Article

Adaptive numerical modelling of tsunami wave generation and propagation with FreeFem++

Georges Sadaka [1,†,*], **Denys Dutykh** [2,†]

[1] Laboratoire de Mathématiques Raphaël Salem, Université de Rouen Normandie, CNRS UMR 6085, Avenue de l'Université, BP 12, F-76801 Saint-Étienne-du-Rouvray, France; Georges.Sadaka@univ-rouen.fr

[2] Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, LAMA, 73000 Chambéry, France; Denys.Dutykh@univ-smb.fr

[*] Correspondence: Georges.Sadaka@univ-rouen.fr; Tel.: +33(0)781978413 (G.S.)

[†] These authors contributed equally to this work.The present paper is also available as a preprint [1].

Academic Editor: Anawat Suppasri
Version August 26, 2020 submitted to Geosciences

Abstract: A simplified nonlinear dispersive Boussinesq system of the Benjamin–Bona–Mahony (BBM)-type, initially derived in [2], is employed here in order to model the generation and propagation of surface water waves over variable bottom. The simplification consists in prolongating the so-called Boussinesq approximation to bathymetry terms as well. Using the finite element method and the `FreeFem++` software [3], we solve numerically this system for three different complexities for the bathymetry function: a flat bottom case, a variable bottom in space, and a variable bottom both in space and in time. The last case is illustrated with the Java 2006 tsunami event. This article is designed to be a pedagogical paper presenting to tsunami wave community a new technology and a novel adaptivity technique along with all source codes necessary to implement it.

Keywords: tsunami wave; finite elements; mesh adaptation; domain adaptation; co-seismic displacements; tsunami wave energy; `FreeFem++`; unstructured meshes

PACS: 47.35.Bb; 02.60.-x

MSC: 76B15; 65N30; 65N50

1. Introduction

Tsunami waves represent undeniably a complex natural process. Moreover, they represent a major risk for exposed coastal areas including also the local populations, infrastructure, *etc*. The present work is devoted to the modelling tsunami generation and propagation processes. Moreover, this article is designed as a tutorial paper in order to show to the readers how easily these processes can be modelled in the framework of the `FreeFem++`, which is a free software (under the `LGPL` license). `FreeFem++` offers a large variety of triangular finite elements (linear and quadratic Lagrangian elements, discontinuous $\mathbb{P}_1$, Raviart–Thomas elements, *etc.*) to solve Partial Differential Equations (PDEs). It is an integrated product with its own high level programming language and a syntax close to mathematical formulations, making the implementation of numerical algorithms very easy. Among the features making `FreeFem++` an easy-to-use and highly adaptive software we recall the advanced automatic mesh generator, mesh adaptation, problem description by its variational formulation, automatic interpolation of data, color display on line, postscript printouts, *etc*. The `FreeFem++` programming framework offers the advantage to hide all technical issues related to the implementation of the finite element method.

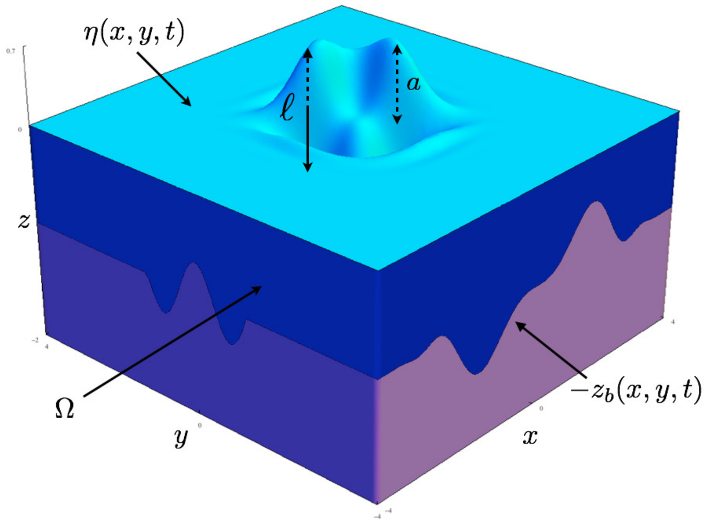


Figure 1. *The sketch of the physical domain* Ω.

Traditionally, tsunami waves are modelled using hydrostatic models [4–7]. In the present manuscript we employ a non-hydrostatic BOUSSINESQ-type system to be specified below. This class of models is distinguished by the application of the so-called BOUSSINESQ approximation [8]. They can be used to study a variety of water wave phenomena in harbors, coastal dynamics and, of course, tsunami generation and propagation problems [9–13].

In this study we consider an sBBM system derived by D. MITSOTAKIS (2009) in 2D over a variable bottom in space $h(x, y)$ and in time $\zeta(x, y, t)$ [2]:

$$\begin{aligned} \eta_t + \nabla \cdot ((h+\eta)V) + \zeta_t + \tilde{A}\nabla \cdot \left(h^2 \nabla \zeta_t\right) + \nabla \cdot \left\{ Ah^2 \left[\nabla\left(\nabla h \cdot V\right) + \nabla h \nabla \cdot V\right] - bh^2 \nabla \eta_t \right\} &= 0\,, \\ V_t + g\nabla\eta + \frac{1}{2}\nabla |V|^2 + Bgh\left[\nabla\left(\nabla h \cdot \nabla\eta\right) + \nabla h \Delta\eta\right] - dh^2 \Delta V_t - Bh\nabla\zeta_{tt} &= 0\,, \end{aligned} \tag{1}$$

where

$$\widehat{a} = \left(\theta - \frac{1}{2}\right), \qquad \widehat{b} = \frac{1}{2}\left((\theta-1)^2 - \frac{1}{3}\right), \qquad \tilde{A} = \nu\widehat{a} - (1-\nu)\widehat{b}, \qquad A = -\widehat{b}, \qquad B = 1-\theta,$$

$$b = \frac{1}{2}\left(\theta^2 - \frac{1}{3}\right)(1-\nu), \qquad d = \frac{1}{2}\left(1-\theta^2\right)(1-\mu).$$

Constants θ, μ, ν are real parameters and g is the acceleration due to gravity. System (1) is an asymptotic approximation to the three-dimensional full EULER equations describing the irrotational free surface flow of an ideal fluid $\Omega \subset \mathbb{R}^3$ [14,15], which is bounded below by $-z_b(x, y, t) = -h(x, y) - \zeta(x, y, t)$ and above by the free surface elevation $\eta(x, y, t)$ (*cf.* Figure 1). The system (1) can be considered as a generalization of the classical BOUSSINESQ system put forward by D. PEREGRINE [16,17].

The variables in (1) are $\mathbf{X} = (x, y) \in \Omega$ and $t > 0$ are proportional to position along the channel and time, respectively. $\eta = \eta(\mathbf{X}, t)$ being proportional to the deviation of the free surface departing from its rest position and $V = V(\mathbf{X}, t) = \begin{pmatrix} u(\mathbf{X}, t) \\ v(\mathbf{X}, t) \end{pmatrix} = (u, v)^\top = (u; v)$ being proportional to the horizontal velocity of the fluid at some height. In our study, we suppose that $\eta = \mathcal{O}(a)$, with the characteristic wave amplitude a (in other words, η is the difference between the water free surface and the still water level). Also we set $\lambda = \mathcal{O}(\ell)$ be the wave length. In addition, we limit ourselves to the case where $\eta + z_b > 0$ (there are no dry zones in our computations).

This paper is organized as follows. In Section 2 we present the space and time discretization of a simplified version (4) of Equations (1). In Section 3, we present the new domain adaptation technique. In Section 4, we establish the convergence of our numerical code, which validates the adequacy of the chosen finite element discretization. Then, with this code we simulate the propagation of a tsunami-like wave generated by the moving bottom (*e.g.* an earthquake). We present several test cases in various regions of the world. First, we take a MEDITERRANEAN sea-shaped computational domain with flat bottom and we solve the sBBM system[1] (1) in it. The mesh in this study is generated from a space image. Then, we consider the JAVA island region with real world bathymetry. Finally, we apply this solver to simulate a realistic example of a tsunami wave near the JAVA island which took place in 2006. The main conclusions of this study are outlined in Section 5.

2. Discretization of the sBBM system

In this section, we present the spatial discretization of (1) using Finite Element Method (FEM) with $\mathbb{P}_1$ continuous piecewise linear elements. For the time marching scheme we use an explicit second order RUNGE–KUTTA method.

2.1. Spatial discretization

We let Ω be a convex, plane domain, and $\mathcal{T}_h$ be a regular, quasi-uniform triangulation of Ω with triangles of maximum size $h < 1$. Setting $V_h = \{v_h \in C^0(\bar{\Omega}); v_h\big|_T \in \mathbb{P}_1(T), \forall T \in \mathcal{T}_h\}$ be a finite-dimensional, where $\mathbb{P}_1$ is the set of all polynomials of degree ≤ 1 with real coefficients and denoting by $\langle \cdot ; \cdot \rangle$ the standard L^2 inner product on Ω, we consider the weak formulation of System (1): find $\eta_h, u_h, v_h \in V_h$ such that $\forall \phi_h^\eta, \phi_h^u, \phi_h^v \in V_h$, we have:

$$\begin{aligned}
&\Big\langle \eta_{ht} - b\nabla\cdot\left(h^2\nabla\eta_{ht}\right) + \nabla\cdot\left(\left(h+\eta_h\right)\left(u_h;v_h\right)\right) + \zeta_t ; \phi_h^\eta \Big\rangle + \Big\langle \tilde{A}\nabla\cdot\left(h^2\nabla\zeta_t\right);\phi_h^\eta \Big\rangle \\
&\qquad\qquad + \Big\langle \nabla\cdot\left\{Ah^2\left[\nabla\left(\nabla h\cdot\left(u_h;v_h\right)\right) + \nabla h\nabla\cdot\left(u_h;v_h\right)\right]\right\};\phi_h^\eta \Big\rangle = 0, \\
&\langle u_{ht} - dh^2\Delta u_{ht} + g\eta_{xh} + u_h u_{hx} + v_h v_{hx} - Bh\zeta_{xtt};\phi_h^u\rangle + Bg\Big\langle h\Big[\left(\nabla h\cdot\nabla\eta_h\right)_x + h_x\Delta\eta_h\Big];\phi_h^u\Big\rangle = 0, \\
&\Big\langle v_{ht} - dh^2\Delta v_{ht} + g\eta_{yh} + u_h u_{hy} + v_h v_{hy} - Bh\zeta_{ytt};\phi_h^u\Big\rangle + Bg\Big\langle h\Big[\left(\nabla h\cdot\nabla\eta_h\right)_y + h_y\Delta\eta_h\Big];\phi_h^v\Big\rangle = 0.
\end{aligned} \tag{2}$$

For simplicity, we set $\phi_h^\eta = \Phi^\eta$, $\phi_h^u = \Phi^u$, $\phi_h^v = \Phi^v$, $\eta_h = \mathcal{E}$, $u_h = \mathcal{U}, v_h = \mathcal{V}$, so that system (2) can be rewritten in the following way:

$$\left\{\begin{array}{rcl}
\Big\langle \partial_t\mathcal{E} - b\nabla\cdot\left(h^2\nabla\partial_t\mathcal{E}\right);\Phi^\eta\Big\rangle & = & -\langle (h+\mathcal{E})\nabla\cdot(\mathcal{U};\mathcal{V}) + (h_x+\mathcal{E}_x)\mathcal{U} + (h_y+\mathcal{E}_y)\mathcal{V} + \zeta_t \\
 & & +\tilde{A}\nabla\cdot\left(h^2\nabla\zeta_t\right) + A\nabla\cdot\left\{h^2\left[\nabla\left(\nabla h\cdot(\mathcal{U};\mathcal{V})\right) + \nabla h\nabla\cdot(\mathcal{U};\mathcal{V})\right]\right\};\Phi^\eta\rangle \\
 & = & \mathcal{F}\left(\mathcal{E},\mathcal{U},\mathcal{V},\Phi^\eta\right), \\
\Big\langle (I_d - dh^2\Delta)\partial_t\mathcal{U};\Phi^u\Big\rangle & = & -\Big\langle g\mathcal{E}_x + \mathcal{U}\mathcal{U}_x + \mathcal{V}\mathcal{V}_x + Bgh\Big[\left(\nabla h\cdot\nabla\mathcal{E}\right)_x + h_x\Delta\mathcal{E}\Big] - Bh\zeta_{xtt};\Phi^u\Big\rangle \\
 & = & \mathcal{G}\left(\mathcal{E},\mathcal{U},\mathcal{V},\Phi^u\right), \\
\Big\langle (I_d - dh^2\Delta)\partial_t\mathcal{V};\Phi^v\Big\rangle & = & -\Big\langle g\mathcal{E}_y + \mathcal{U}\mathcal{U}_y + \mathcal{V}\mathcal{V}_y + Bgh\Big[\left(\nabla h\cdot\nabla\mathcal{E}\right)_y + h_y\Delta\mathcal{E}\Big] - Bh\zeta_{ytt};\Phi^v\Big\rangle \\
 & = & \mathcal{H}\left(\mathcal{E},\mathcal{U},\mathcal{V},\Phi^u\right).
\end{array}\right. \tag{3}$$

However, the model presented above contains some drawbacks. In particular, when the bathymetry function contains steep gradients, it causes instabilities in the numerical solution. We have to mention that this problem is well-known in the framework of BOUSSINESQ-type equations [18]. In order to avoid this kind of problems and to have a robust numerical model, we take two measures. First of all, we perform the smoothing of the bathymetry data which is fed into the model. In this way, we avoid noise in the bathymetry gradient. As a second and more radical step, we neglect higher order

[1] Please, notice that BBM–BBM (1) and sBBM (4) systems coincide over flat bottoms.

