



**HAL**  
open science

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

Georges Sadaka, Denys Dutykh

► **To cite this version:**

Georges Sadaka, Denys Dutykh. Adaptive numerical modelling of tsunami wave generation and propagation with FreeFem++. 2020. hal-02912526

**HAL Id: hal-02912526**

**<https://hal.science/hal-02912526>**

Preprint submitted on 6 Aug 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - ShareAlike 4.0 International License

Article

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

Georges SADAKA <sup>1,\*</sup> , Denys DUTYKH <sup>2,†</sup> 

<sup>1</sup> 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

<sup>2</sup> 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.

Academic Editor: Anawat Suppasri

Version August 6, 2020 submitted to Geosciences

**Abstract:** A simplified nonlinear dispersive system of BBM-type, initially derived by D. MITSOTAKIS, is employed here in order to model the generation and propagation of surface water waves over variable bottom. The simplification consists in applying the so-called BOUSSINESQ approximation. Using the finite element method and the FreeFem++ software, 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 rather as a tutorial paper even if it contains the description of completely new adaptation techniques.

**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

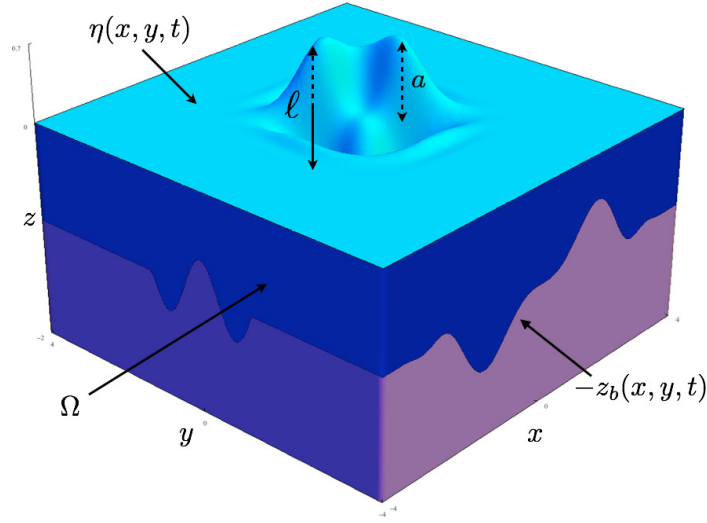
**MSC:** 76B15; 65N30; 65N50

## 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++ open source finite element software. Traditionally, tsunami waves are modelled using hydrostatic models [1–4]. 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 [5]. They can be used to study a variety of water wave phenomena in harbors, coastal dynamics and, of course, tsunami generation and propagation problems [6–10].

In this study we consider a BBM–BBM system derived by D. MITSOTAKIS in 2D over a variable bottom in space  $h(x, y)$  and in time  $\zeta(x, y, t)$  [11]:

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



**Figure 1.** The sketch of the physical domain  $\Omega$ .

where

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

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

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  $\Omega \subset \mathbb{R}^3$  [12,13], 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 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,  $\eta$  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 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) 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. We note that all numerical simulations were done using the FreeFem++ software [14], which is an open source platform to solve Partial Differential Equations (PDEs) numerically, based on Finite Element Methods (FEM). The main conclusions of this study are outlined in Section 5.

## 47 2. Discretization of the BBM–BBM system

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

### 51 2.1. Spatial discretization

We let  $\Omega$  be a convex, plane domain, and  $\mathcal{T}_h$  be a regular, quasi-uniform triangulation of  $\Omega$  with triangles of maximum size  $h < 1$ . Setting  $V_h = \{v_h \in C^0(\bar{\Omega}); v_h|_T \in \mathbb{P}_1(T), \forall T \in \mathcal{T}_h\}$  be a finite-dimensional, where  $\mathbb{P}_1$  is the set of all polynomials of degree  $\leq 1$  with real coefficients and denoting by  $\langle \cdot; \cdot \rangle$  the standard  $L^2$  inner product on  $\Omega$ , 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:

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

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:

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

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

$$\begin{aligned} -\langle b\nabla \cdot (h^2 \nabla \partial_t \mathcal{E}); \Phi^\eta \rangle &= b \langle h^2 \nabla \partial_t \mathcal{E}; \nabla (\Phi^\eta) \rangle - \int_{\Gamma_n} bh^2 \Phi^\eta \frac{\partial(\partial_t \mathcal{E})}{\partial n} \partial \gamma, \\ -\langle dh^2 \Delta \partial_t \mathcal{U}; \Phi^u \rangle &= d \langle h^2 \nabla \partial_t \mathcal{U}; \nabla \Phi^u \rangle + d \langle 2h \nabla h \cdot \nabla \partial_t \mathcal{U}; \Phi^u \rangle - \int_{\Gamma_n} dh^2 \Phi^u \frac{\partial(\partial_t \mathcal{U})}{\partial n} \partial \gamma, \end{aligned}$$

and

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

where  $\Gamma_n$  is the boundary of the domain  $\Omega$ . 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:

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

and

$$\left\langle \nabla \cdot \{h^2 [\nabla (\nabla h \cdot (\mathcal{U}; \mathcal{V})) + \nabla h \nabla \cdot (\mathcal{U}; \mathcal{V})]\}; \Phi^\eta \right\rangle$$

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

On the other hand, we have:

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

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

$$\begin{aligned}
\mathcal{F}(\mathcal{E}, \mathcal{U}, \mathcal{V}, \Phi^\eta) &= - \left\langle (h + \mathcal{E}) \nabla \cdot (\mathcal{U}; \mathcal{V}) + (h_x + \mathcal{E}_x) \mathcal{U} + (h_y + \mathcal{E}_y) \mathcal{V} + \zeta_t; \Phi^\eta \right\rangle \\
&\quad - \tilde{A} \left\langle 2hh_x \zeta_{xt} + h^2 \zeta_{xxt} + 2hh_y \zeta_{yt} + h^2 \zeta_{yyt}; \Phi^\eta \right\rangle - A \left\langle (2hh_x h_{xx} + 2hh_y h_{xy} + h^2 h_{xxy}) \right. \\
&\quad \left. + h^2 h_{xxx} \right\rangle \mathcal{U} + (2hh_x h_{xy} + 2hh_y h_{yy} + h^2 h_{yyy} + h^2 h_{xxy}) \mathcal{V} + h^2 h_{xx} \mathcal{U}_x + h^2 h_{xy} \mathcal{U}_y - 2hh_x h_y \mathcal{V}_x \\
&\quad + (h^2 h_{yy} + 2hh_x^2 + h^2 h_{xx}) \mathcal{V}_y; \Phi^\eta \left. \right\rangle + A \left( \left\langle 2h^2 h_x \mathcal{U}_x + h^2 h_y \mathcal{V}_x; \Phi_x^\eta \right\rangle + \left\langle h^2 h_y \mathcal{U}_x + h^2 h_x \mathcal{U}_y \right. \right. \\
&\quad \left. \left. + h^2 h_x \mathcal{V}_x + 2h^2 h_y \mathcal{V}_y; \Phi_y^\eta \right\rangle \right) - A \int_{\Gamma_n} \left( (3h^2 h_x + h^2 h_y) \Phi^\eta \frac{\partial \mathcal{U}}{\partial n} + (h^2 h_x + 3h^2 h_y) \Phi^\eta \frac{\partial \mathcal{V}}{\partial n} \right) \partial \gamma.
\end{aligned}$$

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

$$\begin{aligned}
\mathcal{G}(\mathcal{E}, \mathcal{U}, \mathcal{V}, \Phi^\mu) &= - \left\langle g \mathcal{E}_x + \mathcal{U} \mathcal{U}_x + \mathcal{V} \mathcal{V}_x + Bgh \left[ (h_x \mathcal{E}_x + h_y \mathcal{E}_y)_x + h_x (\mathcal{E}_{xx} + \mathcal{E}_{yy}) \right] - Bh \zeta_{xtt}; \Phi^\mu \right\rangle \\
&= - \left\langle g \mathcal{E}_x + \mathcal{U} \mathcal{U}_x + \mathcal{V} \mathcal{V}_x + Bg (hh_{xx} \mathcal{E}_x + hh_{xy} \mathcal{E}_y) - Bh \zeta_{xtt}; \Phi^\mu \right\rangle - Bg \left\langle 2hh_x \mathcal{E}_{xx} + hh_y \mathcal{E}_{xy} + hh_x \mathcal{E}_{yy}; \Phi^\mu \right\rangle \\
&= - \left\langle g \mathcal{E}_x + \mathcal{U} \mathcal{U}_x + \mathcal{V} \mathcal{V}_x + Bg (hh_{xx} \mathcal{E}_x + hh_{xy} \mathcal{E}_y) - Bh \zeta_{xtt}; \Phi^\mu \right\rangle + Bg \left\langle 2hh_x \mathcal{E}_x; \Phi_x^\mu \right\rangle + Bg \left\langle (2h_x^2 + 2hh_{xx}) \mathcal{E}_x; \Phi^\mu \right\rangle \\
&\quad + Bg \left\langle hh_y \mathcal{E}_x + hh_x \mathcal{E}_y; \Phi_y^\mu \right\rangle + Bg \left\langle (h_y^2 + hh_{yy}) \mathcal{E}_x + (h_x h_y + hh_{xy}) \mathcal{E}_y; \Phi^\mu \right\rangle - \int_{\Gamma_n} Bg (3hh_x + hh_y) \Phi^\mu \frac{\partial \mathcal{E}}{\partial n} \partial \gamma \\
&= - \left\langle g \left( I_d - B \left( hh_{xx} + 2h_x^2 + hh_{yy} + h_y^2 \right) \right) \mathcal{E}_x + \mathcal{U} \mathcal{U}_x + \mathcal{V} \mathcal{V}_x - Bgh h_y \mathcal{E}_y - Bh \zeta_{xtt}; \Phi^\mu \right\rangle + Bg \left\langle 2hh_x \mathcal{E}_x; \Phi_x^\mu \right\rangle
\end{aligned}$$

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

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

$$\begin{aligned} \mathcal{H}(\mathcal{E}, \mathcal{U}, \mathcal{V}, \Phi^v) &= - \left\langle g \mathcal{E}_y + \mathcal{U} \mathcal{U}_y + \mathcal{V} \mathcal{V}_y + Bgh \left[ (h_x \mathcal{E}_x + h_y \mathcal{E}_y)_y + h_y (\mathcal{E}_{xx} + \mathcal{E}_{yy}) \right] - Bh \zeta_{ytt}; \Phi^v \right\rangle \\ &= - \left\langle g \mathcal{E}_y + \mathcal{U} \mathcal{U}_y + \mathcal{V} \mathcal{V}_y + Bg (hh_{xy} \mathcal{E}_x + hh_{yy} \mathcal{E}_y) - Bh \zeta_{ytt}; \Phi^v \right\rangle - Bg \left\langle hh_y \mathcal{E}_{xx} + hh_x \mathcal{E}_{xy} + 2hh_y \mathcal{E}_{yy}; \Phi^v \right\rangle \\ &= - \left\langle g \mathcal{E}_y + \mathcal{U} \mathcal{U}_y + \mathcal{V} \mathcal{V}_y + Bg (hh_{xy} \mathcal{E}_x + hh_{yy} \mathcal{E}_y) - Bh \zeta_{ytt}; \Phi^v \right\rangle + Bg \left\langle hh_y \mathcal{E}_x; \Phi_x^v \right\rangle + Bg \left\langle (h_x h_y + hh_{xy}) \mathcal{E}_x; \Phi^v \right\rangle \\ &+ Bg \left\langle hh_x \mathcal{E}_x + 2hh_y \mathcal{E}_y; \Phi_y^v \right\rangle + Bg \left\langle (h_x h_y + hh_{xy}) \mathcal{E}_x + (2h_y^2 + 2hh_{yy}) \mathcal{E}_y; \Phi^v \right\rangle - \int_{\Gamma_n} Bg(hh_x + 3hh_y) \Phi^v \frac{\partial \mathcal{E}}{\partial n} \partial \gamma \\ &= - \left\langle -Bg(2h_x h_y + hh_{xy}) \mathcal{E}_x + \mathcal{U} \mathcal{U}_y + \mathcal{V} \mathcal{V}_y + g \left( I_d - B (hh_{yy} - 2h_y^2) \right) \mathcal{E}_y - Bh \zeta_{ytt}; \Phi^v \right\rangle + Bg \left\langle hh_y \mathcal{E}_x; \Phi_x^v \right\rangle \\ &+ Bg \left\langle hh_x \mathcal{E}_x + 2hh_y \mathcal{E}_y; \Phi_y^v \right\rangle - \int_{\Gamma_n} Bg(hh_x + 3hh_y) \Phi^v \frac{\partial \mathcal{E}}{\partial n} \partial \gamma. \end{aligned}$$

