REPOSITORIO UNIVERSIDADE CORUÑA

UNIVERSIDADE DA CORUÑA

This is an ACCEPTED VERSION of the following published document:

A. M. Ferreiro-Ferreiro, J. A. García-Rodríguez, J. G. López-Salas, C. Escalante, y M. J. Castro, «Global optimization for data assimilation in landslide tsunami models», Journal of Computational Physics, vol. 403, p. 109069, feb. 2020, doi: 10.1016/j.jcp.2019.109069.

Link to published version: https://doi.org/10.1016/j.jcp.2019.109069

General rights:

© 2020. This manuscript version is made available under the CC-BY-NC-ND 4.0 license https://creativecommons.org/licenses/by-nc-nd/4.0/.
This version of the article has been accepted for publication in Journal of Computational Physics (1090-2716). The Version of Record is available online at 10.1016/j.jcp.2019.109069.

Global optimization for data assimilation in landslide tsunamis models

A.M. Ferreiro-Ferreiro[a], J.A. García-Rodríguez[a,*], J.G. López-Salas[a], C. Escalante[b], M.J. Castro[b]

[a] *Department of Mathematics, Faculty of Informatics, Campus Elviña s/n, 15071-A Coruña, Spain*
[b] *Department of Análisis Matemático, University of Málaga*

Abstract

The goal of this article is to make automatic data assimilation for a landslide tsunami model, given by the coupling between a non-hydrostatic multi-layer shallow-water and a Savage-Hutter granular landslide model for submarine avalanches. The coupled model is discretized using a positivity preserving second-order path-conservative finite volume scheme. The data assimilation problem is posed in a global optimization framework and we develop and compare parallel metaheuristic stochastic global optimization algorithms, more precisely multi-path versions of the Simulated Annealing algorithm, with hybrid global optimization algorithms based on hybridizing Simulated Annealing with gradient local searchers, like L-BFGS-B.

Keywords:
Landslide tsunamis, non-hydrostatic multi-layer shallow-water model, finite volume method, global optimization, Simulated Annealing, hybrid optimization algorithms, Basin Hopping, multi-path, L-BFGS-B, parallel, multi-CPU.

1. Introduction

The goal of this work is twofold. On the one hand, assessing the feasibility of performing data assimilation for models of tsunamis generated by submarine landslides (also known as submarine mass failures, SMF), when only information/data of the fluid free surface is available: that is, checking whether the data assimilation problem is well posed, i.e. the identifiability of the model parameters. On the other hand, if the former is possible, we also aim at developing a generic data assimilation framework/machinery based on parallel and efficient global optimization algorithms which can deal with landslide tsunami models.

*Corresponding author
Email address: jose.garcia.rodriguez@udc.es (J.A. García-Rodríguez)

The tsunami hazard modeling is of great importance to prevent and forecast the consequences of such events, as they can cause a large number of casualties and huge financial losses. Tsunamis can be generated mainly by earthquakes, storm surges or landslides (subaerial or submarine). The majority of them are caused by an offshore earthquake that pushes the ocean up or down. Nevertheless tsunamis can also be generated in other ways. Underwater landslides, which might accompany an earthquake or occur independently, are a classic example. Traditional warning systems completely miss tsunamis from those types of sources. Once we have a model for these phenomena, the correct calibration of the parameters is of key importance for the accurate simulation of the tsunami. This calibration could be even done in real time, feeding the model with the measures given by the tide-gauges in the ocean, in the first moments of the tsunami. After the calibration, the data can be used to rerun the model and predict the trajectory of the tsunami and the impact areas.

Several types of models can be found in the literature for modeling landslide tsunamis. Their development focuses in three aspects: a physical model for landslide material, a hydrodynamic model that simulates the generation and propagation of resulting waves, and the coupling between both. The hydrodynamics of landslide-induced tsunamis has been extensively studied using numerical models based on different levels of simplification.

The simplest model contemplates the landslide as a rigid solid with fixed landslide shape (see for example [1]). Another approach to simulate landslide-induced tsunamis is to consider both the landslide and the water as two different fluids (see [2, 3, 4, 5, 6, 7]). This approach allows the landslide to deform, and to couple the landslide and the fluid. Although the two-fluid models described above can be reasonably successful in predicting tsunami wave generation, they may fail to determine the landslide motion from initiation to deposition.

Initial steps towards development of granular flow-based models for landslide behavior have usually been based on depth-integrated models pioneered by Iverson (1997, see [8]), Savage and Hutter (1989, see [9]), and others. These models were initially developed for application to shallow subaerial debris flows. In [10] a two-layer Savage-Hutter type model was proposed to simulate submarine landslides, where the hydrostatic pressure assumption is assumed to derive the model.

In [11] a two-phase model for granular landslide motion and tsunami wave generation is developed. The granular phase is modeled by a standard Savage-Hutter type model governed by Coulomb friction and the tsunami wave generation is simulated using a three-dimensional non-hydrostatic wave model, which is capable of capturing wave dispersion efficiently using a small number of discretized vertical layers.

Here, we follow a similar approach, that is, we consider a two-phase model, however we will replace the three-dimensional non-hydrostatic model by the multi-layer non-hydrostatic model recently proposed in [12]. We briefly describe this model in Section 2.

The previous model depends on a set of parameters that need to be calibrated in order to match real data. Note that, having a good model and a

strong and reliable numerical method for solving the problem, is as important as performing a good parameters adjustment of the model according to physical measures. In other words, a good model, together with a good numerical method, can lead to totally wrong results with poorly calibrated parameters. Data assimilation is the tool for embedding reality in numerical simulation. Together with mathematical modeling and development of the proper numerical methods, it could be considered as the third leg supporting the numerical simulation of processes in science and engineering, allowing the model to learn and profit from real measured data, see the pioneering work of J. Lions about the mathematical basis of data assimilation and control, [13]. Data assimilation is of key importance, for example, in atmospheric models for weather forecasting, see [14].

Our work follows the classical approach to calibrate the parameters of a model, i.e. the parameters are adjusted in such a way that the behaviour of the model approximates, as closely and consistently as possible, the observed response of a hydrologic system over some historical period of time. Ultimately, the best parameters are those minimizing the simple least square objective function of the residuals, which accounts for the differences between the model-simulated output and the measured data. This is the right approach as long as the mathematical model is correct (realistic enough), and physical data are measured without error. The uncertainty in the model prediction will be due to the uncertainty in the parameter estimates.

There is a separate line of research [15] arguing that models have structural errors arising from the aggregation of spatially distributed real-world processes into mathematical models. Besides, due to this aggregation process, model parameters usually do not represent directly measurable entities and must therefore be estimated using measurements of the system inputs and outputs, thus adding another source of errors. As a consequence, during the calibration process one should also take also into account input, output and model structural errors. Several methods were firstly proposed to deal with model structural and data errors, like the Bayesian approach, Recursive Parameter Estimation algorithms, multiobjective calibration or stochastic input error models. Bayesian methods treats model parameters as probabilistic variables, in contrast with Frequentists approaches which consider model parameters fixed but unknown. Examples of Bayesian methods in hydrology are the Generalized Likelihood Uncertainty Estimation framework of Beven and Binley [16] and the Bayesian Recursive Estimation approach of Thiemann [17]. Recursive Parameter estimation algorithms help to identify model structural flaws by reducing the temporal aggregation associated with traditional batch processing, like PIMLI and recursive Shuffled Complex Evolution Metropolis algorithms (SCEM-UA) [18, 19]. Multiobjective frameworks in order to better understand the limitation of the models, use complementary criteria in the optimization procedure and analyze the trade off in the fitting of these criteria; MOCOM [20] and MOSCEM-UA [15] being examples of these algorithms. Finally, realistic stochastic input error models, like the Bayesian Total Error Analysis of Kavetski, only account for input errors.

These previously discussed methods were not successful to account for all the referred sources of uncertainty in hydrologic modelling, i.e. parameter, input, output and structural model errors. Later, sequential data assimilation (SDA) techniques, represented by Kalman and extended Kalman filters techniques, for linear and nonlinear models respectively, continuously update the parameters of the model when new measurements are available, in order to improve the model forecast and evaluate the forecast accuracy. Recently, Vrugt et al. in [15] enrich the classical calibration approach with SDA techniques, thus developing the called simultaneous parameter optimization and data assimilation (SODA) method, which combines the strengths of the parameter search efficiency and explorative capabilities of the Shuffled Complex Evolution Metropolis algorithm [21], with the power and computational efficiency of the ensemble Kalman filter, thus accounting for the parameter, input, output and model structural uncertainties in hydrologic modeling.

Another approach aiming to reduce the uncertainty of models and improve their prediction skills consists on identifying the sensitive parameters and then focus on reducing the error of these delicate parameters [22]. For example, in [23], Yuan Shijin et al. studied the sensitivity of wind stress, the viscosity coefficient and the lateral friction for the simulation of the double-gyre variation in the Regional Ocean Modeling System [24], a model that can be used to simulate global waters of any size from basins to oceans. This sensitivity study was carried out not only for single parameters, but also for the combination of multiple parameters, by means of solving the Conditional Nonlinear Optimal Perturbation related to Parameter (CNOP-P) method [25], with the help of a modified Simulated Annealing (SA) algorithm in order to find the optimal solution in an efficient way. These works ([23]) exploring optimal parameters using sensitivity experiments, not only for individual parameters but also taking into account the interdependence between model parameters, are not feasible for models with large number of parameters, due to the fact that the number of necessary experiments increases exponentially with the involved number of model variables. A study of the sensitivities of the parameters for a simplified version of the model we are considering in this work was carried out by means of Multi-Level Monte Carlo in [26], the fluid model component being hydrostatic with just one fluid layer.

In a general setting, the data assimilation problem, for a given model, can be posed as an unconstrained global optimization problem in a bounded domain. Stochastic global metaheuristic algorithms are useful to solve these kind of problems. They have the advantage of needing little information of the function, and also allow to escape from local optima, being their main disadvantage the slow rate of convergence, which is typical of Monte Carlo algorithms. Classical well known examples of these methods are Simulated Annealing (see [27, 28]), Particle Swarm (PS, see [29, 30]) or Differential Evolution (DE, see [31]). Conversely, local optimization algorithms are deterministic and use more information of the function, thus being faster. Their main disadvantages are that, in general, they require some regularity of the cost function, and even more important, they do not guarantee reaching the global optimum, as they can get trapped into a

local minimum. They can be gradient free, for example Pattern Search (PS, see [32]) or Nelder-Mead (NM, see [33]); or gradient based, like steepest descent, Newton method, Conjugate Gradient (CG), Nonlinear CG (NCG, see [34]) or Quasi-Newton methods, for example, BFGS [35, 36, 37, 38], L-BFGS [39] or L-BFGS-B [40]. One idea to profit from the good properties of stochastic (global) and deterministic (local) algorithms, is to hybridize them: this can be done, for example, by nesting the local search inside the global algorithm. One example is the Basin Hopping (BH) algorithm [41, 42, 43]. In this work, in order to calibrate the tsunami model, we follow this idea, using a in-house developed parallel multi-path version of the BH algorithm.

Data assimilation for shallow-water models has been addressed in many works. In these works usually gradient based local optimization methods, like the simplest steepest descent method, have been used to solve the resulting optimization problem. Due to the high computational cost, the gradient is computed by solving the adjoint problem, either by solving directly the adjoint system or computing the adjoint by automatic differentiation (AD, see [44, 45]). For example, in [46] the identification of Manning's roughness coefficients in shallow-water flows is performed, and the authors compare three local optimization algorithms, a n-trust region method, L-BFGS and L-BFGS-B minimizers, where the gradients are computed by solving the adjoint equations. In [47] the variational data assimilation method (4D-VAR) is presented as a tool to forecast floods, in the case of purely hydrological flows: the cost function is a modification of the shallow-water equations to include a simplified sediment transport model and the steepest descent algorithm is then used to find the minimum. The initial and boundary conditions are calibrated. The gradient of the cost function is analytically computed by solving the adjoint equations of the model. In [48] the authors developed a 4D-VAR combining remote sensing data (spatially distributed water levels extracted from spatial images, SAR) and a 2D shallow-water model to identify time-independent parameters (e.g. Manning coefficients and initial conditions) and time-dependent parameters (e.g. inflow). In [49] the authors show the application of the technology developed in [48] to derive water levels with precision from satellite images of a real event. In [50] the authors presented a method to use Lagrangian data along with classical Eulerian observations, in a variational data assimilation process for a river hydraulics 2D shallow-water model, using the trajectories of particles advected by the flow and extracted from video images. In all the cited works AD is applied for computing the gradients, and the data assimilation is performed using gradient local optimization algorithms.

