Computers and Fluids 209 (2020) 104649

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Performance analysis of Volna-OP2 – massively parallel code for tsunami modelling

Daniel Giles[a,*], Eugene Kashdan[a], Dimitra M. Salmanidou[b], Serge Guillas[b], Frédéric Dias[a,c]

[a] School of Mathematics and Statistics, University College Dublin, Dublin, Ireland
[b] Department of Statistical Science, University College London, London, UK
[c] Earth Institute, University College Dublin, Dublin, Ireland

A R T I C L E I N F O

Article history:
Received 16 January 2020
Revised 24 June 2020
Accepted 30 June 2020
Available online 4 July 2020

Keywords:
Tsunami modelling
Finite-volume scheme
Error analysis
GPUs

A B S T R A C T

The software package Volna-OP2 is a robust and efficient code capable of simulating the complete life cycle of a tsunami whilst harnessing the latest High Performance Computing (HPC) architectures. In this paper, a comprehensive error analysis and scalability study of the GPU version of the code is presented. A novel decomposition of the numerical errors into the dispersion and dissipation components is explored. Most tsunami codes exhibit amplitude smearing and/or phase lagging/leading, so the decomposition shown here is a new approach and novel tool for explaining these occurrences.

To date, Volna-OP2 has been widely used by the tsunami modelling community. In particular its computational efficiency has allowed various sensitivity analyses and uncertainty quantification studies. Due to the number of simulations required, there is always a trade-off between accuracy and runtime when carrying out these statistical studies. The analysis presented in this paper will guide the user towards an acceptable level of accuracy within a given runtime.

1. Introduction

The tsunami community relies heavily on mathematical and computational modelling to carry out hazard assessments and/or provide forecasts. There exists a rich literature and database of tsunami codes which have been developed to model tsunami dynamics. Different governing systems such as the Nonlinear Shallow Water Equations (NSWE) or Boussinesq equations have been tackled with a wide variety of numerical methods (Finite Difference, Finite Element and Finite Volume). The regions of applicability and limitations of these governing equations are discussed further in Section 2. A brief mention of related works is given as follows with an emphasis on the Finite Difference, Element and Volume discretisations, while it is acknowledged that other advanced computational techniques such as spectral, pseudo-spectral [1] and smoothed particle hydrodynamics methods have also been successfully applied to the governing systems of tsunami dynamics. An overview of the computational methods utilised in tsunami science can be found here [2].

The most popular governing system of equations are the NSWE. Codes which tackle these by using a finite difference discretisation include NOAA's MOST [3], COMCOT [4] and CENALT [5], while SELFE [6], TsunAWI [7], ASCETE [8] and Firedrake-Fluids [9] utilise a finite element discretisation. Furthermore, GeoClaw [10] and HySEA [11] make use of a finite volume discretisation. Codes such as FUNWAVE [12], COULWAVE [13] and Celerais [14] are capable of capturing the effect of physical dispersion and thus simulate the Boussinesq equations. Most of these previously mentioned codes utilise CPUs. While there has been a concerted effort by the community to develop codes which are capable of leveraging the rapid speed ups of GPUs [14–18], only HySEA and Volna-OP2 are tsunami codes which are capable of supporting clusters of GPUs, to the authors' knowledge.

Volna-OP2, which utilises a finite volume discretisation, has been used for the simulations of tsunami modelling by a number of research groups around the world since its first introduction in 2011 [19]. The driving force for developing the code was a need within the tsunami research community for a solver which was applicable for analysis of realistic tsunami events and aimed to aid operational tsunami research [19,20]. The code solves the depth-averaged NSWE in two horizontal dimensions (x, y) using modern numerical methods for solution of hyperbolic systems. Volna-OP2

* Corresponding author.
E-mail address: daniel.giles@ucdconnect.ie (D. Giles).

https://doi.org/10.1016/j.compfluid.2020.104649

can efficiently simulate the complete life of a tsunami from generation induced by bathymetry displacement, propagation and inundation onshore. It can be used for cases of a simplified bathymetry represented by a mathematical formula but also for the complex bathymetry and topography of the examined geographical region. Owing to the use of an unstructured triangular mesh, irregular bathymetric and topographic features can be efficiently captured and represented. The first operational use of Volna-OP2 for a realistic scenario was for the modelling of sliding and tsunami generation in the St. Lawrence estuary in Canada [21]. The code has been used to model various tsunamigenic episodes [22–24]; in several cases it has been used in conjunction with statistical modelling to perform comprehensive sensitivity analysis tests and uncertainty quantification [25–29].

Both the originally developed and the newly parallelised version of Volna-OP2 have been carefully validated against well known benchmarks available to the tsunami community [19,20]. However, in the present paper, we pay particular attention to the accuracy of the new GPU version of the code with special emphasis on dispersion and dissipation errors as well as its computational efficiency on a general purpose GPU. Gaining a deeper understanding of the numerical errors can lead the user towards more accurate real case simulations. The manuscript is organised as follows: in the next Section 2, we describe the mathematical model and numerical schemes implemented in the parallel version of the code. Section 3 is dedicated to an analytic benchmark used to explore the accuracy of the code and it is followed by the analysis of the dispersion and dissipation errors in Section 4. Section 5 discusses application of the code to real cases and the scalability of its GPU implementation. The paper is wrapped up by concluding remarks and perspective developments in Section 6.

2. Mathematical model and algorithms

The incompressible Euler equations with a free surface capture the full dynamics of tsunamis. However, from a computational perspective, these equations are often too expensive to solve. Thus, by utilising the fact that tsunamis exhibit very large wavelengths in comparison to the depth of the basin over which they propagate, one often reduces and simplifies the Euler equations. By introducing a nonlinear parameter and a dispersion parameter [30], and by carrying out asymptotic analysis, one can generate simpler systems of equations, such as the NSWE or Boussinesq variants. Each system of equations has its own region of applicability. The NSWE retain the first order nonlinear term while neglecting the dispersion terms. The regions of applicability and ultimately the role of dispersion are discussed in Glimsdal et al. [31].

Physical dispersion can play a role in tsunami dynamics, and in particular when dealing with landslide events. However, the downside of including the effects of dispersion is that the computational complexity of the system of equations is increased. The new parallelised version of Volna is built upon a domain-specific language (DSL) OP2 [32]. This DSL is designed for unstructured mesh calculations with explicit temporal schemes. Thus, implementing the implicit schemes necessary in modelling the dispersion terms is not feasible in Volna-OP2. However, for cases where the effect of dispersion can be neglected [31], this increased computational efficiency from the parallelised version is extremely beneficial.