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 [15]. 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 derivatives of the bathymetry function as it was proposed earlier in [11]. Thus, from now on we shall use the following system of equations:

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

with

$$\begin{aligned} \mathcal{F}(\mathcal{E}, \mathcal{U}, \mathcal{V}, \Phi^\eta) &= - \left\langle (h + \mathcal{E}) \nabla \cdot (\mathcal{U}; \mathcal{V}) + (h_x + \mathcal{E}_x) \mathcal{U} + (h_y + \mathcal{E}_y) \mathcal{V} + \zeta_t; \Phi^\eta \right\rangle \\ &- \tilde{A} \left\langle 2hh_x \zeta_{xt} + 2hh_y \zeta_{yt}; \Phi^\eta \right\rangle - A \left\langle -2hh_x h_y \mathcal{V}_x + 2hh_x^2 \mathcal{V}_y; \Phi^\eta \right\rangle + A \left( \left\langle 2h^2 h_x \mathcal{U}_x + h^2 h_y \mathcal{V}_x; \Phi_x^\eta \right\rangle + \left\langle h^2 h_y \mathcal{U}_x \right. \right. \\ &\left. \left. + h^2 h_x \mathcal{U}_y + h^2 h_x \mathcal{V}_x + 2h^2 h_y \mathcal{V}_y; \Phi_y^\eta \right\rangle \right) - A \int_{\Gamma_n} \left( (3h^2 h_x + h^2 h_y) \Phi^\eta \frac{\partial \mathcal{U}}{\partial n} + (h^2 h_x + 3h^2 h_y) \Phi^\eta \frac{\partial \mathcal{V}}{\partial n} \right) \partial \gamma, \\ \mathcal{G}(\mathcal{E}, \mathcal{U}, \mathcal{V}, \Phi^u) &= - \left\langle g \left( I_d - B (2h_x^2 + h_y^2) \right) \mathcal{E}_x + \mathcal{U} \mathcal{U}_x + \mathcal{V} \mathcal{V}_x - Bgh_x h_y \mathcal{E}_y - Bh \zeta_{xtt}; \Phi^u \right\rangle + Bg \left\langle 2hh_x \mathcal{E}_x; \Phi_x^u \right\rangle \\ &+ Bg \left\langle hh_y \mathcal{E}_x + hh_x \mathcal{E}_y; \Phi_y^u \right\rangle - \int_{\Gamma_n} Bg(3hh_x + hh_y) \Phi^u \frac{\partial \mathcal{E}}{\partial n} \partial \gamma, \end{aligned}$$

and

$$\begin{aligned} \mathcal{H}(\mathcal{E}, \mathcal{U}, \mathcal{V}, \Phi^v) &= - \left\langle -2Bg h_x h_y \mathcal{E}_x + \mathcal{U} \mathcal{U}_y + \mathcal{V} \mathcal{V}_y + g \left( I_d - 2Bh_y^2 \right) \mathcal{E}_y - Bh \zeta_{ytt}; \Phi^v \right\rangle + Bg \left\langle hh_y \mathcal{E}_x; \Phi_x^v \right\rangle \\ &+ Bg \left\langle hh_x \mathcal{E}_x + 2hh_y \mathcal{E}_y; \Phi_y^v \right\rangle - \int_{\Gamma_n} Bg(hh_x + 3hh_y) \Phi^v \frac{\partial \mathcal{E}}{\partial n} \partial \gamma. \end{aligned}$$

## 52 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  $\delta t$  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:

$$\begin{cases} \langle \mathcal{E}^{n+1}; \Phi^\eta \rangle = \langle \mathcal{E}^n + \frac{\mathcal{E}^{k1} + \mathcal{E}^{k2}}{2}; \Phi^\eta \rangle, \\ \langle \mathcal{U}^{n+1}; \Phi^u \rangle = \langle \mathcal{U}^n + \frac{\mathcal{U}^{k1} + \mathcal{U}^{k2}}{2}; \Phi^u \rangle, \\ \langle \mathcal{V}^{n+1}; \Phi^v \rangle = \langle \mathcal{V}^n + \frac{\mathcal{V}^{k1} + \mathcal{V}^{k2}}{2}; \Phi^v \rangle, \end{cases} \quad (5)$$

where

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

and

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

## 53 3. New domain adaptation, domains computation and initial data

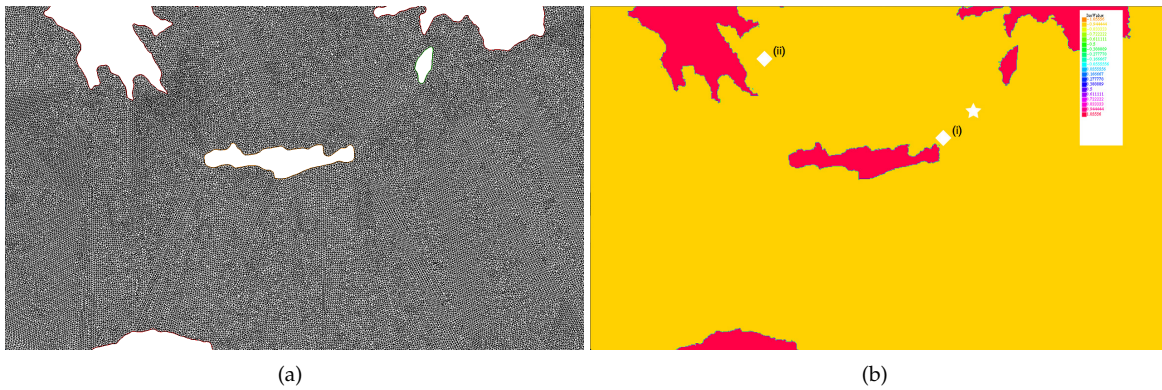
54 We present here the new domain adaptation technique that will be compared in the sequel with  
55 the mesh adaptation used in FreeFem++.

### 56 3.1. New domain adaptation technique

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

66 Firstly, we cut from the initial mesh  $\text{Th}_{\text{init}}$  a circle or a rectangle zone  $\text{Th}$  where our initial solution  
67 lives (using `trunc` in FreeFem++), secondly we compute the initial solution  $u_0$  and we interpolate it to  
68  $u_{\text{adapt}} \in \mathbb{P}_1$  finite element (using `interpolate` in FreeFem++), and for each adaptation we follow this  
69 algorithm:

- 70 • We deduce the limit min max of  $\text{Th}$  on  $x$  and  $y$  direction (using `boundingbox` in FreeFem++).
- 71 • We add `epsadapt` from each side in order to build the new rectangle  $\text{Th}_1$  (cutted from  $\text{Th}_{\text{init}}$ )  
72 that contains  $\text{Th}$  (using `trunc` in FreeFem++).



**Figure 2.** Left (a): the mesh around CRETE island. Right (b): the place of  $\diamond$ : wave gauge and  $\star$ : epicenter.

- 73
- We interpolate  $u_{\text{adapt}} \in \text{Th}$  to  $u_{\text{adapt1}} \in \text{Th1}$  (using `interpolate` in `FreeFem++`).
  - We smooth the function obtained from  $\text{abs}(u_{\text{adapt1}}) \geq \text{err}_{\text{adapt}}$  using:

$$\beta u - \Delta u = \beta f, \quad (8)$$

74 where  $f = (|u_{\text{adapt1}}| \geq \text{err}_{\text{adapt}})$ , with zero DIRICHLET BC only on the new boundary label  
 75 of  $\text{Th1}$  and a NEUMANN BC in the other boundary label.

- 76
- We cut from  $\text{Th1}$ , respecting to  $u > \text{iso}_{\text{adapt}}$ , the final mesh  $\text{Th}_{\text{new}}$  (using `trunc` in `FreeFem++`).

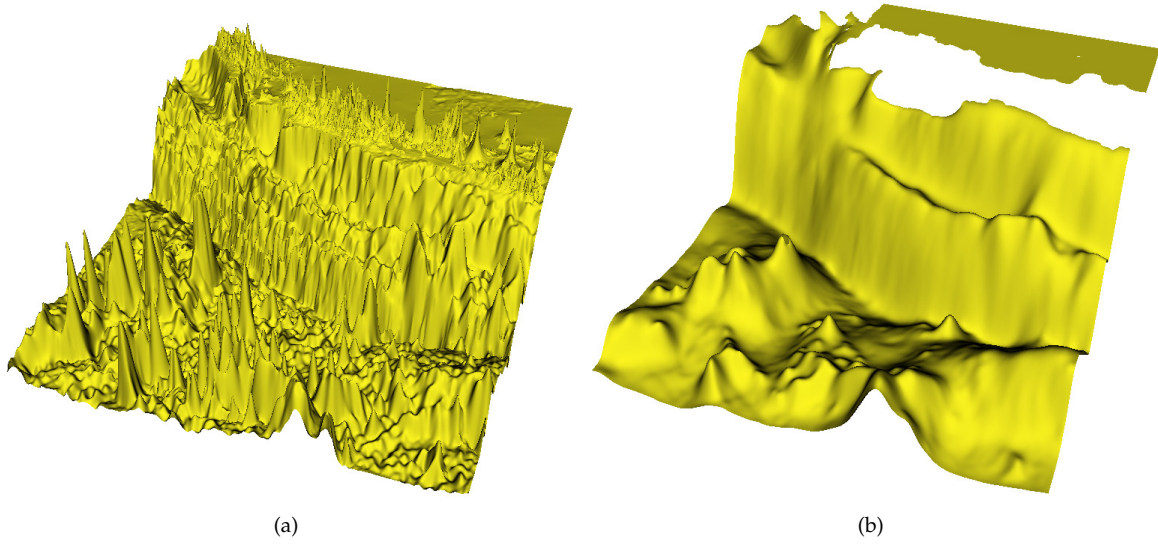
77 Finally, we replace  $\text{Th}$  by the new mesh  $\text{Th}_{\text{new}}$  and we interpolate the solution from the old mesh to  
 78 the new one (using `interpolate` in `FreeFem++`). We use a reflective Boundary Condition (BC) on the  
 79 new boundary, *i.e.* zero NEUMANN BC for  $\eta$  and zero DIRICHLET BC for  $V$ , cause our BBM–BBM  
 80 system gives artificial numerical explosion on the boundary if we do not use any BC or if we use only  
 81 NEUMANN BC for  $\eta$  and  $V$ .

82 For the BBM–BBM system over a flat bottom, we use a mesh generated through a photo of the  
 83 MEDITERRANEAN sea (a cut of the mesh around the CRETE island is shown in Figure 2 at left panel)  
 84 and for the BBM–BBM system over a variable bottom in space and in time, we use a mesh generated  
 85 using an imported bathymetry  $f_{xy}$  for the sea near the JAVA island which can be downloaded from  
 86 the NOAA<sup>1</sup> website where in this case, the mesh generated is for the area, where the amplitude is  
 87 zero. We can smooth the bathymetric data obtained from NOAA (*cf.* Figure 3, left panel) by solving (8)  
 88 with  $f = f_{xy}$ . For all simulations with realistic bathymetry, we use  $\beta = 20$  in (8) to smooth the initial  
 89 bathymetry after the generation of the mesh (*cf.* Figure 3, right panel) in order to ensure the stability  
 90 of the numerical method, we also note that in order to be in a big deep water wave regime for the  
 91 BBM–BBM system we change the depth close to the shoreline to 100 m.

