t, $\mathcal{Q} \cdot n = 0$, $h = 0$, but $u \cdot n \neq 0$ in general.

Therefore, when computing u an indetermination appears which avoids dividing $\mathcal{Q}$ by h. We propose, here, to calculate u with the aid of a supplementary equation,

$$\frac{\partial u}{\partial t} + u \cdot \nabla u + g \nabla \eta = F \quad \text{in } \Omega_t^+,$$

i.e. to use the momentum conservation equation in the variables (u, η) in the region with water Ω_t^+. The boundary conditions, would then be

- $u \cdot n = 0$ in the boundary of Ω, where $\mathcal{Q} \cdot n = 0$,
- where conditions are over η, we keep the same boundary conditions,
- at Σ_t, $\eta = -H$.

For the sake of simplicity, assume $\mathcal{Q} \cdot n = 0$ at $\partial\Omega$. The cost of adding this new equation is not very high. Semi-discretizing in time again using the method of characteristics, yields

$$\frac{1}{\Delta t}u^{n+1}+g\nabla\eta^{n+1}=F+\frac{1}{\Delta t}u^n(X^n),$$

$$u^{n+1}\cdot n=0 \quad \text{at } \partial\Omega,$$

$$\eta^{n+1}=-H \quad \text{at } \Sigma_{t_{n+1}},$$

resulting in

$$\frac{1}{\Delta t}(u^{n+1},z)=g(\eta^{n+1},\nabla\cdot z)+(F,z)+\frac{1}{\Delta t}(u^n(X^n),z)+g\int_{\Sigma t_{n+1}}H(z\cdot n)\,\mathrm{d}\sigma,$$

$$\forall z\in\{z\in H_0(\mathrm{div};\ \Omega):\ z\cdot n=0 \text{ at } \partial\Omega\cap\partial\Omega^0_{t_{n+1}}\}. \tag{17}$$

Notice that this calculation is done after $\mathcal{Q}^{n+1}$ and η^{n+1} are obtained, which means that η^{n+1} is known and the function X^n has been approximated. The mass lumping technique is used.

6. SPACE DISCRETIZATION

To discretize problem (12) in space, Raviart–Thomas [8] mixed finite elements have been used. Let τ_h be a family of triangulations and consider two finite-dimensional vector spaces. V_h is the space of vector functions that are polynomials of degree one on each triangle, which is discontinuous on the edges of the elements but whose normal components are continuous and constant there. The normal components on each side are taken as degrees of freedom.

On the other hand, let M_h be the space of piecewise constant functions:

$$M_h=\{\eta_k:\ \eta_{h|_K}\in P_0,\ \forall K\in\tau_h\}.$$

Then the discretized problem consists of finding $\mathcal{Q}_h^{n+1}\in V_h$ and $\eta_h^{n+1}\in M_h$ such that

$$\begin{cases} a(\mathcal{Q}_h^{n+1},z_h)=\dfrac{g}{2}\displaystyle\int_\Omega p_h^{n+1}\nabla\cdot z_h\,\mathrm{d}x+\langle L_h,z_h\rangle,\\ p_h^{n+1}=G^{w_x}_{x,\lambda_x}\dfrac{g}{2}(\eta_h^n-\Delta t\nabla\cdot\mathcal{Q}_h^{n+1}+\lambda_x p_h^{n+1}),\\ \eta_h^{n+1}=\eta_h^n-\Delta t\nabla\cdot\mathcal{Q}_h^{n+1}. \end{cases} \tag{18}$$

The same discretization is used in (17).

The numerical algorithm that results is

Given u_0, η_0, $\mathcal{Q}_0=u_0(\eta_0+H)$.
For $n=0,1,2,\ldots$

- approximate $X^n=X(x,t^{n+1};t^n)$,
- calculate $\mathcal{Q}_h^{n+1}$, η_h^{n+1} solving (18),
- determine $\Omega^+_{t_{n+1}}$, $\Omega^0_{t_{n+1}}$, $\Sigma_{t_{n+1}}$,
- compute u_h^{n+1} using (17).

From the algorithm point of view, the most delicate part is the detection of $\Omega^0_{t_{n+1}}$. This is done by removing/adding elements. Initially, implementation has been done for the one-dimensional case, producing good results. Presently, we are working on the generalization to two-dimensional domains.

7. RESOLUTION OF THE LINEAR SYSTEMS. PRECONDITIONERS

As we have shown, at each time step t_n, the numerical scheme used here to discretize the shallow-water equations requires the solution of a fixed-point problem (step (5) in Algorithm 2). At each iteration of the fixed-point algorithm we have to solve a linear system of the form

$$A^n \mathcal{Q} = b,$$

where A^n is the matrix appearing in the space discretization of the bilinear form (9). Therefore, A^n is unchanged during the solution of the fixed point problem, but it varies with time through its dependence on w_x and must be freshly computed at each time step. Thus, the coefficients of A^n must be recalculated at each iteration in time. Nevertheless, the zero-structure of the matrices, which depends on the mesh connectivities, remains unchanged. Matrices A^n are regular but not necessarily symmetric and, in general, their conditioning numbers, $K(A^n)$, are high depending upon w_x. This entails an additional difficulty if a generalized conjugate gradient method is going to be used. Consequently, a good preconditioner C^n must be chosen.

To improve the convergence speed of the method, we initially use the preconditioner IC(0), i.e. the preconditioner matrix is an LU incomplete factorization with the same zero-structure as the matrix. Although this preconditioner appears efficient in numerical simulations performed with small time steps and w fixed, the computing time increases considerably with increasing time steps and large variations of w_x, sometimes even leading to problems of convergence. A detailed analysis of the linear problems to be solved lead us to consider other preconditioners.