Data assimilation for tsunamis forecasting and early warning is a very challenging problem, and on top of that some data are even unknown, for example the geometry of the landslide or bottom deformation related to earthquake. Real time data is available in the Tsunami Early Warning Systems (TEWS), for example in the tide-gauges network of Deep-Ocean Assessment and Reporting of Tsunamis (DART) from National Data Buoy Center of the NOAA, or similar systems from other countries, see [51]. Tsunami buoys are not only intended to display the occurrence of the tsunami, but also to provide real time data that

can be assimilated into the tsunami warning system, to improve the accuracy of the tsunami forecasting. Real time data assimilation in tsunamis models is mostly done using optimal interpolation (OI) and tsunami Green functions, which are calculated in advance with linear tsunami propagation models, see for example [52, 53]. Another alternative assimilation method, is to use Kalman filter techniques (see [54, 55]) for wave field reconstructions and forecasts, see [56, 57]. In [58] data assimilation is done using a OI algorithm to both the real observations and virtual stations, in order to construct a complete wave front of tsunami propagation. In [59] tsunami data assimilation of high-density offshore pressure gauges is performed. In [56] a Kalman filter technique is proposed and compared with OI. In [60] the assimilation of Lagrangian data into a primitive equations circulation model of the ocean at basin scale, using the four-dimensional variational technique and the adjoint method, is performed. In [61] retrospectively data assimilation is applied to the tsunami generated in 2011 off the Pacific coast by the Tohoku Earthquake (Mw 9.0). The data assimilation is done using an algorithm of near-field tsunami forecasting with tsunami data recorded at various offshore tsunami stations: these measures were taken between 5 and 10 minutes before the tsunami reached the coastal tide-gauge stations nearest to its origin.

Nevertheless data assimilation in landslide generated tsunamis is not so well-developed. In this work we propose to use global optimization algorithms, that in general produce better results than the local ones. In fact many times the calibrated parameters do not correspond to the global minimum of the involved cost function because the considered local optimizer got stuck in a local minima far from the global solution.

Our work lies in the same vein of the recent works of Sumata et al. [62] and [63]. For example in [63] the authors applied a global minimization algorithm in order to calibrate an Arctic Sea Ice-Ocean model. Their approach consists on minimizing a cost function corresponding to the model-observation misfit of three sea ice quantities (the sea ice concentration, drift and thickness), with a genetic algorithm. The similarities between this work and our approach are the use of bound constrained global stochastic minimization and the method to assess on the optimality of the achieved solution by using a pool of independent and randomly initialized minimization experiments. Nevertheless, the approach we are proposing differs from their strategy in several features. First of all, our goal is to calibrate a tsunami model involving less parameters than the 15 model variables of the sea ice-ocean model calibrated in their article. Besides, the different nature between this model and the tsunami model we are looking at, enforces a different optimization window, a large one (two decades) in their work versus a small one (a few hours at most) in our sketch. On top of that, Sumata et al. performed the optimization of the cost function on a discrete search space, while our approach, allowing a continuous parameter domain, is richer.

Based on their previous work [62], Sumata et al. in [63] support, as our work does, the statement that gradient descent local minimization algorithms are likely to get stuck at local minima for these complicated cost functions.

Therefore, the authors impose the need to use stochastic global minimization algorithms. In fact, in [62] two types of optimization methods were applied to the calibration of a coupled ocean-sea ice model, and a comparison was made to assess the applicability and efficiency of both methods. One was a gradient descent method based on finite differences for computing the gradient, while the other was a genetic algorithm. Also a parallel implementation was carried out to speed up the optimization process. In the case of the gradient descent method, each component of the gradient was computed in parallel. They precisely conclude that the global optimization GA is preferred, because it yields a better optimum, since the gradient local optimizers could get trapped in local optima, even if several launches of the gradient algorithm are launched, in a multistart fashion. This statement exactly coincide with our forthcoming conclusions in Section 4.1 and 4.2 (see Figures 4 and 11).

In our paper, we overcome this disadvantage, by proposing for first time in this field, the use of a parallel hybrid local-global minimization algorithm. More precisely we develop a BH like algorithm. BH consists on hybridizing SA and local gradient searchers, allowing to benefit from both worlds, the global convergence properties of SA and the speed of local optimizers. We go even further by proposing a parallel version of the BH algorithm. For the local searcher ingredient of BH, we use a bounded version of the L-BFGS algorithm used in [62], namely the L-BFGS-B algorithm. This version is able to increase the convergence speed and the success rate of BH. The multistart technique performed in [62] can be seen as computing only one temperature stage of our multi-path BH algorithm. Another advantage of our algorithm is its embarrassingly parallel nature, as we can map each search path to a different parallel thread. In [62] each CPU thread computes one component of the gradient, while in our case, each thread is responsible of one L-BFGS path. We show using an analytical test, that this algorithm improves the multi-start technique, as it is always able to find the global optimum. Besides, in our article not only we compare the efficiency of this multi-path BH, with the equivalent version of a multipath SA (that can be seen as the BH without performing the local searches), but also show that by using the gradient searches the convergence speed of even a multipath SA increased. As mentioned before, a SA algorithm was also used in [23] to effectively solve the CNOP-P of ROMS.

The organization of this paper is as follows. In Section 2 we pose the data assimilation problem. In Section 2.1 we describe the cost function, which is given by the measure of the mismatch between the free surface laboratory data and the computed one, that depends on the parameters we want to assimilate. The optimization of this cost function is a hard problem: on the one hand, the evaluation of the cost function is an expensive computational problem, because it relies in the solution of a time dependent system of partial differential equations. On the other hand, this data assimilation problem gives rise to a global optimization problem. In Section 2.2 we briefly describe the two-phase tsunami model and give some references about the numerical scheme we use. The physical parameters of the system, that need to be calibrated, are the ratio of densities between the grain and the fluid, the Coulomb friction angle and

the Manning friction coefficient. The evaluation of the cost function requires a numerical solution of this two-phase model, computed for a given set of parameters.

In Section 3, we recall the global optimization algorithms that we will use: multi-path Simulated Annealing and multi-path Basin Hopping algorithms. Both algorithms were proposed by the authors in [64] and [65] for accelerating the convergence of SA and BH respectively, and are based in performing synchronized parallel Metropolis searches, or parallel gradient based local searches. They were assessed against the hard benchmarks in the global optimization field, and have been successfully applied to the calibration of models in finance, even in the case where the costly Monte Carlo method is the only alternative to price the involved financial products (see for example [66]). In this work we apply these algorithms for data assimilation in landslide tsunami modeling. One of the objectives of this article is to show that this type of algorithms can be successfully applied for the parameters calibration on challenging geophysical problems.

In Section 4, we present the numerical experiments that we have carried out: Section 4.1 is devoted to validating the methodology using synthetic tests, in which the model is run for fixed sets of parameters, and we generate files with the free surface information. Then, we consider these data as data coming from laboratory, and try to recover the parameters that were used to generate those data, by global optimization in a large domain. After validating the methodology, in Section 4.2 we apply the technique for performing the data assimilation considering real laboratory data.

2. Data assimilation problem

In general, the cost function measures the error, computed in some norm, between the real data and the solution produced by the numerical model. The model will depend on a set of parameters. For example, in the case of a one layer shallow-water model, they can be: one Manning coefficient for the whole domain, or also several Manning coefficients, one per subdomain; the initial conditions; the boundary conditions, etc. These parameters can be even time dependant (boundary conditions, for example).

2.1. Cost function

In this study, the cost function only depends on the free surface elevation because this quantity is easily measurable and perhaps the most important magnitude to predict the tsunami inundation. Thus, to carry out the data assimilation method we can introduce the following cost function using the Hilbert space $L^2(0,T;\Omega)$ norm:

$$f(\boldsymbol{p}) = \|\eta^{\boldsymbol{p}} - \eta^{obs}\|_{L^2(0,T;\Omega)} = \Big(\int_0^T \|\eta^{\boldsymbol{p}}(\cdot,t) - \eta^{obs}(\cdot,t)\|^2_{L^2(\Omega)} dt\Big)^{1/2}, \quad (1)$$

where $\Omega \subset \mathbb{R}$ is the spatial domain, $[0, T]$ is the time domain, $\eta^{\boldsymbol{p}}(x, t)$ is the free surface elevation at the point x and at time t computed with some model using the set of parameters $\boldsymbol{p}$, and η^{obs} are the observed values, that can be obtained from SAR images, sea buoys or laboratory experiments. This leads to an unconstrained global optimization problem in a bounded domain. More precisely, we address problems that can be formulated as

$$\min_{\boldsymbol{p} \in D \subseteq \mathbb{R}^n} f(\boldsymbol{p}),$$

where f is a real valued function, with $\boldsymbol{p} \in \mathbb{R}^n$ the vector of parameters, defined on $D = \prod_{i=1}^{n} [l_i, u_i]$, with l_i and u_i being the lower and upper bounds in direction i, respectively. The solution can be written as:

$$\boldsymbol{p}^* = \arg\min_{\boldsymbol{p} \in D \subseteq \mathbb{R}^n} f(\boldsymbol{p}).$$

In the discrete case, the cost function will have the following expression:

$$f(\boldsymbol{p}) = \sqrt{\sum_{k=1}^{N_T} \sum_{i=1}^{N} \left(\eta_{i,k}^{\boldsymbol{p}} - \eta_{i,k}^{obs} \right)^2},$$

where $\eta_{i,k}^{\boldsymbol{p}} = \eta^{\boldsymbol{p}}(x_i, t_k)$ and $\eta_{i,k}^{obs} = \eta^{obs}(x_i, t_k)$ being x_i the i-th measure point, for $i = 1, \ldots, N$ and t_k the k-th measure time, with $k = 1, \ldots, N_T$.

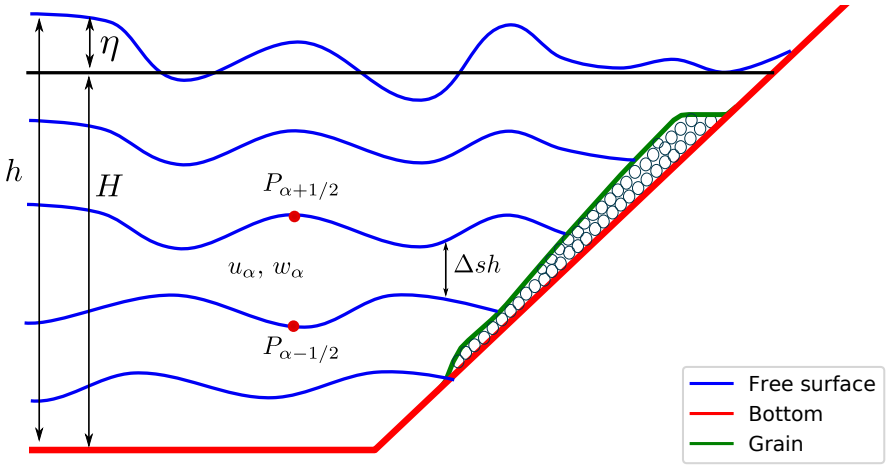


Figure 1: Sketch of the model.

Note that the cost function depends on $\eta^{\boldsymbol{p}}$, which implicitly depends on the parameters to be calibrated. Therefore, a single evaluation of the cost function requires a realization of the numerical model for a given set of parameters. In

the next section we present the equations of the two-phase model, pointing out what are the parameters to be calibrated. Some basic idea about the numerical scheme we use is also sketched.

2.2. Mathematical model

As discussed in the introduction, we use a two-phase model in order to describe the interaction between the submarine landslide and the fluid. In this work, a Savage-Hutter model (see [9]) is considered for the kinematics of the submarine landslide, and a multi-layer non-hydrostatic shallow-water model is used for the evolution of the ambient water (see [12]). Both models are coupled through the boundary conditions at the sea-floor.

At this point, we suppose that the landslide is totally submerged and that the ratio of densities between the ambient fluid and the granular material is given by the parameter r. Usually

$$r = \frac{\rho_f}{(1-\varphi)\rho_s + \varphi\rho_f},$$

where ρ_s is the typical density of the granular material, ρ_f is the density of the fluid ($\rho_s > \rho_f$), and φ is the porosity ($0 \leq \varphi < 1$). Here, we suppose that φ is constant on space and time, and therefore r is also constant. Note that $0 < r < 1$. Finally, let us remark that even on a uniform material, r is difficult to estimate as it depends on porosity φ. Typical values of r are in the interval $[0.3, 0.8]$.

The 1D Savage-Hutter model that we consider in this article is written as follows:

$$\begin{cases} \partial_t z_s + \partial_x(z_s u_s) = 0, \\ \partial_t(z_s u_s) + \partial_x\left(z_s u_s^2 + \dfrac{g(1-r)}{2} z_s^2\right) = g(1-r) z_s \partial_x H + \tau_C, \end{cases} \tag{2}$$

where g is the gravity acceleration ($g = 9.81\ m/s^2$); $H(x)$ is the non-erodible bathymetry measured from a given reference level and unchanged during the simulation; $z_s(x,t)$ is the landslide depth at each point x at time t; and $u_s(x,t)$ the averaged horizontal velocity. τ_C is the Coulomb friction term given by:

$$\tau_C = -g(1-r)\mu z_s \frac{\sqrt{u_s^2}}{u_s}.$$

Note that this term is multi-valuated when $u_s = 0$. The simplest friction law corresponds to a constant friction coefficient:

$$\mu = \tan(\theta), \tag{3}$$

where θ is the friction angle, although more complex friction terms could be used to simulate natural subaerial or submarine landslides (see [67, 68]). Other

definitions, derived from experimental data, have been proposed by Pouliquen (see [69]) where the friction coefficient depends on the velocity and thickness of the granular layer. This law is widely used in the literature and involves at least three parameters to be calibrated (see e.g. [70]).