The bathymetry data downloaded from the NOAA website are in degree coordinate and we need to convert them to meters. 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  $lat_0 = .5(lat_{\text{South}} + lat_{\text{North}})$  and the Longitude  $long_0 = .5(long_{\text{West}} + long_{\text{East}})$ . 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

<sup>1</sup> <https://maps.ngdc.noaa.gov/viewers/wcs-client/>





**Figure 3.** 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).

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:

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

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

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

### 92 3.2. Initial data

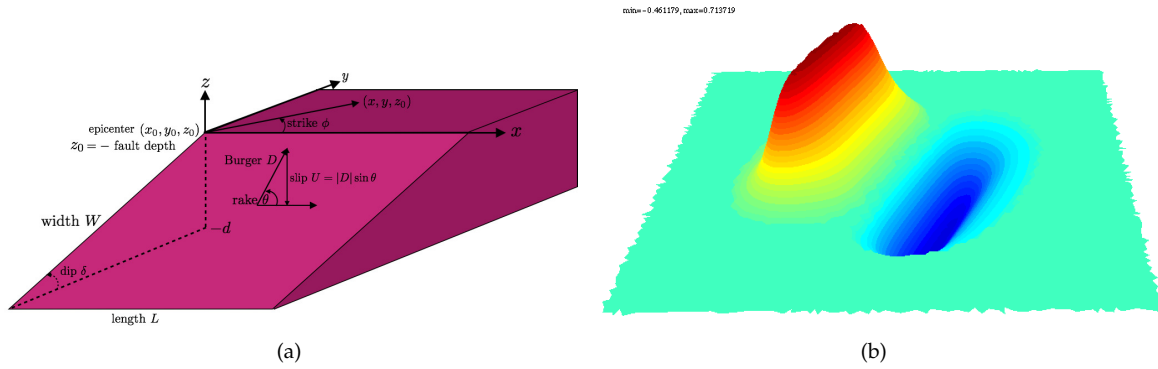
Tsunami waves considered in this study are generated by the co-seismic deformation of the Ocean's or sea's bottom due to an earthquake. The adopted modelling of the tsunami wave generation process is inspired by [8,11,16,17]. The co-seismic displacement is computed according to the celebrated OKADA's solution [18,19]. 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. [16,17]:

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

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

where

$$\begin{aligned} \xi &= (x - x_0) \cos \phi + (y - y_0) \sin \phi, \quad Y = -(x - x_0) \sin \phi + (y - y_0) \cos \phi, \\ p &= Y \cos \delta + d \sin \delta, \quad q = Y \sin \delta - d \cos \delta, \\ \tilde{y} &= \eta \cos \delta + q \sin \delta, \quad \tilde{d} = \eta \sin \delta - q \cos \delta, \\ R^2 &= \xi^2 + \eta^2 + q^2 = \xi^2 + \tilde{y}^2 + \tilde{d}^2, \quad X^2 = \xi^2 + q^2 \end{aligned}$$



**Figure 4.** Geometry of the source model (left) and the initial solution for  $\eta$  (right,  $\min = -0.46$  m,  $\max = 0.71$  m).

and

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

93 Here,  $W$  and  $L$  are the width and the length of the rectangular fault,  $(x, y)$  are the points where we  
 94 computes displacements,  $(x_0, y_0)$  is the epicenter,  $d = \text{fault depth}(x_0, y_0) + W \sin \delta$ ,  $\delta$  is the dip angle,  
 95  $\theta$  is the rake angle,  $D$  is the BURGERS'S vector,  $U = |D| \sin \theta$  is the slip on the fault,  $\phi$  is the strike angle  
 96 which is measured conventionally in the counter-clockwise direction from the North (cf. Figure 4 (left)),  
 97  $\mu$ ,  $\lambda$  are the LAMÉ constants derived from elastic-wave velocities:  $\lambda = \rho_c (V_P^2 - V_S^2)$  and  $\mu = \rho_c V_S^2$ ,  
 98 where  $\rho_c$  is the crust density,  $V_P$  is the compressional-wave ( $P$ -wave) velocity,  $V_S$  is the shear-wave  
 99 ( $S$ -wave) velocity. The Matlab script to compute the OKADA solution can be downloaded at the  
 100 following URL:

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

102 We shall distinguish here the two types of tsunami wave generation mechanisms [20,21]: *active*  
 103 and *passive* generation mechanisms.

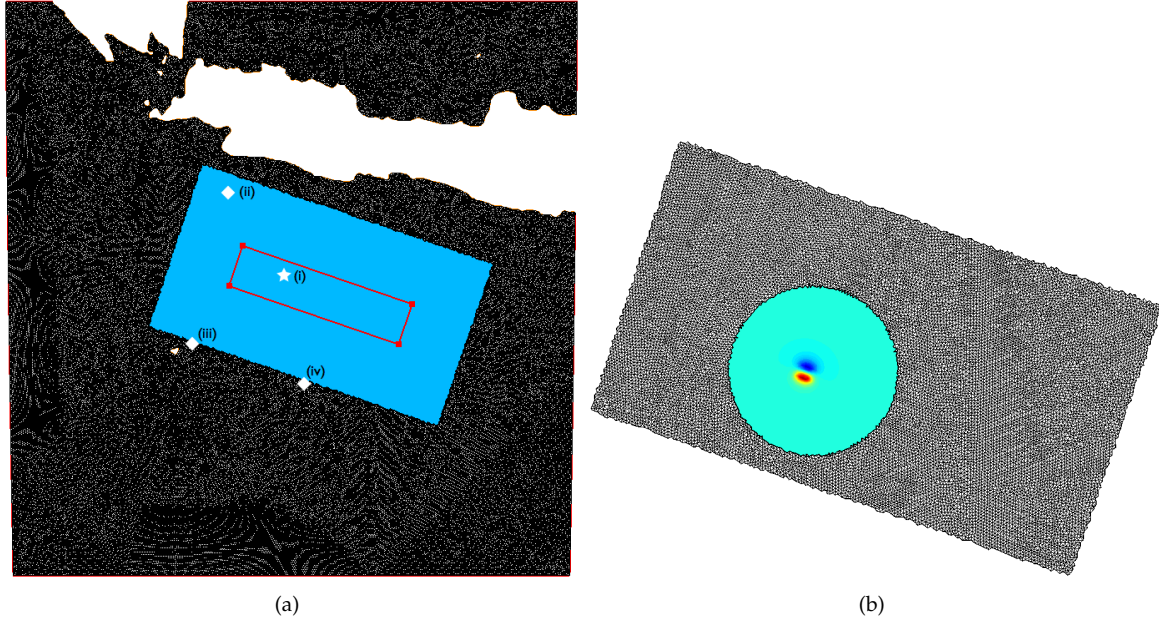
### 104 3.2.1. Passive generation

105 We remind that the passive generation approach consists in transposing the bottom deformation  
 106 on the free surface as an initial condition for tsunami propagation codes. In order to compute the initial  
 107 data for  $\eta(x, y, 0) = \mathcal{O}(x, y)$  in meters (cf. Figure 4 (right)),  $V(x, y, 0) = 0$  which is referred to as a *passive*  
 108 generation of a tsunami wave near the JAVA island, using our domain adaptive technique, we will use  
 109 the fact that the solution is concentrated in the small rectangle  $[x_0 - 3.2W; x_0 + 1.2W] \times [y_0 - L; y_0 + L]$   
 110 where  $L = 100$  km,  $W = 50$  km,  $\delta = 10.35^\circ$ ,  $\phi = 288.94^\circ$ ,  $\theta = 95^\circ$ ,  $U = 2$  m,  $\rho_c = 2700 \text{ kg/m}^3$ ,  $V_P = 6000$   
 111 m/s,  $V_S = 3400$  m/s,  $(x_0; y_0) = (107.345^\circ, -9.295^\circ)$  and the fault depth 10 km. All these geophysical  
 112 parameters can be downloaded from this file hosted by USGS:

113 [https://Earthquake.usgs.gov/archive/product/finite-fault/usp000ensm/us/1486510367579/  
 114 web/p000ensm.param](https://Earthquake.usgs.gov/archive/product/finite-fault/usp000ensm/us/1486510367579/web/p000ensm.param)

### 115 3.2.2. 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 [8,22]. In this case  
 we consider zero initial conditions for both the free surface elevation and the velocity field, and assume



**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).

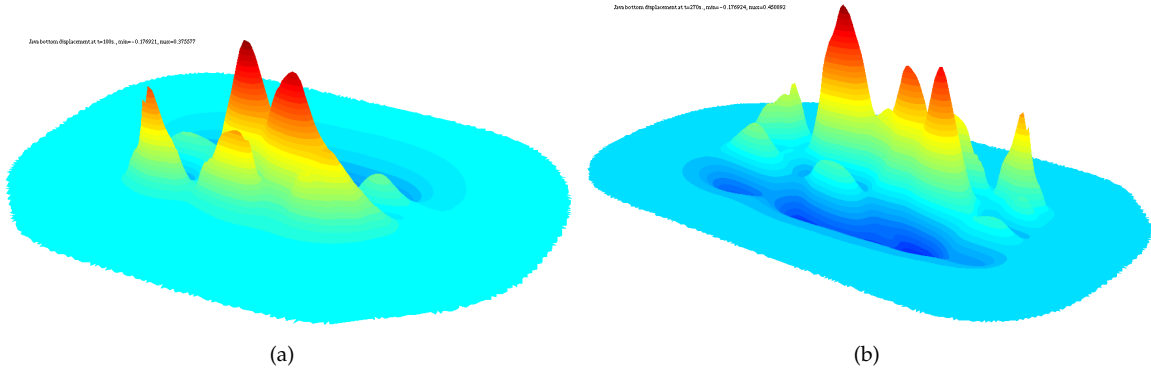
that the bottom is moving in time. This case may be described by considering the bottom motion formula:  $-z_b(x, y, t) = -h(x, y) - \zeta(x, y, t)$  with

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

116 where  $N_x$  sub-faults along strike and  $N_y$  sub-faults down the dip angle,  $\mathcal{H}(t)$  is the Heaviside step  
 117 function and  $\alpha = \log(3)/t_r$ , where  $t_r = 8$  s is the rise time. We choose here an exponential scenario,  
 118 but in practice, various scenarios could be used (instantaneous, linear, trigonometric, *etc.*) and could  
 119 be found in [8,16,17,22,23]. Parameters such as sub-fault location  $(x_i, y_i)$ , depth  $d_i$ , slip  $U$  and rake  
 120 angle  $\theta$  for each segment are given in [8, Table 3]. In this table, we notice that the fault's surface is  
 121 conventionally divided into  $N_x = 21$  sub-faults along strike and  $N_y = 7$  sub-faults down the dip angle,  
 122 leading to a total number of  $N_x \times N_y = 147$  equal segments.

123 For our special domain adaptivity technique, since the fault plane is considered to be the  
 124 rectangle with vertices located at  $(109.20508^\circ$  (Lon),  $-10.37387^\circ$  (Lat)),  $(106.50434^\circ$  (Lon),  $-9.45925^\circ$   
 125 (Lat)),  $(106.72382^\circ$  (Lon),  $-8.82807^\circ$  (Lat)) and  $(109.42455^\circ$  (Lon),  $-9.74269^\circ$  (Lat)), we will consider  
 126 that our bottom displacement is concentrated on the big rectangle which is equidistant of  $1^\circ$  from  
 127 each side of the initial fault plane as in Figure 5 (left panel), then we compute each OKADA solution  
 128  $\mathcal{O}_i$  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  
 129 solution will be interpolated on the big rectangle before starting to compute the vertical displacement  
 130 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  
 131 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  
 132 mesh each 3 iterations *i.e.* each 6 s by using the following value for the domain adaptation  $\text{uadapt} = \zeta$ ,  
 133  $\text{isoadapt}=5e-2$ ,  $\text{erradapt}=1e-4$ ,  $\beta=5e-9$ ,  $\text{epsadapt}=50e3$ .

134 We show in Figure 6, the bottom displacement  $\zeta(x, y, t)$  at time  $t = 100$  s and  $t = 270$  s using our  
 135 domain adaptation technique. We note that after building the OKADA solution  $\mathcal{O}(x, y)$  in the *passive*  
 136 generation or  $\mathcal{O}_i(x, y)$  in the active generation, we can remark that this solution is non-local and decays  
 137 slowly to zero, that is why in our domain adaptation technique we put 0 where the absolute value  
 138 of the solution is less than  $\min(|\min(\mathcal{O}_i(x, y))|, |\max(\mathcal{O}_i(x, y))|) < 9.2$  m. We make the same thing  
 139 without adaptive mesh in order to compare the solution using the same initial data.