From the expression of the bilinear form of (9) it can be proved that

$$K(A^{n+1}) \leq c_n K(A^n),$$

where c_n is a constant depending on the fluctuation of w_x^n and w_x^{n+1}. As w_x^n depends on the depth of the water column (is twice the thickness of the layer), and the fluctuations of this thickness are supposed to be small from one time iteration to the next one, we can expect that $c_n \approx 1$. Keeping this in mind, it stands to reason that if C^n is a good preconditioner for A^n it will also be so for A^{n+1}.

A first approximation consists in determining the zero-structure of all the preconditioners at the first time iteration. This structure is to be conserved along the whole integration. The procedure to build the preconditioners can be split in the following steps:

1. Compute the LU-complete factorization of the matrix A^0 at time $t = 0$.
2. For each diagonal d_i of the matrix L (respectively U) compute the norm

$$\|d_i\| = \sqrt{\frac{\sum_{l=1}^{m-i} a_{l+i,l}^2}{m-i}},$$

   where m represents the number of degrees of freedom and the index i gives the ith diagonal ($i = 0$ for the principal diagonal).
3. The k% of the non-zero elements of L and U (k being a parameter to be provided to the algorithm) is retained. This is done by the elimination of the diagonals with smallest norms.
4. $C^n = C^0$, $\forall n$.

This preconditioner, which we denote as IC(k), is efficient for a suitable choice of k even with large time steps. Nevertheless, if the number of iterations to be performed is too high, problems of convergence may arise; this may happen when $K(A^n)$ strongly differs from $K(A^0)$.

To avoid this problem, the coefficients of the preconditioning matrix C^n can be recalculated each time a certain number of iterations has been completed, but using the zero-structure of the preconditioning matrix determined at the first iteration. In order to do this, the incomplete LU factorization of A^n is performed. By this stratagem, we avoid computing a complete LU factorization and the calculation of the diagonal norms. This modified preconditioner, although computationally slightly more expensive than the previous preconditioner, is very efficient with any time step and does not present convergence problems (see Tables I and II for numerical results). For a detailed description on the preconditioners used see [9].

8. THE MULTI-LAYER MODEL

A sharp density gradient can be generated in the sea water by solar heating of the upper layer or in a strait connecting two basins where waters have different densities, as is the case in the Strait of Gibraltar. In these cases, fresher water flows over more saline, and consequently, heavier water. It may even happen that water in adjacent layers flows in opposite directions. In that situation, a shallow water model is not suitable. On the one hand, density variations are important; on the other hand, large velocity fluctuations along the water column can be expected.

In this section, a model is proposed that considers sea water as composed of several immiscible layers of different constant densities. In such a model, waves appear not only on the surface but also in the interface between the layers. It will be assumed that phenomena to be modelled have wavelengths large enough to make appropriate the shallow water approximation in each layer.

8.1. The general model

In this section, we use a generalization of the notation introduced in Figure 2 for the two-layer case. For this case, two constant reference levels, A_1 and A_2, have to be chosen. More precisely, A_2 is the reference level for the deeper layer and $A = A_1 + A_2$ is the reference level for the upper one (mean level of the interface and sea level respectively). In the general case, as many reference levels as layers must be taken.

The equations obtained in the general case of m layers are (see [10,11] for further details on the derivation of these equations):

(a) First layer

$$\frac{\partial \mathcal{Q}_1}{\partial t} + \nabla\cdot(u_1 \otimes \mathcal{Q}_1) + \frac{g}{2}\nabla(\eta_1^2 + 2\eta_1 H_1) = g\eta_1 \nabla H_1,$$

$$\frac{\partial h_1}{\partial t} + \nabla\cdot\mathcal{Q}_1 = 0.$$

(b) kth layer

$$\frac{\partial \mathcal{Q}_k}{\partial t} + \nabla\cdot(u_k \otimes \mathcal{Q}_k) + \frac{g}{2}\nabla(\bar{\eta}_k^2 + 2\bar{\eta}_k \bar{H}_k) = g\bar{\eta}_k \nabla \bar{H}_k,$$

$$\frac{\partial h_k}{\partial t} + \nabla\cdot\mathcal{Q}_k = 0.$$

where

$$\bar{\eta}_k = \eta_k + \frac{\rho_1}{\rho_k} h_1 + \cdots + \frac{\rho_{k-1}}{\rho_k} h_{k-1}$$

and

$$\bar{H}_k = H_k - \frac{\rho_1}{\rho_k} h_1 - \cdots - \frac{\rho_{k-1}}{\rho_k} h_{k-1}.$$

(c) Lower layer

$$\frac{\partial \mathcal{Q}_m}{\partial t} + \nabla \cdot (u_m \otimes \mathcal{Q}_m) + \frac{g}{2} \nabla(\bar{\eta}_m^2 + 2\bar{\eta}_m \bar{H}_m) = g\bar{\eta}_m \nabla \bar{H}_m,$$

$$\frac{\partial h_m}{\partial t} + \nabla \cdot \mathcal{Q}_m = 0.$$

where

$$\bar{\eta}_m = \eta_m + \frac{\rho_1}{\rho_m} h_1 + \cdots + \frac{\rho_{m-1}}{\rho_m} h_{m-1}$$

and

$$\bar{H}_m = H_m - \frac{\rho_1}{\rho_m} h_1 - \cdots - \frac{\rho_{m-1}}{\rho_m} h_{m-1}.$$

The reason for these new 'overbar' variables is to produce exactly the same equations for any layer as in the one-layer case. Thus, the non-linear operator studied above will appear again in the solution of the equations at each layer, and the problems to be solved become analogous of those already studied.