2.1. Nonlinear shallow water equations

As stated by neglecting the effects of physical dispersion the NSWE yield:

$$\frac{\partial H}{\partial t} + \nabla \cdot (H\vec{u}) = 0, \tag{1}$$

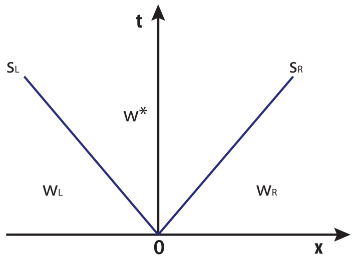


Fig. 1. The approximate Riemann fan.

$$\frac{\partial (H\vec{u})}{\partial t} + \nabla \cdot \left(H\vec{u} \otimes \vec{u} + \frac{g}{2}H^2\mathbf{I}\right) = -gH\nabla h, \tag{2}$$

where $H = (h + \eta)$ is the total water depth, described as the sum of the time-dependent bathymetry $h(x, y, t)$ and the free surface elevation $\eta(x, y, t)$, $\vec{u}(u, v)$ is the fluid velocity in the x and y horizontal directions, $\mathbf{I}$ is the identity matrix and g is the acceleration due to gravity. Provided that $H > 0$ the system is strictly hyperbolic. In the wet/dry transition the system starts to become non-hyperbolic since $H = 0$ in a dry region. To deal with that an algorithm that solves the shoreline Riemann problem developed by [33] is implemented in the code.

2.2. Spatial discretisation

A cell-centered Finite Volume numerical method is used for the spatial discretization in Volna-OP2 [19]. The numerical flux implemented in a numerical algorithm has to ensure that some standard conservation and consistency properties are satisfied: the fluxes from adjacent control volumes sharing an interface exactly cancel when summed and the numerical flux with identical state arguments reduces to the true flux of the same state.

In Volna-OP2 the Harten-Lax-van Leer (HLL) numerical flux was selected to ensure these conditions are met [19]. The HLL approximate Riemann solver was proposed by Harten et al. in 1983 and assumes a two-wave configuration for the exact solution [34]. The wave speed chosen according to [35] yields a very robust approximate Riemann solver. The Riemann solver models two waves that travel with speeds s_L and s_R, the larger signal velocity is represented by s_R and the smaller by s_L; three states are identified (Fig. 1). The subscripts R and L are used to represent the right and left cell values respectively. The intermediate state is denoted by $\vec{w}^*$, where $\vec{w}$ is a vector of the conserved variables (H, Hu, Hv). The numerical flux function of the scheme can be described by:

$$\phi_{HLL}\left(\vec{w}_L, \vec{w}_R\right) := \begin{cases} \vec{F}_L & \text{for } s_L \geq 0, \\ \vec{F}^* & \text{for } s_L < 0 \leq s_R, \\ \vec{F}_R & \text{for } s_R < 0, \end{cases} \tag{3}$$

where $\vec{w}_L, \vec{w}_R$ are the two interface states and $\vec{F}_{L,*,R}$ denotes the true flux at state $\vec{w}_{L,*,R}$ respectively. The right and left states are known. The intermediate state can be determined by applying the Rankine-Hugoniot conditions twice [19]. It then derives that:

$$\vec{w}^* = \frac{s_R\vec{w}_R - s_L\vec{w}_L - \left(\vec{F}_R - \vec{F}_L\right)}{s_R - s_L}, \tag{4}$$

$$\vec{F}^* = \frac{s_R\vec{F}_L - s_L\vec{F}_R + s_L s_R\left(\vec{w}_R - \vec{w}_L\right)}{s_R - s_L}. \tag{5}$$

In Volna-OP2 the wave speeds are computed as:

$$s_L = \min(u_{nL} - c_L, u_n^* - c^*), \tag{6}$$

$$s_R = \max(u_n^* + c^*, u_{nR} + c_R), \tag{7}$$

where $u_{nL} = \vec{u}_L \cdot \vec{n}_{LR}$, $u_{nR} = \vec{u}_R \cdot \vec{n}_{LR}$ and u_n^* and c^* are equal to:

$$u_n^* = \frac{1}{2}(u_{nL} + u_{nR}) + c_L - c_R, \tag{8}$$

$$c^* = \frac{1}{2}(c_L + c_R) - \frac{1}{4}(u_{nR} - u_{nL}), \tag{9}$$

where $c_R = \sqrt{gH_R}$ and $c_L = \sqrt{gH_L}$ are the gravity wave speeds for the right and left state of the system respectively and $\vec{n}_{LR}$ denotes the vector along the shared face between the right and left states. The shortcoming of the HLL scheme is that it cannot resolve isolated contact discontinuities. It can thus become quite dissipative.

2.3. Temporal discretisation

A Strong Stability-Preserving (SSP) method is used in conjunction with a Runge-Kutta method for the temporal discretization in Volna-OP2. In the current version of the code, the optimal second order two stage Runge-Kutta scheme SSP-RK(2,2) is used, with optimal Courant-Friedrichs-Lewy (CFL) condition equal to 1. The scheme is given as follows:

$$\begin{aligned} \vec{w}^{(1)} &= \vec{w}^{(n)} + \Delta t \mathcal{L}\left(\vec{w}^{(n)}\right), \\ \vec{w}^{(n+1)} &= \frac{1}{2}\vec{w}^{(n)} + \frac{1}{2}\vec{w}^{(1)} + \frac{1}{2}\Delta t \mathcal{L}\left(\vec{w}^{(1)}\right), \end{aligned} \tag{10}$$

where $\mathcal{L}(\vec{w})$ is defined as the finite volume space discretization operator. The stability of the scheme is guaranteed if the CFL condition is satisfied. The Runge-Kutta scheme is very robust, especially in handling discontinuities. However, the scheme is both dissipative and numerically dispersive [36]. Dissipativity causes a leak of energy from the system while numerical dispersion leads to either phase lag or phase lead. A full explanation of these phenomena is given in Section 4.

2.4. Second order extension

The classical finite volume schemes are only first order accurate in space, which is insufficient for most modern computational simulations. Simulating a real tsunami case with a first order accurate scheme would require an unfeasible mesh resolution to obtain meaningful results. So in order to yield second order accuracy in space, a reconstruction technique is implemented. One must ensure that the scheme is total variation diminishing (TVD), i.e no artificial maxima and minima are introduced. Thus, within Volna-OP2 a MUSCL (Monotone Upstream-centered Scheme for Conservation Laws) scheme is implemented. The second order spatial accuracy is achieved by reconstructing the conserved variables on the cell interfaces. The reconstruction relies on calculating the gradient of a conserved variable over a cell and projecting the reconstructed value on the interface. Within Volna-OP2, a least squares gradient reconstruction and Barth-Jesperson limiter [37] are implemented. The reconstructed values given below (11) are then used in the numerical flux calculation:

$$\vec{w}\left(\vec{x_f}\right) = \vec{w}_K + \alpha_K \left(\nabla \vec{w}\right)_K \cdot \left(\vec{x_f} - \vec{x_0}\right), \tag{11}$$

where $\vec{w}$ is a vector of conserved variables, $\vec{w}(\vec{x_f})$ is the conserved variable evaluated at the interface, α_K is the cell specific conserved variable limiter, $(\nabla\vec{w})_K$ is the cell centred gradient and $\vec{x_f} - \vec{x_0}$ is a vector pointing from the cell-centre to the face centre. To avoid large gradients being calculated in the wet/dry region of the domain a threshold depth has been introduced to ensure that the code remains stable. When the depth goes below this threshold the scheme doesn't carry out a reconstruction. This threshold depth has been set to be $H_{\text{threshold}} = 10^{-6}$ m. However, it has been found that a conservative value can ensure greater stability, for example in Section 3, $H_{\text{threshold}} = 10^{-5}$ m. For the real case (Section 5), $H_{\text{threshold}} = 10^{-3}$ m. At present these values are found through a trial and error approach but more research is required on the optimisation of this threshold depth.