**Figure 6.** 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).

#### 140 4. Numerical simulations

141 In this section, we study first the rate of convergence of our schemes for the BBM–BBM System (4)  
 142 with non-dimensional and unscaled variables *i.e.*, with  $g = 1$  over a variable bottom in space, which  
 143 establishes the adequacy of the chosen finite element discretization and the used time marching scheme,  
 144 for the flat bottom case, we refer to [24], where we use the same technique as in this paper. Then,  
 145 we simulate the propagation of a wave, that is similar to a real-world tsunami wave generated by  
 146 an earthquake, in the MEDITERRANEAN sea with the BBM–BBM model over a flat bottom. Then, we  
 147 switch to the JAVA island region with real variable bottom in space. Finally, we study the active tsunami  
 148 generation scenario which took place in 2006 near the JAVA island. In all numerical simulations we  
 149 used  $\mathbb{P}_1$  continuous piecewise linear functions for  $\eta$ ,  $u$ ,  $v$ ,  $h$  and  $\zeta$ .

##### 150 4.1. Rate of convergence

We present the evidence here, following the work done for the 1D case of the BBM–BBM system  
 in [25], that the second order RUNGE–KUTTA time scheme considered for the BBM–BBM 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 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  $\eta_h$ ,  $u_h$  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:

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

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

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

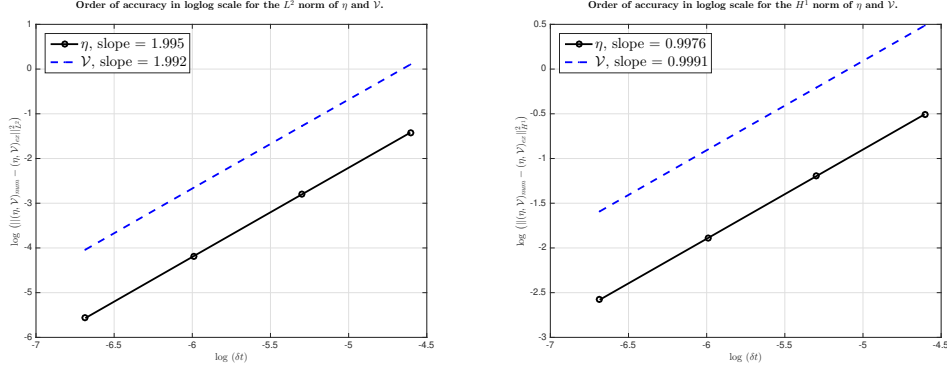
$$N_{L^2}(\eta) = \|\eta_h - \eta_{ex}\|_{L^2}, N_{H^1}(\eta) = \|\eta_h - \eta_{ex}\|_{H^1}, N_{L^2}(V) = \|u_h - u_{ex}\|_{L^2} + \|v_h - v_{ex}\|_{L^2}$$

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

151 and we end up with the results reported in Table 1. So, the  $L^2$  rates for  $\eta$  and  $V$  is of order  $\sim 2$  and the  
 152  $H^1$  rates for  $\eta$  and  $V$  is of order  $\sim 1$  as shown in the Figure 7 and which confirms the convergence of  
 153 the second-order RUNGE–KUTTA scheme in time for the BBM–BBM system with variable bottom in  
 154 space.

**Table 1.**  $L^2$  norm of the error for  $\eta$  and  $V$ .

N	$\delta t$	$N_{L^2}(\eta)$	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

**Figure 7.** Rate of convergence for sBBM system with variable bottom in space.

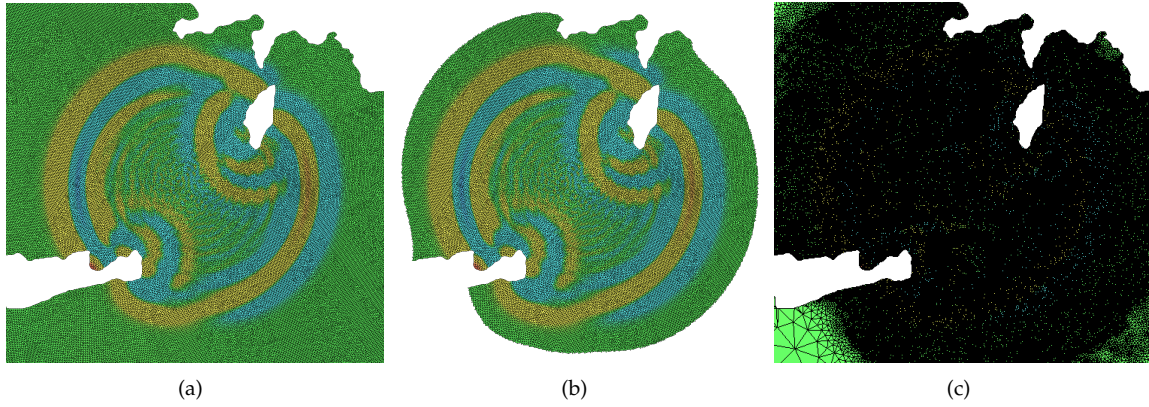
#### 155 4.2. Propagation of a tsunami wave in the Mediterranean sea with a flat bottom

156 We simulate here, the propagation of a wave that looks like a tsunami wave generated by an  
 157 earthquake in the MEDITERRANEAN sea with the BBM–BBM System (4) with a flat bottom  $-h(x, y) =$   
 158  $-1,5$  km which is the average depth of the MEDITERRANEAN sea. This wave was defined above in  
 159 the *passive* generation part of the Section 3 where, in this case, the initial solution is concentrated in  
 160 the small rectangle  $[x_0 - 5W; x_0 + 4W] \times [y_0 - 1.5L; y_0 + 2.5L]$  and we take these following values:  
 161  $L = 20$  Km,  $W = 10$  km,  $\delta = 7^\circ$ ,  $\phi = 0^\circ$ ,  $\theta = 90^\circ$ ,  $E = 9.5$  GPA is the YOUNG’S modulus,  $\nu = 0,27$   
 162 is the POISSON’S ratio,  $U = 2.5$  m,  $(x_0; y_0) = (2390. * scale, 590. * scale)$  and the fault depth 10 km.  
 163 In this example, we will take the fact that the LAMÉ constants  $\mu$  and  $\lambda$  are given by the formulas  
 164  $\mu = E/2(1 + \nu)$  and  $\lambda = E\nu/(1 + \nu)(1 - 2\nu)$ . We also use the following settings: for the step time  
 165  $\delta t = 1$  s, a reflective BC for all the boundary, for the adaptmesh of FreeFem++:

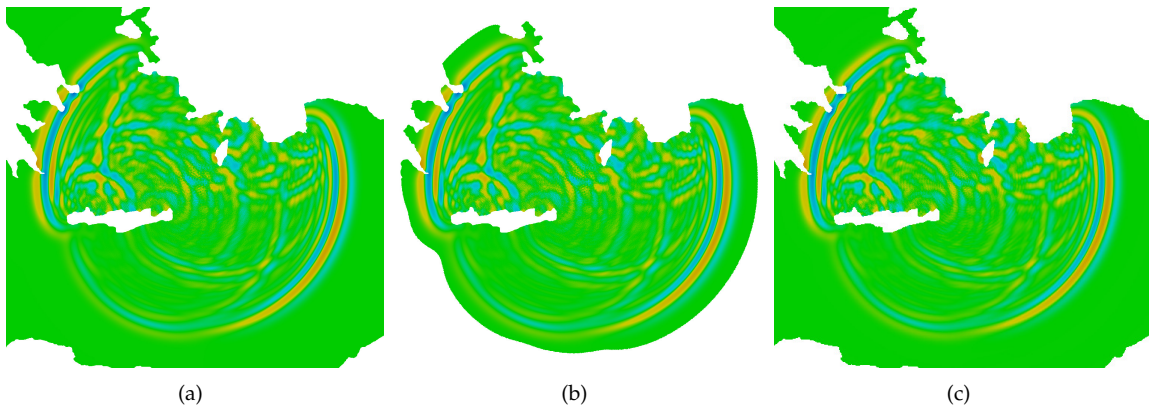
```
166 fespace Vhinit(Thinit, P0); Vhinit hT=hTriangle;
167 real Dx=hT[ ].min;
168 Th=adaptmesh(Th, uadapt, err=1.e-7, errg=1.e-2, hmin=Dx, iso=true, nbvx=1e8);
```

169 and for our domain adapt technique:  $isoadapt=5e-2$ ,  $erradapt=1e-7$ ,  $\beta=5e-3$ ,  $epsadapt=2e-2$ . We  
 170 note that, we adapt the mesh around the solution each 100 iterations *i.e.* each 10 s by using  $uadapt =$   
 171  $\eta + u + v$ .

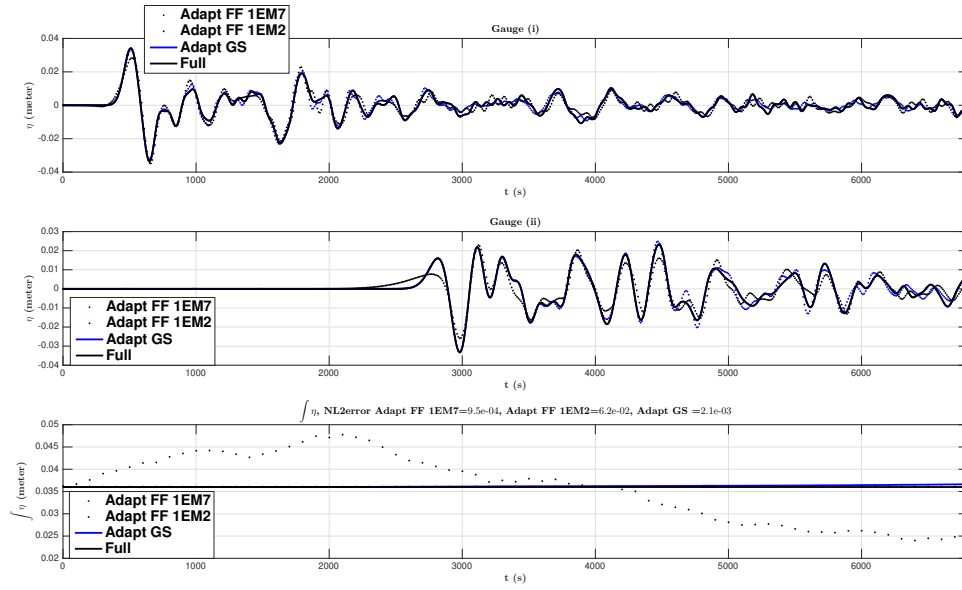
172 In order to compare the results between adaptmesh of FreeFem++, our new domain adaptation  
 173 technique and without using mesh adaptation, we plot in addition to the free surface elevation  $\eta$   
 174 in the Figures 8  $\rightarrow$  9, the variation of  $\eta$  vs time in Figure 10 at two wave ‘gauges’ placed at the positions  
 175 represented by  $\diamond$  in Figure 2 at right and the mass of the water  $\int \eta$ . Specifically, gauges were placed at  
 176 the points (i) :  $(2350. * scale, 550. * scale)$ , (ii) :  $(2104. * scale, 665. * scale)$ . In Figure 11, we represent  
 177 the comparison between the three methods: complete, domain adaptation and FreeFem++ internal  
 178 mesh adaptivity of the maximum of the propagation of the solution at time  $t = 6800$  s. We also plot  
 179 the computation time for each adapt mesh, the computation time of the simulation, the number of  
 180 degree of freedom in Figure 12. We can see in Figures 10 and 12 that the adaptmesh of FreeFem++ with  
 181  $err=1.e-2$  is the fastest method but unfortunately it does not preserve the mass invariant  $\int \eta$ . On  
 182 the other hand, our new domain adaptation technique preserves the mass invariant throughout the  
 183 simulation with an error of order  $2.1e - 3$  and an important time computation difference with the one



**Figure 8.** The mesh and 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 9.** 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  $err=1. e-7$  at the right panel (c) ( $min = -5.5e^{-2}$  m,  $max = 4.7e^{-2}$  m, for the three case).