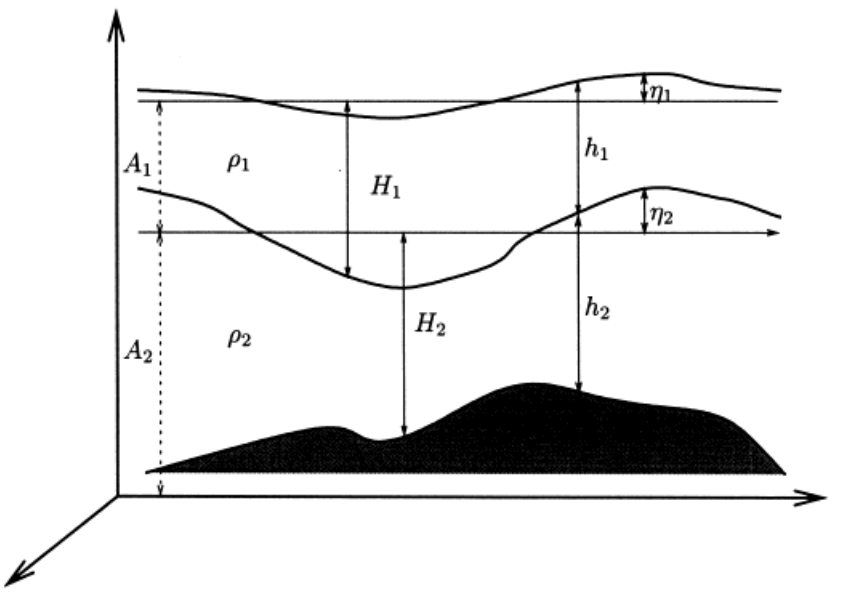

Figure 2. The two-layer model scheme.

8.2. Numerical resolution

For the sake of simplicity, in this section we present the numerical resolution of the two-layer system. The generalization to the multi-layer case is straightforward and has already been implemented [10]. The algorithm used for solving the equations at each layer is the same as that presented for the one-layer case.

Let Δt be the given time step and let $\mathcal{Q}_i^n$, η_i^n be Q_i and η_i approximations at time $n\Delta t$. If $\mathcal{Q}_i^n$, η_i^n, $i=1,2$ are known, then $\mathcal{Q}_i^{n+1}$, η_i^{n+1}, $i=1,2$ are computed as follows:

(a) Second layer
Using the same as in the lower layer (here the second) the following equality holds:

$$\frac{\partial h_2}{\partial t}=\frac{\partial \eta_2}{\partial t},$$

and discretizing the mass conservation equation for this layer equation by the implicit Euler method, we have

$$\eta_2^{n+1}=\eta_2^n-\Delta t\nabla\cdot\mathcal{Q}_2^{n+1}.$$

The time discretization of the momentum equation is carried out using the method of characteristics. Then the unknowns $\bar{\eta}_2^{n+1}$ and $\bar{H}_2^{n+1}$ are computed, where $\bar{\eta}_2$ and $\bar{H}_2$ are given as above. However, the thickness of the first layer at time $(n+1)\Delta t$, which has not yet been computed, appears in these expressions. In order to uncouple the resolution of the equations corresponding to each layer, the following explicit approximation is considered:

$$\bar{H}_2^{n+1}=H_2-\frac{\rho_1}{\rho_2}h_1^n,$$

$$\bar{\eta}_2^{n+1}=\eta_2^{n+1}-\frac{\rho_1}{\rho_2}h_1^n=\eta_2^n-\Delta t\nabla\cdot\mathcal{Q}_2^{n+1}+\frac{\rho_1}{\rho_2}h_1^n.$$

With this approximation, the non-linear boundary problem, which must be solved to compute $\mathcal{Q}_2^{n+1}$ is identical to those arising in the one-layer model. It is just possible that this explicit approximation may affect the stability properties of the algorithm: theoretical studies of the question remain to be done. In any event, in the numerical computations done so far no problem of stability has occurred.

(b) First layer
Using the same as that for any layer not the lower, we have

$$\frac{\partial h_k}{\partial t}=\frac{\partial \eta_k}{\partial t}-\frac{\partial \eta_{k+1}}{\partial t}.$$

In the two-layer case, this gives

$$\frac{\partial h_1}{\partial t}=\frac{\partial \eta_1}{\partial t}+\nabla\cdot\mathcal{Q}_2.$$

Using this expression, the continuity equation for the first layer can be rewritten as

$$\frac{\partial \eta_1}{\partial t}=-\nabla\cdot\mathcal{Q}_1-\nabla\cdot\mathcal{Q}_2.$$

The implicit Euler discretization then gives

$$\eta_1^{n+1} = \eta_1^n - \Delta t \nabla \cdot \mathcal{Q}_1^{n+1} - \Delta t \nabla \cdot \mathcal{Q}_2^{n+1}. \tag{19}$$

Notice that, since the computations begin from the deeper layer, $\mathcal{Q}_2^{n+1}$ is known at this algorithm step; therefore, a new explicit approximation is not necessary.

Again, the momentum equation is discretized using the method of characteristics. On applying (19) to the resulting expression, a boundary problem is obtained with $\mathcal{Q}_1^{n+1}$ as the only unknown. This problem is analogous to that occurring in the discretization of the second layer equation, and is solved using the same algorithm.

9. NUMERICAL EXAMPLES

9.1. A one-dimensional test case

In order to demonstrate the capability of the numerical algorithm presented in this contribution in handling regions without water, in this first example we show some results for a one-dimensional case. The domain is the interval [0, 10] and the depth is given by the function

$$H(x) = 1 + \cos\left(\frac{2\pi x}{5}\right).$$

Initially, the water is at rest. Then, water leaves the domain through the edges $x = 0$ and $x = 10$, where outflowing boundary conditions are prescribed. Two regions without water develop on the slopes of the hills, while the water within the valley remains at rest. Figure 3 depicts the time evolution of the water in this situation. In these figures, the water surface is represented by the solid line, land topography by the dashed line and the dotted line depicts the level of the water at rest.