derivatives of the bathymetry function as it was proposed earlier in [2]. Thus, from now on we shall use the following system of equations:

$$\left\{\begin{array}{rcl} \left\langle \partial_t\mathcal{E};\Phi^\eta\right\rangle + b\left\langle h^2\nabla\partial_t\mathcal{E};\nabla(\Phi^\eta)\right\rangle - \displaystyle\int_{\Gamma_n} bh^2\Phi^\eta\frac{\partial(\partial_t\mathcal{E})}{\partial n}\partial\gamma & = & \mathcal{F}\left(\mathcal{E},\mathcal{U},\mathcal{V},\Phi^\eta\right) \\ \left\langle \partial_t\mathcal{U};\Phi^u\right\rangle + d\left\langle h^2\nabla\partial_t\mathcal{U};\nabla\Phi^u\right\rangle + d\left\langle 2h\nabla h\cdot\nabla\partial_t\mathcal{U};\Phi^u\right\rangle - \displaystyle\int_{\Gamma_n} dh^2\Phi^u\frac{\partial(\partial_t\mathcal{U})}{\partial n}\partial\gamma & = & \mathcal{G}\left(\mathcal{E},\mathcal{U},\mathcal{V},\Phi^u\right) \\ \left\langle \partial_t\mathcal{V};\Phi^v\right\rangle + d\left\langle h^2\nabla\partial_t\mathcal{V};\nabla\Phi^v\right\rangle + d\left\langle 2h\nabla h\cdot\nabla\partial_t\mathcal{V};\Phi^v\right\rangle - \displaystyle\int_{\Gamma_n} dh^2\Phi^v\frac{\partial(\partial_t\mathcal{V})}{\partial n}\partial\gamma & = & \mathcal{H}\left(\mathcal{E},\mathcal{U},\mathcal{V},\Phi^v\right) \end{array}\right. \tag{4}$$

with

$$\mathcal{F}\left(\mathcal{E},\mathcal{U},\mathcal{V},\Phi^\eta\right) = -\left\langle (h+\mathcal{E})\nabla\cdot(\mathcal{U};\mathcal{V}) + (h_x+\mathcal{E}_x)\mathcal{U} + (h_y+\mathcal{E}_y)\mathcal{V} + \zeta_t;\Phi^\eta\right\rangle$$

$$-\tilde{A}\left\langle 2hh_x\zeta_{xt} + 2hh_y\zeta_{yt};\Phi^\eta\right\rangle - A\left\langle -2hh_xh_y\mathcal{V}_x + 2hh_x^2\mathcal{V}_y;\Phi^\eta\right\rangle + A\left(\left\langle 2h^2h_x\mathcal{U}_x + h^2h_y\mathcal{V}_x;\Phi^\eta_x\right\rangle + \left\langle h^2h_y\mathcal{U}_x\right.\right.$$

$$\left.\left. +h^2h_x\mathcal{U}_y + h^2h_x\mathcal{V}_x + 2h^2h_y\mathcal{V}_y;\Phi^\eta_y\right\rangle\right) - A\int_{\Gamma_n}\left((3h^2h_x + h^2h_y)\Phi^\eta\frac{\partial\mathcal{U}}{\partial n} + (h^2h_x + 3h^2h_y)\Phi^\eta\frac{\partial\mathcal{V}}{\partial n}\right)\partial\gamma,$$

$$\mathcal{G}\left(\mathcal{E},\mathcal{U},\mathcal{V},\Phi^u\right) = -\left\langle g\left(I_d - B\left(2h_x^2 + h_y^2\right)\right)\mathcal{E}_x + \mathcal{U}\mathcal{U}_x + \mathcal{V}\mathcal{V}_x - Bgh_xh_y\mathcal{E}_y - Bh\zeta_{xtt};\Phi^u\right\rangle + Bg\left\langle 2hh_x\mathcal{E}_x;\Phi^u_x\right\rangle$$

$$+Bg\left\langle hh_y\mathcal{E}_x + hh_x\mathcal{E}_y;\Phi^u_y\right\rangle - \int_{\Gamma_n} Bg(3hh_x + hh_y)\Phi^u\frac{\partial\mathcal{E}}{\partial n}\partial\gamma,$$

and

$$\mathcal{H}\left(\mathcal{E},\mathcal{U},\mathcal{V},\Phi^v\right) = -\left\langle -2Bgh_xh_y\mathcal{E}_x + \mathcal{U}\mathcal{U}_y + \mathcal{V}\mathcal{V}_y + g\left(I_d - 2Bh_y^2\right)\mathcal{E}_y - Bh\zeta_{ytt};\Phi^v\right\rangle + Bg\left\langle hh_y\mathcal{E}_x;\Phi^v_x\right\rangle$$

$$+Bg\left\langle hh_x\mathcal{E}_x + 2hh_y\mathcal{E}_y;\Phi^v_y\right\rangle - \int_{\Gamma_n} Bg(hh_x + 3hh_y)\Phi^v\frac{\partial\mathcal{E}}{\partial n}\partial\gamma.$$

Several intermediate computations are reported in Appendix A. We would like to underline the fact that the performed simplification allows us to gain in numerical model stability and robustness at the price of some higher order bathymetry effects.

2.2. Time marching scheme

Our method is based on the explicit second order RUNGE–KUTTA scheme. For that, let us denote by $(\mathcal{E}^{n+1},\mathcal{U}^{n+1},\mathcal{V}^{n+1})$ and $(\mathcal{E}^n,\mathcal{U}^n,\mathcal{V}^n)$ the approximate values at time $t = t^{n+1}$ and $t = t^n$, respectively and by δt the time step size. Then, owing to (4), the unknown fields at time $t = t^{n+1}$ are defined as the solution of the following system:

$$\left\{\begin{array}{l} \langle\mathcal{E}^{n+1};\Phi^\eta\rangle = \langle\mathcal{E}^n + \dfrac{\mathcal{E}^{k1}+\mathcal{E}^{k2}}{2};\Phi^\eta\rangle, \\ \langle\mathcal{U}^{n+1};\Phi^u\rangle = \langle\mathcal{U}^n + \dfrac{\mathcal{U}^{k1}+\mathcal{U}^{k2}}{2};\Phi^u\rangle, \\ \langle\mathcal{V}^{n+1};\Phi^v\rangle = \langle\mathcal{V}^n + \dfrac{\mathcal{V}^{k1}+\mathcal{V}^{k2}}{2};\Phi^v\rangle, \end{array}\right. \tag{5}$$

where

$$\begin{aligned}
\left\langle \mathcal{E}^{k1};\Phi^{\eta}\right\rangle + b\left\langle h^2\nabla\mathcal{E}^{k1};\nabla(\Phi^{\eta})\right\rangle - \int_{\Gamma_n} bh^2\Phi^{\eta}\frac{\partial(\mathcal{E}^{k1})}{\partial n}\partial\gamma &= \delta t\cdot\mathcal{F}\left(\mathcal{E}^n,\mathcal{U}^n,\mathcal{V}^n,\Phi^{\eta}\right),\\
\left\langle \mathcal{U}^{k1}+2dh\nabla h\cdot\nabla\mathcal{U}^{k1};\Phi^{u}\right\rangle + d\left\langle h^2\nabla\mathcal{U}^{k1};\nabla\Phi^{u}\right\rangle - \int_{\Gamma_n} dh^2\Phi^{u}\frac{\partial(\mathcal{U}^{k1})}{\partial n}\partial\gamma &= \delta t\cdot\mathcal{G}\left(\mathcal{E}^n,\mathcal{U}^n,\mathcal{V}^n,\Phi^{u}\right),\\
\left\langle \mathcal{V}^{k1}+2dh\nabla h\cdot\nabla\mathcal{V}^{k1};\Phi^{v}\right\rangle + d\left\langle h^2\nabla\mathcal{V}^{k1};\nabla\Phi^{v}\right\rangle - \int_{\Gamma_n} dh^2\Phi^{v}\frac{\partial(\mathcal{V}^{k1})}{\partial n}\partial\gamma &= \delta t\cdot\mathcal{H}\left(\mathcal{E}^n,\mathcal{U}^n,\mathcal{V}^n,\Phi^{v}\right)
\end{aligned}\tag{6}$$

and

$$\begin{aligned}
\left\langle \mathcal{E}^{k2};\Phi^{\eta}\right\rangle + b\left\langle h^2\nabla\mathcal{E}^{k2};\nabla(\Phi^{\eta})\right\rangle - \int_{\Gamma_n} bh^2\Phi^{\eta}\frac{\partial(\mathcal{E}^{k2})}{\partial n}\partial\gamma = \delta t\cdot\mathcal{F}\left(\mathcal{E}^n+\mathcal{E}^{k1},\mathcal{U}^n+\mathcal{U}^{k1},\mathcal{V}^n+\mathcal{V}^{k1},\Phi^{\eta}\right),\\
\left\langle \mathcal{U}^{k2}+2dh\nabla h\cdot\nabla\mathcal{U}^{k2};\Phi^{u}\right\rangle + d\left\langle h^2\nabla\mathcal{U}^{k2};\nabla\Phi^{u}\right\rangle - \int_{\Gamma_n} dh^2\Phi^{u}\frac{\partial(\mathcal{U}^{k2})}{\partial n}\partial\gamma = \delta t\cdot\mathcal{G}\left(\mathcal{E}^n+\mathcal{E}^{k1},\mathcal{U}^n+\mathcal{U}^{k1},\mathcal{V}^n+\mathcal{V}^{k1},\Phi^{u}\right),\\
\left\langle \mathcal{V}^{k2}+2dh\nabla h\cdot\nabla\mathcal{V}^{k2};\Phi^{v}\right\rangle + d\left\langle h^2\nabla\mathcal{V}^{k2};\nabla\Phi^{v}\right\rangle - \int_{\Gamma_n} dD^2\Phi^{v}\frac{\partial(\mathcal{V}^{k2})}{\partial n}\partial\gamma = \delta t\cdot\mathcal{H}\left(\mathcal{E}^n+\mathcal{E}^{k1},\mathcal{U}^n+\mathcal{U}^{k1},\mathcal{V}^n+\mathcal{V}^{k1},\Phi^{v}\right).
\end{aligned}\tag{7}$$

3. New domain adaptation, domains computation and initial data

We present here the new domain adaptation technique that will be compared in the sequel with the mesh adaptation used in `FreeFem++`. In order to complete the literature review, we would like to mention that alternative approaches exist, see *e.g.* [19,20].

3.1. New domain adaptation technique

Since some computation domains for many applications (here for tsunami waves [21]) may be huge and the initial data is concentrated in a small domain, a circle $\mathcal{C}(O,R)$ or a rectangle $[a,b]\times[c,d]$, before starting to propagate in the domain, we present here an idea to build a moving computation domain around the solution only, as when we use a mesh adaptation. The difference between these two methods is that the moving domain will be a cut from the initial one; *i.e.* all initials vertices, edges and boundary labels are conserved and a new label is defined for the new boundary; while the mesh adaptation technique don't conserve the initials vertices and edges, so when we make interpolation of solution from old to new mesh we will lose some information in the mesh adaptation technique but not with the moving domain.

Firstly, we cut from the initial mesh `Thinit` a circle or a rectangle zone `Th` where our initial solution lives (using `trunc` in `FreeFem++`, see (a) in Figure 2), we let `uadapt` be the initial solution used for the domain adaptation, and we follow this algorithm:

- We deduce the limit min max of `Th` on x and y direction (using `boundingbox` in `FreeFem++`).
- We increase the mesh from `Th` to `Th1` by adding layers from the original mesh (using `trunc` in `FreeFem++`), the added zone is a size of `epsadapt` from each side and we mapped `uadapt` to `unew` (using `interpolate` in `FreeFem++`, see (b) in Figure 2).
- We define a HEAVISIDE function `unewadapt` which have a 1 value if the absolute value of `unew` is grater then or equal to a defined error (`erradapt`) by the user and 0 otherwise (see (c) in Figure 2).
- We smooth the function `unewadapt` (see (d) in Figure 2)) solving the following problem:

$$\beta\,\texttt{usmadapt} - \Delta\,\texttt{usmadapt} = \beta\,\texttt{f},\tag{8}$$

  with zero DIRICHLET BC only on the new boundary label of `Th1` and a NEUMANN BC in the other boundary label. Here β is the real coefficient that control the smoothness of the solution and `f=unewadapt`.
- We define a HEAVISIDE function `ufinal` which have a 1 value if the absolute value of `usmadapt` is grater then or equal to a defined error (`erradapt`) by the user and 0 otherwise (see (c) in Figure 2).

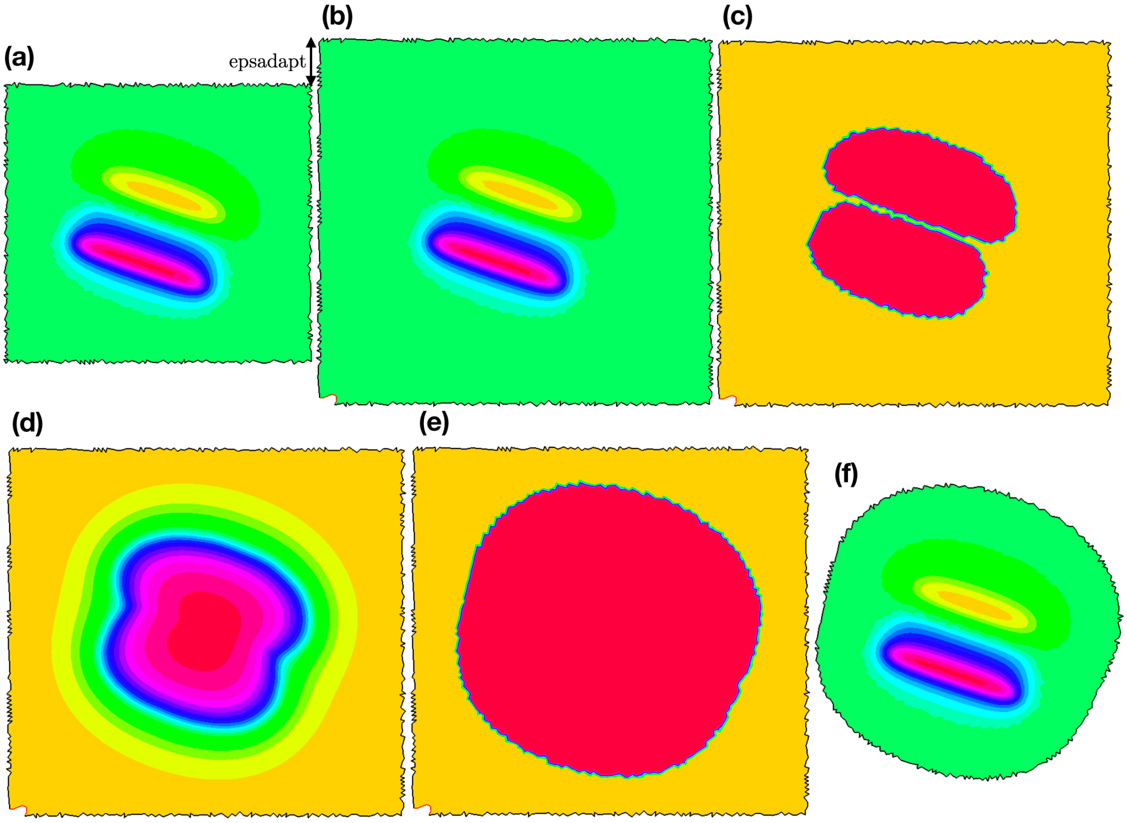


Figure 2. *Domain adaptation technique step : (a) : initial solution, (b) : the solution* `unew` *mapped from the initial solution over a mesh augmented by* `epsadapt`*, (c) : the new* HEAVISIDE *function* `unewadapt` *which have a value of 1 (the red part in the figure) if* $|\texttt{unew}| \geq \texttt{erradapt}$ *and 0 otherwise, (d) : the function* `usmadapt` *which is the smoothness of the* `unewadapt`*, (e) : the new* HEAVISIDE *function* `ufinal` *which have a value of 1 (the red part in the figure) if* $|\texttt{usmadapt}| \geq \texttt{erradapt}$ *and 0 otherwise, (f) : mapped of the initial solution to the final domain adapted.*