2.5. Boundary conditions - wall/solid boundary

For a full explanation on the treatment of boundary conditions the reader is referred to [19]. However, the case of a wall/solid boundary is given here. For all boundary conditions a ghost cell technique is used. This approach allows one to reconstruct the conserved variables on the boundary and thus preserve second order accuracy. Values of the conserved variables on the ghost cells are defined based on the type of boundary condition needed. In the following, cell L is defined to be inside and cell R (ghost cell) is outside of the domain. The boundary of the domain is the common edge between cell L and cell R.

For a wall/solid boundary, $\vec{u} \cdot \vec{n} = 0$, where $\vec{u}$ is the flow velocity and $\vec{n}$ is the normal vector to the boundary edge. To ensure that this is satisfied, the tangential ($\vec{u}^{\parallel}$) and normal ($\vec{u}^{\perp}$) velocities for the ghost cell are set to be equal and opposite to those of the interior cell respectively:

$$\begin{aligned} \vec{u}_R^{\perp} &= -\vec{u}_L^{\perp}, \\ \vec{u}_R^{\parallel} &= \vec{u}_L^{\parallel}. \end{aligned} \tag{12}$$

3. Benchmark test with analytical solution

The two dimensional case of a radially symmetric paraboloid is implemented following the analytic solution initially proposed by Thacker [38]. This solution is available in the SWASHES (Shallow-Water Analytic Solutions for Hydraulic and Environmental Studies) library [39]. The major aim of SWASHES is to aid numerical modellers to validate shallow water equation solvers. The oscillatory motion of the paraboloid is described by a periodic solution in which damping is assumed to be negligible. The morphology of the domain is a paraboloid of revolution given by:

$$z(r) = -h_0\left(1 - \frac{r^2}{\alpha^2}\right), \tag{13}$$

where $r = \sqrt{x^2 + y^2}$ for each $(x, y) \in [-\frac{L}{2}, \frac{L}{2}] \times [-\frac{L}{2}, \frac{L}{2}]$, where L is the length of the domain, h_0 is the water depth at the central point of the domain when the shoreline elevation is zero and α is the horizontal distance from the central point to the shoreline with zero elevation (Fig. 2).

The free surface elevation $h(r, t)$ and the velocity components $u(x, y, t)$ and $v(x, y, t)$ are then given by:

$$h(r,t) = h_0\left\{\frac{\sqrt{1-A^2}}{1 - A\cos(\omega t)} - 1 - \frac{r^2}{\alpha^2}\left[\frac{1-A^2}{(1 - A\cos(\omega t))^2} - 1\right]\right\} - z(r), \tag{14}$$

$$u(x,y,t) = \frac{1}{1 - A\cos(\omega t)}\left[\frac{\omega A \sin(\omega t)}{2}\left(x - \frac{L}{2}\right)\right], \tag{15}$$

$$v(x,y,t) = \frac{1}{1 - A\cos(\omega t)}\left[\frac{\omega A \sin(\omega t)}{2}\left(y - \frac{L}{2}\right)\right], \tag{16}$$

where $\omega = \sqrt{8gh_0}/\alpha$ is the frequency, r_0 is the distance from the central point of the domain to the initial shoreline location and

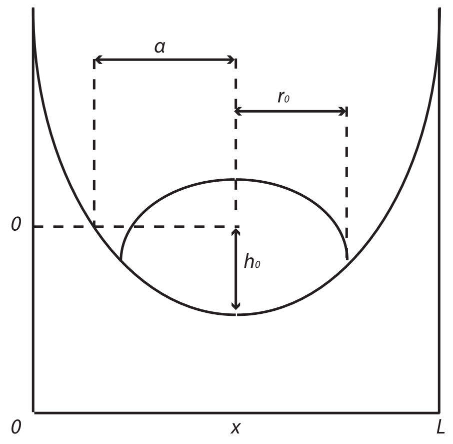


Fig. 2. Geometry of the domain used for the analytic solution following [39].

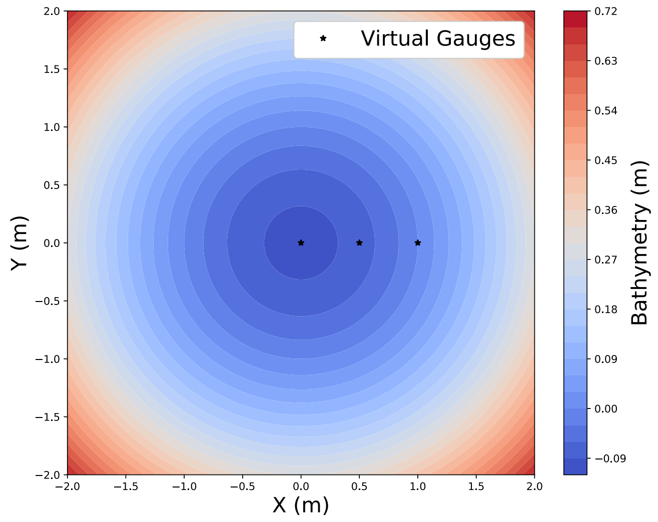


Fig. 3. Plot of the bathymetry (parabolic bowl) using the parameters outlined by Delestre [39]. The colour coding matches the height of the bathymetry. The locations of the virtual gauges are marked by the black stars.

$A = (\alpha^2 - r_0^2)/(\alpha^2 + r_0^2)$. To model the solution we follow the values proposed in Delestre [39], where $\alpha = 1$ m, $r_0 = 0.8$ m, $h_0 = 0.1$ m, $L = 4$ m, and $T = 3(2\pi/\omega)$.

We record the free surface elevation in three positions $(x_1, y_1) = (0, 0)$m (centre of the domain), $(x_2, y_2) = (0.5, 0)$m, and $(x_3, y_3) = (1, 0)$m (shoreline). Fig. 3 highlights a top-down view of the bathymetry and the locations of the wave gauges. In numerical simulations with Volna-OP2 we model the free surface elevation at the three positions with various spatial resolutions as a function of time up to $t_{fin} = 10$ s. An analytic solution at time $t = 0$ is chosen as an initial condition and $\Delta t = 0.45\Delta x$ for the simulations. This was chosen as it was found to be stable for all the mesh resolutions.

The results of the numerical simulations are shown in (Figs. 4–6). It is noted that at the shoreline position (Fig. 6) we see a discrepancy between the numerical and analytical solutions. For the gauge points Volna-OP2 outputs the water height referenced to the resting water level. When the water level runs down and the water height becomes zero, the surface elevation in Volna-OP2 is set to a threshold value (equal to the topographic height) while the analytical solution allows for unfeasible negative surface elevations to be produced. The initialisation step of the simulation can also be seen in (Fig. 6) at $t_0 = 0$, where Volna-OP2 takes the analytical solution for the surface elevation and sets it equal to the threshold value.