9.2. A channel

This section presents a comparison between Algorithm 2 and the Bermúdez–Moreno algorithm with constant coefficients (Algorithm 1). The computational domain for this comparison is a 64 km long (x_1), 21 km wide (x_2) channel (dimensions comparable with those of the Strait of Gibraltar). A 1 Sv (1 Sv $= 10^6$ m^3 s^{-1}) input flow is imposed at the edge of the channel located at $x_1 = 0$. The same output is imposed at the other end of the channel located at $x_1 = 64$. At the channel walls, the normal component of the flow is set equal to zero. Initially, the water is taken at rest and with no elevation. The depth of the channel is given by the following sinusoidal function:

$$H(x_1, x_2) = 90 \cos\left(\frac{x_1}{4000} + \frac{\pi}{2}\right) + 100.$$

Problem (2) has been solved on the domain described above for times leading to a stationary state using Algorithms 1 and 2. Two different meshes have been considered for this comparison: first, a coarse mesh containing 1111 nodes, and second, a finer mesh with 4950 nodes. In Algorithm 1, the constant values of λ and w are taken as the mean of the two extreme optimal values obtained when the stationary state is reached with Algorithm 2. The time step is $\Delta t = 120$ s, and with both algorithms the stationary state is achieved after 800 iterations (≈ 26.6 h).

Table I summarizes some data referring to Test 1: a channel problem with the coarse mesh computations. The total CPU time consumed by both algorithms (using a machine with a Pentium 150 MHz processor and the Linux operating system) is displayed for a range of values of the parameter k. Also shown are the mean number of iterations needed by the iterative method to solve the linear systems and by the duality method used to solve the non-linear problems. The number of non-zero elements in the preconditioning matrices is also shown. Notice that, in spite of the additional cost due to matrix and preconditioner computations for Algorithm 2 (matrices depends on w_x and the coefficients of the preconditioning matrix are recalculated every ten iterations) its superior rate of convergence results in a much lower total CPU time consumed than for Algorithm 1.

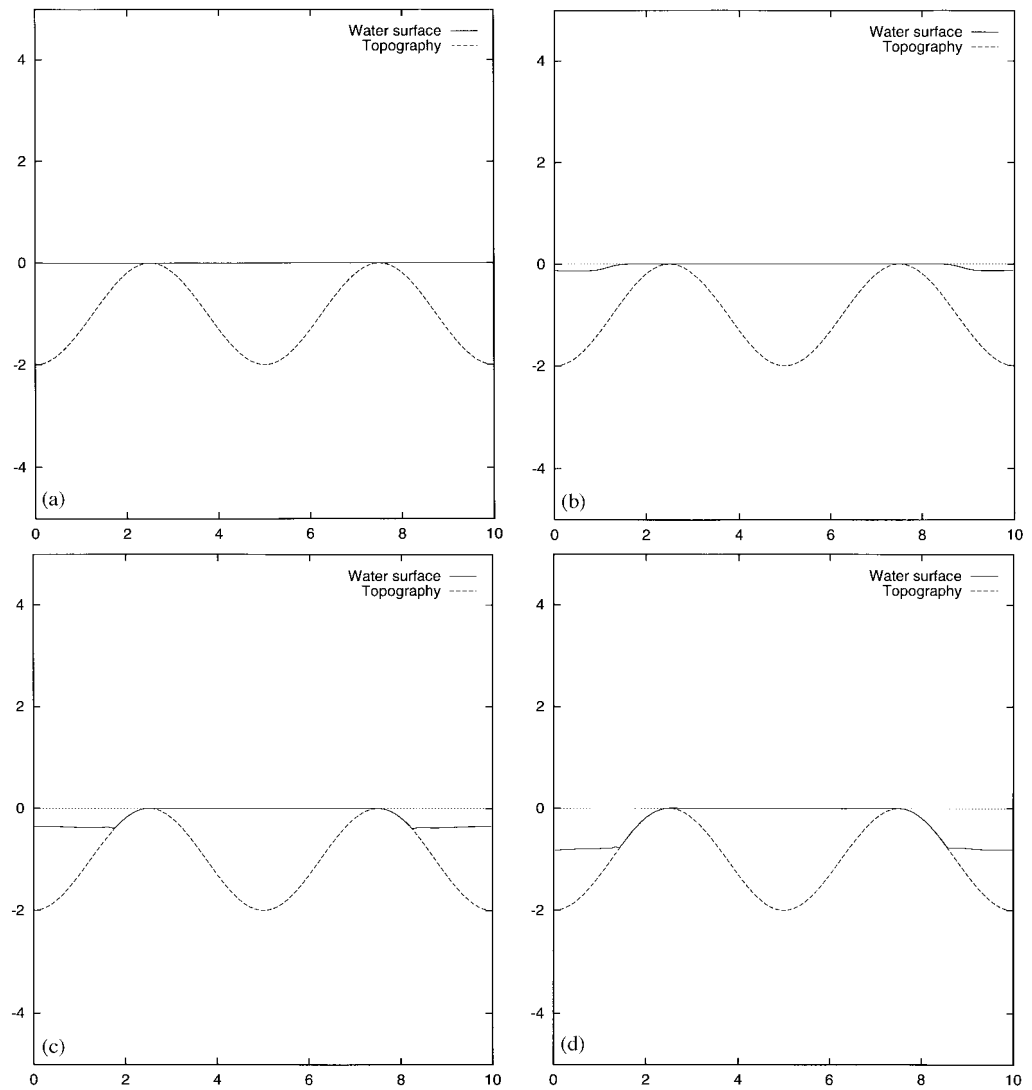

Figure 3. Emergence of land areas in a one-layer system: initial state and time steps 25, 100 and 150.