The Coulomb friction term τ_C is quite relevant, as it controls the motion of the landslide. In particular, it is defined in terms of the friction angle θ, which is a parameter to calibrate in order to fit the simulation with the experimental data. Finally, let us mention that in the derivation of the previous model we have supposed a rigid-lid assumption with respect to the free surface of the ambient fluid: that is, the pressure variations induced by the fluctuation on the free surface of the ambient fluid over the landslide are neglected. Nevertheless, the buoyancy effects have been taken into account.

The ambient fluid is supposed to be modeled by a multi-layer non-hydrostatic shallow-water system recently proposed in [12]. This system is obtained by a process of depth-averaging of the incompressible Euler equations. More precisely, it can be seen as a particular semi-discretization with respect to the vertical variable of the incompressible Euler equations. Total pressure is decomposed into a sum of a hydrostatic and a non-hydrostatic component. In this process, vertical velocities are assumed to have a linear vertical profile, whilst the horizontal velocities are supposed to have a constant vertical profile. The resulting multi-layer model admits an exact energy balance, and when the number of layers increases, the linear dispersion relation of the linear model converges to the same of Airy's theory. The model proposed in [12] can be written in compact form as

$$
\begin{cases}
\partial_t h + \partial_x(h\bar{u}) = 0, \\[2ex]
\partial_t(hu_\alpha) + \partial_x\Big(hu_\alpha^2 + \dfrac{g}{2}h^2\Big) - gh\partial_x(H - z_s) \\
\qquad\qquad + u_{\alpha+1/2}\Gamma_{\alpha+1/2} - u_{\alpha-1/2}\Gamma_{\alpha-1/2} = -h(\partial_x p_\alpha + \sigma_\alpha \partial_z p_\alpha) - \tau_\alpha, \\
\partial_t(hw_\alpha) + \partial_x(hu_\alpha w_\alpha) + w_{\alpha+1/2}\Gamma_{\alpha+1/2} - w_{\alpha-1/2}\Gamma_{\alpha-1/2} = -h\partial_z p_\alpha, \\
\partial_x u_{\alpha-1/2} + \sigma_{\alpha-1/2}\partial_z u_{\alpha-1/2} + \partial_z w_{\alpha-1/2} = 0,
\end{cases}
\tag{4}
$$

for $\alpha \in \{1, 2, \ldots, L\}$, being L the number of layers. In the previous system, we have used the following notation:

$$u_{\alpha+1/2} = \frac{1}{2}(u_{\alpha+1} + u_\alpha), \quad \partial_z u_{\alpha+1/2} = \frac{1}{h\Delta s}(u_{\alpha+1} - u_\alpha),$$

$$w_{\alpha+1/2} = \frac{1}{2}(w_{\alpha+1} + w_\alpha), \quad \partial_z w_{\alpha+1/2} = \frac{1}{h\Delta s}(w_{\alpha+1} - w_\alpha),$$

$$p_\alpha = \frac{1}{2}(p_{\alpha+1/2} + p_{\alpha-1/2}), \quad \partial_z p_\alpha = \frac{1}{h\Delta s}(p_{\alpha+1/2} - p_{\alpha-1/2}),$$

$$\sigma_\alpha = \partial_x(H - z_s - h\Delta s(\alpha - 1/2)), \quad \sigma_{\alpha-1/2} = \partial_x(H - z_s - h\Delta s(\alpha - 1)). \tag{5}$$

As depicted in Figure 1, the flow depth h is split along the vertical axis into $L \geq 1$ layers and $\Delta s = 1/L$. u_α and w_α are the depth averaged velocities in the x and z directions respectively, and g is the gravitational acceleration. The term $p_{\alpha+1/2}$ is the non-hydrostatic pressure at the interface $z_{\alpha+1/2}$. The free surface elevation measured from the still-water level is $\eta = h - H + z_s$, where again $H(x)$ is the unchanged non-erodible bathymetry measured from a given reference level. $\tau_\alpha = 0$, $\alpha > 1$ and τ_1 is the Manning friction term that is only present at the lowest layer ($\alpha = 1$) given by

$$\tau_1 = gh\frac{n^2}{h^{4/3}}u_1|u_1|.$$

Finally, for $\alpha = 1, \ldots, L-1$, $\Gamma_{\alpha+1/2}$ account for the mass transfer across interfaces and are defined by

$$\Gamma_{\alpha+1/2} = \sum_{\beta=\alpha+1}^{L} \partial_x(h\Delta s(u_\beta - \bar{u})), \quad \bar{u} = \sum_{\alpha=1}^{L} \Delta s u_\alpha.$$

Here we suppose that $\Gamma_{1/2} = \Gamma_{L+1/2} = 0$, that is, there is no mass transfer through the bottom nor the free-surface.

In order to close the system, the following boundary conditions are considered: $p_{L+1/2} = 0$, $u_0 = 0$ and $w_0 = \partial_t z_s$. Note that the last two conditions enter into the incompressibility condition for the lowest layer ($\alpha = 1$), given by

$$\partial_x u_{1/2} + \sigma_{1/2}\partial_z u_{1/2} + \partial_z w_{1/2} = 0.$$

Observe that both models are coupled through the unknown z_s, present in the equations and in the boundary condition ($w_0 = \partial_t z_s$).

Note that the two-phase model depends on three coefficients (that are the ones to be calibrated), namely the vector of coefficient is $\boldsymbol{p} = (r, \theta, n)$, where r is the ratio of densities between the fluid and the granular phase, θ the Coulomb friction angle, and n the friction (Manning) coefficient. In particular the first two are quite relevant for the landslide motion and therefore, for the induced tsunami water waves.

System (2) could be written in the following compact way:

$$\partial_t U_s + \partial_x F_s(U_s) = G_s(U_s)\partial_x H - S_s(U_s), \tag{6}$$

being

$$U_s = \begin{bmatrix} z_s \\ u_s z_s \end{bmatrix},\ F_s(U_s) = \begin{bmatrix} z_s u_s \\ z_s u_s^2 + \dfrac{g(1-r)}{2} z_s^2 \end{bmatrix},$$

$$G_s(U_s) = \begin{bmatrix} 0 \\ g(1-r) z_s \end{bmatrix},\ S_s(U_s) = \begin{bmatrix} 0 \\ \tau_C \end{bmatrix}.$$

The multi-layer non-hydrostatic shallow-water system could also be expressed in a similar way:

$$\begin{cases} \partial_t U_f + \partial_x F_f(U_f) + B_f(U_f)\partial_x U_f = G_f(U)\partial_x(H - z_s) + \mathcal{T}_{NH} - S_f(U_f), \\ B(U_f, (U_f)_x, H, H_x, z_s, (z_s)_x) = 0, \end{cases} \tag{7}$$

where

$$U_f = \begin{bmatrix} h \\ hu_1 \\ \vdots \\ hu_L \\ hw_1 \\ \vdots \\ hw_L \end{bmatrix}, F_f(U_f) = \begin{bmatrix} h\bar{u} \\ hu_1^2 + \dfrac{1}{2}gh^2 \\ \vdots \\ hu_L^2 + \dfrac{1}{2}gh^2 \\ hu_1 w_1 \\ \vdots \\ hu_L w_L \end{bmatrix}, G_f(U_f) = \begin{bmatrix} 0 \\ gh \\ \vdots \\ gh \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

$B_f(U_f)\partial_x(U_f)$ contains the non-conservative products involving the momentum transfer across the interfaces

$$B_f(U_f)\partial_x(U_f) = \begin{bmatrix} 0 \\ u_{3/2}\Gamma_{3/2} \\ u_{5/3}\Gamma_{5/2} - u_{3/2}\Gamma_{3/2} \\ \vdots \\ -u_{L-1/2}\Gamma_{L-1/2} \\ w_{3/2}\Gamma_{3/2} \\ w_{5/3}\Gamma_{5/2} - w_{3/2}\Gamma_{3/2} \\ \vdots \\ -w_{L-1/2}\Gamma_{L-1/2} \end{bmatrix},$$

$S_f(U_f)$ contains the Manning friction term

$$S_f(U_f) = \begin{bmatrix} 0 \\ \tau_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

The non-hydrostatic corrections in the momentum equations are given by

$$\mathcal{T}_{NH} = \mathcal{T}_{\mathcal{NH}}(h, h_x, H, H_x, z_s, (z_s)_x, p, p_x) = -\begin{bmatrix} 0 \\ h(\partial_x p_1 + \sigma_1 \partial_z p_1) \\ \vdots \\ h(\partial_x p_L + \sigma_L \partial_z p_L) \\ h\partial_z p_1 \\ \vdots \\ h\partial_z p_L \end{bmatrix},$$

and finally, the operator related with the incompressibility condition at each layer is given by:

$$B(U_f, (U_f)_x, H, H_x, z_s, (z_s)_x) = \begin{bmatrix} \partial_x u_{1/2} + \sigma_{1/2}\partial_z u_{1/2} + \partial_z w_{1/2} \\ \vdots \\ \partial_x u_{L-1/2} + \sigma_{L-1/2}\partial_z u_{L-1/2} + \partial_z w_{L-1/2} \end{bmatrix}.$$

The discretization of systems (6) and (7) becomes difficult. In this article, we have considered the natural extension of the numerical schemes proposed in [71] and [72], where a splitting technique has been described. Firstly, the systems (6) and (7) can be expressed as the following non-conservative hyperbolic system:

$$\begin{cases} \partial_t U_s + \partial_x F_s(U_s) = G_s(U_s)\partial_x H, \\ \partial_t U_f + \partial_x F_f(U_f) + B_f(U_f)\partial_x(U_f) = G_f(U_f)\partial_x(H - z_s). \end{cases} \tag{8}$$

Both equations are solved simultaneously using the same *time step*, by means of a second order HLL, positivity-preserving and well-balanced, path-conservative finite volume scheme (see [73]). The synchronization of time steps is done taking into account the CFL condition of the complete system (8). A first order estimation of the maximum of the wave speed for system (8) is the following:

$$\lambda_{\max} = \max(|u_s| + \sqrt{(g(1-r)h_s}, |\bar{u}| + \sqrt{gh}).$$

Next, the non-hydrostatic pressure corrections $p_{1/2}, \cdots, p_{L-1/2}$ at the vertical interfaces are computed from

$$\begin{cases} \partial_t U_f = \mathcal{T}_{NH}(h, h_x, H, H_x, z_s, (z_s)_x, p, p_x), \\ B(U_f, (U_f)_x, H, H_x, z_s, (z_s)_x) = 0. \end{cases}$$

This requires the discretization of an elliptic operator by means of standard second order central finite differences. The resulting linear system is solved using an iterative Scheduled Jacobi method (see [74]). Finally, the horizontal and vertical momentum at each layer are updated using the computed non-hydrostatic corrections. At this stage, the frictions $S_s(U_s)$ and $S_f(U_f)$ are also discretized (see [71, 72]). We refer the reader to [10] for the discretization of the Coulomb friction term.

3. Multi-path BH global optimization

In this section we describe the optimization algorithms multi-path SA ($\mathrm{SA_M}$) and multi-path BH ($\mathrm{BH_M}$), that can be seen as a modification of the sequential BH algorithm, introducing a parallel multi-path searching technique.

The BH algorithm is a hybrid between the Metropolis algorithm and some kind of gradient local optimization method, in order to profit from the speed and accuracy of the local optimizer, while retaining the global convergence properties of the stochastic one. The seminal idea was presented by Navon and Robertson et al. in [41, 42] for finding the global minimum of Potential Energy Surfaces (PES) related to structures of mixed Argon-Xenon clusters. The authors developed the finite-temperature lattice based Monte Carlo method and compared the use of three different limited memory Quasi-Newton-like conjugate gradient methods as local minimizers, the L-BFGS against two others, being L-BFGS the better performing one. Seven years later, a similar idea was also successfully applied by Wales and Doye (see [43]) in order to minimize the PES, for finding Lennard-Jones clusters using a nonlinear conjugate gradient method (Polak Ribière [34]) as the local optimizer. In the latter reference the authors named the method Basin-Hopping; this name became widely accepted for referring to these kind of global optimization methods. Nowadays, the term BH encompasses a family of algorithms obtained by combining different local (NCG, BFGS, ...) and global stochastic algorithms (Metropolis or SA): quasi-Newton methods (BFGS and descendants) are the most common choice for the local component. BH methods have been extensively studied by Locatelli et al., see [75, 76, 77, 78], and Leary [79]. In the BH method, the local optimizer can be seen as an operator that transforms the original function $f(\boldsymbol{x})$, returning a new piece-wise constant function, $L(\boldsymbol{x}) = f(\mathcal{LS}(\boldsymbol{x}))$, being $\mathcal{LS}(\boldsymbol{x})$ the point where a local minimum of f is obtained from a starting point $\boldsymbol{x}$. The resulting global optimization problem for the function $L(\boldsymbol{x})$, is much more tractable for the global optimizer component, as the barriers between local minima have been softened.