**Figure 10.** Comparison between the three methods: full, *FreeFem++* adaptivity and the domain adaptation of the free surface elevations (in meters) vs time (in seconds), computed numerically at two wave gauges and of the mass conservation.

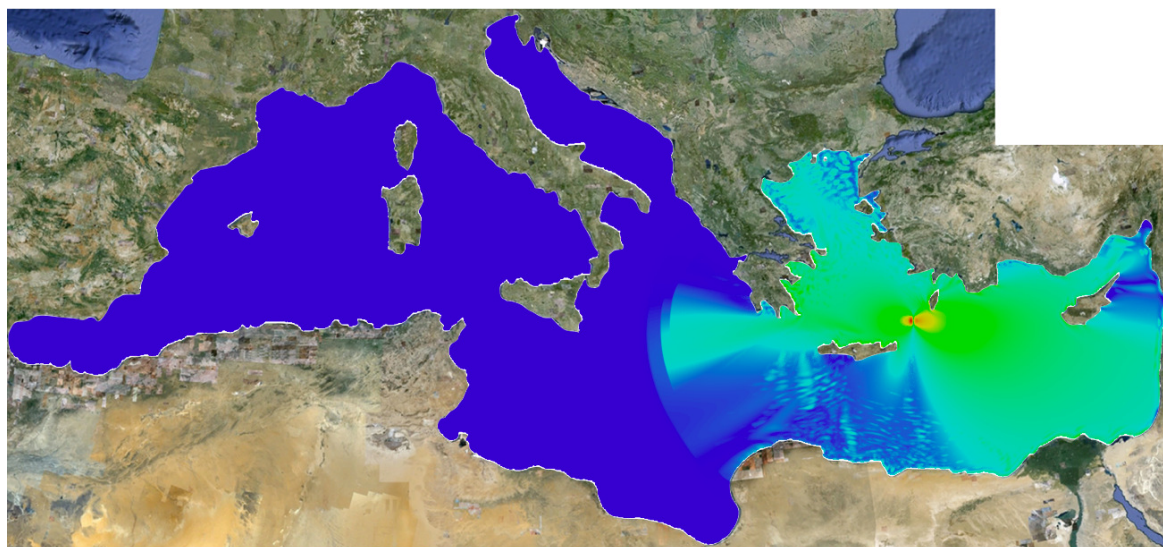
184 without mesh adaptation which is very promising method for the tsunami wave propagation. For the  
 185 adaptmesh of *FreeFem++* with  $\text{err}=1.e-7$  and  $\text{errg}=1.e-2$ , we also almost get the mass conservation  
 186 with an error of order  $9.5e-4$ , but we obtain some difference in wave gauge with the full method  
 187 which is due to the refinement mesh adaptation and the interpolation of the solution, although the  
 188 computation time is almost the double of the new domain adaptation technique. Thus, we can go  
 189 faster with our new domain adaptation technique if we can also deduce the mass matrix after cutting  
 190 the mesh, of course, if the mass matrix does not change along the simulation of the full mesh. This is  
 191 an outgoing project.

#### 192 4.3. Propagation of a tsunami wave near the Java island: passive generation

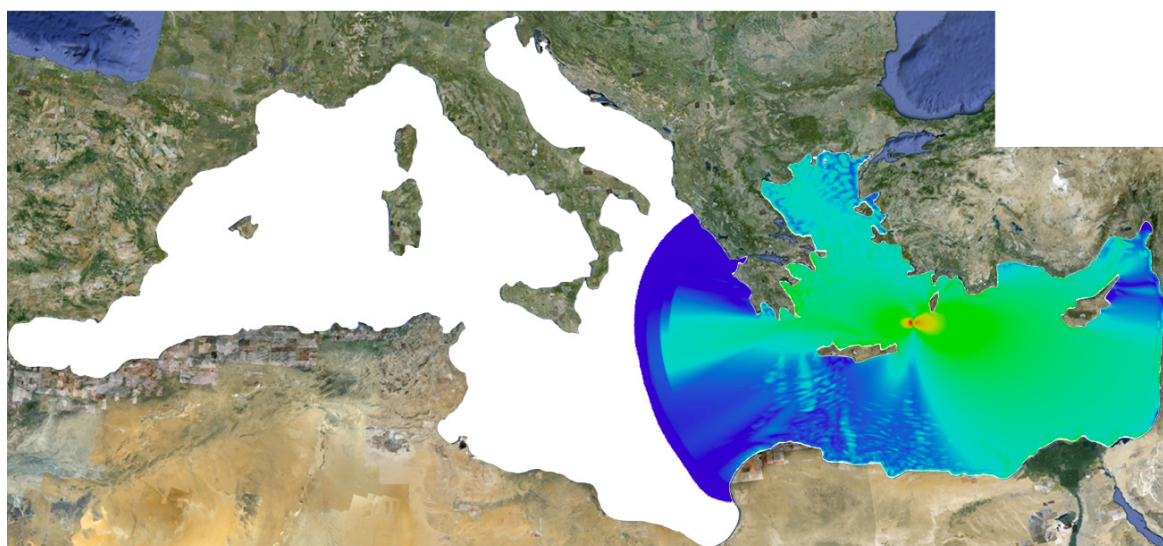
193 We will take here the same initial data as defined above in the passive generation part of Section 3,  
 194 we take  $\delta t = 1$  s as the time step size and we note that, we adapt the mesh after computing the  
 195 initial data for  $\eta$  and then every 50 s by using the following value for the domain adaptation  $\text{uadapt} =$   
 196  $\eta + u + v$ ,  $\text{isoadapt}=3e-2$ ,  $\text{erradapt}=1e-4$ ,  $\beta=5e-9$ ,  $\text{epsadapt}=30e3$ . We compare here the results  
 197 between our new domain adaptation technique and without using mesh adaptation. To this end, we  
 198 plot the free surface elevation  $\eta$  in the Figures 13 and 14, the variation of  $\eta$  vs time (in Figure 15) at four  
 199 numerical wave gauges placed at the following locations: (i)  $(107.345^\circ, -9.295^\circ)$ , (ii)  $(106.5^\circ, -8^\circ)$ , (iii)  
 200  $(105.9^\circ, -10.35^\circ)$  and (iv)  $(107.7^\circ, -11^\circ)$  (see Figure 5 (left)) where (i) is the position of the epicenter.  
 201 However, because of the large variations of the bottom, shorter waves were generated, especially  
 202 around CHRISTMAS Island (southwest of JAVA) and around the undersea canyon near the earthquake  
 203 epicenter.

Finally, we present a comparison of the kinetic, potential and total energies with the full mesh  
 (in Figure 16, top left panel) and with the domain adaptivity method (in Figure 16, top right panel)  
 defined in [26] as follows:

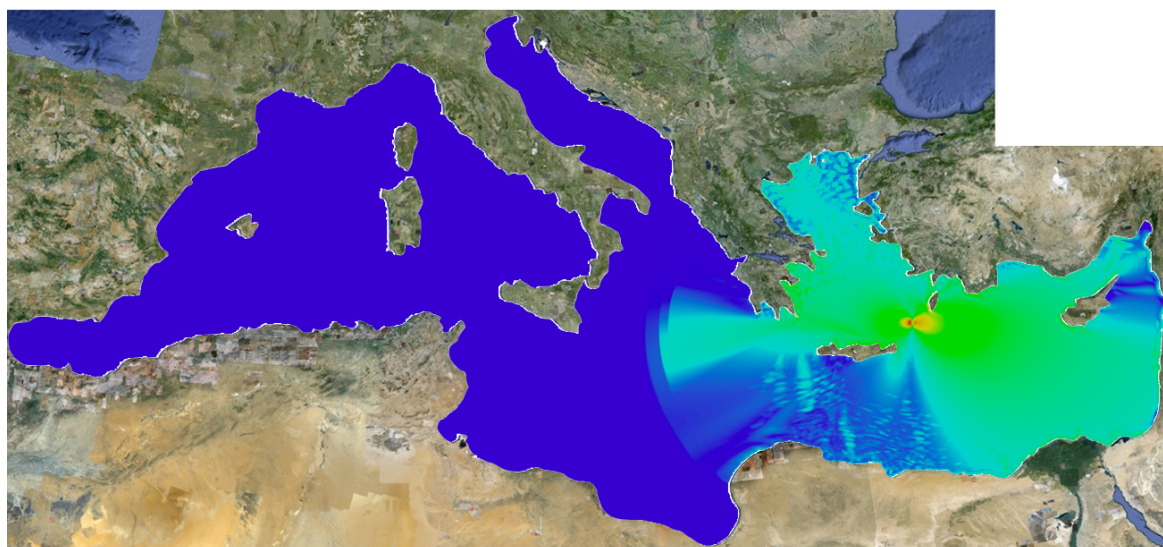
$$E_c = \frac{1}{2} \rho_w \int_{\Omega} \left( \int_{-h(x,y)}^{\eta} |V|^2 dz \right) dx dy, \quad E_p = \frac{1}{2} \rho_w \cdot g \int_{\Omega} \eta^2 dx dy, \quad (9)$$



(a)



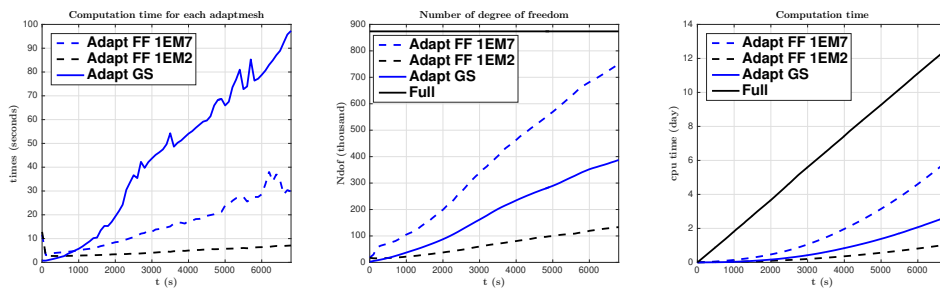
(b)



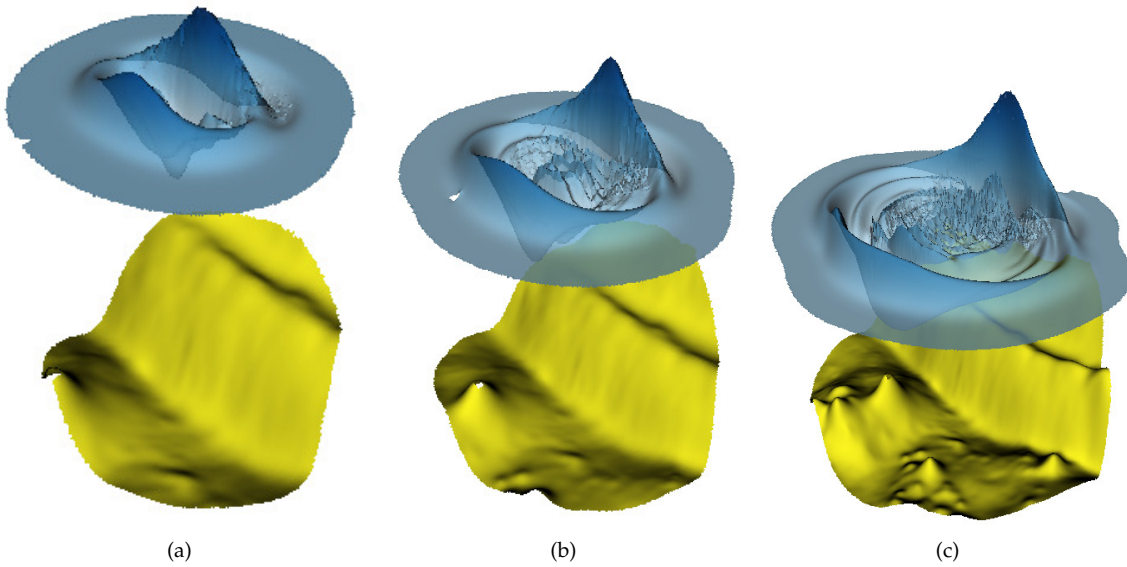
(c)

**Figure 11.** Comparison between the three method full (top panel), domain adaptation (middle panel) and *FreeFem++* adaptation (down panel) 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).