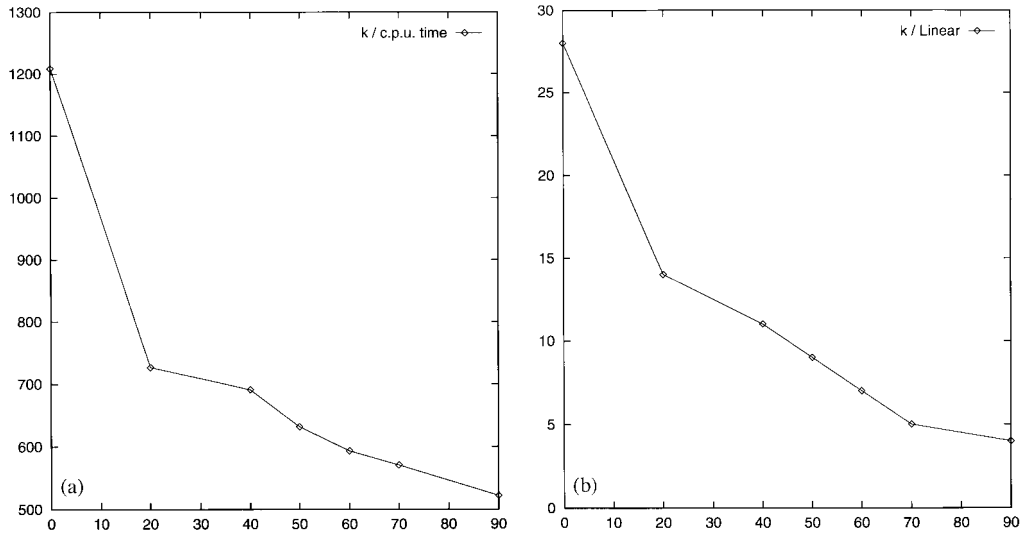

Figure 4. Test 1: coarse mesh. CPU time (left panel) and mean number of iterations needed to solve the linear problems (right panel) as a function of k for Algorithm 2.

Figure 4 shows the CPU time and mean number of iterations needed to solve the linear problems as functions of the parameter k for Algorithm 2. In both graphs, notice the decrease with increasing k. This feature need not always occur; an inflection in these curves may appear at some other value of k. Notice, nevertheless, that the reduction in computing time and number of iterations is steepest for values of k between 0 and 20.

A comparison of convergence curves for Algorithms 1 and 2 ($k = 40$) at time steps 100 (left panel) and 225 (left panel) is displayed in Figure 5. The abscissa represents the number of iterations of the duality algorithm and the ordinate is the error in logarithmic scale.

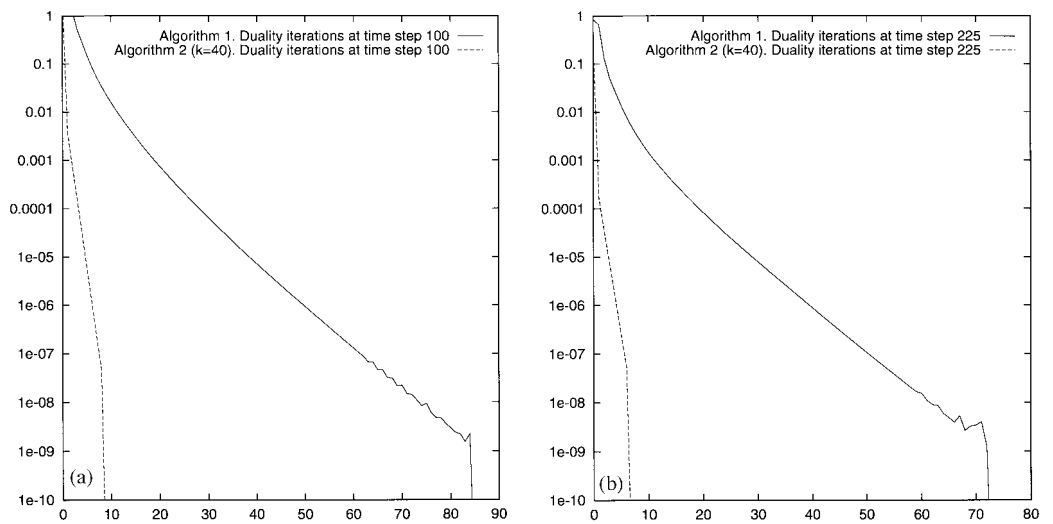

Figure 5. Convergence curves for the duality algorithm. Solid lines for Algorithm 1 and dashed lines for Algorithm 2 with $k = 40$. The left graphic is for iteration 100 and the right graphic for iteration 225.

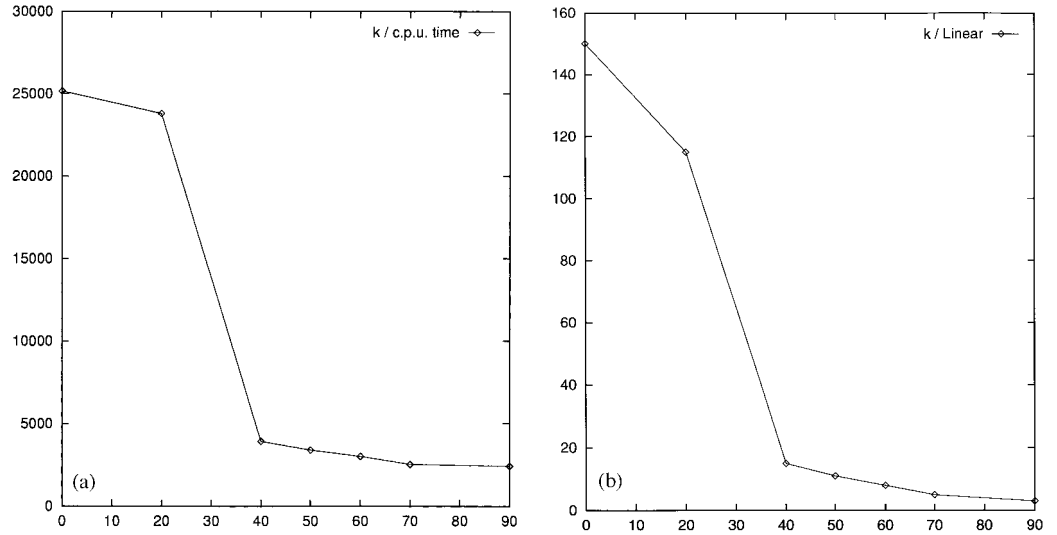

Figure 6. Test 2: refined mesh. CPU time (left panel) and mean number of iterations needed to solve the linear problems (right panel) as a function of k for Algorithm 2.