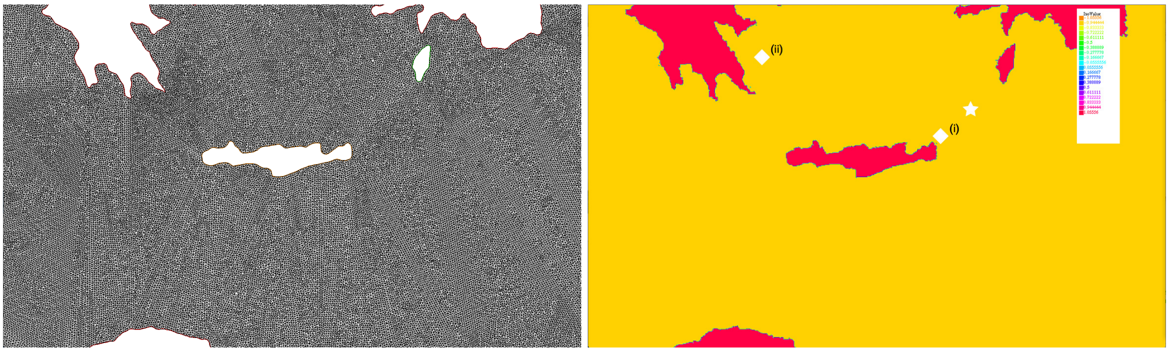
- We cut from `Th1`, the region where `ufinal` is grater then a defined isoline `isoadapt` by the user in order to obtain the final mesh `Thnew` (using `trunc` in `FreeFem++`), then we obtain the initial solution mapped over the final mesh using `interpolate` in `FreeFem++`, see (f) in Figure 2).

We use a reflective Boundary Condition (BC) on the new boundary, *i.e.* homogeneous NEUMANN BC for η and homogeneous DIRICHLET BC for the velocity V. This choice is justified theoretically over flat bottom case in [22]. Moreover, the homogeneous NEUMANN BC for η can be shown to hold *exactly* in the full EULER equations on solid vertical walls, see [23, Section §2.1.4] for the proof.

3.2. Domains computation

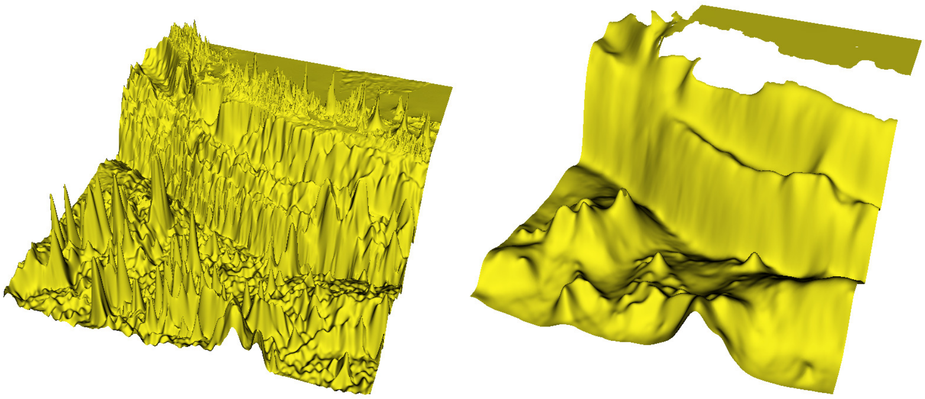
For the BBM–BBM system over a flat bottom, we use a mesh generated through a photo of the MEDITERRANEAN sea (a cut of the mesh around the CRETE island is shown in Figure 3 at left panel) and for the sBBM system over a variable bottom in space and in time, we use a mesh generated using an imported bathymetry f_{xy} for the sea near the JAVA island which can be downloaded from the NOAA[2] website where in this case we remove the dry zone from our mesh and we keep only the wet zone. We can smooth the bathymetric data obtained from NOAA (*cf.* Figure 4, left panel) by solving (8) with $\texttt{f} = f_{xy}$. For all simulations with realistic bathymetry, we use $\beta = 20$ in (8) to smooth the initial

[2] https://maps.ngdc.noaa.gov/viewers/wcs-client/



(a) (b)

Figure 3. *Left (a): the mesh around the* CRETE *island. Right (b): the place of ◇: wave gauge and ⋆: epicenter.*



(a) (b)

Figure 4. *Left (a): Bathymetry downloaded from the NOAA website, (min* $= -7239$ m *and max* $= 3002$ m*). Right (b): smoothed bathymetry with* $\beta = 20$ *in* (8)*, (min* $= -6207$ m *and max* $= -100$ m*).*

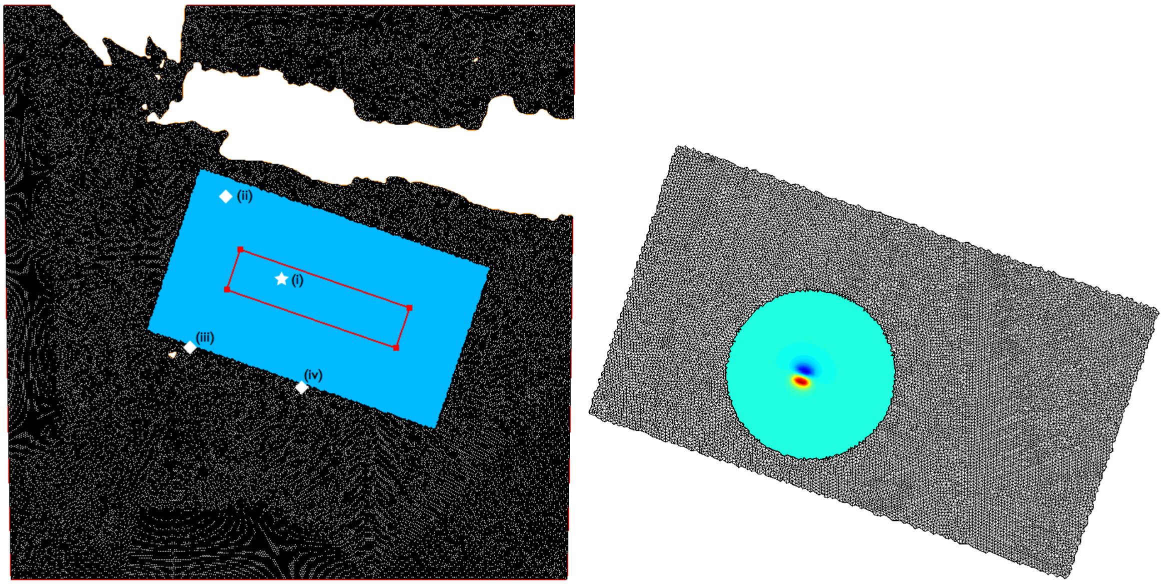
bathymetry after the generation of the mesh (*cf.* Figure 4, right panel) in order to ensure the stability of the numerical method. We also mention that we change the depth close to the shoreline to 100 m in order to avoid the run-up problem in this study. Finally, all types of meshes used in our computations are depicted in Figure 9.

The bathymetry data downloaded from the NOAA website are in geographical degrees coordinates and we need to convert them back to meters and a CARTESIAN system. So, on the first hand, we must know the degree of Latitude (South and North) and of Longitude (West and East) of our domain where we can deduce the Latitude $lat0 = .5(lat_{South} + lat_{North})$ and the Longitude $long0 = .5(long_{West} + long_{East})$. On the other hand, we must take into account the spherical shape of the EARTH, even if it does not play significant role because of the small spatial scale of the experiments. So, we know that the radius of the EARTH near the equator is $R_{equator} = 6378,137$ km, and near to the pole $R_{pole} = 6356,752$ km, thus the radius of our domain equals to:

$$R = \sqrt{\frac{\left(R^2_{equator}\cos(lat0\cdot\pi/180)\right)^2 + \left(R^2_{pole}\sin(lat0\cdot\pi/180)\right)^2}{\left(R_{equator}\cos(lat0\cdot\pi/180)\right)^2 + \left(R_{pole}\sin(lat0\cdot\pi/180)\right)^2}}.$$

So, we move the mesh of our domain using the following translation (coefl0 $= \pi R/180$):

$$[x;y] \longrightarrow [(x - lon0)\cos(\pi y/180)\text{coefl0};(y - lat0)\text{coefl0}].$$



(a) (b)

Figure 5. *Left panel (a): Surface projection of the fault's plane and the mesh around,* $\diamond$*: wave gauge,* $\star$*: epicenter. Right panel (b): the* 14*-th Okada solution (min=* -0.09 *m, max=* 0.17 *m).*

Finally, for the active generation case, since the fault plane is considered to be the rectangle with vertices located at (109.20508° (Lon),−10.37387° (Lat)), (106.50434° (Lon),−9.45925° (Lat)), (106.72382° (Lon),−8.82807° (Lat)) and (109.42455° (Lon),−9.74269° (Lat)), we will consider that our bottom displacement is concentrated on the big rectangle which is equidistant of 1° from each side of the initial fault plane as in Figure 5 (left panel).

3.3. Initial data

Tsunami waves considered in this study are generated by the co-seismic deformation of the Ocean's or sea's bottom due to an earthquake. The adopted modelling of the tsunami wave generation process is inspired by [2,11,24,25]. The co-seismic displacement is computed according to the celebrated OKADA's solution [26,27]. We assume the dip-slip dislocation process underlying the earthquake. The vertical component of displacement vector $\mathcal{O}(x,y)$ is given by the following formulas employing CHINNERY's notation, *cf.* [24,25]:

$$f(\xi,\eta)\,||= f\,(\xi,p) - f\,(\xi,p-W) - f\,(\xi-L,p) + f\,(\xi-L,p-W)\ ,$$

$$\mathcal{O}(x,y) = -\frac{U}{2\pi}\left(\frac{\tilde{d}q}{R(R+\xi)} + \sin\delta\arctan\frac{\xi\eta}{qR} - I\sin\delta\cos\delta\right)\Bigg|\Bigg|,$$

where

$$\xi = (x-x_0)\cos\phi + (y-y_0)\sin\phi,\ Y = -(x-x_0)\sin\phi + (y-y_0)\cos\phi,$$

$$p = Y\cos\delta + d\sin\delta,\ q = Y\sin\delta - d\cos\delta,$$

$$\tilde{y} = \eta\cos\delta + q\sin\delta,\ \tilde{d} = \eta\sin\delta - q\cos\delta,$$

$$R^2 = \xi^2 + \eta^2 + q^2 = \xi^2 + \tilde{y}^2 + \tilde{d}^2,\ X^2 = \xi^2 + q^2$$

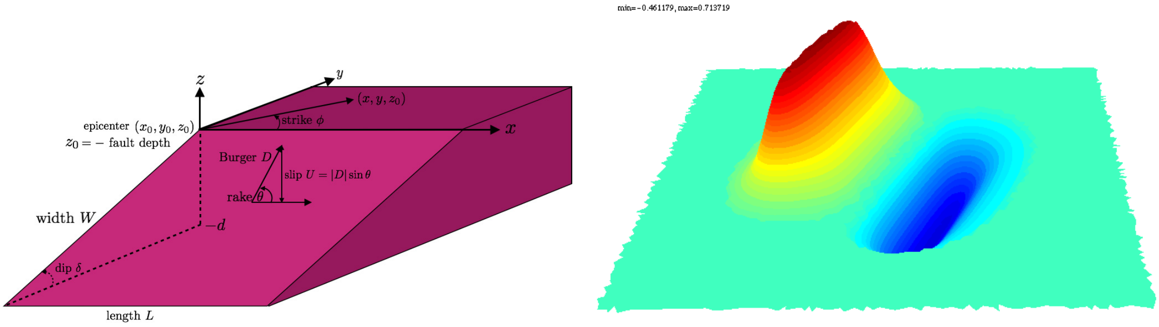
and

$$I = \begin{cases} \dfrac{\mu}{\lambda+\mu}\dfrac{2}{\cos\delta}\arctan\dfrac{\eta(X+q\cos\delta)+X(R+X)\sin\delta}{\xi(R+X)\cos\delta} & \text{if } \cos\delta\neq 0, \\ \dfrac{\mu}{\lambda+\mu}\dfrac{\xi\sin\delta}{R+\tilde{d}} & \text{if } \cos\delta = 0. \end{cases}$$

Here, W and L are the width and the length of the rectangular fault, (x, y) are the points where we computes displacements, (x_0, y_0) is the epicenter, $d =$ fault depth$(x_0, y_0) + W \sin\delta$, δ is the dip angle, θ is the rake angle, D is the BURGERS's vector, $U = |D| \sin\theta$ is the slip on the fault, ϕ is the strike angle which is measured conventionally in the counter-clockwise direction from the North (*cf.* Figure 6 (left)), μ, λ are the LAMÉ constants derived from elastic-wave velocities: $\lambda = \rho_c \left(V_P^2 - V_S^2\right)$ and $\mu = \rho_c V_S^2$, where ρ_c is the crust density, V_P is the compressional-wave ($P-$wave) velocity, V_S is the shear-wave ($S-$wave) velocity. The `Matlab` script to compute the OKADA solution can be downloaded at the following URL:

https://mathworks.com/matlabcentral/fileexchange/39819-okada-solution/

We shall distinguish here the two types of tsunami wave generation mechanisms [28,29]: *active* and *passive* generation mechanisms.