**Figure 12.** Comparison between the three methods: full, *FreeFem++* adaptation and domain adaptation of the computation time of each *adaptmesh*, the number of degree of freedom and the computation time of the simulation.

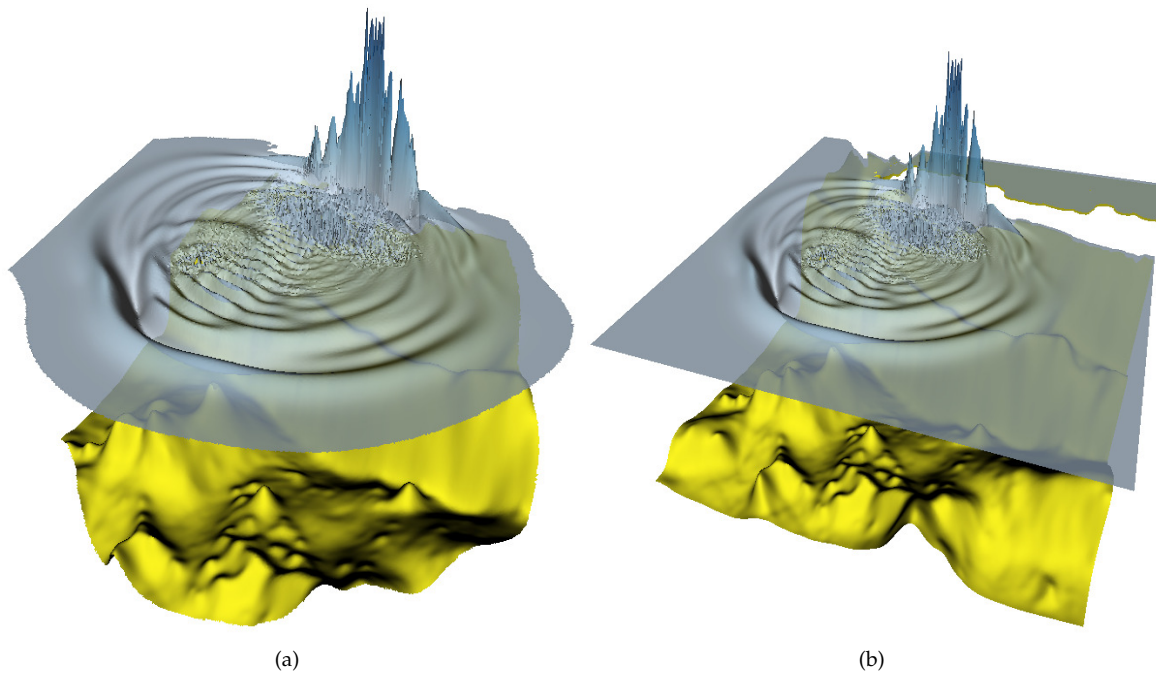


**Figure 13.** Passive generation: the bottom and the solution at  $t = 250$  s (left, solution ( $\min = -0.36$  m,  $\max = 0.38$  m), bottom ( $\min = -6207$  m,  $\max = -2096$  m)),  $t = 500$  s (center, solution ( $\min = -0.26$  m,  $\max = 0.35$  m), bottom ( $\min = -6207$  m,  $\max = -243$  m)) and  $t = 1000$  s (right, solution ( $\min = -0.21$  m,  $\max = 0.29$  m), bottom ( $\min = -6207$  m,  $\max = -100$  m)), with the domain adaptivity method.

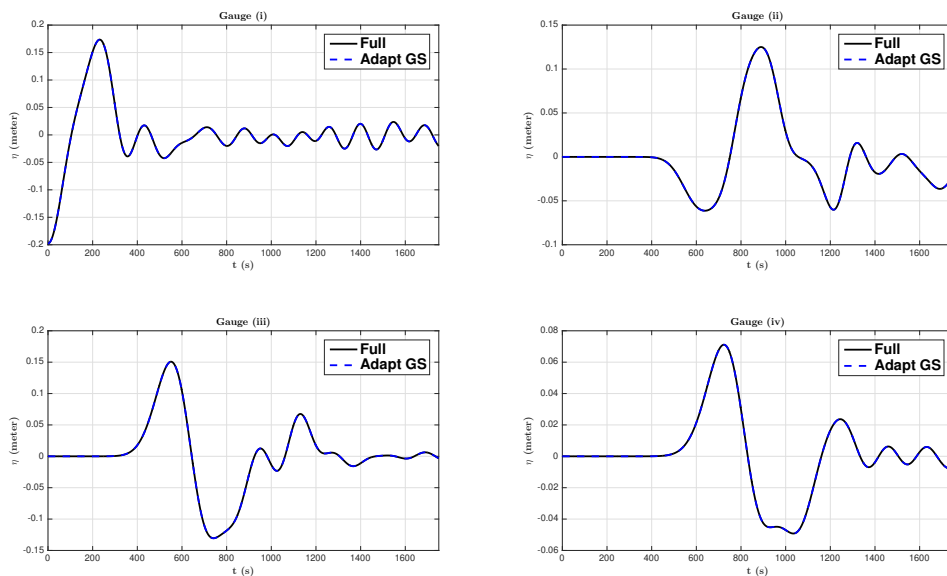
204 where  $\rho_w = 1027$  kg/m<sup>3</sup> is the ocean water density, the number of degrees of freedom (in Figure 16,  
 205 down left panel) and the computation time of the simulation (in Figure 16, down right panel). We  
 206 obtain here an error of order  $2.6e - 4$  between the total energy with domain adaptivity and without  
 207 any adaptation. We present in Figure 17 the comparison of the maximum of the propagation of the  
 208 solution between the full and domain adaptivity methods at  $t = 1750$  s.

#### 209 4.4. Propagation of a tsunami wave near the Java island: active generation

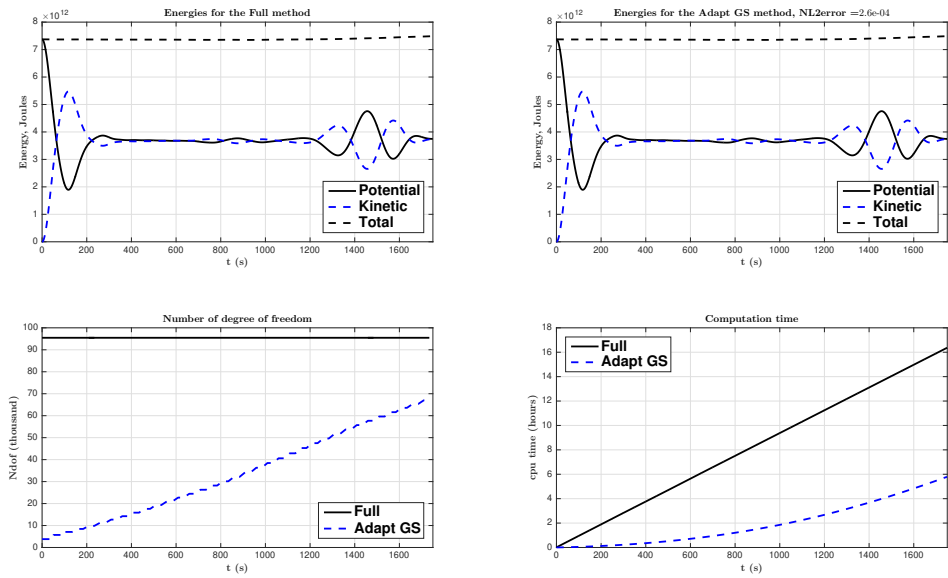
210 For a more realistic case as in the JAVA 2006 event, we use the active generation in order to model  
 211 the generation of a tsunami wave as in [8,22]. In this case we consider zero initial conditions for both  
 212 the surface elevation and the velocity field, we take  $\delta t = 2$  s as the time step size, we assume that the  
 213 bottom described in the Section 3 is moving in time and we note that we adapt the mesh, before the  
 214 end of the generation time  $t = 270$  s, every three iterations *i.e.* every 6 s by using the following value for  
 215 the domain adaptation  $u_{\text{adapt}} = \eta + u + v$ ,  $\text{isoadapt} = 5e-2$ ,  $\text{erradapt} = 1e-4$ ,  $\beta = 5e-9$ ,  $\text{epsadapt} = 50e3$   
 216 and then for  $t > 270$  s every 25 iterations *i.e.* each 50 s. We compare here only the results between  
 217 our new domain adaptation technique and without using mesh adaptation. To this end, we plot the  
 218 free surface elevation  $\eta$  in the Figures 18  $\rightarrow$  20. However, as in the passive case, because of the large  
 219 variations of the bottom, shorter waves were generated, especially around the CHRISTMAS Island



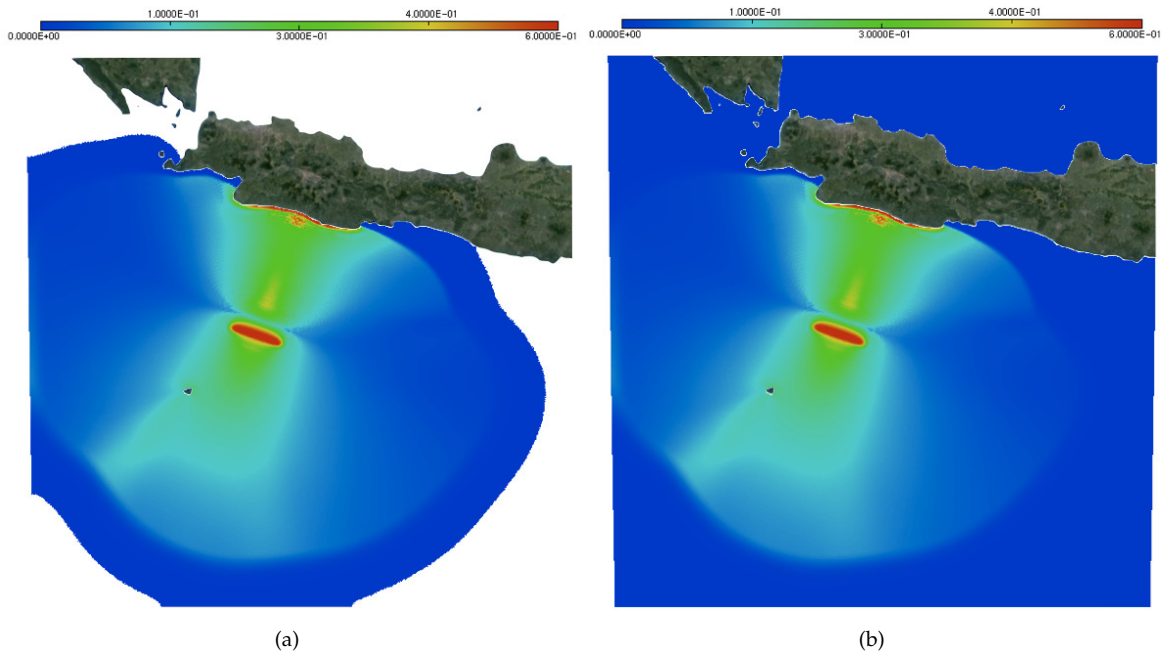
**Figure 14.** *Passive generation: comparison between the bottom and the solution at  $t = 1500$  s, with the domain adaptation method (left, solution (min =  $-0.38$  m, max =  $1.07$  m), bottom (min =  $-6207$  m, max =  $-100$  m)) and with the full one (right, solution (min =  $-0.38$  m, max =  $1.07$  m), bottom (min =  $-6207$  m, max =  $-100$  m)).*



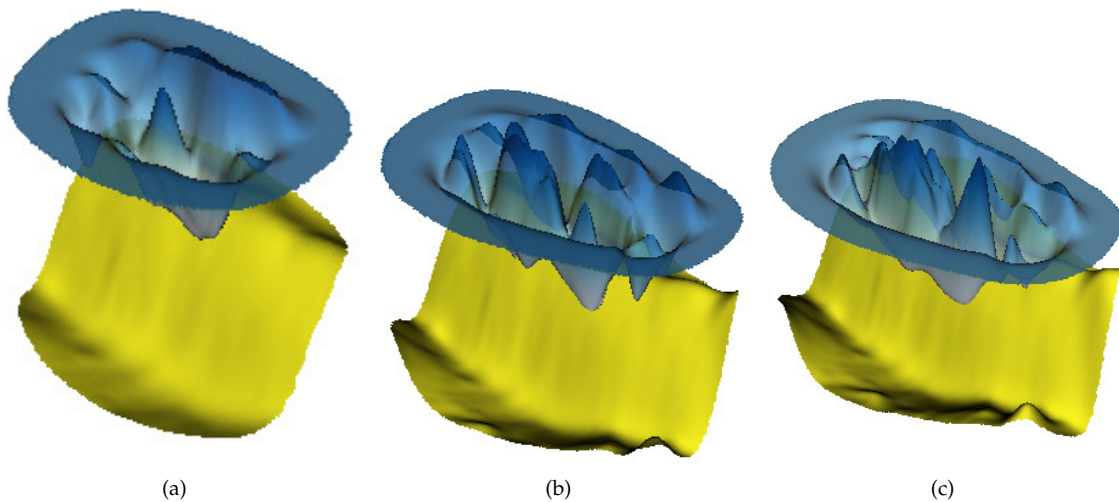
**Figure 15.** *Passive generation: comparison between the two methods the full one and domain adaptivity of the free surface elevations (in meters) vs time (in seconds), computed numerically at four wave gauges where the gauge (i) corresponds to the epicenter.*



**Figure 16.** Passive generation: comparison between the two methods the full one and domain adaptivity of the kinetic, potential and total energies, the number of degree of freedom and the computation time of the simulation.