Table II gathers the data from the numerical experiments performed with the refined mesh: Test 2. Note that if Algorithm 1 is used, the total CPU time is as much as 303731 units, while if Algorithm 2 with $k = 40$ is used, the computer time is only 3921 units. This represents a reduction in CPU time of over 1/75 for an increase of just five times of the memory required.

Finally, Figure 6 shows for Algorithm 2 the total CPU time (left panel) and the mean number of iterations needed to solve the linear problems (right panel) as a function of the parameter k. Again, a larger rate of decrease is seen in both quantities for values of k between 0 and 40.

9.3. The Alboran Sea

Observational data (satellite, aircraft and *in situ* data) have shown that two anti-cyclonic gyres, the western and the eastern gyres, are major ocean features of the Alboran Sea [12]. A large variation in the structure, size and position of these gyres have been observed. It may even happen that one or the other gyre disappears during certain periods, although the disappearance of both gyres at the same time has never been observed. This changing structure is due to several causes: wind, bottom and Coriolis effects, properties of the incoming Atlantic water (vorticity, angle, internal waves, etc.).

The results presented in this section are part of a larger set of numerical experiments aimed to study the gyre and its variations with different constant winds (see [6]). Bottom and Coriolis effects were taken into account. The two-layer model was used: the upper layer (initially at 80 m depth) represents for the inflowing Atlantic water that enters through the Strait of Gibraltar and exits into the western Mediterranean basin. In this layer, the constant value for the water density is taken to be equal to 1027 kg m^{-3}. The lower layer stands for the Mediterranean water pouring out from the eastern Mediterranean into the Atlantic. The constant density value taken in this layer was 1029 kg m^{-3}.

The physical domain considered presents four different boundaries. Two 'natural' boundaries, corresponding to the Spanish and Moroccan coasts, and two 'artificial' boundaries

limiting the computational domain to the east and west. The western boundary near Tarifa, in the Strait of Gibraltar, and the eastern limit constituted by the sides of a large rectangle. These sides follow the orientation of the Spanish and African coasts at that part of the Mediterranean. The meshes were generated from digitalized cartographic data provided by the Instituto Español de Oceanografía (IEO). When real bottom topography was considered, the bathymetry function H was computed from digital cartographic data by means of an automatic interpolation process over the mesh vertices. From this discrete function and the first layer mesh, the second and subsequent layer meshes are automatically constructed suppressing the spare triangles, i.e. the elements k in the first layer, with $H(k) > -80$ m are suppressed to obtained the second-layer mesh.

The boundary conditions considered were as follows: at the first layer, the input flux was imposed over the Strait of Gibraltar. A total flux of 1 Sv was taken, which is the estimate corresponding to the annual mean of the Atlantic input flux (see [13]). The profile considered for this input flow was designed to fulfil a certain criterion of conservation of the potential vorticity (see [6,14] for further details). On the sides of the rectangle, coast conditions were imposed at both layers. For the second layer at the Strait, an output symmetric to the first layer input was imposed. For preserving the mass conservation at each layer was necessary to consider an exchange between layers of 1 Sv. This exchange was imposed just in the most eastern part of the domain, the region limited by the rectangle. As external forcing, a 20 km coastal eastern wind of 10 m s^{-1} was imposed, trying to simulate the effects of coastal winds on the development of the gyres. The time step was 15 min.

As initial conditions, the results shown here restart from a previous experiment where a stationary state was reached. This experiment was performed with the same boundary conditions described above but without wind, and started from a resting sea (i.e. elevation and fluxes initially set to zero).

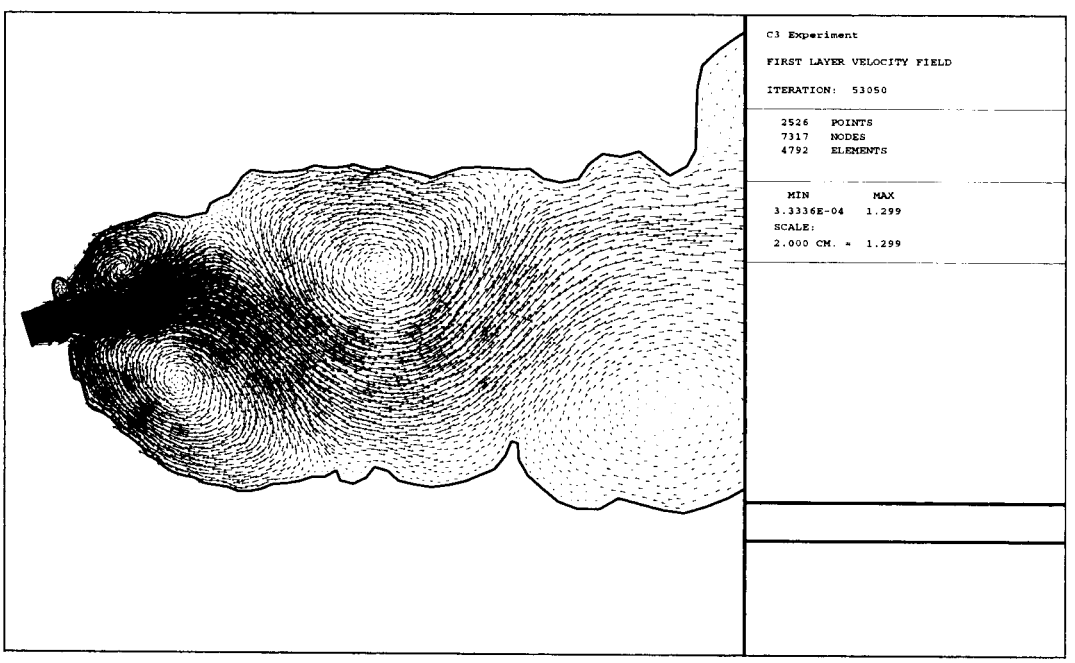

Figure 7. First layer simulated velocity field (see text for details).