The finest mesh ($\Delta x = \Delta y = 0.006$ m) yields a representation closest to the surface elevation given by the analytic solution. The discrepancies between the meshes can only be seen by zooming in on the plots (Fig. 7). Focusing on the centre of the domain, we plot the absolute difference between the analytic and the numerical free surface elevation over time (Fig. 8). The centre point is chosen as it remains 'wet' for the whole simulation. This negates the possibility of errors due to incomparable solutions at wet/dry points. Further, an investigation into the errors obtained at points surrounding the centre yielded results similar to the results at the centre, but with the absolute error being less. This can be explained by the fact that the set up is radially symmetric, and the errors propagate towards and coincide at the centre point.

Turning to (Fig. 8), the numerical error is always larger for the coarser meshes. There appears to be a temporal regularity in the occurrence of the large spikes. We will return to this point in Section 4. In order to gain an idea on the numerical order of the scheme, a convergence study was carried out. The L_∞ norm is calculated at 10 s over the whole 2D domain, thus including the wetting/drying points, for the various mesh resolutions (0.048–0.003)m and then plotted versus the characteristic mesh size (Fig. 9). As the solution at $t = 10$ s has run up the sides of the parabolic basin, no false errors with respect to negative surface elevations from the analytical solution are calculated for the wet/dry points.

As the absolute difference between the numerical and analytical solution decreases with mesh resolution (Fig. 8) and the slope of the L_∞ norm, calculated for the whole 2D domain and plotted against the mesh size (Fig. 9), approximately equals 2, the errors of the numerical scheme are found to be of the order of $O(h^2)$. Thus, the results shown in (Figs. 8 and 9) highlight that the scheme is second order accurate in space. However, the role of numerical dispersion and dissipation has not been explored and they should be accounted for when running long time tsunami simulations. We come back to this discussion in Section 4.

To check the temporal discretization error we keep the mesh size constant at $\Delta x = \Delta y = 0.006$ m and vary the time step by adjusting the constant connecting the spatial and temporal resolutions. We choose three values: $\Delta t = (0.333, 1, 1.2)\Delta x$ and run the test for a longer time $t_{fin} = 100$ s. The results of the simulations are shown in Figs 10 and 11.

Changes in time-step within the stability limits while keeping the same spatial resolution almost do not affect the accuracy of the solution (Fig. 10). This is highlighted in Fig. 11 where the plots of the numerical solution with various time steps overlap each other. However, comparing the subplots of Fig. 11 as time goes on, one observes a damping of the numerical signal and a phase shift. The Runge-Kutta scheme implemented in the code is dissipative and we could expect a leak of the energy from the system, thus we can expect the damping of the numerical signal. Reasons for the phase shift in the signal will be explained in Section 4. Overall, the results from this section show that the space discretisation has a strong influence on the solution, while reducing the time step has no visible effect.

4. Error analysis

The exact solution of the discretized equations satisfies a PDE which is generally different from the one to be solved. The original equation is replaced with the modified equation $Au^{n+1} = Bu^n$, or, in other words

$$\frac{\partial\omega}{\partial t} + \mathcal{L}\omega = 0 \quad \text{becomes} \quad \frac{\partial\omega}{\partial t} + \mathcal{L}\omega = \sum_{p=1}^{\infty} \alpha_{2p} \frac{\partial^{2p}\omega}{\partial x^{2p}} + \sum_{p=1}^{\infty} \alpha_{2p+1} \frac{\partial^{2p+1}\omega}{\partial x^{2p+1}}. \tag{17}$$

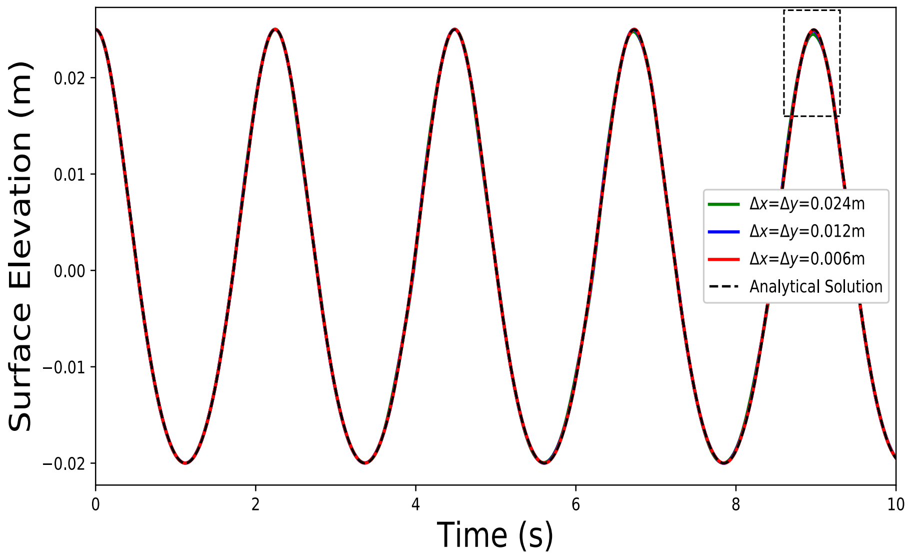


Fig. 4. Comparing numerical and analytic solutions in time at $(x_1, y_1) = (0, 0)$m (centre of the domain), where the analytic solution – black dashed, $\Delta x = \Delta y = 0.024$ m – green, $\Delta x = \Delta y = 0.012$ m – blue, and $\Delta x = \Delta y = 0.006$ m – red. In the numerical simulations, $\Delta t = 0.45\Delta x$. The dashed box is the boundaries of (Fig. 7). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

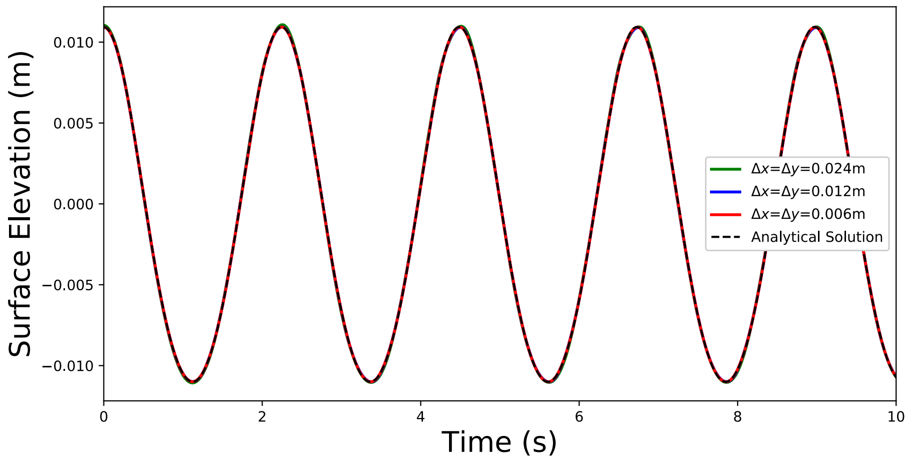


Fig. 5. Comparing numerical and analytic solutions in time at $(x_2, y_2) = (0.5, 0)$m, where the analytic solution – black dashed, $\Delta x = \Delta y = 0.024$ m – green, $\Delta x = \Delta y = 0.012$ m – blue, and $\Delta x = \Delta y = 0.006$ m – red. In the numerical simulations, $\Delta t = 0.45\Delta x$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