(a) (b)

Figure 6. *Geometry of the source model (left) and the initial solution for η (right, min= -0.46 m, max= 0.71 m).*

3.3.1. Passive generation

We remind that the passive generation approach consists in transposing the bottom deformation on the free surface as an initial condition for tsunami propagation codes. In order to compute the initial data for $\eta(x, y, 0) = \mathcal{O}(x, y)$ in meters (*cf.* Figure 6 (right)), $V(x, y, 0) = 0$ which is referred to as a *passive* generation of a tsunami wave near the JAVA island, using our domain adaptive technique, we will use the fact that the solution is concentrated in the small rectangle $[x_0 - 3.2W; x_0 + 1.2W] \times [y_0 - L; y_0 + L]$ where $L = 100$ km, $W = 50$ km, $\delta = 10.35°$, $\phi = 288.94°$, $\theta = 95°$, $U = 2$ m, $\rho_c = 2700\text{kg}/\text{m}^3$, $V_P = 6000$ m/s, $V_S = 3400$ m/s, $(x_0; y_0) = (107.345°, -9.295°)$ and the fault depth 10 km. All these geophysical parameters can be downloaded from this file hosted by USGS:

https://Earthquake.usgs.gov/archive/product/finite-fault/usp000ensm/us/1486510367579/web/p000ensm.param

3.3.2. Active generation

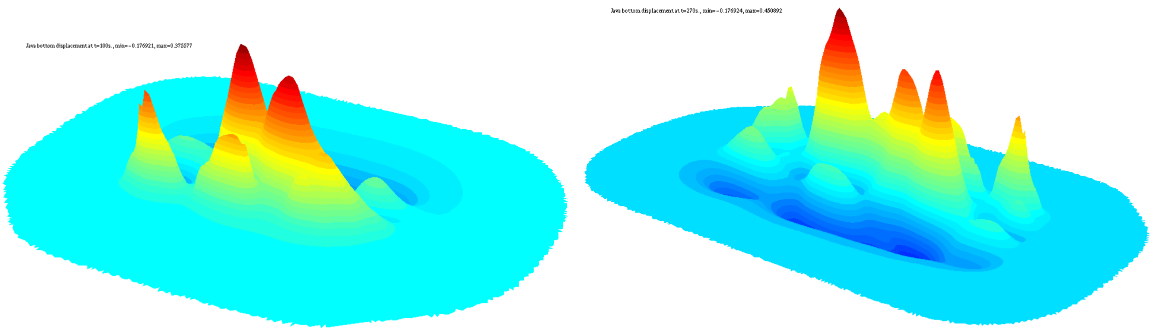
In contrast to passive generation, the active generation approach consists in generating a tsunami waves by computing fluid layer interaction with moving bottom. For a more realistic case of the JAVA 2006 event, we use precisely this so-called *active* generation approach by following [11,30]. In this case we consider zero initial conditions for both the free surface elevation and the velocity field, and assume that the bottom is moving in time. This case may be described by considering the bottom motion formula: $-z_b(x, y, t) = -h(x, y) - \zeta(x, y, t)$ with

$$\zeta(x, y, t) = \sum_{i=1}^{N_x \cdot N_y} \mathcal{H}(t - t_i) \cdot \left(1 - e^{-\alpha(t - t_i)}\right) \cdot \mathcal{O}_i(x, y),$$

where N_x sub-faults along strike and N_y sub-faults down the dip angle, $\mathcal{H}(t)$ is the HEAVISIDE step function and $\alpha = \log(3)/t_r$, where $t_r = 8$ s is the rise time. We choose here an exponential scenario, but in practice, various scenarios could be used (instantaneous, linear, trigonometric, *etc.*) and could be found in [11,24,25,30,31]. Parameters such as sub-fault location (x_i, y_i), depth d_i, slip U and rake angle θ for each segment are given in [11, Table 3]. In this table, we notice that the fault's surface is conventionally divided into $N_x = 21$ sub-faults along strike and $N_y = 7$ sub-faults down the dip angle, leading to a total number of $N_x \times N_y = 147$ equal segments.

We compute each OKADA solution $\mathcal{O}_i$ on a circle of center $(x_i - 10m, y_i - 10m)$ and of radius $6\max(L, W)$ and at the end all the OKADA solution will be interpolated on the big rectangle before starting to compute the vertical displacement of the bottom $\zeta(x, y, t)$, in Figure 5 (right panel) we plot $\mathcal{O}_{14}$. For the computation of $\zeta(x, y, t)$, we start the mesh by a circle of center $(x_c - 5m, y_c - 5m)$ and of radius $4\max(L, W)$ and we adapt the mesh each 3 iterations *i.e.* each 6 s by using the following value for the domain adaptation $\texttt{uadapt} = \zeta$, $\texttt{isoadapt}= 5 \times 10^{-2}$, $\texttt{erradapt}= \times 10^{-4}$, $\beta = 5 \times 10^{-9}$, $\texttt{epsadapt}= 5 \times 10^{4}$.

We show in Figure 7, the bottom displacement $\zeta(x, y, t)$ at time $t = 100$ s and $t = 270$ s using our domain adaptation technique. We note that after building the OKADA solution $\mathcal{O}(x, y)$ in the *passive* generation or $\mathcal{O}_i(x, y)$ in the active generation, we can remark that this solution is non-local and decays slowly to zero, that is why in our domain adaptation technique we put 0 where the absolute value of the solution is less then $\min(|\min(\mathcal{O}_i(x, y))|, |\max(\mathcal{O}_i(x, y))|) < 9.2$ m. We make the same thing without adaptive mesh in order to compare the solution using the same initial data.



(a)

(b)

Figure 7. *Bottom displacement at $t = 100$ s (left, min $= -0.18$ m, max $= 0.38$ m) and at $t = 270$ s (right, min $= -0.18$ m, max $= 0.45$ m).*

4. Numerical simulations

In this section, we study first the rate of convergence of our schemes for the sBBM System (4) with non-dimensional and unscaled variables *i.e.*, with $g = 1$ over a variable bottom in space, which establishes the adequacy of the chosen finite element discretization and the used time marching scheme, for the flat bottom case, we refer to [32], where we use the same technique as in this paper. Then, we simulate the propagation of a wave, that is similar to a real-world tsunami wave generated by an earthquake, in the MEDITERRANEAN sea with the BBM–BBM model over a flat bottom. Then, we switch to the JAVA island region with real variable bottom in space. Finally, we study the active tsunami generation scenario which took place in 2006 near the JAVA island. In all numerical simulations we used $\mathbb{P}_1$ continuous piecewise linear functions for η, u, v, h and ζ.

4.1. Rate of convergence

We present the evidence here, following the work done for the 1D case of the BBM–BBM system in [33], that the second order RUNGE–KUTTA time scheme considered for the sBBM variable bottom in space is of order 2. We note that the function $\zeta(x, y, t)$ is only used for the generation of tsunami

Table 1. *L^2 norm of the error for η and V.*

N	δt	$N_{L^2}(\eta)$	rate	$N_{L^2}(V)$	rate	$N_{H^1}(\eta)$	rate	$N_{H^1}(V)$	rate
2^5	$.01/2^0$	0.24145	-	1.10773	-	0.60317	-	1.62575	-
2^6	$.01/2^1$	0.06078	1.990	0.28016	1.983	0.30196	0.998	0.81276	1.000
2^7	$.01/2^2$	0.01524	1.996	0.07038	1.993	0.15119	0.998	0.40696	0.999
2^8	$.01/2^3$	0.00381	1.999	0.01760	1.999	0.07578	0.998	0.20355	0.999

wave and, thus, will not be taken into account in the convergence rate test. In this example, we take bi-periodic Boundary Conditions (BC) for η_h, u_h and v_h on the whole boundary of the square $[0, 2L] \times [0, 2L]$, where $L = 50$ and we consider the following exact solutions:

$$\eta_{ex} = .2\cos(2\pi x/L - t)\cos(2\pi y/L - t), u_{ex} = .5\sin(2\pi x/L - t)\cos(2\pi y/L - t),$$

$$v_{ex} = .5\cos(2\pi x/L - t)\sin(2\pi y/L - t), \qquad h(x, y) = 1 - .5\cos(2\pi x/L)\cos(2\pi y/L),$$

adding an appropriate function to the right-hand side to make these solutions exact. We measure at time $T = 1$ and for $\theta^2 = \dfrac{2}{3}$, $\delta t = \dfrac{0.01}{2^n}$ and $\delta x = \dfrac{2L}{N} = \dfrac{2L}{2^{n+5}}$ $\forall n \in \{0, 1, 2, 3, 4\}$, the following errors

$$N_{L^2}(\eta) = \|\eta_h - \eta_{ex}\|_{L^2},$$

$$N_{H^1}(\eta) = \|\eta_h - \eta_{ex}\|_{H^1},$$

$$N_{L^2}(V) = \|u_h - u_{ex}\|_{L^2} + \|v_h - v_{ex}\|_{L^2},$$

$$N_{H^1}(V) = \|u_h - u_{ex}\|_{H^1} + \|v_h - v_{ex}\|_{H^1}$$

and we end up with the results reported in Table 1. So, the L^2 rates for η and V is of order ~ 2 and the H^1 rates for η and V is of order ~ 1 as shown in the Figure 8 and which confirms the convergence of the second-order RUNGE–KUTTA scheme in time for the sBBM system with variable bottom in space.

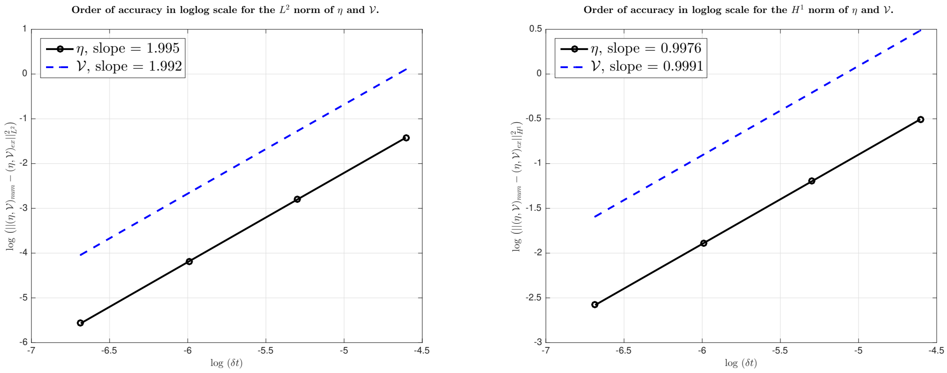


Figure 8. *Rate of convergence for sBBM system (4) with variable bottom in space.*

4.2. Propagation of a tsunami wave in the Mediterranean sea with a flat bottom

We simulate here, the propagation of a wave that looks like a tsunami wave generated by an earthquake in the MEDITERRANEAN sea with the sBBM System (4) with a flat bottom $-h(x, y) = -1,5$ km which is the average depth of the MEDITERRANEAN sea. This wave was defined above in the *passive* generation part of the Section 3 where, in this case, the initial solution is concentrated in the small rectangle $[x_0 - 5W; x_0 + 4W] \times [y_0 - 1.5L; y_0 + 2.5L]$ and we take these following values: $L = 20$ Km, $W = 10$ km, $\delta = 7°$, $\phi = 0°$, $\theta = 90°$, $E = 9.5$ GPA is the YOUNG's modulus, $\nu = 0,27$ is the POISSON's ratio, $U = 2.5$ m, $(x_0; y_0) = (2390. * scale, 590. * scale)$ and the fault depth 10 km. In this example, we

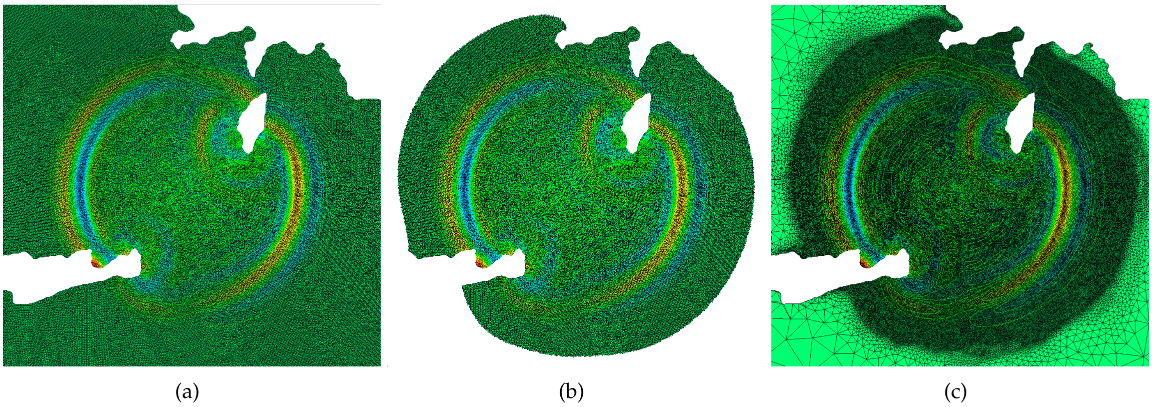


Figure 9. *The mesh and the numerical isoline of the solution at $t = 1000$ s, with the full method at the left panel (a), the domain adaptive method at the center panel (b) and the* `FreeFem++` *adaptation with* `err=1.e-7` *at the right panel (c).*

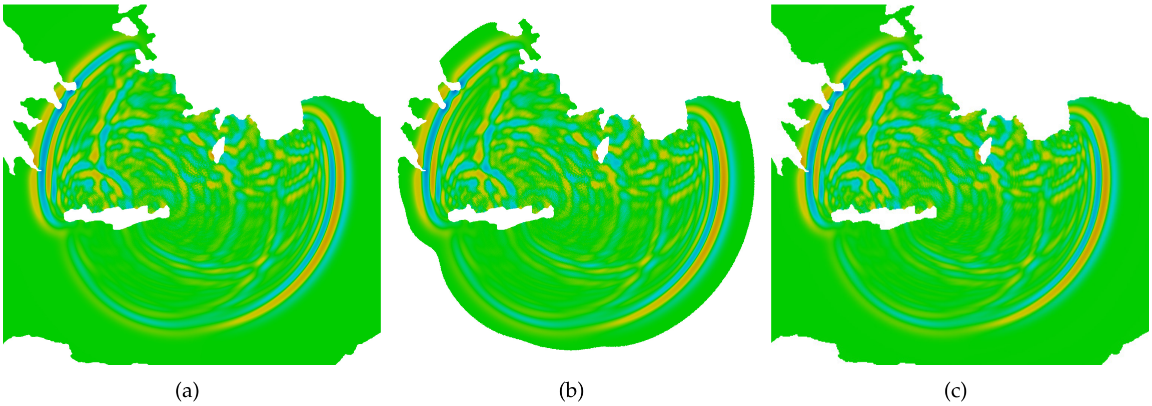


Figure 10. *The solution at $t = 3000$ s, with the full method at the left panel (a), the domain adaptive method at the center panel (b) and the* `FreeFem++` *adaptation with* `err`$= 10^{-7}$ *at the right panel (c) (min$= -5.5 \times 10^{-2}$ m, max$= 4.7 \times 10^{-2}$ m, for the three case).*

will take the fact that the LAMÉ constants μ and λ are given by the formulas $\mu = E/2(1+\nu)$ and $\lambda = E\nu/(1+\nu)(1-2\nu)$.