Figure 7 shows the first layer velocity field for the experiment described above after 3000 iterations (≈ 1 month real time). It illustrates a 'typical' two anti-cyclonic (clockwise in the Northern Hemisphere) gyre configuration as found in many observational studies, with the western anti-cyclonic gyre confined to the western most part of the Alboran Sea and to the south of the inflowing Atlantic water. To the north of the inflowing current, cyclonic (anti-clockwise) structures often form: they are also present in this simulation. Finally, a second anti-cyclonic gyre usually forms in the eastern basin, to the east of Cape Tres Forcas as shown in Figure 7. For further details on the Alboran Sea dynamics, see the introduction to [6] and references therein.

The simulated elevation of the sea surface is presented in Plate 1, while Plate 2 shows the movement of the interface. The anti-cyclonic gyres tend to accumulate water at their interior, producing an elevation of the sea surface and a sinking of the interface. The opposite occurs for cyclonic gyres. These features can be observed in the simulation shown here.

10. FINAL REMARKS

It has been shown that the duality algorithm considered in this paper greatly improves convergence speed and, consequently, the CPU time needed to solve the shallow-water problems that have been proposed when the automatic choice of parameters and the family of preconditioners described in previous sections are used.

We remarked that the problems put forward in this contribution arose from our interest in modelling the dynamic features of the Alboran Sea. The numerical model chosen seems able to simulate a number of the phenomena taking place in this region of the Mediterranean (see [6] for further details) with good accuracy and at a reasonable computational cost.

In order to improve the model, we are carrying out the following research activities:

- Dealing with vanishing thickness regions. The use of penalization techniques to improve convergence, and the generalization of the results presented here for the one-dimensional example to the two-dimensional case and, as a final goal, to the multi-layer system, are the main lines of research on this topic.
- Generalizing the choice of boundary conditions. Studying the possibility of using data assimilation techniques.
- Implementation of adaptative mesh techniques.
- Implementation and comparison of other time and/or space discretizations, including second-order schemes.
- Testing the model under more realistic wind and boundary conditions. Comparing the results with previous works and experimental measurements.
- Application of the model to the study of the variability of the Alboran Sea gyres.
- Analyzing the mathematical model. At the present moment, we are trying, in collaboration with P. Orenga (Université de Corse), to generalize to the multi-layer case the results proved for the one-layer system (cf. [15]).

ACKNOWLEDGMENTS

This research have been partially supported by the C.I.C.Y.T. (project MAR97-1055-C02-01).

APPENDIX A

Let E be a real Hilbert space and w a real number. A multi-valued operator G in E is called a *maximal-$M(w)$* operator if $G + wI$ is a maximal monotone, where I is the identity operator (see [16]).

Let $\lambda > 0$ be a real number such that $0 < \lambda w < 1$. Then, the operator $J_\lambda = (I + \lambda G)^{-1}$ is defined over all E and is univalued. Moreover, it is a Lipschitz function with constant $(1 - \lambda w)^{-1}$. In this case, we can define the Yosida approximation of G by

$$G_\lambda = \frac{I - J_\lambda}{\lambda}. \tag{20}$$

The following results can be shown (see [17]):

1. If $\lambda w \leq \frac{1}{2}$, G_λ is a Lipschitz continuous function with constant $1/\lambda$.
2. If $\lambda w < 1$, the following statements are equivalent:
   - $\theta \in G(u)$;
   - $\theta = G_\lambda(u + \lambda_\theta)$.

REFERENCES

1. A. Bermúdez, C. Rodríguez and M.A. Vilar, 'Solving shallow water equations by a mixed implicit finite element method', *IMA J. Numer. Anal.*, **11**, 79–97 (1991).
2. A. Bermúdez and C. Moreno, 'Duality methods for solving variational inequalities', *Comp. Maths. Appl.*, **7**, 43–58 (1981).
3. O. Pironneau, *Méthodes des Éléments Finis pour les Fluides*, vol. 7 of RMA, Masson, Paris, 1988.
4. J.F. Rodrigues, *Obstacle Problems in Mathematical Physics*, North-Holland, Amsterdam, 1987.
5. I. Ekeland and R. Temam, *Convex Analysis and Variational Problems*, *Studies in Mathematics and its Applications*, vol. 1, North-Holland, Paris, 1976.
6. J. Macías, 'Some topics in numerical modelling in oceanography', *PhD Thesis*, University of Paris VI, Paris, 1998.
7. C. Pares, J. Macías and M.J. Castro, 'Duality methods with automatic choice of parameters. Application to shallow-water equations in conservative form', *Internal Journal 0115*, Group on Differential Equations, Numerical Analysis and Applications, University of Malaga, Spain, 1998; also submitted to *Numerische Mathematik*.
8. P.A. Raviart and J.M. Thomas, 'A mixed finite element method for 2nd order elliptic problems', in *Mathematical Aspects of Finite Element Methods. Lecture Notes in Mathematics*, vol. 606, Springer, Berlin, 1977, pp. 292–313.
9. M.J. Castro, J. Macías and C. Parés, 'The design of an IC-type preconditioner. Description and applications', *Internal Journal 0130*, group on Differential Equations, Numerical Analysis and Applications, University of Málaga, Spain, 1998.
10. M.J. Castro and J. Macías, *Modelo Matemático de las Corrientes Forzadas por el Viento en el Mar de Alborán*, vol. 5, Publicaciones del Grupo de Análisis Matemático Aplicado de la Universidad de Málaga, 1994.
11. M.J. Castro, J. Macías and C. Pares, 'A multilayer shallow water model. Applications to the modelling of the Alboran Sea and the Strait of Gibraltar', in J.I. Díaz (ed.), *The Mathematics of Models for Climatology and Environment*, Heidelberg, Germany, 1996, pp. 367–394; NATO ASI Series I: Global Environmental Change, No. 48, Springer, New York.
12. G.W. Heburn and P.E. La Violette, 'Variations in the structure of the anticyclonic gyres found in the Alboran Sea', *J. Geophys. Res.*, **95**, 1599–1613 (1990).
13. H. Bryden, J. Candela and T.H. Kinder, 'Exchange through the Strait of Gibraltar', *Prog. Oceanogr.*, **33**, 201–248 (1994).
14. J. Macías, 'Deduction of a boundary condition for the input flow through the Strait of Gibraltar by a conservation of the potential vorticity approach', *Internal Journal 1027*, group on Differential Equations, Numerical Analysis and Applications, University of Málaga, 1997.
15. P. Orenga, 'Un théorème d'existence de solutions d'un problème de shallow water', *Arch. Rational Mech. Anal.*, **130**, 183–204 (1995).
16. A. Pazy, 'On the asymptotic behaviour of iterates of nonexpansive mappings in Hilbert spaces', *Israel J. Math.*, **26**, 197–204 (1977).
17. H. Brézis, *Operateurs Maximaux Monotones et Semi-groupes de Contractions dans les Espaces de Hilbert*, North-Holland, Amsterdam, 1973.