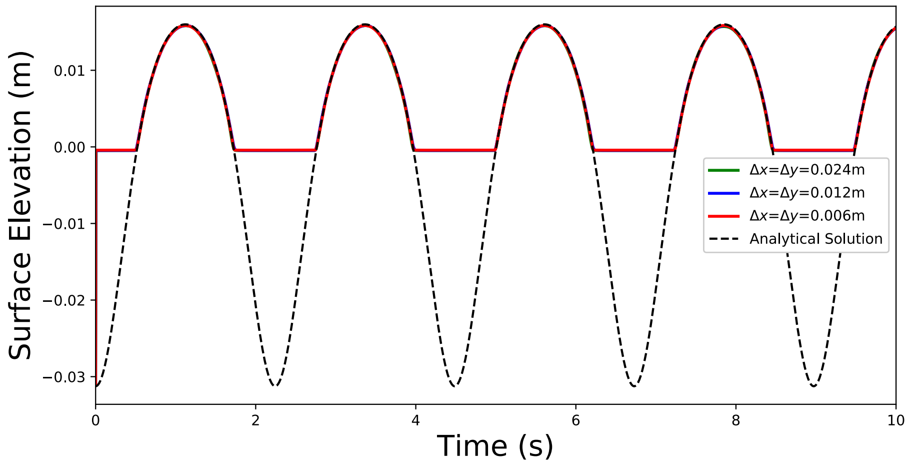


Fig. 6. Comparing numerical and analytic solutions in time at $(x_3, y_3) = (1, 0)$m (shoreline), where the analytic solution – black dashed, $\Delta x = \Delta y = 0.024$ m – green, $\Delta x = \Delta y = 0.012$ m – blue, and $\Delta x = \Delta y = 0.006$ m – red. In the numerical simulations, $\Delta t = 0.45\Delta x$. The shoreline forbids the numerical solutions to go below zero. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

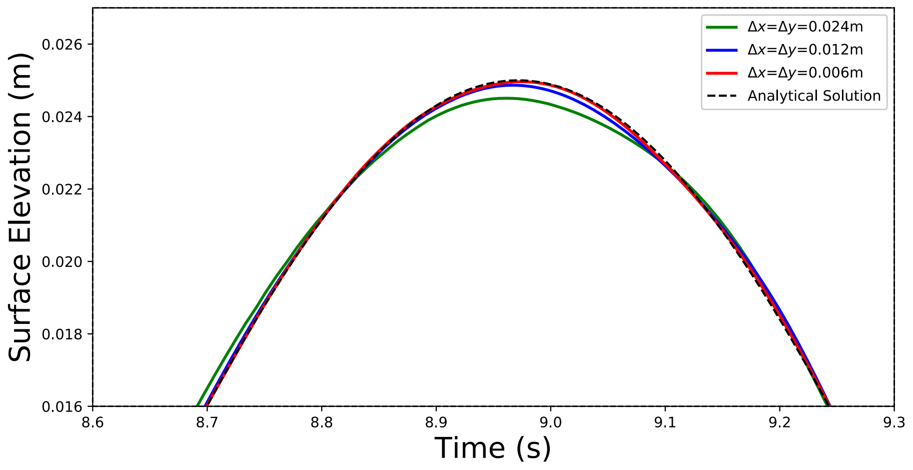


Fig. 7. Zoom in on the plot of the gauge at the centre of the domain $(x_1, y_1) = (0, 0)$m, one can observe the effect of the mesh size on the accuracy.

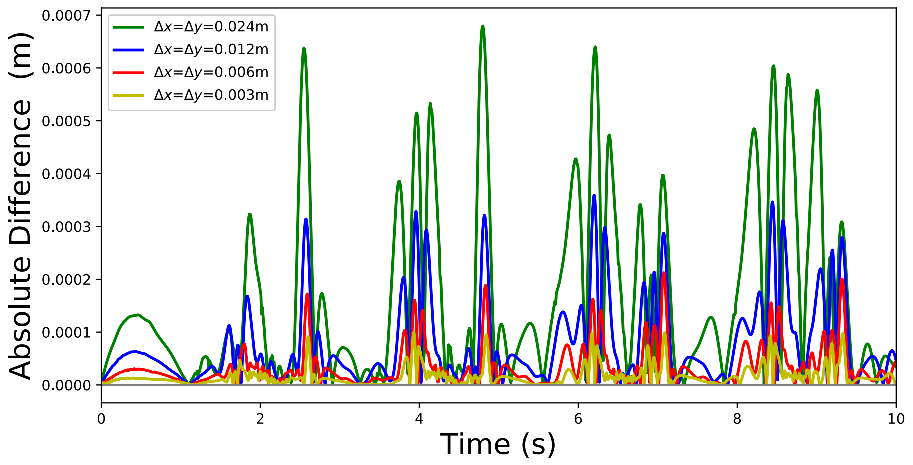


Fig. 8. Absolute difference between the analytic and the numerical free surface elevation over time at the centre of the domain $(x = y = 0)$m for spatial resolution: $\Delta x = \Delta y = 0.003$ m – yellow line, $\Delta x = \Delta y = 0.006$ m – red, $\Delta x = \Delta y = 0.012$ m – blue, $\Delta x = \Delta y = 0.024$ m – green; $\Delta t = 0.45\Delta x$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

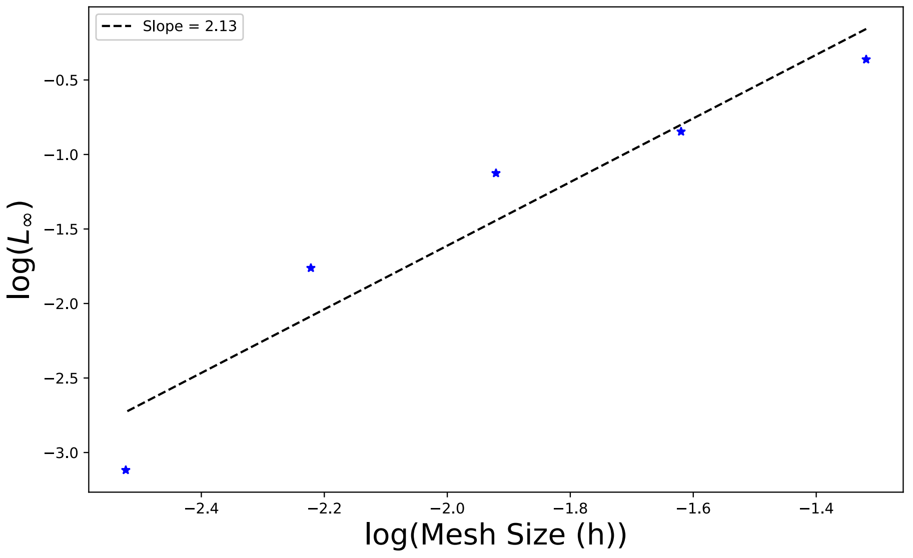


Fig. 9. Convergence rate of the L_∞ norm, calculated over the whole 2D domain at $t = 10$ s. As the slope approximates to the value 2, Volna-OP2 with the MUSCL extension is 2nd order accurate in space, i.e the error is $O(h^2)$.