An efficient mesh adaptivity algorithm using metrics control is used to adapt the mesh every 50 time steps, we use the standard function (`adaptmesh`) which is an efficient tool offered by `FreeFem++` to efficiently adapt 2D meshes by metrics control [34], see [35, Section §3.2] for more details. We also use the following settings: for the step time $\delta t = 0.1$ s, a reflective BC for all the boundary, for the `adaptmesh` of `FreeFem++`:

```
Th=adaptmesh(Th,uadapt,err=1.e-7,errg=1.e-2,hmin=Dx,iso=true,nbvx=1e8);
```

where `err` : is the interpolation error level inside the geometry, `errg` : is the interpolation error level on the boundary, `hmin` : the minimum edge size, `iso` : forces the metric to be isotropic or not and `nbvx` : is the maximum number of vertices allowed in the mesh generator. Finally, for our domain adapt technique: `isoadapt`$= 5 \times 10^{-2}$, `erradapt`$= 1 \times 10^{-7}$, $\beta = 5 \times 10^{-3}$, `epsadapt`$= 2 \times 10^{-2}$. We note that, we adapt the mesh around the solution each 100 iterations *i.e.* each 10 s by using `uadapt`$= \eta + u + v$.

In order to compare the results between adaptive mesh generated by `FreeFem++`, our new domain adaptation technique and without using mesh adaptation, we plot in addition to the free surface elevation η in the Figures 9 → 10, the variation of η vs time in Figure 12 at two wave 'gauges' placed at the positions represented by $\diamond$ in Figure 3 at right and the mass of the water $\int \eta$. Specifically, gauges

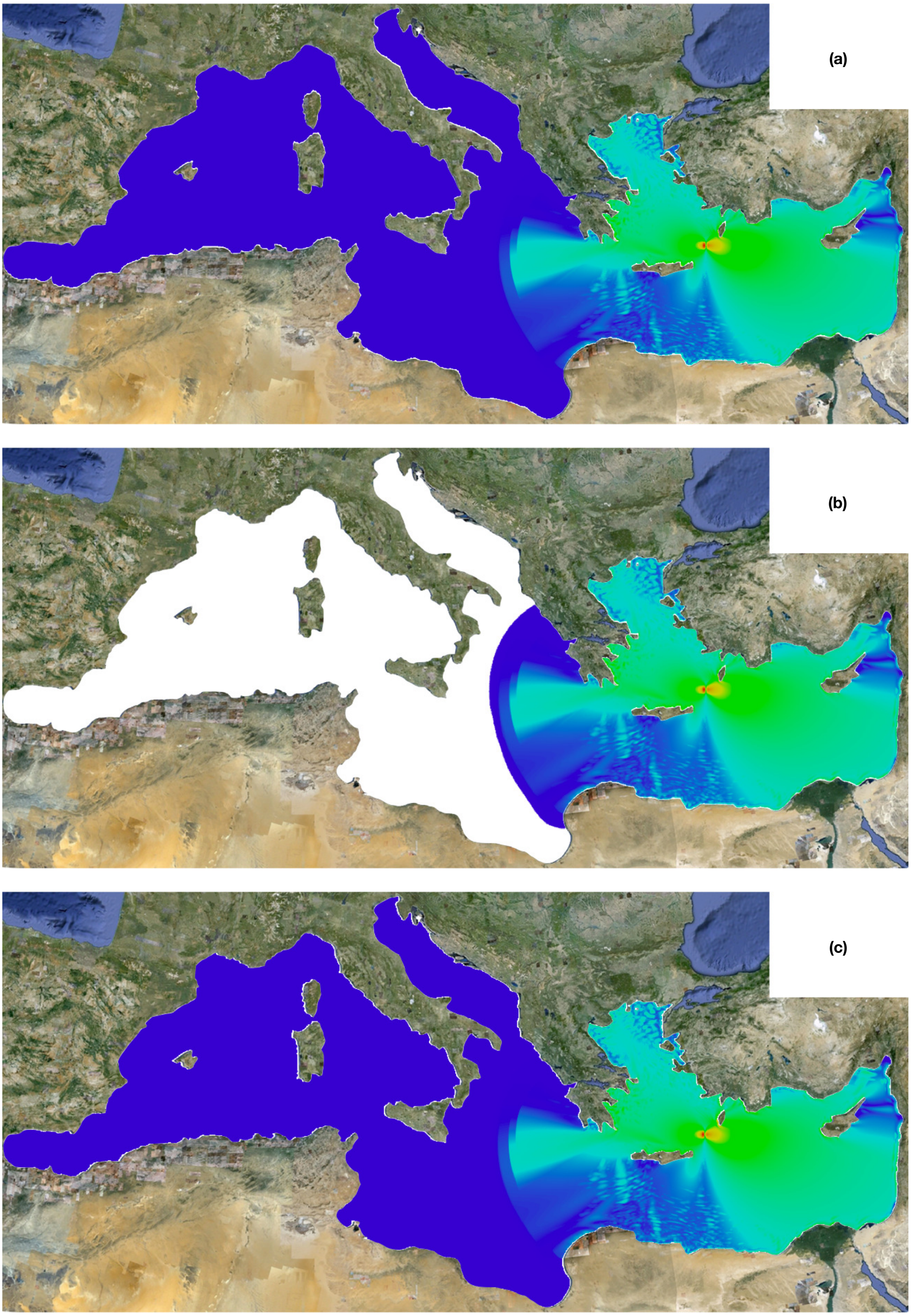


Figure 11. *Comparison between the three method full (a), domain adaptation (b) and adaptive mesh generated by* `FreeFem++` *with* `err`$= 10^{-7}$ *(c) of the maximum of the propagation of the solution of a tsunami wave in the* MEDITERRANEAN *sea for* $t = 6800$ s *(min*$= 0$ *m, max*$= .4$ *m, for three cases).*

were placed at the points $(i) : (2350. * scale, 550. * scale)$, $(ii) : (2104. * scale, 665. * scale)$. In Figure 11, we represent the comparison between the three methods: full mesh, domain adaptation and FreeFem++ internal mesh adaptivity of the maximum of the propagation of the solution at time $t = 6800$ s. We also plot the computation time for each adapt mesh, the computation time of the simulation, the number of degree of freedom in Figure 13. We can see in Figures 12 and 13 that the adaptive mesh generated by FreeFem++ with err$= 10^{-2}$ is the fastest method but unfortunately it does not preserve the mass invariant $\int \eta$. On the other hand, our new domain adaptation technique preserves the mass invariant throughout the simulation with an error of order 2.1×10^{-3} and an important time computation difference with the one without mesh adaptation which is very promising method for the tsunami wave propagation. For the adaptive mesh generated by FreeFem++ with err=1.e-7 and errg$= 10^{-2}$, we also almost get the mass conservation with an error of order 9.5×10^{-4}, but we obtain some difference in wave gauge with the full method which is due to the refinement mesh adaptation and the interpolation of the solution, although the computation time is almost the double of the new domain adaptation technique. Thus, we can go faster with our new domain adaptation technique if we can also deduce the mass matrix after cutting the mesh, of course, if the mass matrix does not change along the simulation of the full mesh. This is an outgoing project.

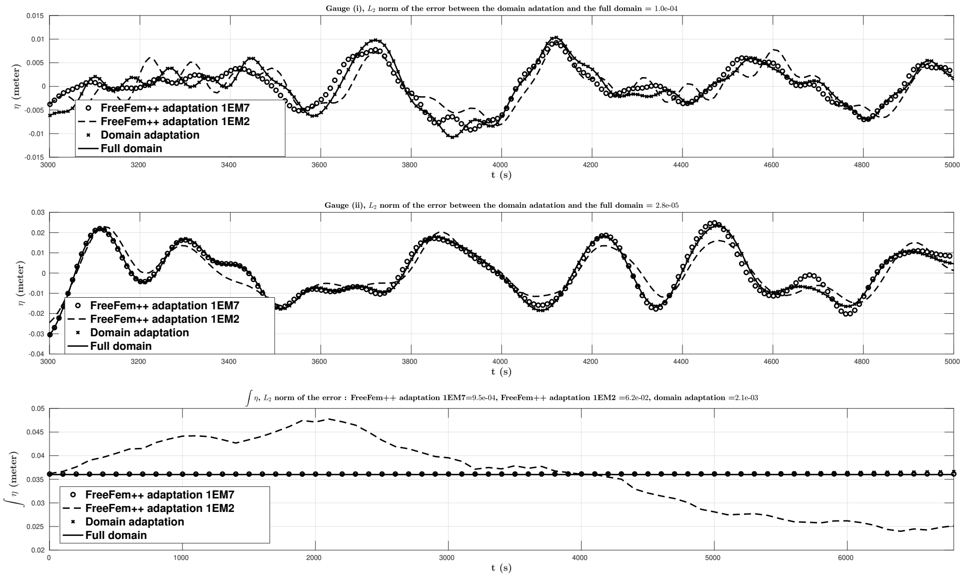


Figure 12. *Comparison between the three methods: full, FreeFem++ adaption with err$= 10^{-2}$ and err$= 10^{-7}$ and the domain adaptation of the free surface elevations (in meters) vs time (in seconds), computed numerically at two wave gauges (up and middle) and of the mass conservation (down).*

4.3. *Propagation of a tsunami wave near the Java island: passive generation*

We will take here the same initial data as defined above in the passive generation part of Section 3, we take $\delta t = 1$ s as the time step size and we note that, we adapt the mesh after computing the initial data for η and then every 50 s by using the following value for the domain adaptation uadapt$= \eta + u + v$, isoadapt$= 3 \times 10^{-2}$, erradapt$= 10^{-4}$, $\beta = 5 \times 10^{-9}$, epsadapt$= 3 \times 10^{4}$. We compare here the results between our new domain adaptation technique and without using mesh adaptation. To this end, we plot the free surface elevation η in the Figures 14 and 15, the variation of η vs time (in Figure 16) at four numerical wave gauges placed at the following locations: (i) $(107.345°, -9.295°)$, (ii) $(106.5°, -8°)$, (iii) $(105.9°, -10.35°)$ and (iv) $(107.7°, -11°)$ (see Figure 5 (left)) where (i) is the position of the epicenter. However, because of the large variations of the bottom, shorter waves were generated,

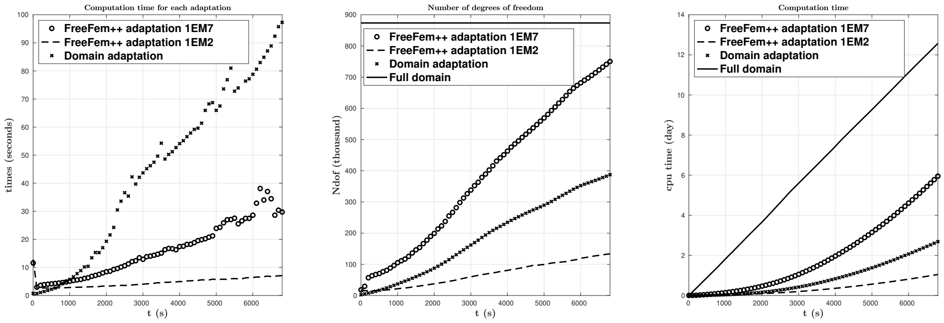
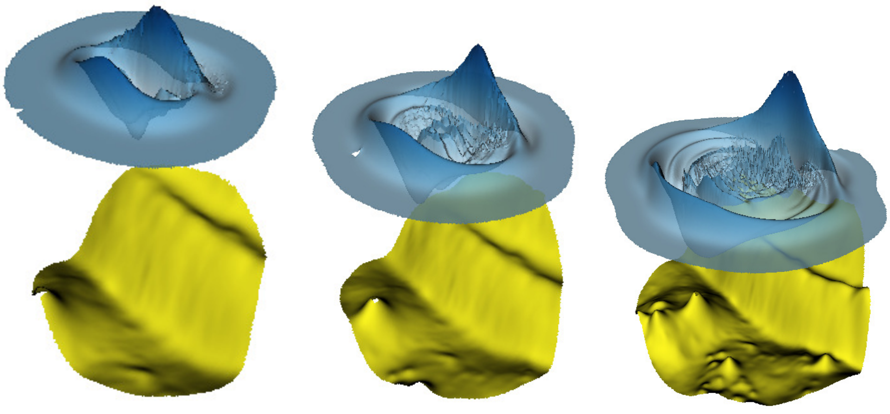


Figure 13. *Comparison between the three methods: full,* `FreeFem++` *adaption with* `err`$= 10^{-2}$ *and* `err`$= 10^{-7}$ *and domain adaptation of the computation time of each mesh/domain adaptation (left), the number of degrees of freedom (middle) and the computation time of the simulation (right).*



(a) $t = 250$ s (b) $t = 500$ s (c) $t = 1000$ s

Figure 14. *Passive generation: the bottom together with the free surface elevation at different instances of time obtained with the proposed domain adaptivity method.*

especially around CHRISTMAS Island (southwest of JAVA) and around the undersea canyon near the earthquake epicenter.

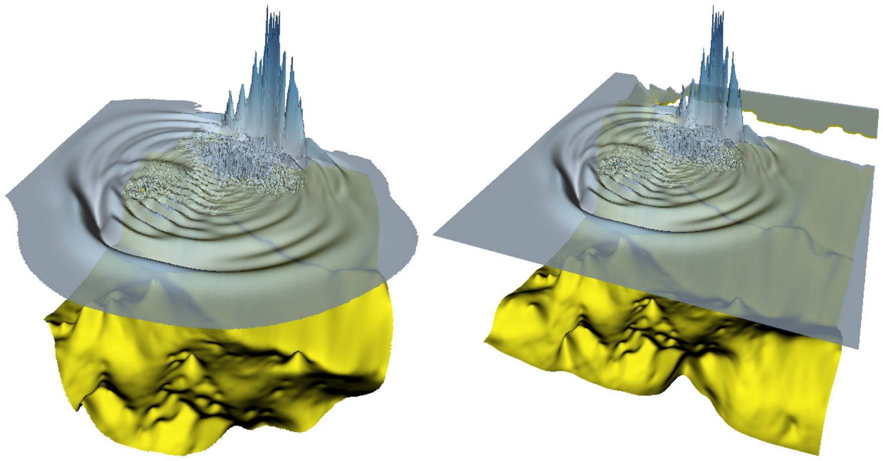
Finally, we present a comparison of the kinetic, potential and total energies with the full mesh (in Figure 17, top left panel) and with the domain adaptivity method (in Figure 17, top right panel) defined in [36] as follows:

$$E_c = \frac{1}{2}\rho_w \int_\Omega \left(\int_{-h(x,y)}^{\eta} |V|^2 \mathrm{d}z \right) \mathrm{d}x\mathrm{d}y, \qquad E_p = \frac{1}{2}\rho_w \cdot g \int_\Omega \eta^2 \mathrm{d}x\mathrm{d}y, \tag{9}$$

where $\rho_w = 1027\ \mathrm{kg/m^3}$ is the ocean water density, the number of degrees of freedom (in Figure 17, down left panel) and the computation time of the simulation (in Figure 17, down right panel). We obtain here an error of order 2.6×10^{-4} between the total energy with domain adaptivity and without any adaptation. We present in Figure 18 the comparison of the maximum of the propagation of the solution between the full and domain adaptivity methods at $t = 1750$ s.