**Figure 17.** Passive generation: comparison between the maximum of the solution at  $t = 1750$  s, with the domain adaptivity method (left panel) and with the full one (right panel).



**Figure 18.** Active generation: the bottom and the solution at  $t = 100$  s (left, solution ( $\min = -0.17$  m,  $\max = 0.07$  m), bottom ( $\min = -6207$  m,  $\max = -2589$  m)),  $t = 200$  s (center, solution ( $\min = -0.19$  m,  $\max = 0.08$  m), bottom ( $\min = -6207$  m,  $\max = -2285$  m)) and  $t = 270$  s (right, solution ( $\min = -0.14$  m,  $\max = 0.10$  m), bottom ( $\min = -6207$  m,  $\max = -2084$  m)), with the domain adaptivity method.

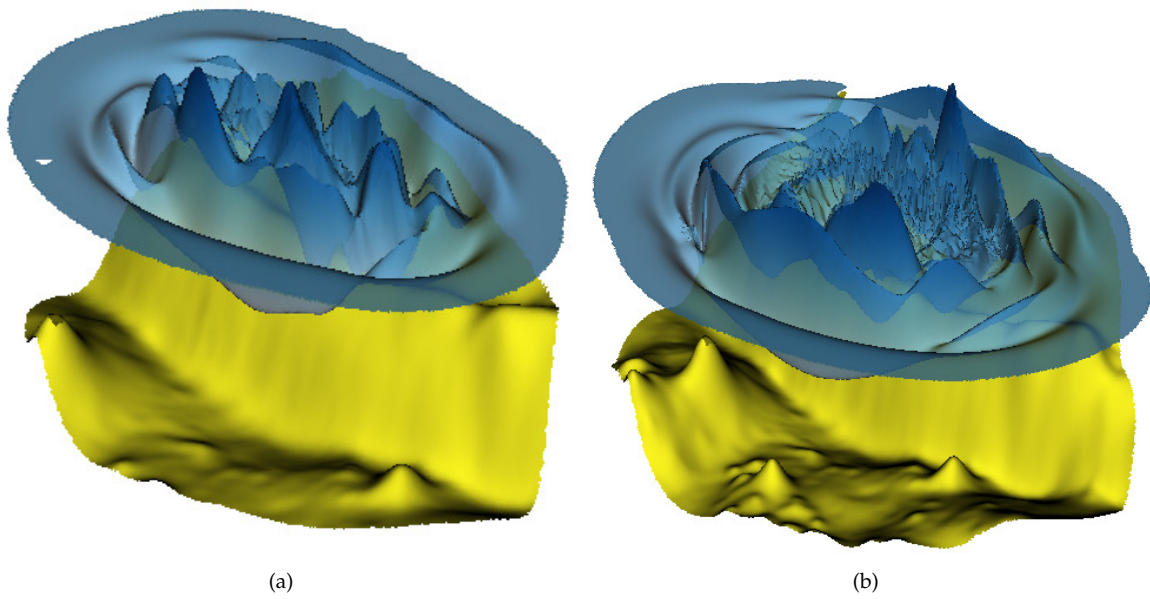
220 (southwest of JAVA island) and around the undersea canyon near the earthquake epicenter. We plot the  
 221 variation of  $\eta$  vs time (in Figure 21) at four numerical wave gauges placed at the following locations:  
 222 (i) ( $107.345^\circ, -9.295^\circ$ ), (ii) ( $106.5^\circ, -8^\circ$ ), (iii) ( $105.9^\circ, -10.35^\circ$ ) and (iv) ( $107.7^\circ, -11^\circ$ ) (see Figure 5  
 223 (left panel)) where (i) is the position of the epicenter. Finally, we present a comparison of the kinetic,  
 224 potential and total energies with the full mesh (in Figure 22, top left panel) and with the domain  
 225 adaptivity method (in Figure 22, top right) defined in (9), the number of the degrees of freedom (in  
 226 Figure 22, lower left panel) and the computation time of the simulation (in Figure 22, lower right  
 227 panel). We obtain here an error of order  $2e - 5$  between the total energy with domain adaptivity and  
 228 without any adaptation. We present in Figure 23 the comparison of the maximum of the propagation  
 229 of the solution between the full and domain adaptivity method at  $t = 1750$  s.

## 230 5. Conclusion and Outlook

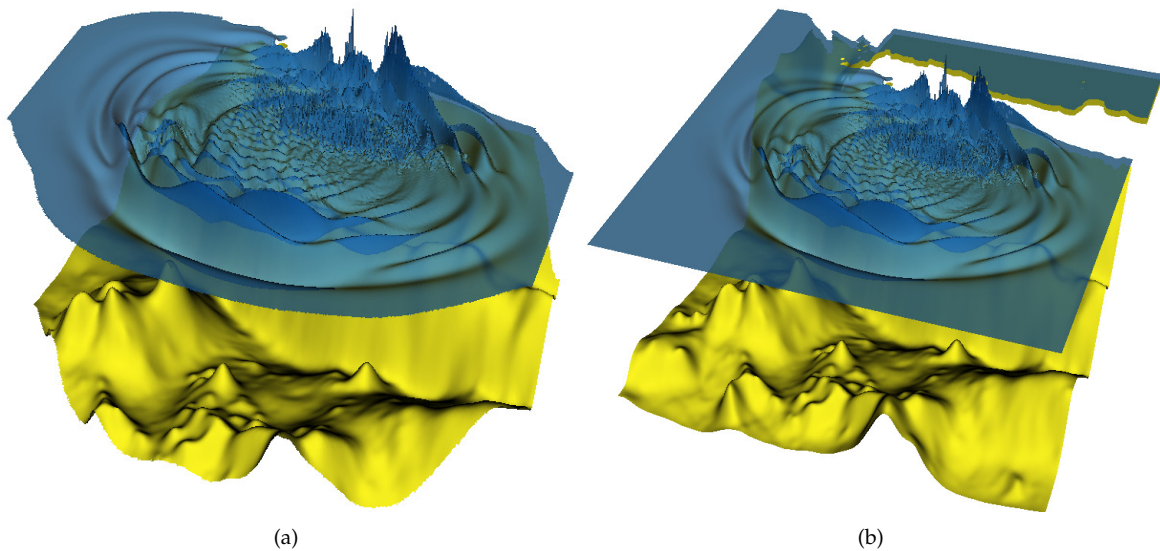
231 In this manuscript we demonstrated how to discretize the BBM–BBM system (1) using the FEM  
 232 and dedicated open-source software FreeFem++. The use of this numerical technique was demonstrated  
 233 in view of applications to tsunami wave modelling [17,27]. The concrete cases of wave propagation  
 234 in the MEDITERRANEAN sea and in JAVA island region (INDONESIA) were considered. The digital  
 235 computing environment that we developed allows the integration of realistic data (bathymetry and  
 236 geography) in a relatively simple software framework. The codes used in this study are made freely  
 237 available for all our readers. Moreover, a novel mesh and domain adaptation technique was proposed  
 238 to speed-up substantially the computations. The gain in terms of the CPU time after applying this  
 239 technique can be clearly seen in Figure 22. The accuracy of the ‘accelerated’ solution is more than  
 240 acceptable to make this technique useful in a variety of tsunami propagation problems. It goes without  
 241 saying that this technique can be applied to other events and other regions of the world with minimal  
 242 changes in the provided codes.

243 Regarding the perspectives of this study, we would like to develop also the parallel version of  
 244 this code together with the domain adaptation technique to make computations practically *faster* than  
 245 the real time tsunami wave propagation. However, we underline that even the current version can be  
 246 efficiently run even on a modest laptop personal computer.

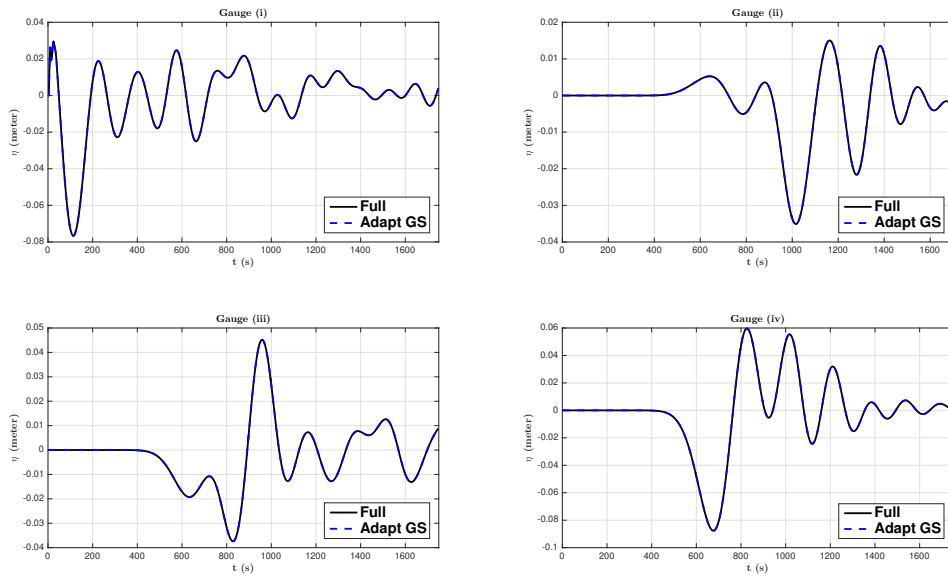
247 **Author Contributions:** Conceptualization, G.S. and D.D.; methodology, G.S.; software, G.S.; validation, G.S. and  
 248 D.D.; investigation, G.S.; writing–original draft preparation, G.S.; writing–review and editing, D.D.; visualization,  
 249 G.S.; supervision, D.D. All authors have read and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



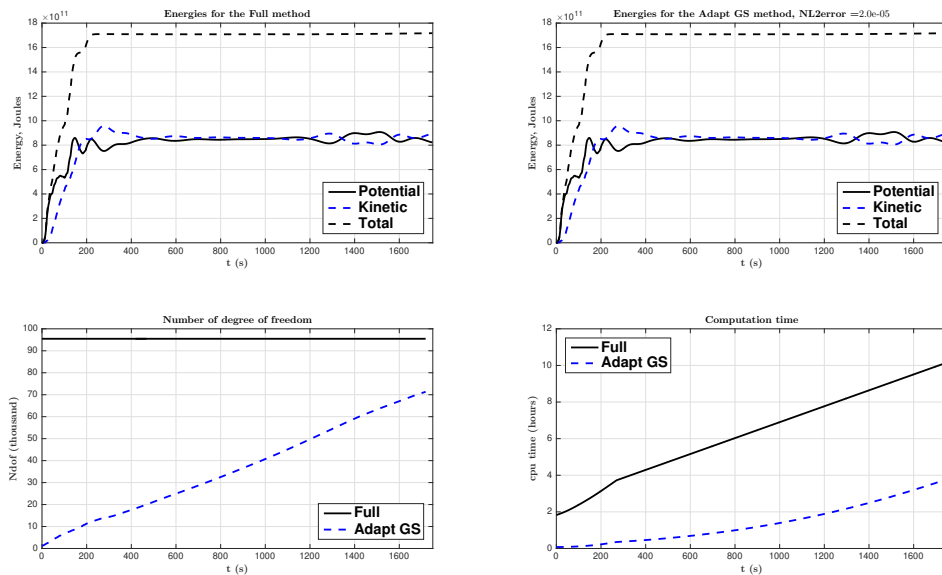
**Figure 19.** Active generation: the bottom and the solution at  $t = 500$  s (left, solution ( $\min = -0.15$  m,  $\max = 0.10$  m), bottom ( $\min = -6207$  m,  $\max = -260$  m)) and  $t = 1000$  s (right, solution ( $\min = -0.14$  m,  $\max = 0.09$  m), bottom ( $\min = -6207$  m,  $\max = -100$  m)), with the domain adaptivity method.