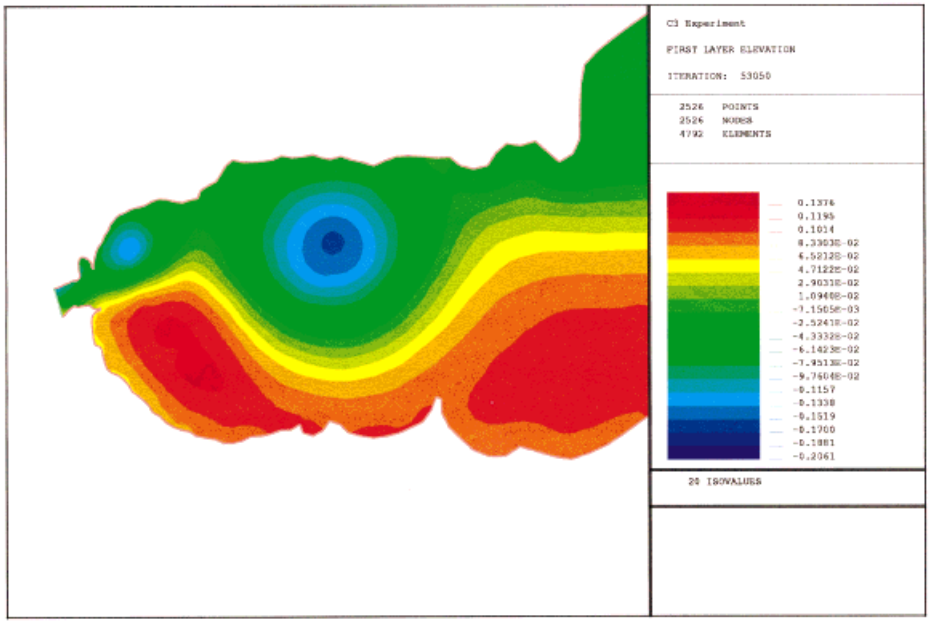

Plate 1. Simulated elevation of the sea surface.

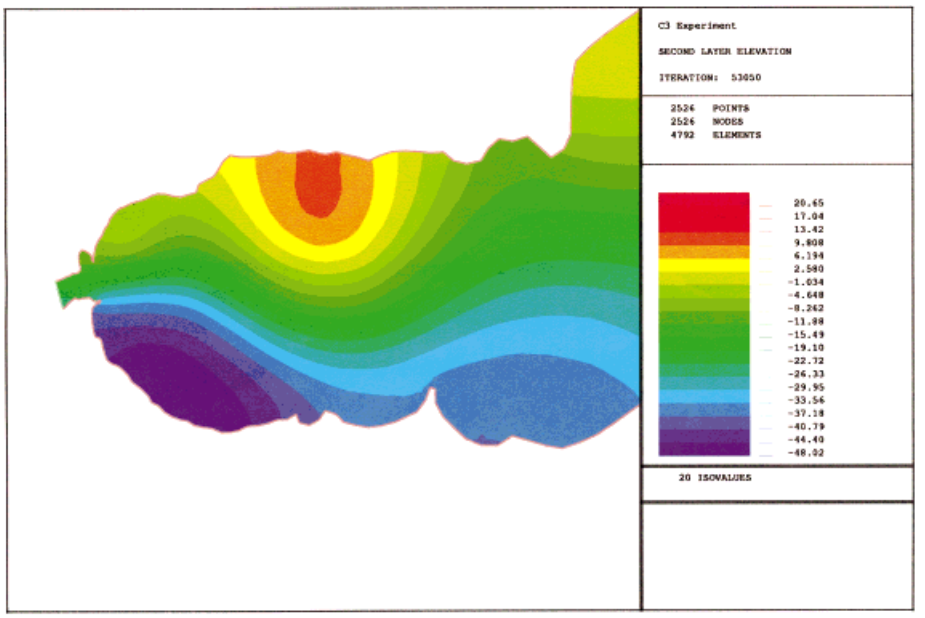

Plate 2. Simulated elevation/sinking of the interface above/under its mean level, located this at 80 m depth.