4.4. Propagation of a tsunami wave near the Java island: active generation

For a more realistic case as in the JAVA 2006 event, we use the active generation in order to model the generation of a tsunami wave as in [11,30]. In this case we consider zero initial conditions for both the surface elevation and the velocity field, we take $\delta t = 2$ s as the time step size, we assume that the bottom described in the Section 3 is moving in time and we note that we adapt the mesh, before the end of the generation time $t = 270$ s, every three iterations *i.e.* every 6 s by using the following value



(a) Domain adaptivity (b) Full mesh

Figure 15. *Passive generation: comparison between the bottom and the free surface elevation at $t = 1500$ s between the domain adaptation method and the full mesh.*

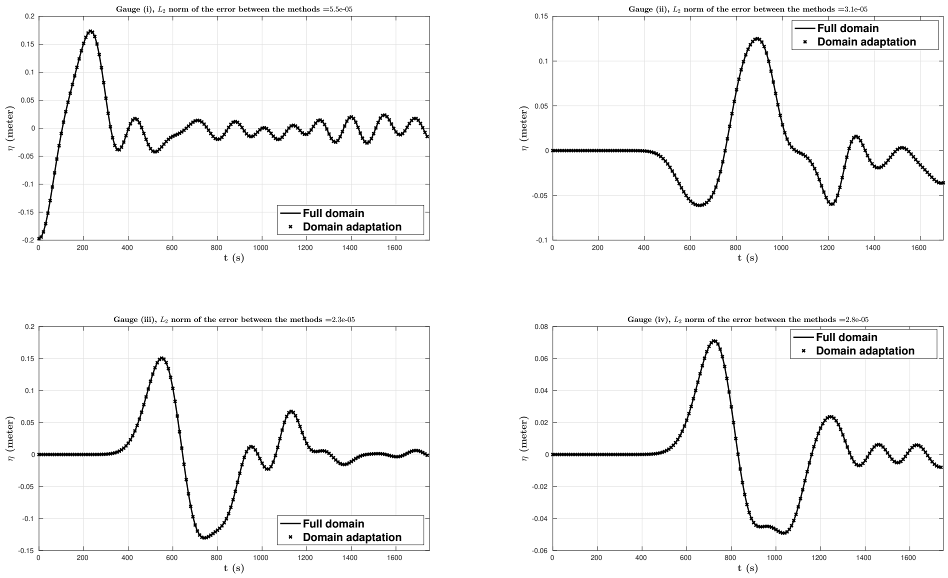


Figure 16. *Passive generation: comparison between the two methods the full one and domain adaptivity of the free surface elevations (in meters) vs time (in seconds), computed numerically at four wave gauges where the gauge (i) corresponds to the epicenter.*

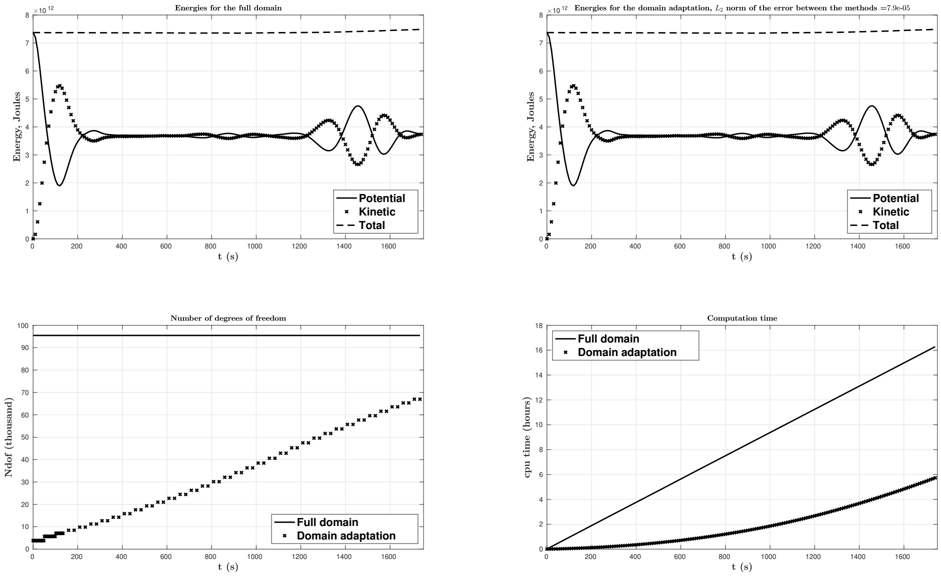
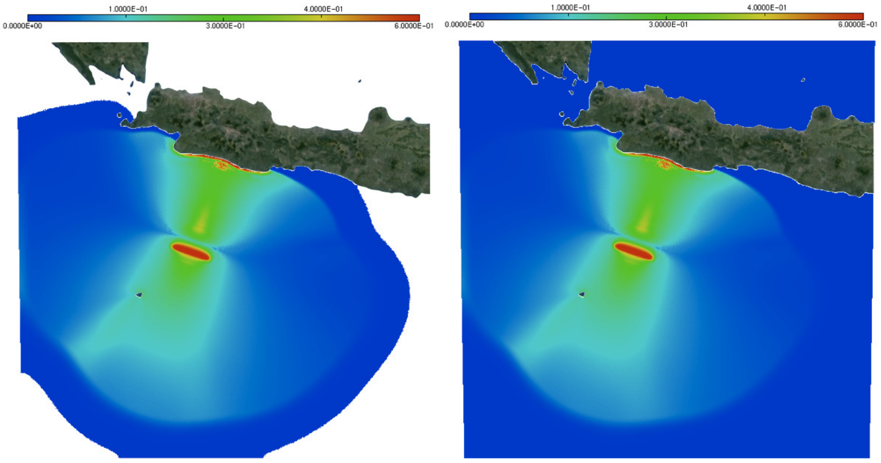


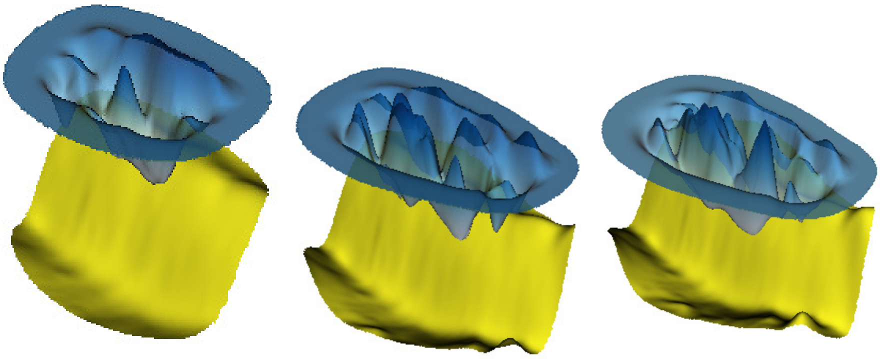
Figure 17. *Passive generation: comparison between the two methods the full one and domain adaptivity of the kinetic, potential and total energies, the number of degrees of freedom and the computation time of the simulation.*



(a)

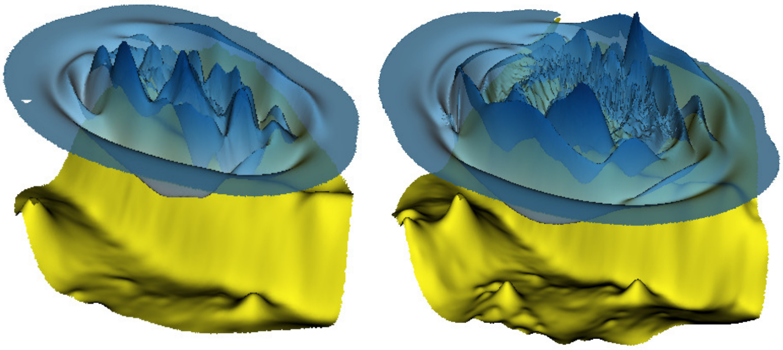
(b)

Figure 18. *Passive generation: comparison between the maximum of the solution at $t = 1\,750$ s, with the domain adaptivity method (left panel) and with the full one (right panel).*



(a) $t = 100$ s (b) $t = 200$ s (c) $t = 270$ s

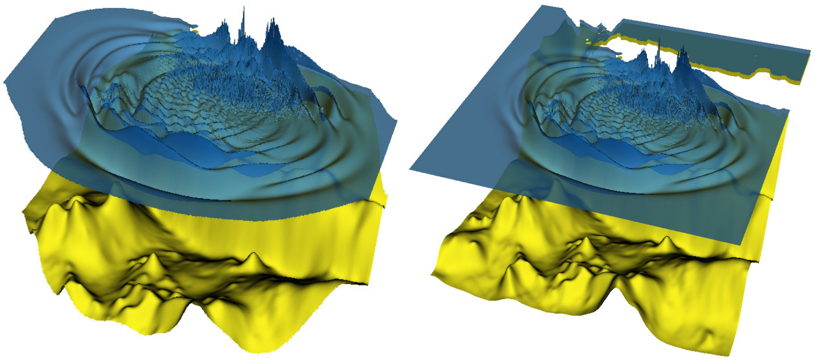
Figure 19. *Active generation: the bottom and the free surface elevation computed with the domain adaptivity method.*



(a) $t = 500$ s (b) $t = 1\,000$ s

Figure 20. *Active generation: the bottom and the free surface elevation computed with the domain adaptivity method.*

for the domain adaptation `uadapt` $= \eta + u + v$, `isoadapt`$= 5 \times 10^{-2}$, `erradapt`$= 10^{-4}$, $\beta = 5 \times 10^{-9}$, `epsadapt`$= 5 \times 10^{4}$ and then for $t > 270$ s every 25 iterations *i.e.* each 50 s. We compare here only the results between our new domain adaptation technique and without using mesh adaptation. To this end, we plot the free surface elevation η in the Figures 19 → 21. However, as in the passive case, because of the large variations of the bottom, shorter waves were generated, especially around the CHRISTMAS Island (southwest of JAVA island) and around the undersea canyon near the earthquake epicenter. We plot the variation of η *vs* time (in Figure 22) at four numerical wave gauges placed at the following locations: (i) $(107.345°, -9.295°)$, (ii) $(106.5°, -8°)$, (iii) $(105.9°, -10.35°)$ and (iv) $(107.7°, -11°)$ (see Figure 5 (left panel)) where (i) is the position of the epicenter. Finally, we present a comparison of the kinetic, potential and total energies with the full mesh (in Figure 23, top left panel) and with the domain adaptivity method (in Figure 23, top right) defined in (9), the number of the degrees of freedom (in Figure 23, lower left panel) and the computation time of the simulation (in Figure 23, lower right panel). We obtain here an error of order 2×10^{-5} between the total energy with domain adaptivity and without any adaptation. We present in Figure 24 the comparison of the maximum of the propagation of the solution between the full and domain adaptivity method at $t = 1\,750$ s.



(a) Domain adaptivity (b) Full mesh

Figure 21. *Active generation: comparison between the bottom and the free surface elevation at* $t = 1\,500$ *s between the domain adaptivity method and the full mesh.*

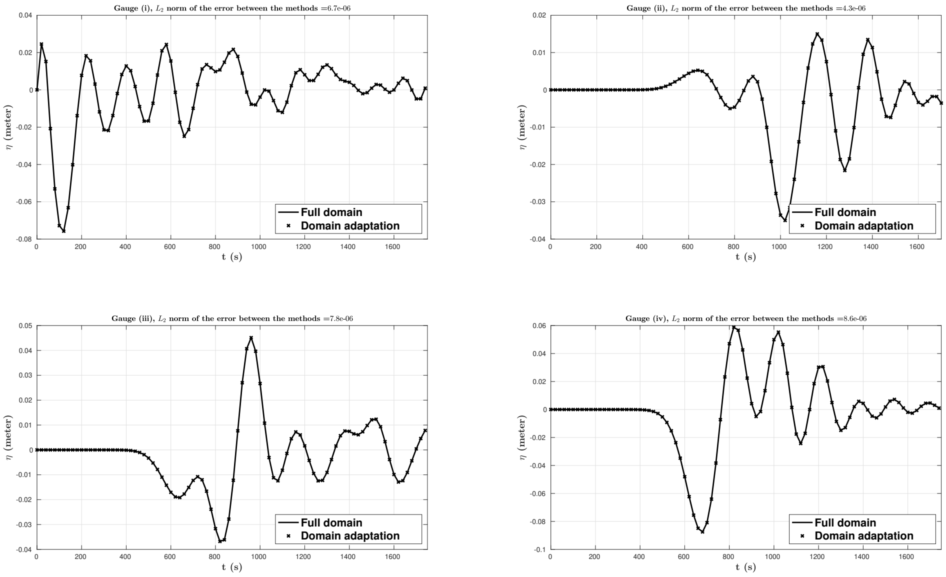


Figure 22. *Active generation: comparison between the two methods (the full one and domain adaptivity) of the free surface elevations (in meters) vs time (in seconds), computed numerically at four wave gauges where the gauge (i) correspond to the epicenter.*