The idea of BH is to use a temperature process like in SA: we denote by $\mathcal{T}$ and $\mathcal{T}_{min}$ the current and minimum temperatures, we consider the temperature reduction schedule, $\mathcal{T}_{k-1} = \rho\mathcal{T}_k$, being ρ the cooling rate, and we perform a Metropolis process with N steps at each temperature level. More precisely, at temperature level $\mathcal{T}_k$, being $\boldsymbol{x}_k$ the starting point, first, we generate a random neighbor, $\boldsymbol{y}_k$, inside a ball with radius r_k and centered in $\boldsymbol{x}_k$, $\boldsymbol{y}_k \in B(\boldsymbol{x}_k, r_k)$. Next, we perform a gradient local search starting from $\boldsymbol{y}_k$, in order to obtain a

local minimum, and we decide whether to accept or discard it, using the Boltzmann law. Finally, we advance to the next temperature level. The algorithm stops when the temperature reaches $\mathcal{T}_{min}$, or the number of successive rejections exceeds J. The radius r_k is updated after a certain interval, by using the 50% acceptance rule [80]. A nice property is that BH can also be seen as a generalization of SA: SA can be recovered by skipping the local optimization phase in BH.

In [64, 65] the authors proposed a synched multiple Metropolis path approach for SA and BH-like algorithms, respectively. The idea is to perform not one, but M Metropolis searches at each temperature level $\mathcal{T}_k$, from the same initial point $\boldsymbol{x}_k$ (see Algorithm 1). In the simplified case with $N = 1$, the algorithm consists of launching M gradient local searchers (see Figure 2), starting from the corresponding set of random neighbors $\boldsymbol{y}_k^l$, $l = 1, \ldots, \mathrm{M}$, of the current minimum point, and thus the Metropolis searches are entirely replaced by local searches. After performing the N steps of Metropolis at each path, and before advancing to the next temperature level, we gather the final information, keeping the best of the attained minima, so that $\boldsymbol{x}_{k+1}^{best} = min(\boldsymbol{x}_*^l)$ (see Figure 2 and Algorithm 1). We will refer to this algorithm as $\mathrm{BH_M}$, M being the number of paths (number of Metropolis processes with local searchers; or just the number of local searches, if $N = 1$) launched at each temperature optimization step. Note that if besides $M = 1$, then $\mathrm{BH_1}$ corresponds with the classical BH (only one Metropolis path, or only one local search). Also, if we replace the local search operator, $\mathcal{LS}$, with the identity, id, we recover the multi-path SA algorithms, $\mathrm{SA_M}$ [64]; furthermore $\mathrm{SA_1}$ corresponds to the classical single path SA. These multi-path $\mathrm{BH_M}$ algorithms have two interesting properties: on the one hand, they are highly parallelizable; on the other hand they improve convergence properties, both the convergence speed and the success rate of the classical SA and BH.

This approach has the advantage of being easily parallelizable, because the multiple search paths can be computed asynchronously at the same time. For example, if we have a multi-CPU architecture, each CPU thread can take care of computing one local search, and after that the results have to be synchronized (see Figure 2). In this paper, we will use this multi-path implementation in a multi-CPU setting, each CPU thread will take care of a search path.

Regarding the convergence properties, in [65] the study of the optimal number of multi-searches, both from the convergence rate and the success rate viewpoint, is done empirically. According to the results, increasing the number of searchers improves the convergence rate, although this increase in convergence rate is not unlimited. For example, if the problem is simple and/or the dimension is low, by increasing the number of searchers one would only obtain a marginal increase in convergence speed. Nevertheless, even in those cases, the computing time can be lower because the evaluations can be done in parallel, and thus this increase in the number of search paths comes almost for free. Even if the problem is computationally hard, it always comes a point where the optimal convergence rate is achieved and a further increase in the number of searchers will not have any advantage. Usually this number of searches for

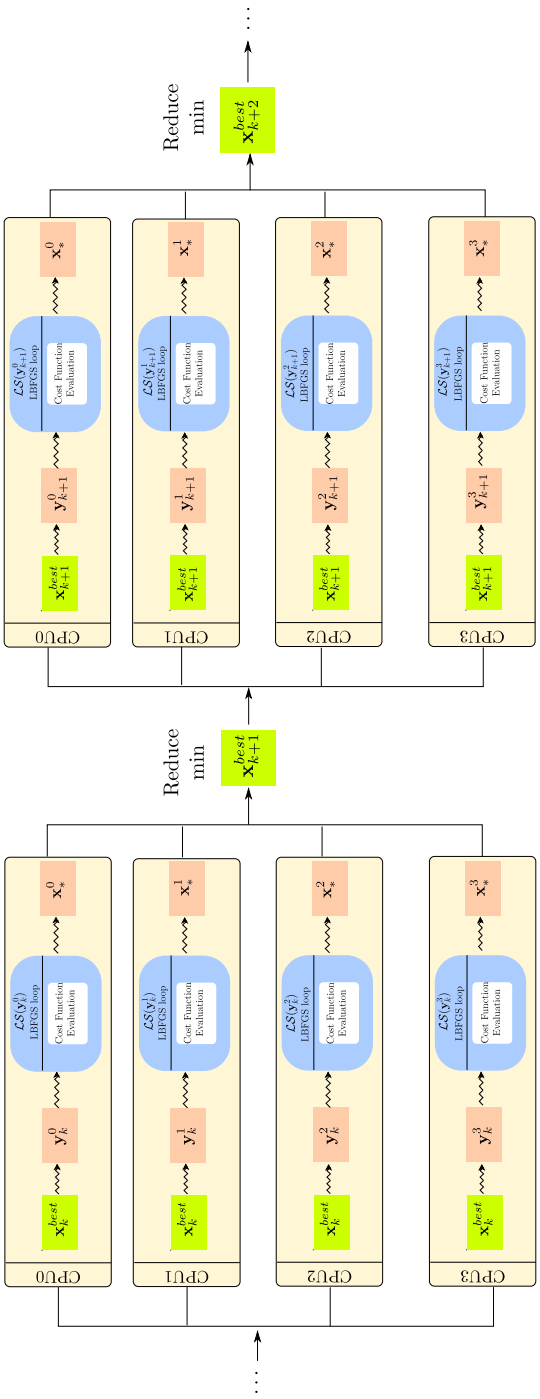


Figure 2: Schematic visualization of the BHM algorithm (with M = 4).

Algorithm 1: Synched multi L-BFGS-B BH, pseudocode.

$\boldsymbol{y}$ = random uniform in D;
Set # successive rejections: $j = 0$;
Iteration number: $k = 0$;
Initial position: $\boldsymbol{x}_0 = \boldsymbol{x}_* = \mathcal{LS}(\boldsymbol{y})$;
while *($j < J$)* ***or*** *($\mathcal{T} < \mathcal{T}_{min}$)* **do**
  for $l = 1{:}M$ **do**
    for $i = 1{:}N$ **do**
      $\boldsymbol{y}_i^l$ = random uniform in $B(\boldsymbol{x}_i^l, r_k)$;
      $\boldsymbol{u}$ = random uniform in $[0, 1]$;
      $\Delta = L(\boldsymbol{y}_i^l) - L(\boldsymbol{x}_*^l)$;
      if $\boldsymbol{u} < exp(-\Delta/\mathcal{T})$ **then**
        $\boldsymbol{x}_*^l = \boldsymbol{x}_{i+1}^l = \mathcal{LS}(\boldsymbol{y}_i^l)$;
        $j = 0$;
      else
        $j = j + 1$;
      end if
    end for
  end for
  Synchronization: $\boldsymbol{x}_{k+1}^{best} = min(\boldsymbol{x}_*^l)$;
  for $l = 1{:}M$ **do**
    $\boldsymbol{x}_*^l = \boldsymbol{x}_{k+1}^{best}$;
  end for
  $k = k + 1$;
  Update r_k;
  $\mathcal{T} = \rho \cdot \mathcal{T}$;
end while

obtaining an optimal convergence rate is moderate: the optimization problem has to be really tough in order to demand a high number of local searchers. The good properties of the proposed algorithm also apply to the success rate and the same conclusions can be obtained. Usually, it comes a point when a 100% success rate is achieved, more number of searchers will not have any advantage. Besides, the number of searches for obtaining this 100% success rate is, once more, normally moderate. For tough problems, the advantage of performing a large number of local searches becomes more evident.

In this work, for the local optimizer we will use the very robust L-BFGS-B algorithm. This minimizer is intended for problems in which information on the Hessian matrix is difficult to obtain. It was presented by Nocedal in [81] as an extension of the L-BFGS minimizer, being a limited-memory quasi-Newton algorithm (it does not need to store the Hessian matrix) that allows to solve nonlinear optimization problems with restrictions given by simple bounds on the variables of the function to be optimized.

In our work since the parameters are known to vary between given bounds, and we need to ensure that the optimizer would never explode by following a wrong path outside the physical domain, we used the L-BFGS-B bounded gradient method. If one uses a non bounded gradient local optimizer, some search paths could reach points outside the physical domain, where the equations could stop making sense. In that case the evaluation of the cost function (a finite volume solver) may explode, either by crashing or by entering in a very low Δt state (imposed by the CFL condition). As a consequence the assimilation process will crash or never end. We preferred to stay safe with the bounded algorithm, as it has almost the same computational cost as the unbounded L-BFGS version.

In order to compute the partial derivatives with respect to the variables to be identified, needed for the gradient of the objective function, we can use either algorithms based on the so-called adjoint method or the standard finite-difference method. Both techniques have their own advantages and disadvantages. In this article we opted for the finite difference procedure attending to the reasons that will be discussed hereafter.

There are two different approaches for tackling the adjoint problem. One technique is the classical approach developed by Lions (see [13]) and applied for the simpler 2D one layer shallow water model by Monnier et al. in [48]. It consists in computing the adjoint PDE system, and then solving it by numerical methods. This is a very challenging problem even for the simpler shallow water model assimilated by Monnier, and even much more for our problem at hand: we emphasize that we are dealing with a coupled model involving an arbitrary number of fluid layers (denoted by L in the PDE system (4)) of "shallow-water type systems", along with the Savage-Hutter equations, thus resulting in a large hyperbolic system of coupled conservation laws. The mentioned system can only be numerically approximated by means of very involved finite volume numerical discretizations, thus dealing with the corresponding stability issues related to high nonlinearities involved in hyperbolic problems along with spatial-temporal discretization issues. As a consequence, the adjoint method will lead to a system of conservation laws with source terms and non-conservative products, for which it would not be clear the hyperpolicity region. Besides, the numeric approximation of this adjoint system will be very sophisticated. One wonders if all this challenging work, even if feasible, is worth it for calibrating just this particular model. On the other hand, a way to circumvent those difficulties and avoid computing the adjoint system, is to compute the partial derivatives by means of Automatic Differentiation (AD). As in the close future we pretend to tackle real two dimensional problems, which involve much higher computational cost, and consequently even more for the adjoint AD procedure, speeding up on GPUs the cost function evaluation (i.e. the solution of the system) becomes compulsory. In this scenario, also the automatic differentiation algorithm should be carried out in the GPU side. Therefore, an AD library for GPUs in needed, something that can be an obstacle due to the fact that these tools are not always available, specially for massively parallel architectures like GPUs. Furthermore, the code should be rewritten from the very basics using the overloaded operators

provided by the AD library. On top of that, more memory will be needed in this adjoint setting, which is again an issue in GPUs.

Having in mind all the previously discussed issues, in this article we opted for the direct numerical approximation of the partial derivatives involved in the gradient using finite differences. In our case this has several advantages when compared to the adjoint computation. First of all, one of our goals is to develop a data assimilation framework/machinery for landslide tsunami models, generic enough in the sense that it should be directly applicable if one wants to enrich the here considered model with further characteristics or even fully replace it with other models. This machinery should endowe us with a tool for comparing the accuracy of (possibly quite) different models, and this is a reason for not developing an algorithm that is too tailored/tight for a particular model or numerical scheme. In this sense, by computing the gradient via finite differences we gain generality, since the method can be easily applied to models of all kinds without changes in the calibration procedure (in the same vein of [62]); one will just need to invoke it by plug in the new model solver (no changes are needed in the solver, unlike with the adjoint method). Hence we are well positioned in order to face the calibration of the previously mentioned oncoming richier two dimensional model to real data. Additionally, our technique is able to cope with the strongly nonlinear relation between model state and parameters, for which other approaches based on Kalman filter have difficulties. Finally, regarding the computational efficiency, finite-difference method for computing the derivatives of the cost function with respect to the parameters to be calibrated is not much more computing time demanding than the adjoint method if the number of optimization variables is short; indeed this is the case we are dealing with, our goal is to calibrate three parameters, namely the ratio r of densities between the fluid and the granular phase, the Coulomb friction angle θ, and the friction Manning coefficient n. Last but not least, nowadays, thanks to the available high computational power, the numerical computation of the gradient could be directly addressed making use of parallel codes that combine multi-CPU implementation of the optimizer and multi-GPU implementation of the numerical solver used to evaluate the cost function.