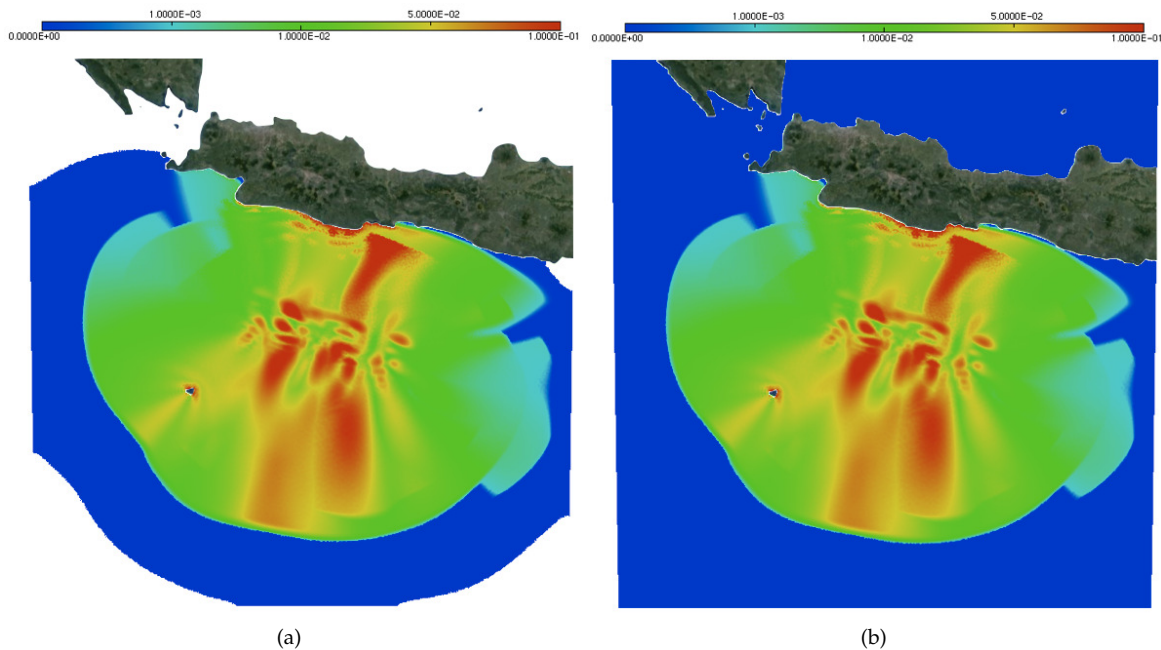
**Figure 20.** Active generation: comparison between the bottom and the solution at  $t = 1500$  s, with the domain adaptivity method (left, solution ( $\min = -0.29$  m,  $\max = 0.13$  m), bottom ( $\min = -6207$  m,  $\max = -100$  m)) and with the full one (right, solution ( $\min = -0.29$  m,  $\max = 0.13$  m), bottom ( $\min = -6207$  m,  $\max = -100$  m)).



**Figure 21.** 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 22.** 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.



**Figure 23.** Active generation: comparison between the maximum of the solution at  $t = 1750$  s, with the domain adaptivity method (left panel) and with the full one (right panel).

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

252 **Acknowledgments:** This work would not be possible without a precious help of Professors Frédéric HECHT,  
 253 Dimitrios MITSOTAKIS, and Olivier PANTZ.

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

## 255 Abbreviations

256 The following abbreviations are used in this manuscript:

257	2D	Two-dimensional
	BBM	BENJAMIN-BONA-MAHONY
	BC	Boundary Condition
	FEM	Finite Element Method
258	FreeFem++	Free Finite Element Method
	MOST	Method Of Splitting Tsunami
	NOAA	National Oceanic and Atmospheric Administration
	PDE	Partial Differential Equation
	USGS	United States Geological Survey

## 259 References

- 260 1. Imamura, F. Simulation of wave-packet propagation along sloping beach by TUNAMI-code. In *Long-wave*  
 261 *Runup Models*; Yeh, H.; Liu, P.L.F.; Synolakis, C.E., Eds.; World Scientific: Singapore, 1996; pp. 231–241.  
 262 pages 1
- 263 2. Titov, V.V.; González, F.I. Implementation and testing of the method of splitting tsunami (MOST) model.  
 264 Technical Report ERL PMEL-112, Pacific Marine Environmental Laboratory, NOAA, 1997. pages
- 265 3. Dutykh, D.; Poncet, R.; Dias, F. The VOLNA code for the numerical modeling of tsunami waves:  
 266 Generation, propagation and inundation. *Eur. J. Mech. B/Fluids* **2011**, *30*, 598–615, [1002.4553].  
 267 doi:10.1016/j.euromechflu.2011.05.005. pages

- 268 4. Khakimzyanov, G.S.; Dutykh, D.; Mitsotakis, D.; Shokina, N.Y. Numerical simulation of conservation  
269 laws with moving grid nodes: Application to tsunami wave modelling. *Geosciences* **2019**, *9*, 197.  
270 doi:10.3390/geosciences9050197. pages 1
- 271 5. Boussinesq, J.V. Théorie générale des mouvements qui sont propagés dans un canal rectangulaire horizontal.  
272 *C. R. Acad. Sc. Paris* **1871**, *73*, 256–260. pages 1
- 273 6. Wu, T.Y. Long Waves in Ocean and Coastal Waters. *Journal of Engineering Mechanics* **1981**, *107*, 501–522.  
274 pages 1
- 275 7. Dias, F.; Dutykh, D.; O'Brien, L.; Renzi, E.; Stefanakis, T. On the Modelling of Tsunami Generation and  
276 Tsunami Inundation. *Procedia IUTAM* **2014**, *10*, 338–355. doi:10.1016/j.piutam.2014.01.029. pages
- 277 8. Dutykh, D.; Mitsotakis, D.; Gardeil, X.; Dias, F. On the use of the finite fault solution for tsunami generation  
278 problems. *Theor. Comput. Fluid Dyn.* **2013**, *27*, 177–199. doi:10.1007/s00162-011-0252-8. pages 8, 9, 10, 16
- 279 9. Guesmia, M.; Heinrich, P.H.; Mariotti, C. Numerical simulation of the 1969 Portuguese tsunami by a finite  
280 element method. *Natural Hazards* **1998**, *17*, 31–46. pages
- 281 10. Lynett, P.J.; Borrero, J.C.; Liu, P.L.F.; Synolakis, C.E. Field Survey and Numerical Simulations: A  
282 Review of the 1998 Papua New Guinea Tsunami. *Pure Appl. Geophys.* **2003**, *160*, 2119–2146.  
283 doi:10.1007/s00024-003-2422-0. pages 1
- 284 11. Mitsotakis, D.E. Boussinesq systems in two space dimensions over a variable bottom for the generation  
285 and propagation of tsunami waves. *Math. Comp. Simul.* **2009**, *80*, 860–873. pages 1, 5, 8
- 286 12. Whitham, G.B. *Linear and Nonlinear Waves*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1999; p. 656.  
287 doi:10.1002/9781118032954. pages 2
- 288 13. Stoker, J.J. *Water Waves: The Mathematical Theory with Applications*; John Wiley and Sons, Inc.: Hoboken, NJ,  
289 USA, 1992; p. 600. doi:10.1002/9781118033159. pages 2
- 290 14. Hecht, F.; Auliac, S.; Pironneau, O.; Morice, J.; Le Hyaric, A.; Ohtsuka, K.; Jolivet, P. *FreeFem++*, 3 ed.;  
291 Laboratoire Jacques-Louis Lions, Université Pierre et Marie Curie: Paris, 2018; p. 424. pages 2
- 292 15. Løvholt, F.; Pedersen, G. Instabilities of Boussinesq models in non-uniform depth. *Int. J. Num. Meth. Fluids*  
293 **2009**, *61*, 606–637. doi:10.1002/flid.1968. pages 5
- 294 16. Dutykh, D.; Dias, F. Water waves generated by a moving bottom. In *Tsunami and Nonlinear Waves*; Kundu,  
295 A., Ed.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2007; pp. 65–95. doi:10.1007/978-3-540-71256-5\_4.  
296 pages 8, 10
- 297 17. Dutykh, D. Mathematical modelling of tsunami waves. Phd thesis, École Normale Supérieure de Cachan,  
298 2007. pages 8, 10, 19
- 299 18. Okada, Y. Surface deformation due to shear and tensile faults in a half-space. *Bull. Seism. Soc. Am.* **1985**,  
300 *75*, 1135–1154. pages 8
- 301 19. Okada, Y. Internal deformation due to shear and tensile faults in a half-space. *Bull. Seism. Soc. Am.* **1992**,  
302 *82*, 1018–1040. pages 8
- 303 20. Kervella, Y.; Dutykh, D.; Dias, F. Comparison between three-dimensional linear and nonlinear tsunami  
304 generation models. *Theor. Comput. Fluid Dyn.* **2007**, *21*, 245–269. doi:10.1007/s00162-007-0047-0. pages 9
- 305 21. Dutykh, D.; Dias, F.; Kervella, Y. Linear theory of wave generation by a moving bottom. *Comptes Rendus*  
306 *Mathématique* **2006**, *343*, 499–504. doi:10.1016/j.crma.2006.09.016. pages 9
- 307 22. Dutykh, D.; Mitsotakis, D.; Chubarov, L.B.; Shokin, Y.I. On the contribution of the horizontal  
308 sea-bed displacements into the tsunami generation process. *Ocean Modelling* **2012**, *56*, 43–56.  
309 doi:10.1016/j.ocemod.2012.07.002. pages 9, 10, 16
- 310 23. Hammack, J. A note on tsunamis: their generation and propagation in an ocean of uniform depth. *J. Fluid*  
311 *Mech.* **1973**, *60*, 769–799. pages 10
- 312 24. Sadaka, G. Solution of 2D Boussinesq systems with FreeFem++: the flat bottom case. *Journal of Numerical*  
313 *Mathematics* **2012**, *20*, 303–324. doi:10.1515/jnum-2012-0016. pages 11
- 314 25. Senthilkumar, A. On the influence of wave reflection on shoaling and breaking solitary Waves. *Proceedings*  
315 *of the Estonian Academy of Sciences* **2016**, *65*, 414–430. doi:10.3176/proc.2016.4.06. pages 11
- 316 26. Dutykh, D.; Dias, F. Energy of tsunami waves generated by bottom motion. *Proc. R. Soc. Lond. A* **2009**,  
317 *465*, 725–744. doi:10.1098/rspa.2008.0332. pages 14
- 318 27. Sadaka, G. Etude mathématique et numérique d'équations d'ondes aquatiques amorties. Phd thesis,  
319 Université de Picardie Jules Verne, 2011. pages 19



320 © 2020 by the authors. Submitted to *Geosciences* for possible open access publication  
321 under the terms and conditions of the Creative Commons Attribution (CC BY) license  
322 (<http://creativecommons.org/licenses/by/4.0/>).