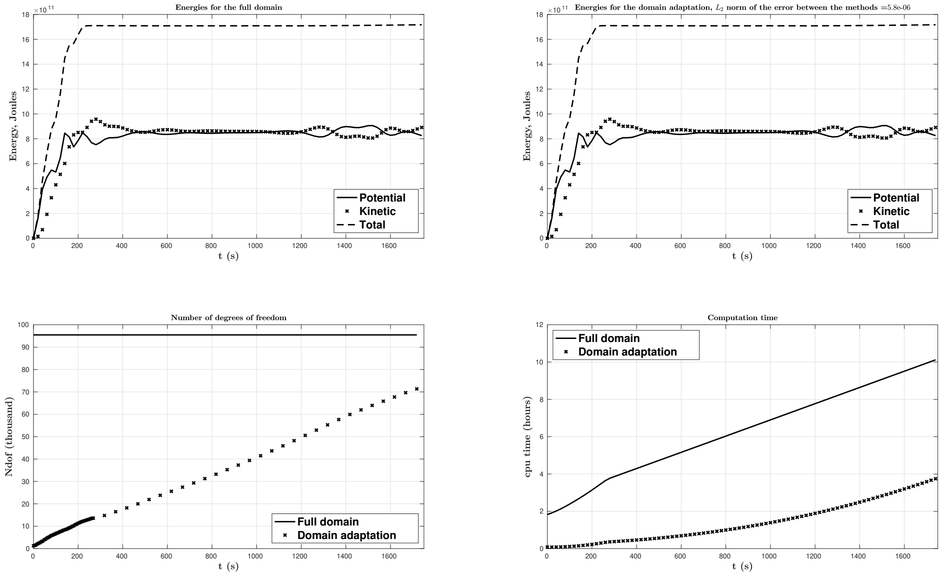
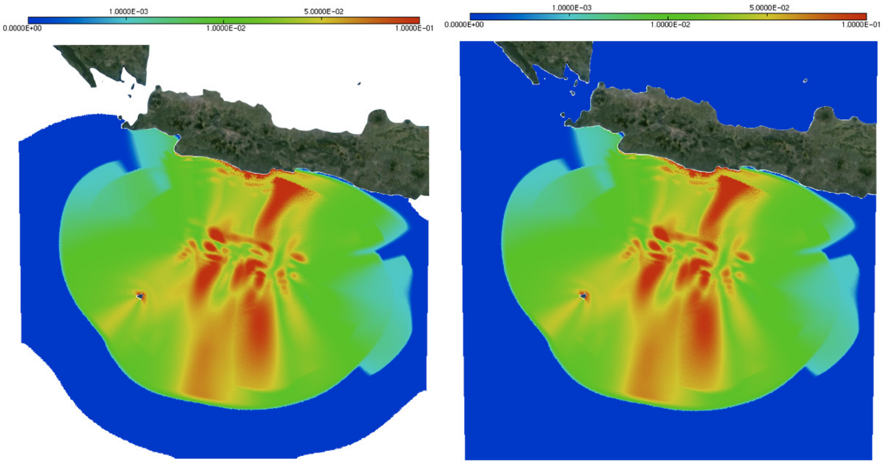


Figure 23. *Active generation: comparison between the two methods (the full one and the domain adaptivity) of the kinetic, potential and total energies, the number of degrees of freedom and the computation time of the simulation.*



(a) (b)

Figure 24. *Active generation: comparison between the maximum of the solution at $t = 1\,750$ s, with the domain adaptivity method (left panel) and with the full one (right panel).*

5. Conclusion and Outlook

In this manuscript we demonstrated how to discretize a simplified version of the BBM–BBM system (1) using the FEM and dedicated open-source software `FreeFem++`. The use of this numerical technique was demonstrated in view of applications to tsunami wave modelling [25,37]. The concrete cases of wave propagation in the MEDITERRANEAN sea and in JAVA island region (INDONESIA) were considered. The digital computing environment that we developed allows the integration of realistic data (bathymetry and geography) in a relatively simple software framework. The codes used in this study are made freely available for all our readers. Moreover, a novel mesh and domain adaptation technique was proposed to speed-up substantially the computations. The gain in terms of the CPU time after applying this technique can be clearly seen in Figure 23. The accuracy of the 'accelerated' solution is more than acceptable to make this technique useful in a variety of tsunami propagation problems. It goes without saying that this technique can be applied to other events and other regions of the world with minimal changes in the provided codes.

Regarding the perspectives of this study, we would like to develop also the parallel version of this code together with the domain adaptation technique to make computations practically *faster* than the real time tsunami wave propagation. However, we underline that even the current version can be efficiently run even on a modest laptop personal computer. There is another direction that we can see to improve the proposed method. Namely, the idea could be called the 'un-adaptivity', which consists in removing the portions of the mesh once the wave passed by. This would allow us to keep only the portions of the computational domain where something is going on.

Author Contributions: Conceptualization, G.S. and D.D.; methodology, G.S.; software, G.S.; validation, G.S. and D.D.; investigation, G.S.; writing–original draft preparation, G.S.; writing–review and editing, D.D.; visualization, G.S.; supervision, D.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the French National Research Agency, through Investments for Future Program (ref. ANR−18−EURE−0016 — Solar Academy).

Acknowledgments: This work would not be possible without a precious help of Professors Frédéric HECHT, Dimitrios MITSOTAKIS, and Olivier PANTZ. The Authors would like to thank also both Referees for their invaluable comments which allowed us to improve our manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

2D	Two-dimensional
BBM	BENJAMIN–BONA–MAHONY
BC	Boundary Condition
FEM	Finite Element Method
`FreeFem++`	Free Finite Element Method
LGPL	GNU Lesser General Public License
MOST	Method Of Splitting Tsunami
NOAA	National Oceanic and Atmospheric Administration
PDE	Partial Differential Equation
USGS	United States Geological Survey

Appendix A. Simplified system derivation

After integrating by parts, the left hand side of (3) becomes:

$$-\left\langle b\nabla\cdot\left(h^2\nabla\partial_t\mathcal{E}\right);\Phi^\eta\right\rangle = b\left\langle h^2\nabla\partial_t\mathcal{E};\nabla(\Phi^\eta)\right\rangle - \int_{\Gamma_n} bh^2\Phi^\eta\frac{\partial(\partial_t\mathcal{E})}{\partial n}\partial\gamma,$$

$$-\left\langle dh^2\Delta\partial_t\mathcal{U};\Phi^u\right\rangle = d\left\langle h^2\nabla\partial_t\mathcal{U};\nabla\Phi^u\right\rangle + d\left\langle 2h\nabla h\cdot\nabla\partial_t\mathcal{U};\Phi^u\right\rangle - \int_{\Gamma_n} dh^2\Phi^u\frac{\partial(\partial_t\mathcal{U})}{\partial n}\partial\gamma,$$

and

$$-\left\langle dh^2\Delta\partial_t\mathcal{V};\Phi^v\right\rangle = d\left\langle h^2\nabla\partial_t\mathcal{V};\nabla\Phi^v\right\rangle + d\left\langle 2h\nabla h\cdot\nabla\partial_t\mathcal{V};\Phi^v\right\rangle - \int_{\Gamma_n} dh^2\Phi^v\frac{\partial(\partial_t\mathcal{V})}{\partial n}\partial\gamma,$$

where Γ_n is the boundary of the domain Ω. Dealing with the right-hand side $\mathcal{F}(\mathcal{E},\mathcal{U},\mathcal{V},\Phi^\eta)$ of the first equation in System (3), we expand the two complex terms which are multiplied by A and $\tilde{A}$ such as:

$$\left\langle\nabla\cdot\left(h^2\nabla\zeta_t\right);\Phi^\eta\right\rangle = \left\langle\left(h^2\zeta_{xt}\right)_x + \left(h^2\zeta_{yt}\right)_y;\Phi^\eta\right\rangle = \left\langle 2hh_x\zeta_{xt} + h^2\zeta_{xxt} + 2hh_y\zeta_{yt} + h^2\zeta_{yyt};\Phi^\eta\right\rangle,$$

and

$$\begin{aligned}
&\left\langle\nabla\cdot\left\{h^2\left[\nabla\left(\nabla h\cdot(\mathcal{U};\mathcal{V})\right) + \nabla h\nabla\cdot(\mathcal{U};\mathcal{V})\right]\right\};\Phi^\eta\right\rangle\\
&= \left\langle\nabla\cdot\left\{h^2\left[\left((h_x\mathcal{U}+h_y\mathcal{V})_x;(h_x\mathcal{U}+h_y\mathcal{V})_y\right) + \left(h_x\nabla\cdot(\mathcal{U};\mathcal{V});h_y\nabla\cdot(\mathcal{U};\mathcal{V})\right)\right]\right\};\Phi^\eta\right\rangle\\
&= \left\langle\nabla\cdot\left(h^2h_{xx}\mathcal{U} + h^2h_x\mathcal{U}_x + h^2h_{xy}\mathcal{V} + h^2h_y\mathcal{V}_x + h^2h_x\nabla\cdot(\mathcal{U};\mathcal{V});h^2h_{xy}\mathcal{U} + h^2h_x\mathcal{U}_y\right.\right.\\
&\qquad\left.\left. + h^2h_{yy}\mathcal{V} + h^2h_y\mathcal{V}_y + h^2h_y\nabla\cdot(\mathcal{U};\mathcal{V})\right);\Phi^\eta\right\rangle\\
&= \left\langle(2hh_xh_{xx} + 2hh_yh_{xy} + h^2h_{xyy} + h^2h_{xxx})\mathcal{U} + (2hh_xh_{xy} + 2hh_yh_{yy} + h^2h_{yyy}\right.\\
&\quad + h^2h_{xxy})\mathcal{V} + (4hh_x^2 + 3h^2h_{xx} + 2hh_y^2 + h^2h_{yy})\mathcal{U}_x + 2(h^2h_{xy} + hh_xh_y)\mathcal{U}_y + (4hh_y^2\\
&\quad \left. + 3h^2h_{yy} + 2hh_x^2 + h^2h_{xx})\mathcal{V}_y + 2(hh_xh_y + h^2h_{xy})\mathcal{V}_x;\Phi^\eta\right\rangle + \left(\left\langle 2h^2h_x\mathcal{U}_{xx};\Phi^\eta\right\rangle\right.\\
&\quad \left. + \left\langle h^2h_y\mathcal{U}_{xy};\Phi^\eta\right\rangle + \left\langle h^2h_x\mathcal{U}_{yy};\Phi^\eta\right\rangle + \left\langle h^2h_y\mathcal{V}_{xx};\Phi^\eta\right\rangle + \left\langle h^2h_x\mathcal{V}_{xy};\Phi^\eta\right\rangle + \left\langle 2h^2h_y\mathcal{V}_{yy};\Phi^\eta\right\rangle\right).
\end{aligned}$$

On the other hand, we have:

$$\begin{aligned}
\left\langle 2h^2h_x\mathcal{U}_{xx};\Phi^\eta\right\rangle &= -\left\langle 2h^2h_x\mathcal{U}_x;\Phi^\eta_x\right\rangle - \left\langle(4hh_x^2 + 2h^2h_{xx})\mathcal{U}_x;\Phi^\eta\right\rangle + \int_{\Gamma_n} 2h^2h_x\Phi^\eta\frac{\partial\mathcal{U}}{\partial n}\partial\gamma,\\
\left\langle h^2h_y\mathcal{U}_{xy};\Phi^\eta\right\rangle &= -\left\langle h^2h_y\mathcal{U}_x;\Phi^\eta_y\right\rangle - \left\langle(2hh_y^2 + h^2h_{yy})\mathcal{U}_x;\Phi^\eta\right\rangle + \int_{\Gamma_n} h^2h_y\Phi^\eta\frac{\partial\mathcal{U}}{\partial n}\partial\gamma,\\
\left\langle h^2h_x\mathcal{U}_{yy};\Phi^\eta\right\rangle &= -\left\langle h^2h_x\mathcal{U}_y;\Phi^\eta_y\right\rangle - \left\langle(2hh_xh_y + h^2h_{xy})\mathcal{U}_y;\Phi^\eta\right\rangle + \int_{\Gamma_n} h^2h_x\Phi^\eta\frac{\partial\mathcal{U}}{\partial n}\partial\gamma,\\
\left\langle h^2h_y\mathcal{V}_{xx};\Phi^\eta\right\rangle &= -\left\langle h^2h_y\mathcal{V}_x;\Phi^\eta_x\right\rangle - \left\langle(2hh_xh_y + h^2h_{xy})\mathcal{V}_x;\Phi^\eta\right\rangle + \int_{\Gamma_n} h^2h_y\Phi^\eta\frac{\partial\mathcal{V}}{\partial n}\partial\gamma,\\
\left\langle h^2h_x\mathcal{V}_{xy};\Phi^\eta\right\rangle &= -\left\langle h^2h_x\mathcal{V}_x;\Phi^\eta_y\right\rangle - \left\langle(2hh_xh_y + h^2h_{xy})\mathcal{V}_x;\Phi^\eta\right\rangle + \int_{\Gamma_n} h^2h_x\Phi^\eta\frac{\partial\mathcal{V}}{\partial n}\partial\gamma,\\
\left\langle 2h^2h_y\mathcal{V}_{yy};\Phi^\eta\right\rangle &= -\left\langle 2h^2h_y\mathcal{V}_y;\Phi^\eta_y\right\rangle - \left\langle(4hh_y^2 + 2h^2h_{yy})\mathcal{V}_y;\Phi^\eta\right\rangle + \int_{\Gamma_n} 2h^2h_y\Phi^\eta\frac{\partial\mathcal{V}}{\partial n}\partial\gamma,
\end{aligned}$$

and, consequently, we deduce the final form of $\mathcal{F}(\mathcal{E},\mathcal{U},\mathcal{V},\Phi^\eta)$ as follows:

$$\begin{aligned}
\mathcal{F}(\mathcal{E},\mathcal{U},\mathcal{V},\Phi^\eta) &= -\left\langle(h+\mathcal{E})\nabla\cdot(\mathcal{U};\mathcal{V}) + (h_x+\mathcal{E}_x)\mathcal{U} + (h_y+\mathcal{E}_y)\mathcal{V} + \zeta_t;\Phi^\eta\right\rangle\\
&\quad -\tilde{A}\left\langle 2hh_x\zeta_{xt} + h^2\zeta_{xxt} + 2hh_y\zeta_{yt} + h^2\zeta_{yyt};\Phi^\eta\right\rangle - A\left\langle(2hh_xh_{xx} + 2hh_yh_{xy} + h^2h_{xyy}\right.\\
&\quad + h^2h_{xxx})\mathcal{U} + (2hh_xh_{xy} + 2hh_yh_{yy} + h^2h_{yyy} + h^2h_{xxy})\mathcal{V} + h^2h_{xx}\mathcal{U}_x + h^2h_{xy}\mathcal{U}_y - 2hh_xh_y\mathcal{V}_x
\end{aligned}$$

$$+(h^2h_{yy}+2hh_x^2+h^2h_{xx})\mathcal{V}_y;\Phi^\eta\Big\rangle + A\left(\Big\langle 2h^2h_x\mathcal{U}_x+h^2h_y\mathcal{V}_x;\Phi^\eta_x\Big\rangle + \Big\langle h^2h_y\mathcal{U}_x+h^2h_x\mathcal{U}_y\right.$$

$$\left.+h^2h_x\mathcal{V}_x+2h^2h_y\mathcal{V}_y;\Phi^\eta_y\Big\rangle\right) - A\int_{\Gamma_n}\left((3h^2h_x+h^2h_y)\Phi^\eta\frac{\partial\mathcal{U}}{\partial n}+(h^2h_x+3h^2h_y)\Phi^\eta\frac{\partial\mathcal{V}}{\partial n}\right)\partial\gamma.$$