All in all, the gradient of the cost function will be numerically computed, using first order progressive finite differences

$$\frac{\partial f}{\partial p_i}(\boldsymbol{p}) = \frac{f(\boldsymbol{p} + \varepsilon \boldsymbol{e}_i) - f(\boldsymbol{p})}{\varepsilon},$$

with $\varepsilon = 10^{-6}$, and $\boldsymbol{e}_i = (0, \ldots, 1, \ldots, 0)$ the unitary vector of direction i.

Regarding the implementation of the algorithms, the whole implementation of both the cost function (finite volume solver) and its gradients, and the optimization algorithms, is custom made. Both algorithms have been integrated in an efficient code using C++, and OpenMP is used for the parallel implementation of the optimization codes (see Figure 2). Also we want to emphasize that the cost function is integrated with the optimization tool, so that it is called on the fly for each set of parameters during the optimization process. Therefore

no intermediate results need to be discharged from RAM to the hard drive for computing the value of the cost function, thus resulting in an efficient code. Furthermore, during the whole optimization process the laboratory data is read only once at the beginning.

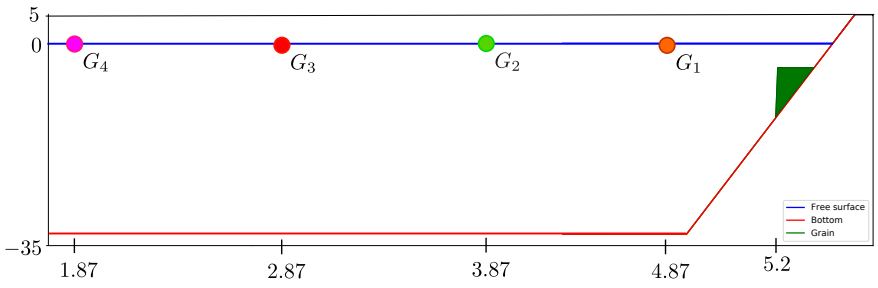


Figure 3: Sketch of the channel, initial condition and position of the tide-gauges.

4. Numerical results

In this section we present two sets of numerical examples. The first one in Section 4.1 is a pool of synthetic tests with known solutions, that are used to validate the proposed algorithms and methodology, to discuss about the identifiability of the problem, and to show the convergence results and computational speedup. The second one in Section 4.2 shows an application of the proposed methodology to the assimilation of real laboratory data.

The laboratory experiment that will be calibrated in this article was presented in [82], and the data can be accessed at [83]. In that work, the authors design different laboratory experiments and perform numerical simulations to validate a landslide tsunami model and to asses how tsunami hazard from SMFs is affected by slide kinematics and rheology.

In [84] the Tsunami-HySEA model is used to perform some of the numerical benchmark problems proposed in [82]. The obtained results are documented in the "Proceedings and results of the 2011 NTHMP Model Benchmarking Workshop".

In our article we focus in one of the experiments performed in [83]: the benchmark 4 (deformable submarine landslide). For both the analytical and the laboratory experiments, the physical conditions of this benchmark are considered. The length of the channel is 6 meters, and its sketch can be seen in Figure 3. The initial condition is water at rest with $\eta = 0$ and a triangular block of sediments, whose geometry is depicted in Figure 3. In our numerical results, we will have four tide-gauges, $N = 4$, where laboratory measures have been taken each 5 milliseconds, thus generating four tidal series. These buoys are located at the positions 1.87, 2.87, 3.87 and 4.87 meters, and they are depicted in Figure 3. We take $g = 9.81\ m/s^2$ and $L = 5$ layers of fluid in the model.

The calibration tests are run until $T = 8$ seconds both for the synthetic test and the laboratory experiment. For the finite volume method we consider 200 cells in the analytical test and 800 cells in the laboratory essay with $CFL = 0.5$.

We recall that the parameters are three, $\boldsymbol{p} = (r, \theta, n)$, where r is the ratio of densities between the fluid and the sediment, θ the Coulomb angle, and n the friction coefficient. The search domain for all the experiments in this section is $D = [0.3, 0.8] \times [5, 45] \times [10^{-5}, 10^{-3}]$, which is quite a broad domain.

Concerning the hardware configuration, all tests have been performed in a server with 16 CPU cores (two Intel Xeon E5-2620 v4 clocked at 2.10GHz, accounting 32 logical threads) and 16 GB of RAM.

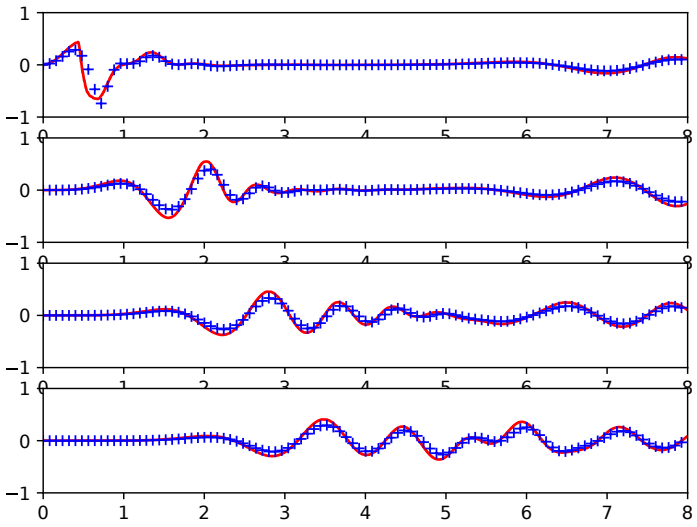


Figure 4: Synthetic generated series vs calibrated ones with the multi-start L-BFGS-B. Target series in red, simulated series in blue.

	r	θ	n
Target values	0.55	12°	0.0002
Obtained values	0.55	12°	0.0002

Table 1: Target and obtained values of the parameters.

4.1. Synthetic test

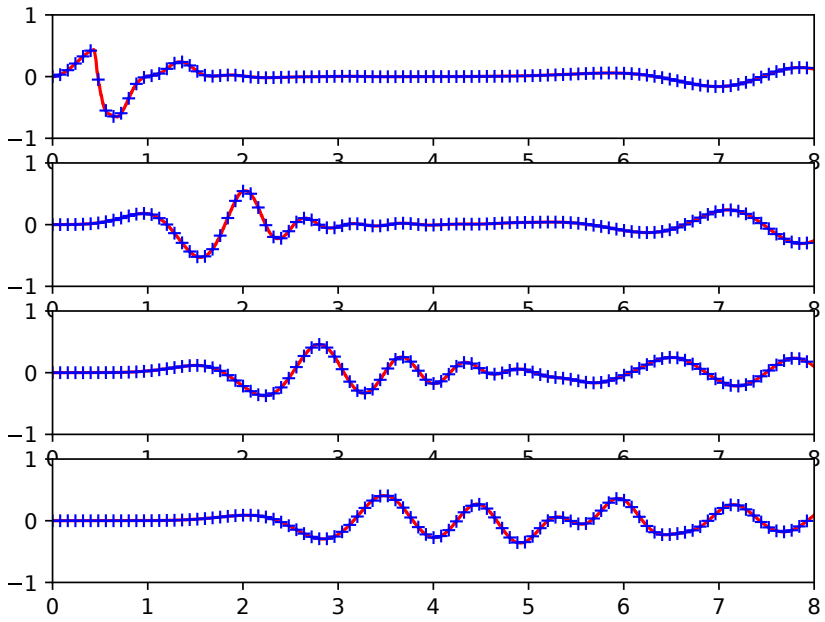


Figure 5: Synthetic generated series vs calibrated ones. Target series in red, simulated series in blue.

The notion of identifiability addresses the question of whether it is at all possible to obtain unique solutions of the inverse problem for unknown parameters of interest in a model from data collected in the spatial and temporal domains [85, 86]. As we have seen, data assimilation problems deal in the end with the search of the global minimum of a cost function. The exploration of the global minimum is a nontrivial task as long as the cost function has a complicated structure, and, on top of that, ensuring that the involved cost function has a unique global minimum is a extremely difficult goal, mainly due to the fact that sophisticated numerical methods are needed to simulate from landslide tsunami models being able to recover realistic physical phenomena, as discussed in Section 2.2.

Analyzing parameter identifiability is precisely one of the aims of this work. We seek to check whether the data assimilation problem for landslide tsunami models is well posed when using only information of the fluid free surface. In fact, our goal in this first set of numerical experiments is precisely to empirically discuss the problem of identifiability and uniqueness of the here proposed parameters calibration strategy. We will observe that the parameters are iden-

tifiable using only data from the free surface, which is something unexpected and eye catching from our point of view, because at first sight one could expect that information of the lower layers, the sediment layer or the speed of the fluid should be required in order to assimilate the data into the model. Nevertheless, in practice the information of the free surface proofs to be enough.

In this work, as in the article [63], the uniqueness of the minimum of the cost function will be discussed by invoking results from the following additional optimization experiments.

4.1.1. Synthetic test 1

First of all, we designed a synthetic experiment, where given the unknown set of parameters, we created the observations numerically, which were then assimilated into the model to retrieve the original set of parameters. The values of the parameters were set at $r = 0.55$, $\theta = 12^\circ$ and $n = 0.0002$. The test was run for 8 seconds. With these data we computed the simulation and stored the series corresponding with the free surface at each measure point in an interval of 0.005 seconds. Then, we supposed that the parameters were unknown and tried to recover them using our optimization algorithms. There is no doubt about the uniqueness of the global minimum: the value of the cost function at this unique global minimum is zero, since the observations are perfect because they arise from the model. This problem has a very similar level of complexity from the optimization point of view to the real one we want to tackle, although it has the advantage of being easier to handle, as the exact solution is known. Moreover, this benchmark allowed us to test and compare the different algorithms with different number of parallel search paths.

First we show that if a local optimization algorithm, like L-BFGS-B is applied, which can be seen as performing only one path and one temperature step of the hybrid algorithm, no convergence to global minimum is obtained. Thus, after executing a local L-BFGS-B searcher, starting from a random point of the search domain, the obtained set of parameters is $(r, \theta, n) = (6.826989 \times 10^{-01}, 10.68841753^o, 8.178492 \times 10^{-4})$, the value of the cost function being 5.273542×10^{-02}. The simulation obtained with this set of parameters is shown and compared with the exact solution at Figure 4. Therefore, a robust global optimization algorithm should be used to compute the global minimum of this problem.

Figure 5 shows the results obtained if both hybrid multi-path SA or BH algorithm are applied. Now, the parameters are computed exactly (see Table 1), and a perfect agreement between the signals is observed.

We can also use this benchmark to assess the convergence and efficiency of the two proposed hybrid multi-path global optimization algorithms. In Figures 6 and 7, we show a comparison of the convergence of $\mathrm{SA_M}$ and $\mathrm{BH_M}$ algorithms, respectively, using different number of paths ranging from 1 to 16. At each temperature, the value of the cost function at the best point visited so far by the algorithm is shown. Note that the current state of the minimizer at each stage could be different to the referred best visited point owing to the stochastic nature of the SA and BH algorithms.

	#Threads	$\mathcal{T}$	#Func Evals	Cost Function
Simulated Annealing	1	1	101	3.25×10^{-2}
		0.48	3.61×10^{3}	2.31×10^{-2}
		10^{-4}	3.92×10^{5}	1.11×10^{-3}
	2	1	201	3.51×10^{-2}
		0.48	7.21×10^{3}	2.16×10^{-2}
		10^{-4}	7.83×10^{5}	4.58×10^{-4}
	4	1	401	2.38×10^{-2}
		0.48	1.44×10^{4}	1.82×10^{-2}
		10^{-4}	1.57×10^{6}	9.11×10^{-5}
	8	1	801	1.73×10^{-2}
		0.48	2.88×10^{4}	1.24×10^{-2}
		10^{-4}	3.13×10^{6}	1.33×10^{-4}
	16	1	1601	7.88×10^{-3}
		0.48	5.76×10^{4}	7.20×10^{-3}
		10^{-4}	6.27×10^{6}	3.50×10^{-5}
Basin Hopping	#Threads	$\mathcal{T}$	#Func Evals	Cost Function
	1	1	82	2.28×10^{-2}
		0.48	1.72×10^{3}	1.47×10^{-2}
		10^{-4}	1.96×10^{5}	6.33×10^{-4}
	2	1	103	2.29×10^{-2}
		0.48	3.48×10^{3}	1.24×10^{-2}
		10^{-4}	3.15×10^{5}	1.50×10^{-3}
	4	1	285	9.83×10^{-3}
		0.48	6.99×10^{3}	2.71×10^{-3}
		10^{-4}	6.49×10^{5}	1.33×10^{-4}
	8	1	369	2.08×10^{-2}
		0.48	1.30×10^{4}	7.04×10^{-4}
		10^{-4}	1.38×10^{6}	1.04×10^{-4}
	16	1	825	1.17×10^{-2}
		0.48	2.51×10^{4}	3.75×10^{-4}
		10^{-4}	2.67×10^{6}	9.92×10^{-6}

Table 2: Parallel SA (SA_M) vs. parallel BH (BH_M). The column labeled as "Cost Function" shows the value of the cost function at the best point visited so far by the minimization algorithm.