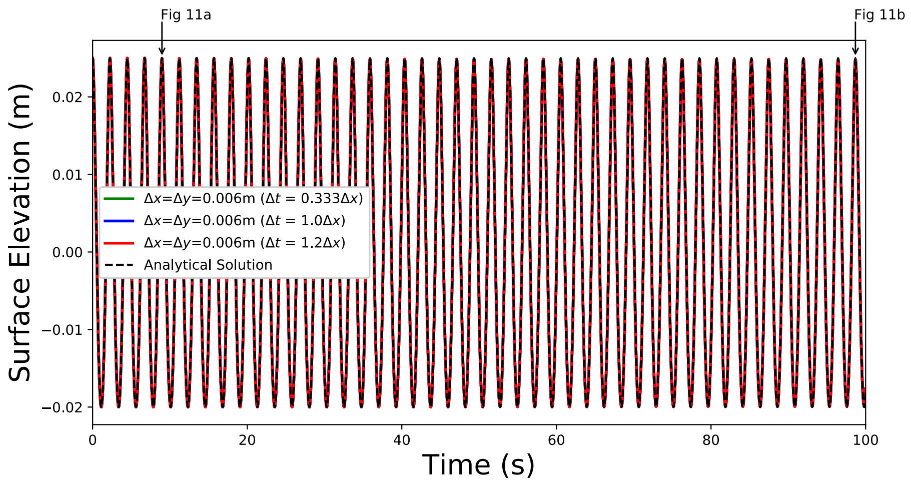


Fig. 10. Solution at the centre of the domain ($x = y = 0$ m) for various time steps with fixed spatial resolution $\Delta x = \Delta y = 0.006$ m: the analytic solution – black dash line, $\Delta t = 0.333\Delta x$ – green, $\Delta t = \Delta x$ – blue, and $\Delta t = 1.2\Delta x$ – red. Arrows point towards the areas highlighted in the subplots (Figs 11a and b). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The even-order derivatives on the right-hand side produce an amplitude error, or numerical dissipation. The odd-order derivatives on the right-hand side produce a wave-number-dependent phase error known as numerical dispersion. In the long time simulations, the numerical behaviour of the scheme largely depends on the role played by the dispersive and dissipative effects also known as "wiggles" (phase errors) and "smearing" (amplitude errors) respectively. A negative dispersion coefficient corresponds to phase lagging (i.e. harmonics travel too slowly), while positive dispersion coefficients yield phase leading with spurious oscillations occurring ahead of the wave.

According to the *Lax-Richtmyer Equivalence Theorem* [40], if a scheme has a truncation error of order (p, q) and the scheme is stable, then the difference between the analytic solution and the numerical solution in an appropriate norm is of the order $(\Delta t)^p + h^q$ for all finite time. It has been observed numerically (see Fig. 9) that the numerical solution is second order accurate. Taking into account that the time step is proportional to the spatial resolution that we call h, we can write that the total error is of the order $O(h^2)$. To analyse the role played by the dissipation and dispersion errors, we rewrite the error as

$$E(t) = \sum_{p=2}^{\infty} C_{p-1}(t) h^p, \tag{18}$$

where $C_{p-1}(t)$ incorporates all the constants included in the error formula. We choose the three leading terms of this expansion:

$$E(t) \approx C_1(t)h^2 + C_2(t)h^3 + C_3(t)h^4. \tag{19}$$

The first term in this expansion corresponds to the truncation error (also the leading dissipation error). It is followed by the leading dispersion and the secondary dissipation error terms. We assume that the remaining terms are significantly smaller and can be neglected. Next we go back to the simulations with various spatial resolution discussed in Section 3. For each of the three grids, we have the absolute error as a function of time. If we define the spatial resolution $h = 0.024$ m, two other grids have the resolution $h/2$ and $h/4$. The system of equations has the form:

$$\sum_{p=2}^{4} C_{p-1}(t) \frac{h^p}{\left(2^{k-1}\right)^p} = E_k(t), \quad k = 1, 2, 3, \tag{20}$$

and its solution is

$$C_1(t) = \frac{E_1(t) - 24E_2(t) + 128E_3(t)}{3h^2}, \tag{21}$$

$$C_2(t) = -\frac{2[E_1(t) - 20E_2(t) + 64E_3(t)]}{h^3}, \tag{22}$$

$$C_3(t) = \frac{8[E_1(t) - 12E_2(t) + 32E_3(t)]}{3h^4}. \tag{23}$$

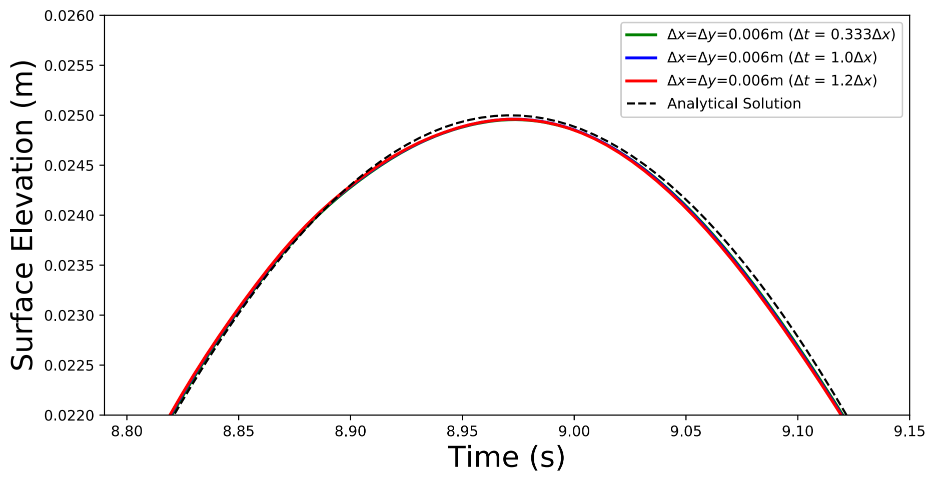
Figs. 12–14 show the composition of the total error for each of the grids. In order to give an idea on the temporal occurrence of the errors, the surface elevation at the center is scaled by the maximum value for the errors and plotted on the same figure.