For the right-hand side $\mathcal{G}\left(\mathcal{E},\mathcal{U},\mathcal{V},\Phi^u\right)$ of the second equation in System (3), we have:

$$\mathcal{G}\left(\mathcal{E},\mathcal{U},\mathcal{V},\Phi^u\right) = -\left\langle g\mathcal{E}_x+\mathcal{U}\mathcal{U}_x+\mathcal{V}\mathcal{V}_x+Bgh\left[\left(h_x\mathcal{E}_x+h_y\mathcal{E}_y\right)_x+h_x(\mathcal{E}_{xx}+\mathcal{E}_{yy})\right]-Bh\zeta_{xtt};\Phi^u\right\rangle$$

$$= -\left\langle g\mathcal{E}_x+\mathcal{U}\mathcal{U}_x+\mathcal{V}\mathcal{V}_x+Bg\left(hh_{xx}\mathcal{E}_x+hh_{xy}\mathcal{E}_y\right)-Bh\zeta_{xtt};\Phi^u\right\rangle - Bg\left\langle 2hh_x\mathcal{E}_{xx}+hh_y\mathcal{E}_{xy}+hh_x\mathcal{E}_{yy};\Phi^u\right\rangle$$

$$= -\left\langle g\mathcal{E}_x+\mathcal{U}\mathcal{U}_x+\mathcal{V}\mathcal{V}_x+Bg\left(hh_{xx}\mathcal{E}_x+hh_{xy}\mathcal{E}_y\right)-Bh\zeta_{xtt};\Phi^u\right\rangle + Bg\left\langle 2hh_x\mathcal{E}_x;\Phi^u_x\right\rangle + Bg\left\langle (2h_x^2+2hh_{xx})\mathcal{E}_x;\Phi^u\right\rangle$$

$$+Bg\left\langle hh_y\mathcal{E}_x+hh_x\mathcal{E}_y;\Phi^u_y\right\rangle + Bg\left\langle (h_y^2+hh_{yy})\mathcal{E}_x+(h_xh_y+hh_{xy})\mathcal{E}_y;\Phi^u\right\rangle - \int_{\Gamma_n} Bg(3hh_x+hh_y)\Phi^u\frac{\partial\mathcal{E}}{\partial n}\partial\gamma$$

$$= -\left\langle g\left(I_d-B\left(hh_{xx}+2h_x^2+hh_{yy}+h_y^2\right)\right)\mathcal{E}_x+\mathcal{U}\mathcal{U}_x+\mathcal{V}\mathcal{V}_x-Bgh_xh_y\mathcal{E}_y-Bh\zeta_{xtt};\Phi^u\right\rangle + Bg\left\langle 2hh_x\mathcal{E}_x;\Phi^u_x\right\rangle$$

$$+Bg\left\langle hh_y\mathcal{E}_x+hh_x\mathcal{E}_y;\Phi^u_y\right\rangle - \int_{\Gamma_n} Bg(3hh_x+hh_y)\Phi^u\frac{\partial\mathcal{E}}{\partial n}\partial\gamma.$$

Finally, for the right-hand side $\mathcal{H}\left(\mathcal{E},\mathcal{U},\mathcal{V},\Phi^v\right)$ of the third equation in System 3, we have:

$$\mathcal{H}\left(\mathcal{E},\mathcal{U},\mathcal{V},\Phi^v\right) = -\left\langle g\mathcal{E}_y+\mathcal{U}\mathcal{U}_y+\mathcal{V}\mathcal{V}_y+Bgh\left[\left(h_x\mathcal{E}_x+h_y\mathcal{E}_y\right)_y+h_y(\mathcal{E}_{xx}+\mathcal{E}_{yy})\right]-Bh\zeta_{ytt};\Phi^v\right\rangle$$

$$= -\left\langle g\mathcal{E}_y+\mathcal{U}\mathcal{U}_y+\mathcal{V}\mathcal{V}_y+Bg\left(hh_{xy}\mathcal{E}_x+hh_{yy}\mathcal{E}_y\right)-Bh\zeta_{ytt};\Phi^v\right\rangle - Bg\left\langle hh_y\mathcal{E}_{xx}+hh_x\mathcal{E}_{xy}+2hh_y\mathcal{E}_{yy});\Phi^v\right\rangle$$

$$= -\left\langle g\mathcal{E}_y+\mathcal{U}\mathcal{U}_y+\mathcal{V}\mathcal{V}_y+Bg\left(hh_{xy}\mathcal{E}_x+hh_{yy}\mathcal{E}_y\right)-Bh\zeta_{ytt};\Phi^v\right\rangle + Bg\left\langle hh_y\mathcal{E}_x;\Phi^v_x\right\rangle + Bg\left\langle (h_xh_y+hh_{xy})\mathcal{E}_x;\Phi^v\right\rangle$$

$$+Bg\left\langle hh_x\mathcal{E}_x+2hh_y\mathcal{E}_y;\Phi^v_y\right\rangle + Bg\left\langle (h_xh_y+hh_{xy})\mathcal{E}_x+(2h_y^2+2hh_{yy})\mathcal{E}_y;\Phi^v\right\rangle - \int_{\Gamma_n} Bg(hh_x+3hh_y)\Phi^v\frac{\partial\mathcal{E}}{\partial n}\partial\gamma$$

$$= -\left\langle -Bg(2h_xh_y+hh_{xy})\mathcal{E}_x+\mathcal{U}\mathcal{U}_y+\mathcal{V}\mathcal{V}_y+g\left(I_d-B\left(hh_{yy}-2h_y^2\right)\right)\mathcal{E}_y-Bh\zeta_{ytt};\Phi^v\right\rangle + Bg\left\langle hh_y\mathcal{E}_x;\Phi^v_x\right\rangle$$

$$+Bg\left\langle hh_x\mathcal{E}_x+2hh_y\mathcal{E}_y;\Phi^v_y\right\rangle - \int_{\Gamma_n} Bg(hh_x+3hh_y)\Phi^v\frac{\partial\mathcal{E}}{\partial n}\partial\gamma.$$

References

1. Sadaka, G.; Dutykh, D. Adaptive numerical modelling of tsunami wave generation and propagation with FreeFem++. *Preprint* **2020**, *hal-029125*, 1–24. pages 1
2. Mitsotakis, D.E. Boussinesq systems in two space dimensions over a variable bottom for the generation and propagation of tsunami waves. *Math. Comp. Simul.* **2009**, *80*, 860–873. pages 1, 2, 4, 8
3. Hecht, F.; Auliac, S.; Pironneau, O.; Morice, J.; Le Hyaric, A.; Ohtsuka, K. *FreeFem++ (manual)*, 3 ed.; www.freefem.org: Paris, 2007; p. 424. pages 1
4. Imamura, F. Simulation of wave-packet propagation along sloping beach by TUNAMI-code. In *Long-wave Runup Models*; Yeh, H.; Liu, P.L.F.; Synolakis, C.E., Eds.; World Scientific: Singapore, 1996; pp. 231–241. pages 2
5. Titov, V.V.; González, F.I. Implementation and testing of the method of splitting tsunami (MOST) model. Technical Report ERL PMEL-112, Pacific Marine Environmental Laboratory, NOAA, 1997. pages
6. Dutykh, D.; Poncet, R.; Dias, F. The VOLNA code for the numerical modeling of tsunami waves: Generation, propagation and inundation. *Eur. J. Mech. B/Fluids* **2011**, *30*, 598–615, [1002.4553]. doi:10.1016/j.euromechflu.2011.05.005. pages
7. Khakimzyanov, G.S.; Dutykh, D.; Mitsotakis, D.; Shokina, N.Y. Numerical simulation of conservation laws with moving grid nodes: Application to tsunami wave modelling. *Geosciences* **2019**, *9*, 197. doi:10.3390/geosciences9050197. pages 2

8. Boussinesq, J.V. Théorie générale des mouvements qui sont propagés dans un canal rectangulaire horizontal. *C. R. Acad. Sc. Paris* **1871**, *73*, 256–260. pages 2

9. Wu, T.Y. Long Waves in Ocean and Coastal Waters. *Journal of Engineering Mechanics* **1981**, *107*, 501–522. pages 2

10. Dias, F.; Dutykh, D.; O'Brien, L.; Renzi, E.; Stefanakis, T. On the Modelling of Tsunami Generation and Tsunami Inundation. *Procedia IUTAM* **2014**, *10*, 338–355. doi:10.1016/j.piutam.2014.01.029. pages

11. Dutykh, D.; Mitsotakis, D.; Gardeil, X.; Dias, F. On the use of the finite fault solution for tsunami generation problems. *Theor. Comput. Fluid Dyn.* **2013**, *27*, 177–199. doi:10.1007/s00162-011-0252-8. pages 8, 9, 10, 15

12. Guesmia, M.; Heinrich, P.H.; Mariotti, C. Numerical simulation of the 1969 Portuguese tsunami by a finite element method. *Natural Hazards* **1998**, *17*, 31–46. pages

13. Lynett, P.J.; Borrero, J.C.; Liu, P.L.F.; Synolakis, C.E. Field Survey and Numerical Simulations: A Review of the 1998 Papua New Guinea Tsunami. *Pure Appl. Geophys.* **2003**, *160*, 2119–2146. doi:10.1007/s00024-003-2422-0. pages 2

14. Whitham, G.B. *Linear and Nonlinear Waves*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1999; p. 656. doi:10.1002/9781118032954. pages 2

15. Stoker, J.J. *Water Waves: The Mathematical Theory with Applications*; John Wiley and Sons, Inc.: Hoboken, NJ, USA, 1992; p. 600. doi:10.1002/9781118033159. pages 2

16. Peregrine, D.H. Calculations of the development of an undular bore. *J. Fluid Mech.* **1966**, *25*, 321–330. doi:10.1017/S0022112066001678. pages 2

17. Peregrine, D.H. Long waves on a beach. *J. Fluid Mech.* **1967**, *27*, 815–827. pages 2

18. Løvholt, F.; Pedersen, G. Instabilities of Boussinesq models in non-uniform depth. *Int. J. Num. Meth. Fluids* **2009**, *61*, 606–637. doi:10.1002/fld.1968. pages 3

19. Liang, Q.; Hou, J.; Amouzgar, R. Simulation of Tsunami Propagation Using Adaptive Cartesian Grids. *Coastal Engineering Journal* **2015**, *57*, 1550016–1–1550016–30. doi:10.1142/S0578563415500163. pages 5

20. Popinet, S. A vertically-Lagrangian, non-hydrostatic, multilayer model for multiscale free-surface flows. *J. Comp. Phys.* **2020**, *418*, 109609. doi:10.1016/j.jcp.2020.109609. pages 5

21. Dias, F.; Dutykh, D. Dynamics of tsunami waves. In *Extreme Man-Made and Natural Hazards in Dynamics of Structures*; Ibrahimbegovic, A.; Kozar, I., Eds.; Springer Netherlands, 2007; pp. 35–60. doi:10.1007/978-1-4020-5656-7_8. pages 5

22. Dougalis, V.A.; Mitsotakis, D.E.; Saut, J.C. On initial-boundary value problems for a Boussinesq system of BBM-BBM type in a plane domain. *Discrete Contin. Dyn. Syst.* **2009**, *23*, 1191–1204. doi:10.3934/dcds.2009.23.1191. pages 6

23. Khakimzyanov, G.; Dutykh, D. Long Wave Interaction with a Partially Immersed Body. Part I: Mathematical Models. *Commun. Comput. Phys.* **2020**, *27*, 321–378. doi:10.4208/cicp.OA-2018-0294. pages 6

24. Dutykh, D.; Dias, F. Water waves generated by a moving bottom. In *Tsunami and Nonlinear Waves*; Kundu, A., Ed.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2007; pp. 65–95. doi:10.1007/978-3-540-71256-5_4. pages 8, 10

25. Dutykh, D. Mathematical modelling of tsunami waves. Phd thesis, École Normale Supérieure de Cachan, 2007. pages 8, 10, 21

26. Okada, Y. Surface deformation due to shear and tensile faults in a half-space. *Bull. Seism. Soc. Am.* **1985**, *75*, 1135–1154. pages 8

27. Okada, Y. Internal deformation due to shear and tensile faults in a half-space. *Bull. Seism. Soc. Am.* **1992**, *82*, 1018–1040. pages 8

28. Kervella, Y.; Dutykh, D.; Dias, F. Comparison between three-dimensional linear and nonlinear tsunami generation models. *Theor. Comput. Fluid Dyn.* **2007**, *21*, 245–269. doi:10.1007/s00162-007-0047-0. pages 9

29. Dutykh, D.; Dias, F.; Kervella, Y. Linear theory of wave generation by a moving bottom. *Comptes Rendus Mathématique* **2006**, *343*, 499–504. doi:10.1016/j.crma.2006.09.016. pages 9

30. Dutykh, D.; Mitsotakis, D.; Chubarov, L.B.; Shokin, Y.I. On the contribution of the horizontal sea-bed displacements into the tsunami generation process. *Ocean Modelling* **2012**, *56*, 43–56. doi:10.1016/j.ocemod.2012.07.002. pages 9, 10, 15

31. Hammack, J. A note on tsunamis: their generation and propagation in an ocean of uniform depth. *J. Fluid Mech.* **1973**, *60*, 769–799. pages 10

32. Sadaka, G. Solution of 2D Boussinesq systems with FreeFem++: the flat bottom case. *Journal of Numerical Mathematics* **2012**, *20*, 303–324. doi:10.1515/jnum-2012-0016. pages 10
33. Senthilkumar, A. On the influence of wave reflection on shoaling and breaking solitary Waves. *Proceedings of the Estonian Academy of Sciences* **2016**, *65*, 414–430. doi:10.3176/proc.2016.4.06. pages 10
34. Hecht, F. New development in Freefem++. *Journal of Numerical Mathematics* **2012**, *20*, 251–266. doi:10.1515/jnum-2012-0013. pages 12
35. Rakotondrandisa A., S.G.; I., D. A finite-element toolbox for the simulation of solid-liquid phase-change systems with natural convection. *Computer Physics Communications* **2020**, *253*, 107–188. doi:10.1016/j.cpc.2020.107188. pages 12
36. Dutykh, D.; Dias, F. Energy of tsunami waves generated by bottom motion. *Proc. R. Soc. Lond. A* **2009**, *465*, 725–744. doi:10.1098/rspa.2008.0332. pages 15
37. Sadaka, G. Etude mathématique et numérique d'équations d'ondes aquatiques amorties. Phd thesis, Université de Picardie Jules Verne, 2011. pages 21