Number of cores	1	2	4	8	16
Time (seconds)	872.64	1640.57	3035.07	5493.48	9338.92
Speedup	1	1.88	3.47	6.30	10.70

Table 3: Multi-path BH_{16}: speedup using multi-CPU implementation.

In Table 2 we show the convergence of the multi-path algorithms when increasing the number of paths. The convergence speed is shown in terms of the

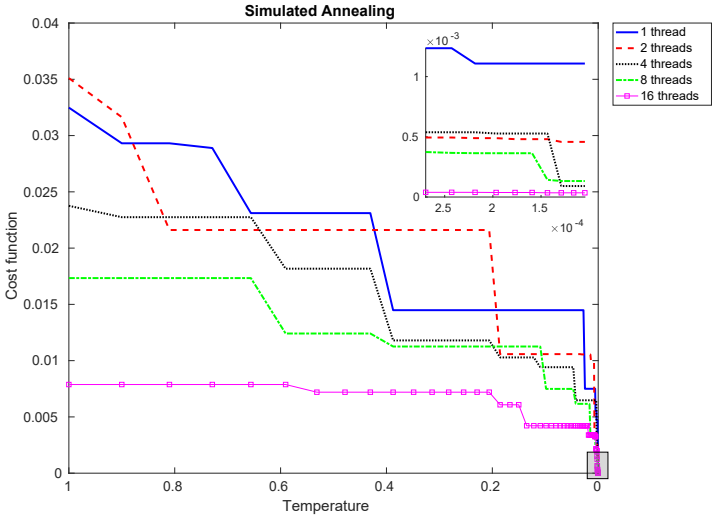


Figure 6: Convergence of multi-path SA, with 1,2,4,8 and 16 search paths.

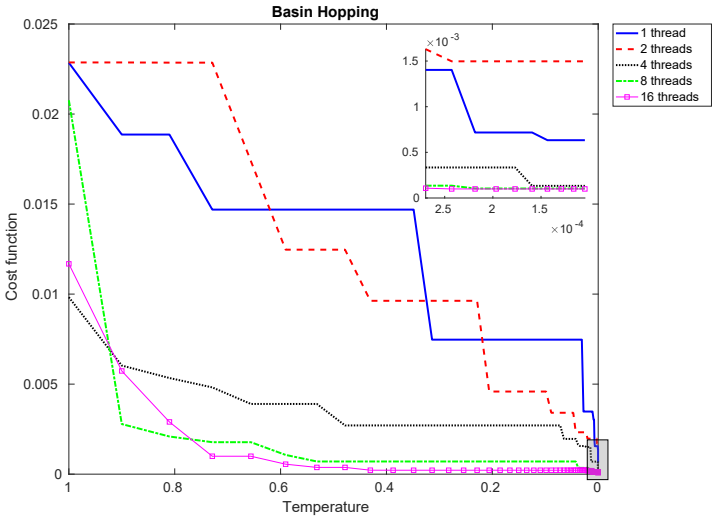


Figure 7: Convergence of multi-path BH, with 1,2,4,8 and 16 search paths.

	Parameters			
Gauges	r	θ	n	Cost func.
$G1$-$G2$-$G3$-$G4$	0.55	12^o	2×10^{-4}	9.923×10^{-6}
$G3$-$G4$	5.493439×10^{-1}	11.8404944^o	2.045592×10^{-4}	6.234×10^{-4}
$G4$	5.529343×10^{-1}	11.2507678^o	2.132503×10^{-4}	1.648×10^{-3}

Table 4: Obtained values of the parameters and value of cost function.

number of function evaluations performed by the algorithm. This number of evaluations is shown at different levels of temperatures in the annealing process, $\mathcal{T} = 1, 0.48, 10^{-4}$, and for different number of search paths, ranging from 1 to 16. The computing time of each evaluation for this test is 2.8 seconds in our hardware configuration. We want to remind that when doing more than one search, the searches are distributed among the number of CPU cores, and that for $\mathrm{BH_M}$ this number of evaluations include the three extra evaluations performed for computing the gradients. In Table 3 we show the parallel computational efficiency, in terms of the speedup, when using multiple cores for performing 16 search paths, with a number of threads ranging from 1 to 16.

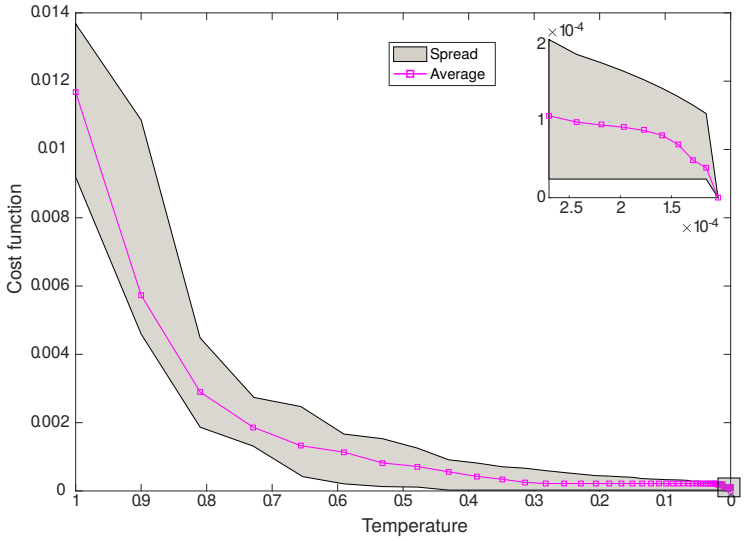


Figure 8: Evolution of the cost function for 20 optimization experiments. The gray shade denotes the spread (the range of maximum-minimum cost) and the squared line the average of the 20 experiments.

Next, we check the convergence of the algorithm to the global optimum when a lower number of measure points is used. We made the experiment of considering the time series of the free surface only at tide-gauge G4, or only at

r	θ	n	Cost func.
0.35	12	0.0002	1.19×10^{-5}
0.35	12	0.0004	9.34×10^{-5}
0.35	25	0.0002	2.90×10^{-5}
0.35	25	0.0004	6.53×10^{-5}
0.35	37	0.0002	3.50×10^{-5}
0.35	37	0.0004	6.41×10^{-5}
0.55	12	0.0002	1.89×10^{-5}
0.55	12	0.0004	8.75×10^{-5}
0.55	25	0.0002	6.89×10^{-5}
0.55	25	0.0004	2.75×10^{-5}
0.55	37	0.0002	2.06×10^{-5}
0.55	37	0.0004	8.33×10^{-5}
0.75	12	0.0002	5.49×10^{-5}
0.75	12	0.0004	8.61×10^{-5}
0.75	25	0.0002	4.81×10^{-5}
0.75	25	0.0004	8.96×10^{-5}
0.75	37	0.0002	4.62×10^{-5}
0.75	37	0.0004	2.21×10^{-5}

Table 5: Values of the cost function for several data assimilations.

tide-gauges G3-G4. In Table 4 we show the value of the cost function together with the obtained set of parameters using only data from G4, and the same information when calibrating against tide-gauges G3-G4. As expected, the value of the cost function is better when taking the four tide-gauges.

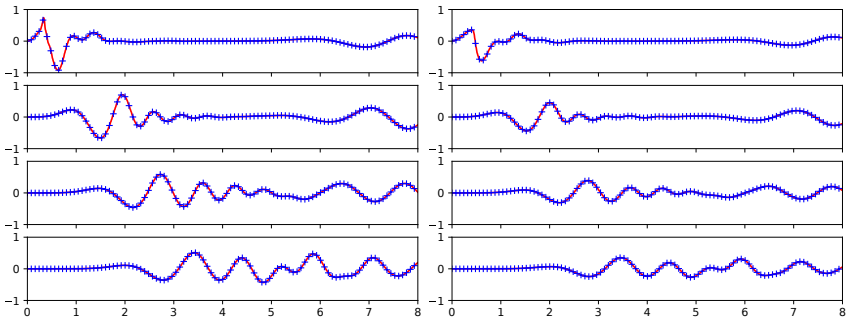
4.1.2. Synthetic test 2

Secondly, we ran a pool of 20 independent optimization experiments with our set up, each optimization starting from different initial parameter values, randomly chosen in the search domain, and each test used a different seed for the creation of the random numbers consumed by the algorithm in order to explore the search domain, i.e. each experiment performed a seek of the minimum from a different starting point along a different search path. Figure 8 shows the evolution of the cost function and its spread for the 20 optimization experiments. The spread is defined by the range of the maximum value of the cost function and its minimum in the set of the 20 optimizations at each temperature step. The average of the 20 realizations was also computed. More precisely at each temperature, the worse, the best, and the average of the best points visited by each one of the twenty minimizers up to the current temperature, are shown. All experiments show an asymptotic reduction of the values of the cost function toward the same zero value, and none of the optimizations ends up in a local minimum. Therefore, this study clearly shows that this stochastic approach (hybrid local-global optimization) is suitable to find the global minimum of a

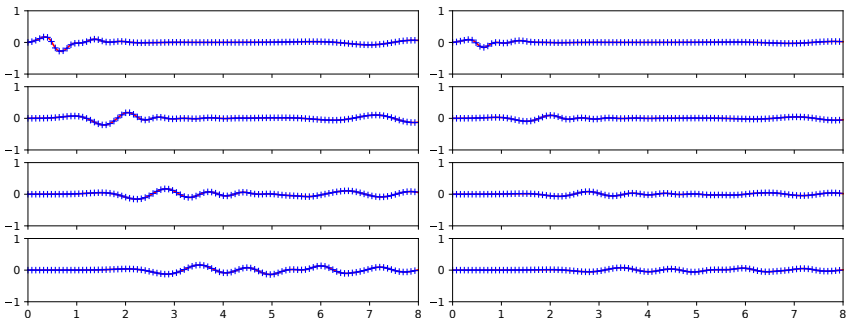
structurally complicated cost function.

4.1.3. Synthetic test 3

Finally, we sampled the search domain with 18 sets of parameters, for all of them, once more we generated the corresponding synthetic tests and performed a successful data assimilation, these experiments being summarized on Table 5. We note that with this pool of data we swimmingly calibrated the model to all types of waves varying from those with very high amplitudes to the flat ones, see Figure 9. Notice that this smoothing effect was obtained by increasing more and more the ratio of densities r and the Coulomb angle θ. Therefore, the issue of identifiability is accomplished for the very different types of possible waves.



(a) $r = 0.3$, $\theta = 12$, $n = 0.0002$. (b) $r = 0.5$, $\theta = 22$, $n = 0.0002$.



(c) $r = 0.6$, $\theta = 31$, $n = 0.0004$. (d) $r = 0.8$, $\theta = 39$, $n = 0.0004$.

Figure 9: Synthetic generated series vs calibrated ones. In red, target series that are a priori generated with a set of known parameters. In blue, simulated series obtained with the assimilated parameters achieved with the global optimizer.

4.2. Application to a laboratory test with real data

In this experiment we performed the data assimilation for a real situation where laboratory series of the free surface for four measure points were given.

The experiment was performed at École Centrale de Marseille (IRPHE), France, [82]. The positions of the measure buoys were once more 1.87, 2.87, 3.87 and 4.87 meters. The time series for these points are shown in Figure 10. These time series, together with the description of the experiment and some videos, are available in the web page [83].

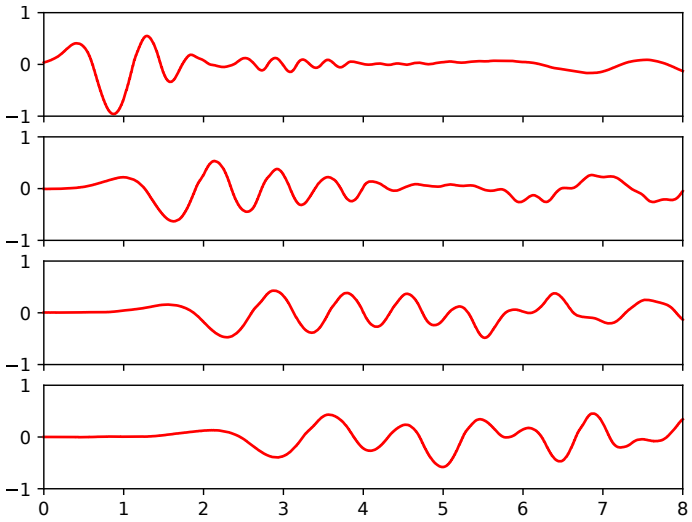


Figure 10: Series measured in laboratory experiments.

One more time, first we show that the results obtained with a multi-start algorithm are worse than those obtained with a hybrid multi-path algorithm. For example, if we apply a multi-start L-BFGS-B to this problem, the obtained solution does not match adequately the laboratory data (see Figure 11). This experiment corresponds with launching only one temperature step of $\mathrm{BH_M}$ with 32 paths, and the set of obtained parameters is $(r, \theta, n) = (4.632742 \times 10^{-01}, 9.48064720^{o}, 8.176018 \times 10^{-1})$, for which the value of the cost function is 1.321301×10^{-01}.

After global calibration with $\mathrm{BH_M}$, the results can be seen in Figures 12 and 13. The obtained values for the parameters are shown in Table 6 and the value of the cost function is 1.224355×10^{-1}.