In all the error decompositions, the components tend to cancel each other, which results in the total error being less than the individual components. However, the leading dispersion error is a dominant component for all the total errors. The leading dispersion error exhibits large negative spikes. This points towards phase lagging, particularly when the surface elevation is transitioning between negative and positive values (or vice-versa) at the center of the domain. These large negative spikes in the leading dispersion error coincide with positive spikes in either the truncation or 4th order dissipative errors. For finer grids the truncation error (leading dissipation error) plays a dominant role. For the coarser grid the negative spikes are offset by the 4th order dissipative error (Fig. 12). Despite this cancellation, we still see the largest total errors coinciding with these large spikes in the leading dispersion error. The main culprit for this is the radially symmetric nature of the domain. The errors (dominated by the phase lagging) from surrounding points coalesce at the centre when the surface elevation transitions through the 0 level.

This error analysis is important when considering real cases – see Section 5 – as any error could be dominated by either the leading dispersion or dissipation terms. However, the appearance of numerical dispersion could compensate for the fact that physical dispersion is neglected in the nonlinear shallow water equations, as shown by Burwell et al. [41].

5. Real cases

In this section we discuss an actual tsunami simulation carried out on a general purpose GPU (NVIDIA® Tesla® V100 card, with 5120 CUDA cores, 16GB max memory size). The domain size is 800 × 1000 km and the physical simulation time is 1 h. The simulation corresponds to a hypothetical scenario of edge volume collapse (765 km^3) at the Rockall Bank Slide Complex. A geophysical study of the event taking into account volumetric, rheolog-



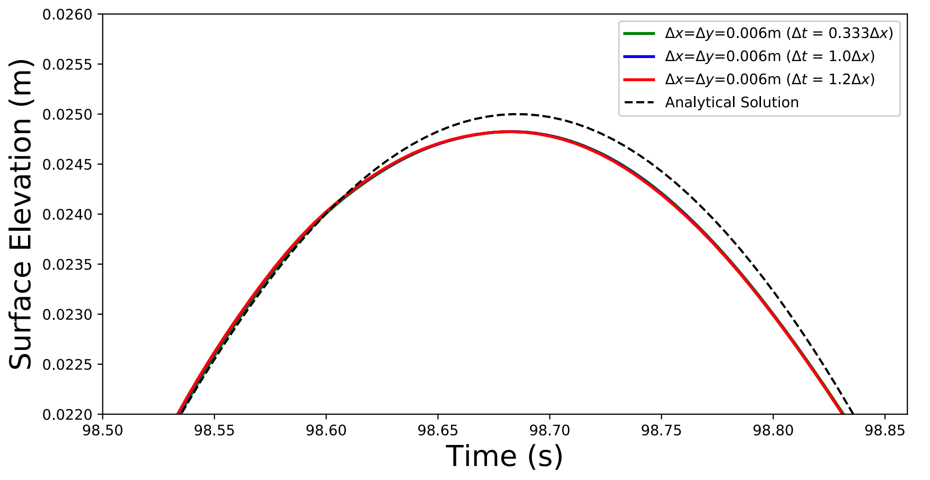


Fig. 11. Zoom in on Fig. 10 for two different time windows (Top (a): $t = 8.79 - 9.15$ s and Bottom (b): $t = 98.5 - 98.86$ s). The difference between the numerical solution with various time steps is not noticeable in each subplot. However, one can see a damping and phase shifting of the numerical signal when comparing the bottom (11 b) and top subplot (11 a).

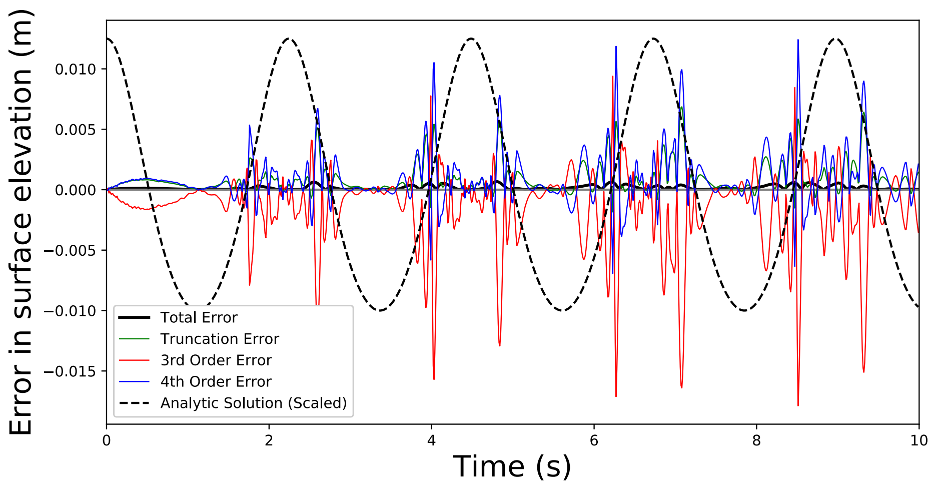


Fig. 12. Decomposition of the absolute error at the centre of the domain ($x = y = 0$ m) for a spatial grid with resolution of $\Delta x = \Delta y = 0.024$ m and CFL=0.45. The colours correspond to: the truncation error – green, the third order error – red, the fourth order error – blue, the total error – black. The black dashed line is a scaled plot of the surface elevation at the centre of the domain over time. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

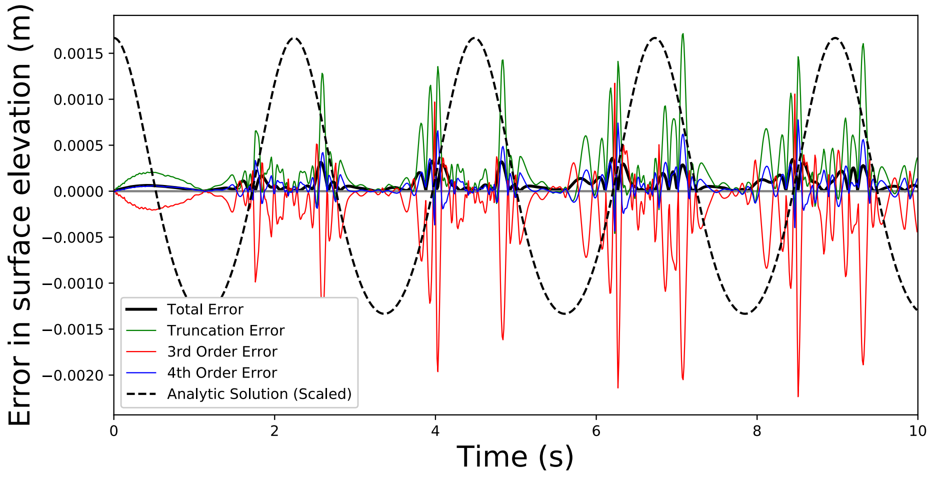


Fig. 13. Decomposition of the absolute error at the centre of the domain ($x = y = 0$ m) for a spatial grid with resolution of $\Delta x = \Delta y = 0.012$ m and CFL=0.45. The colours correspond to: the truncation error – green, the third order error – red, the fourth order error – blue, the total error – black. The black dashed line is a scaled plot of the surface elevation at the centre of the domain over time. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