In those figures we can see that with the calibrated set of parameters a good agreement in the signals amplitudes and pulses is obtained, between laboratory and simulated data. The approximation is even better up to the 4th second (see Figure 12). The matching is quite good at initial seconds, and it becomes worse as time evolves. Also we see a better agreement for the farthest tide-gauges, $G3$ and $G4$, and it becomes worse the closer we are to the the initial position of the landslide, close to tide-gauge $G1$. The amplitudes of the signal are very well captured by the model. The period (pulses, maximums and minima of the signal) is well captured for the last three tide-gauges until the fourth second,

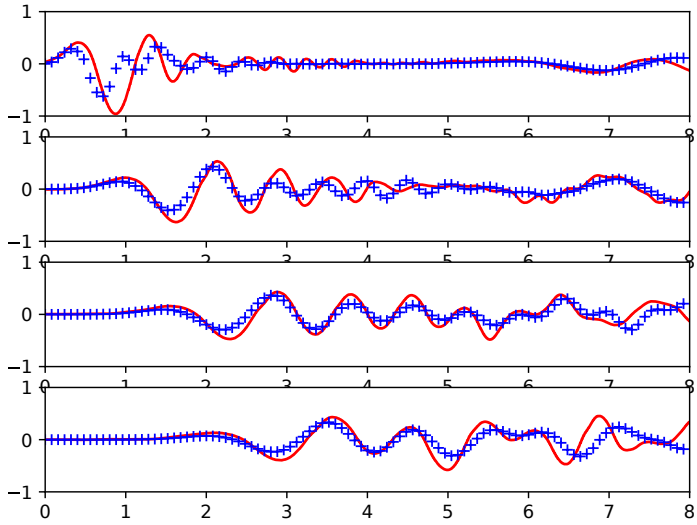


Figure 11: Multi start solution with 32 L-BFGS-B local searches: computed signals in blue and laboratory data in red.

and there is a little gap from that time on. The first tide-gauge is difficult to be captured by the model. Further investigation should be done. In fact, at this early stage, compaction and dilatancy effects are quite important, and they are not taken into account in the here considered landslide model. Therefore, a more accurate model for the landslide motion is needed to better simulate this early stages of the landslide motion.

Newly in this laboratory experiment we repeated the practice of using a lower number of measure points. We made the test of considering the free surface series only at tide-gauge $G4$, or only at tide-gauges $G3$-$G4$, the results can be seen in Table 6. The obtained error considering the four series until time $T = 8$ seconds, using the parameters calibrated with only the last tide-gauge $G4$, is 1.31589×10^{-1}. Besides, the obtained error considering the four series, using the parameters calibrated with only the last two tide-gauges $G3$-$G4$ is 1.24279×10^{-1}. The result is not too poor when considering only the measures of the last tide-gauge, nevertheless it is off course much better when considering $G3$-$G4$. Using the last two tide-gauges, the free surface series are quite close to the best obtained result using the four tide-gauges, and also interesting is the fact that the set of parameters gets closer to the ones obtained with the four tide-gauges.

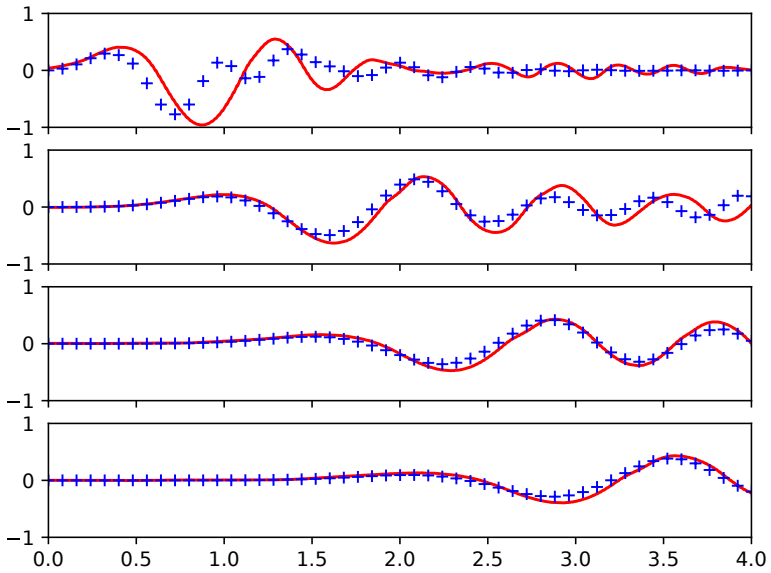


Figure 12: Laboratory series vs calibrated ones. Lab series in red, simulated series in blue. From top to bottom, free surface at tide-gauges $G1$, $G2$, $G3$ and $G4$.

	Parameters			
Gauges	r	θ	n	Cost func.
$G1$-$G2$-$G3$-$G4$	0.6501164	6.03510265^o	4.3690×10^{-4}	1.224355×10^{-1}
$G3$-$G4$	7.080885×10^{-1}	5.38770216^o	3.144702×10^{-4}	1.24279×10^{-1}
$G4$	7.633579×10^{-1}	5.16240342^o	2.397312×10^{-4}	1.31589×10^{-1}

Table 6: Obtained values of the parameters and value of cost function.

5. Conclusions

We have shown that hybrid multi-path global optimization algorithms can be suitable for solving the data assimilation problem for models of submarine avalanches.

Besides, we have assessed the identifiability of the model, if only data of the free surface is available, i.e. we have checked that the data assimilation problem is well posed when calibrating only against measures of the fluid free surface.

We have discussed that using a local optimizer or a multi-start technique produces poor results, and that the consideration of global optimization algorithms is more suitable for this kind of problems. We have also exhibited that

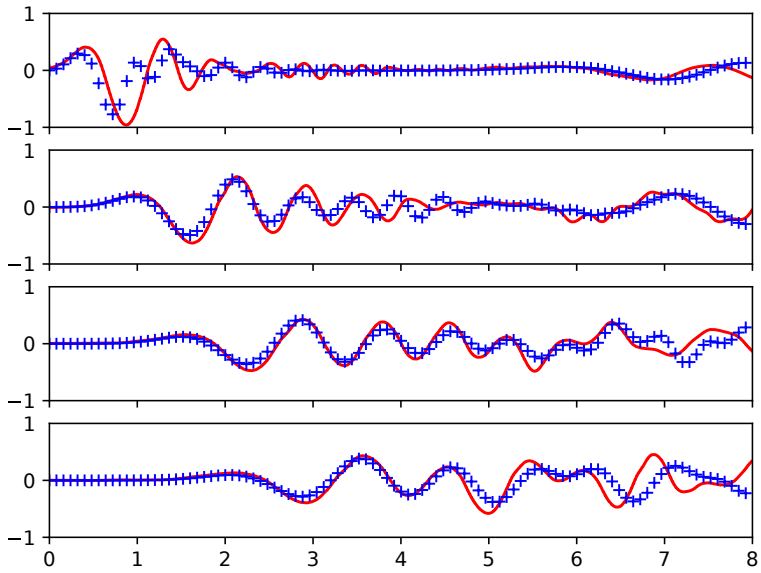


Figure 13: Laboratory series vs calibrated ones. Lab series in red, simulated series in blue. From top to bottom, free surface at tide-gauges $G1$, $G2$, $G3$ and $G4$.

the problem can be solved using gradient numerical optimization algorithms in the local part.

This calibration procedure/technique results also interesting because it allows to measure the quality of the model: the quality of two different models can be quantitative (not only qualitative) compared attending to the result of the calibration. It provides us with a machinery for comparing the good properties of different models. The one with the lowest minimum, can be quantitative said to better approximate the real physical problem.

The laboratory experiment is quite challenging. The obtained results look promising, although a perfect match between laboratory data and the calibrated model has not been achieved due to limitations of the underlying model. In any case, we have shown that the multi-path BH algorithm could be used to calibrate this kind of problems. Moreover, this opens the door to the use of this global optimization machinery for real problems, and in particular, for helping in developing better models for landslide tsunamis and assessing their precision and adjustment to the laboratory data.

6. Acknowledgements

The authors want to acknowledge the designers of the experiment ([11]), for making the data publicly available. The authors also wish to thank the anonymous reviewers for their through review of the article and their constructive advises.

This research has been financially supported by Spanish Government Ministerio de Economía y Competitividad through the research projects MTM2016-76497-R and MTM2015-70490-C2-1-R.

References

[1] S. T. Grilli, P. Watts, Tsunami Generation by Submarine Mass Failure. I: Modeling, Experimental Validation, and Sensitivity Analyses, J. Waterw Port Coast 131 (6) (2005) 283–297.

[2] I. V. Fine, A. B. Rabinovich, B. D. Bornhold, R. E. Thomson, E. A. Kulikov, The grand banks landslide-generated tsunami of november 18, 1929: preliminary analysis and numerical modeling, Marine Geology 215 (1) (2005) 45 – 57.

[3] A. Skvortsov, B. Bornhold, Numerical simulation of the landslide-generated tsunami in Kitimat Arm, British Columbia, Canada, 27 april 1975, J. Geophys. Res. 112.

[4] Abadie, S. M. and Harris, J. C. and Grilli, S. T. and Fabre, R., Numerical modeling of tsunami waves generated by the flank collapse of the Cumbre Vieja Volcano (La Palma, Canary Islands): Tsunami source and near field effects, Journal of Geophysical Research: Oceans 117 (C5).

[5] J. Horrillo, A. Wood, G.-B. Kim, A. Parambath, A simplified 3-D Navier-Stokes numerical model for landslide-tsunami: Application to the Gulf of Mexico, Journal of Geophysical Research: Oceans 118 (12) (2013) 6934–6950.

[6] S. Assier Rzadkiewicz, C. Mariotti, P. Heinrich, Numerical simulation of submarine landslides and their hydraulic effects, Journal of Waterway, Port, Coastal and Ocean Engineering 123 (4) (1997) 149–157.

[7] G. Ma, J. T. Kirby, F. Shi, Numerical simulation of tsunami waves generated by deformable submarine landslides, Ocean Modelling 69 (2013) 146–165.

[8] R. M. Iverson, The physics of debris flows, Reviews of Geophysics 35 (3) (1997) 245–296.

[9] S. B. Savage, K. Hutter, The motion of a finite mass of granular material down a rough incline, Journal of Fluid Mechanics 199 (1989) 177–215.

[10] E. Fernández-Nieto, F. Bouchut, D. Bresch, M. C. Díaz, A. Mangeney, A new Savage-Hutter type model for submarine avalanches and generated tsunami, Journal of Computational Physics 227 (16) (2008) 7720–7754.

[11] G. Ma, J. T. Kirby, T.-J. Hsu, F. Shi, A two-layer granular landslide model for tsunami wave generation: Theory and computation, Ocean Modelling 93 (2015) 40–55.

[12] E. Fernández-Nieto, M. Parisot, Y. Penel, J. Sainte-Marie, A hierarchy of dispersive layer-averaged approximations of Euler equations for free surface flows, Communications in Mathematical Sciences 16 (5) (2018) 1169–1202.

[13] J. Lions, Optimal control of systems governed by partial differential equations, Spinger Verlag, 1971.

[14] Kalnay, E., Atmospheric modeling, data assimilation and predictability, Cambridge: Cambridge University Press, 2003.

[15] Vrugt, Jasper A. and Diks, Cees G. H. and Gupta, Hoshin V. and Bouten, Willem and Verstraten, Jacobus M., Improved treatment of uncertainty in hydrologic modeling: Combining the strengths of global optimization and data assimilation, Water Resources Research 41 (1).

[16] K. Beven, A. Binley, The future of distributed models: Model calibration and uncertainty prediction, Hydrological Processes 6 (3) (1992) 279–298.

[17] Thiemann, M. and Trosset, M. and Gupta, H. and Sorooshian, S., Bayesian recursive parameter estimation for hydrologic models, Water Resources Research 37 (10) (2001) 2521–2535.

[18] Vrugt, Jasper A. and Bouten, Willem and Gupta, Hoshin V. and Sorooshian, Soroosh, Toward improved identifiability of hydrologic model parameters: The information content of experimental data, Water Resources Research 38 (12) (2002) 48–1–48–13.

[19] Vrugt, Jasper A. and Gupta, Hoshin V. and Bouten, Willem and Sorooshian, Soroosh, A Shuffled Complex Evolution Metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters, Water Resources Research 39 (8) (2003) .

[20] P. O. Yapo, H. V. Gupta, S. Sorooshian, Multi-objective global optimization for hydrologic models, Journal of Hydrology 204 (1) (1998) 83 – 97.

[21] Vrugt, Jasper A. and Gupta, Hoshin V. and Bouten, Willem and Sorooshian, Soroosh, A Shuffled Complex Evolution Metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters, Water Resources Research 39 (8).

[22] X. Yin, B. Wang, J. Liu, X. Tan, Evaluation of conditional non-linear optimal perturbation obtained by an ensemble-based approach using the Lorenz-63 model, Tellus A: Dynamic Meteorology and Oceanography 66 (1) (2014) 22773.

[23] Yuan, Shijin and Zhang, Huazhen and Li, Mi and Mu, Bin, CNOP-P-based parameter sensitivity for double-gyre variation in ROMS with simulated annealing algorithm, Journal of Oceanology and Limnology 37 (3) (2019) 957–967.

[24] D. B. Haidvogel, H. G. Arango, K. Hedstrom, A. Beckmann, P. Malanotte-Rizzoli, A. F. Shchepetkin, Model evaluation experiments in the north atlantic basin: Simulations in nonlinear terrain-following coordinates, Dyn. Atmos. Oceans 32 (2000) 239–281.

[25] M. Mu, W. Duan, Q. Wang, R. Zhang, An extension of conditional nonlinear optimal perturbation approach and its applications, Nonlinear Processes in Geophysics 17 (2) (2010) 211–220.

[26] C. Sánchez-Linares and M. de la Asunción and M.J. Castro and S. Mishra and J. Šukys, Multi-level Monte Carlo finite volume method for shallow water equations with uncertain parameters applied to landslides-generated tsunamis, Applied Mathematical Modelling 39 (23) (2015) 7211 – 7226.

[27] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by simulated annealing., Science 220 (1983) 671–680.

[28] E. Aarts, P. van Laarhoven, Statistical cooling: A general approach to combinatorial optimization problems., Philips J. of Research 40 (1985) 193–226.

[29] A. I. F. Vaz, L. N. Vicente, A particle swarm pattern search method for bound constrained global optimization, International J. of Computer Mathematics 39 (2) (2007) 197–219.

[30] A. I. F. Vaz, L. N. Vicente, PSwarm: a hybrid solver for linearly constrained global derivative-free optimization, Optim. Methods and Software 24 (4-5) (2009) 669–685.

[31] R. Storn, K. Price, Differential Evolution - A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, Journal of Global Optimization 11 (4) (1997) 341–359.

[32] R. Hooke, T. A. Jeeves, Direct search solution of numerical and statistical problems, J. ACM 8 (2) (1961) 212–229.

[33] J. A. Nelder, R. Mead, A simplex method for function minimization, Computer J. 7 (1965) 308–313.

[34] E. Polak, G. Ribière, Note sur la convergence de méthodes des directions conjuguées, Rev. Francaise Informat. Recherche Operationelle 3e Année 16 (1969) 35–43.

[35] C. G. Broyden, The convergence of a class of double rank minimization algorithms: 2. The new algorithm, IMA Journal of Applied Mathematics 6 (3) (1970) 222–231.

[36] R. Fletcher, A new approach to variable metric algorithms, Computer J. 13 (1970) 317–322.

[37] D. Goldfarb, A family of variable metric methods derived by variational means, Math. Comp. 24 (109) (1970) 23–26.

[38] D. F. Shanno, Conditioning of quasi-newton methods for function minimization, Math. Comp. 24 (111) (1970) 647–650.

[39] D. C. Liu, J. Nocedal, On the Limited Memory Method for Large Scale Optimization, Mathematical Programming B 45 (3) (1989) 503–528.

[40] R. H. Byrd, P. Lu, J. Nocedal, C. Zhu, A limited memory algorithm for bound constrained optimization, SIAM J. on Scientific Computing 16 (5) (1995) 1190–1208.

[41] D.H. Robertson and B.F. Brown and I.M. Navon, Determination of the Structure of Mixed Argon-Xenon Clusters Using a Finite-Temperature, Lattice-Based Monte-Carlo Method, Journal of Chemical Physics 90 (1989) 3221–3229.

[42] I.M. Navon and F.B. Brown and D.H. Robertson, A Combined Simulated-Annealing and Quasi-Newton-Like Conjugate Gradient Method for Determining The Structure of Mixed Argon-Xenon Clusters, Computers and Chemistry 14 (1990) 305–311.

[43] D. J. Wales, J. P. K. Doye, Global optimization by Basin-Hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 Atoms, J. Phys. Chem. A 101 (1997) 5111–5116.

[44] R. E. Wengert, A simple automatic derivative evaluation program, Commun. ACM 7 (8) (1964) 463–464.

[45] Automatic Differentiation Engine, TAPENADE, `https://www-sop.inria.fr/tropics/tapenade.html`, [Online; accessed 26-09-2019].

[46] D. Yan, J. Yafei, W. S. S. Y., Identification of Mannings Roughness Coefficients in Shallow Water Flows, Journal of Hydraulic Engineering 130 (6) (2004) 501–510.

[47] E. Bélanger, A. Vincent, Data assimilation (4D-VAR) to forecast flood in shallow-waters with sediment erosion, Journal of Hydrology 300 (1) (2005) 114–125.

[48] X. Lai, J. Monnier, Assimilation of spatially distributed water levels into a shallow-water flood model. Part I: Mathematical method and test case, Journal of Hydrology 377 (1) (2009) 1–11.

[49] R. Hostache, X. Lai, J. Monnier, C. Puech, Assimilation of spatially distributed water levels into a shallow-water flood model. Part II: Use of a remote sensing image of Mosel River, Journal of Hydrology 390 (3) (2010) 257–268.

[50] M. Honnorat, J. Monnier, F.-X. Le Dimet, Lagrangian data assimilation for river hydraulics simulations, Computing and Visualization in Science 12 (5) (2009) 235–246.

[51] E. Bernard, V. Titov, Evolution of tsunami warning systems and products, Phil. Trans. R. Soc. A 373.

[52] Y. Wang, K. Satake, T. Maeda, A. R. Gusman, Data assimilation with dispersive tsunami model: a test for the Nankai Trough, Earth, Planets and Space 70 (1) (2018) 131.

[53] Y. Wang, K. Satake, T. Maeda, A. R. Gusman, Green's Function-Based Tsunami Data Assimilation: A Fast Data Assimilation Approach Toward Tsunami Early Warning, Geophysical Research Letters 44 (20) (2017) 10,282–10,289.

[54] J. Li, D. Xiu, On numerical properties of the ensemble kalman filter for data assimilation, Computer Methods in Applied Mechanics and Engineering 197 (43) (2008) 3574 – 3583, stochastic Modeling of Multiscale and Multiphysics Problems.

[55] A. Narayan, Y. Marzouk, D. Xiu, Sequential data assimilation with multiple models, Journal of Computational Physics 231 (19) (2012) 6401 – 6418.

[56] Y. Yang, E. M. Dunham, G. Barnier, M. Almquist, Tsunami Wavefield Reconstruction and Forecasting Using the Ensemble Kalman Filter, Geophysical Research Letters 46 (2) (2019) 853–860.

[57] T. Takagi, K. Inamoto, M. Kawahara, Estimation of Wave Propagation using a Kalman Filter, International Journal of Computational Fluid Dynamics 9 (1) (1998) 77–84.

[58] Y. Wang, T. Maeda, K. Satake, M. Heidarzadeh, H. Su, A. F. Sheehan, A. R. Gusman, Tsunami Data Assimilation Without a Dense Observation Network, Geophysical Research Letters 46 (4) (2019) 2045–2053.

[59] A. R. Gusman, A. F. Sheehan, K. Satake, M. Heidarzadeh, I. E. Mulia, T. Maeda, Tsunami data assimilation of Cascadia seafloor pressure gauge records from the 2012 Haida Gwaii earthquake, Geophysical Research Letters 43 (9) (2016) 4189–4196.

[60] Maëlle Nodet, Variational assimilation of lagrangian data in oceanography, Inverse Problems 22 (1) (2006) 245–263.

[61] H. Tsushima, K. Hirata, Y. Hayashi, Y. Tanioka, K. Kimura, S. Sakai, M. Shinohara, T. Kanazawa, R. Hino, K. Maeda, Near-field tsunami forecasting using offshore tsunami data from the 2011 off the Pacific coast of Tohoku Earthquake, Earth, Planets and Space 63 (7) (2011) 56.

[62] Sumata, H. and Kauker, F. and Gerdes, R. and Köberle, C. and Karcher, M., A comparison between gradient descent and stochastic approaches for parameter optimization of a sea ice model, Ocean Science 9 (4) (2013) 609–630.

[63] Hiroshi Sumata and Frank Kauker and Michael Karcher and Rdiger Gerdes, Simultaneous Parameter Optimization of an Arctic Sea Ice-Ocean Model by a Genetic Algorithm, Monthly Weather Review 147 (6) (2019) 1899 – 1926.

[64] A. M. Ferreiro, J. A. García-Rodríguez, J. López-Salas, C. Vázquez, An efficient implementation of parallel Simulated Annealing algorithm in GPUs, J. of Global Optimization 57 (3) (2013) 863–890.

[65] A. M. Ferreiro, J. García-Rodríguez, L. Souto, C. Vázquez, Basin Hopping with synched multi L-BFGS local searches. Parallel implementation in multi-CPU and GPUs, Applied Mathematics and Computation 356 (2019) 282–298.

[66] A. M. Ferreiro, J. A. García-Rodríguez, J. G. López-Salas, C. Vázquez, SABR/LIBOR market models: Pricing and calibration for some interest rate derivatives, Applied Mathematics and Computation 242 (2014) 65 – 89.

[67] A. Mangeney, F. Bouchut, N. Thomas, J. P. Vilotte, M. O. Bristeau, Numerical modeling of self-channeling granular flows and of their levee-channel deposits, Journal of Geophysical Research: Earth Surface 112 (F2).

[68] M. Pirulli, A. Mangeney, Results of back-analysis of the propagation of rock avalanches as a function of the assumed rheology, Rock Mechanics and Rock Engineering 41 (1) (2008) 59–84.

[69] O. Pouliquen, Scaling laws in granular flows down rough inclined planes, Physics of Fluids 11 (3) (1999) 542–548.

[70] M. Brunet, L. Moretti, A. Le Friant, A. Mangeney, E. D. Fernández Nieto, F. Bouchut, Numerical simulation of the 30–45ka debris avalanche flow of Montagne Pelée volcano, Martinique: from volcano flank collapse to submarine emplacement, Natural Hazards 87 (2) (2017) 1189–1222.

[71] C. Escalante, T. Morales, M. Castro, Non-hydrostatic pressure shallow flows: GPU implementation using finite-volume and finite-difference scheme, Applied Mathematics and Computation 338 (2018) 631–659.

[72] C. Escalante, E. D. Fernández-Nieto, T. Morales de Luna, M. J. Castro, An Efficient Two-Layer Non-hydrostatic Approach for Dispersive Water Waves, Journal of Scientific Computing.

[73] M. Castro Díaz, E. Fernández-Nieto, A Class of Computationally Fast First Order Finite Volume Solvers: PVM Methods, SIAM Journal on Scientific Computing 34 (4) (2012) A2173–A2196.

[74] J. Adsuara, I. Cordero-Carrion, P. Cerda-Duran, M. Aloy, Scheduled relaxation Jacobi method: Improvements and applications, Journal of Computational Physics 321 (2016) 369–413.

[75] M. Locatelli, On the multilevel structure of global optimization problems, Computational Optimization and Applications 30 (2005) 5–22.

[76] M. Locatelli, F. Schoen, Global Optimization: Theory, Algorithms, and Applications, SIAM, MOS-SIAM Series on optimization, 2013.

[77] B. Addis, M. Locatelli, F.Schoen, Local optima smoothing for global optimizations, Optim. Methods and Software 20 (2005) 417–437.

[78] B. Addis, Global optimization using local searches, Phd. Thesis, Universitá degli studi di Firenze, 2004.

[79] R. H. Leary, Global optimization on funneling landscapes, Journal of Global Optimization 18 (2000) 367–383.

[80] W. L. Goffe, SIMANN: A Global Optimization Algorithm using Simulated Annealing, Studies in Nonlinear Dynamics & Econometrics 1 (1996) 1–9.

[81] C. Zhu, R. H. Byrd, P. Lu, J. Nocedal, Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-scale Bound-constrained Optimization, ACM Trans. Math. Softw. 23 (4) (1997) 550–560.

[82] S. T. Grilli, M. Shelby, O. Kimmoun, G. Dupont, D. Nicolsky, G. Ma, J. T. Kirby, F. Shi, Modeling coastal tsunami hazard from submarine mass failures: effect of slide rheology, experimental validation, and case studies off the us east coast, Natural Hazards 86 (1) (2017) 353–391.

[83] Landslide Tsunami Model Benchmarking Workshop, `http://www1.udel.edu/kirby/landslide/problems.html`, [Online; accessed 26-02-2019].

[84] J. Macías, M. J. Castro, S. Ortega, C. Escalante, J. M. González-Vida, Performance Benchmarking of Tsunami-HySEA Model for NTHMP's Inundation Mapping Activities, Pure and Applied Geophysics 174 (8) (2017) 3147–3183.

[85] I. M. Navon, Practical and Theoretical Aspects of Adjoint Parameter Estimation and Identifiability in Meteorology and Oceanography, Dynamics of Atmospheres and Oceans. Special Issue in honor of Richard Pfeffer 27 (1 - 4) (1998) 55 – 79.

[86] D. G. Cacuci, I. M. Navon, M. Ionescu-Bujor, Computational Methods for Data Evaluation and Assimilation, Chapman and Hall/CRC, 2013.