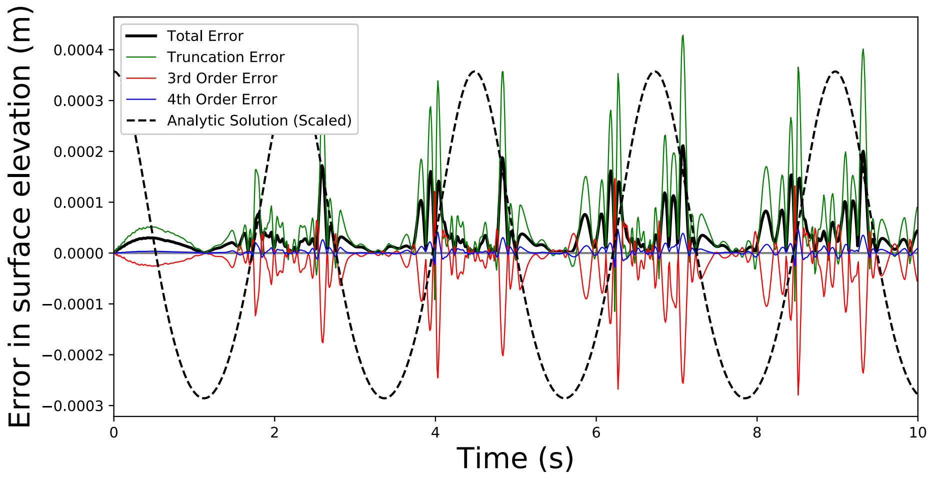


Fig. 14. Decomposition of the absolute error at the centre of the domain ($x = y = 0$ m) for a spatial grid with resolution of $\Delta x = \Delta y = 0.006$ m and CFL=0.45. The colours correspond to: the truncation error – green, the third order error – red, the fourth order error – blue, the total error – black. The black dashed line is a scaled plot of the surface elevation at the centre of the domain over time. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

ical and multiphase collapse considerations has been done under a different framework [42]. Convergence using a simple approach of material with visco-plastic rheology sliding in one go is demonstrated. The collapse takes place in the submarine domain. To reconstruct the pre-slide morphology two centres of collapse were considered in the North upper and lower slope regions (as identified by Georgiopoulou et al. [43]). The pre-slide mass ($t_0 = 0$) is represented by two Gaussians spreading symmetrically on the slope region. The one-fluid version of the code VolcFlow [44] was employed for the underwater flow simulations. The time-dependent bathymetry displacement was used as an input in Volna-OP2. This time-dependent bathymetry is incorporated into Volna-OP2 by reading in updated bathymetry files at specific time stamps. As one has to ensure that the temporal occurrence of the bathymetry changes is consistent for each simulation, there is a constraint on the time stepping for each simulation. For this study we have chosen four gauges marked by the red dots on Fig. 15 to present the evolution of the tsunami as a function of time in Fig. 16.

The role of physical dispersion for this landslide event has been discussed in Salmanidou et al. [28]. Drawing on the findings of [31] it was shown that the effects of physical dispersion warrant further exploration. However this is beyond the scope of the present study. As we present relative differences in mesh resolutions, it is not of interest here.

Overall, the numerical simulations with varying spatial discretisation behave similarly. Notable differences can be seen at gauge 4 (Fig. 16d), where unresolved bathymetric features of the continental shelf play a role. To investigate the numerical differences we will focus on the output at gauge 3 (Fig. 16c). The reason for choosing this gauge is to minimise the effects of these unresolved bathymetric features as the bathymetry between this gauge location and the landslide source is relatively flat. The maximum wave amplitude of the initial tsunami wave at gauge 3 is highlighted in Fig. 16c. As there is no true solution to compare with for this real case we will take the simulation results from the finest mesh $\Delta x = \Delta y = 450$ m as the *ground truth*. Relative differences between this *ground truth* and the other simulations are presented in Fig. 17 and Table 1. When comparing the signals (Fig. 17), the coarser meshes exhibit phase lagging and/or damping of the signal, i.e. the maximum tsunami wave arrives later and its amplitude is diminished. This behaviour was highlighted in the previous

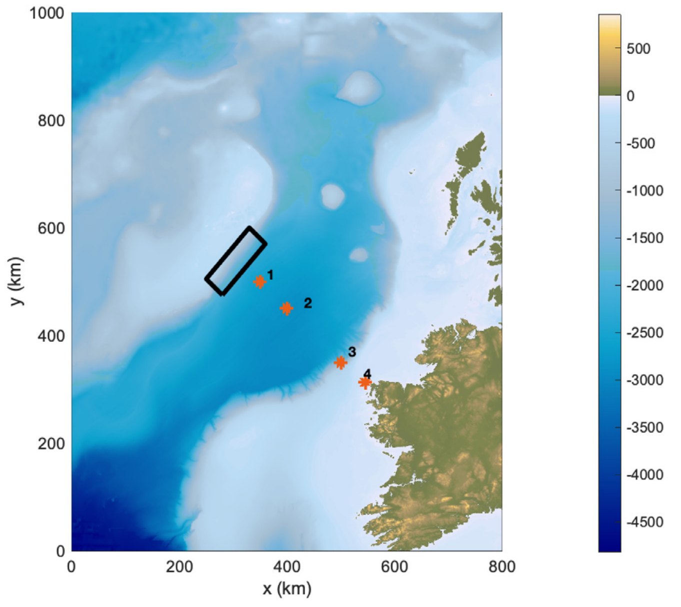


Fig. 15. Computational domain used in the simulation with the region of collapse marked by the black box and the four gauges marked with red dots. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

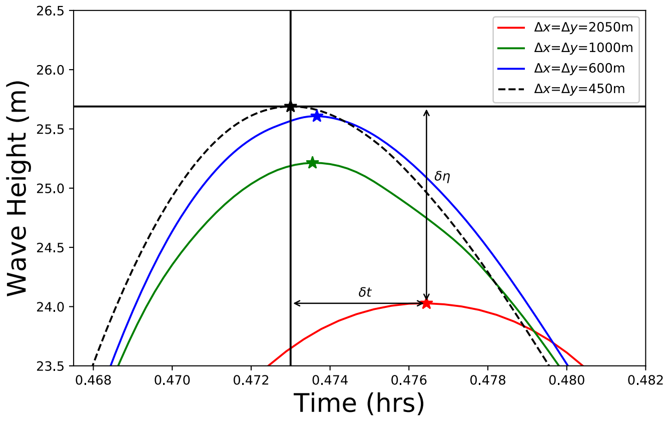


Fig. 17. Zoom in on the maximum wave amplitudes of the initial tsunami wave at gauge 3 (Fig. 16c) simulated using the varying mesh resolutions.

Table 1
Relative differences in the maximum wave height and its arrival time at gauge 3 between the coarser meshes and finest one (Fig. 16). $\delta\eta$ = the difference in maximum wave height and δt = the time delay in the arrival of the maximum wave.

Resolution	$\delta\eta$ [m]	δt [s]
600	0.08	2.4
1000	0.47	2.0
2050	1.66	12.4

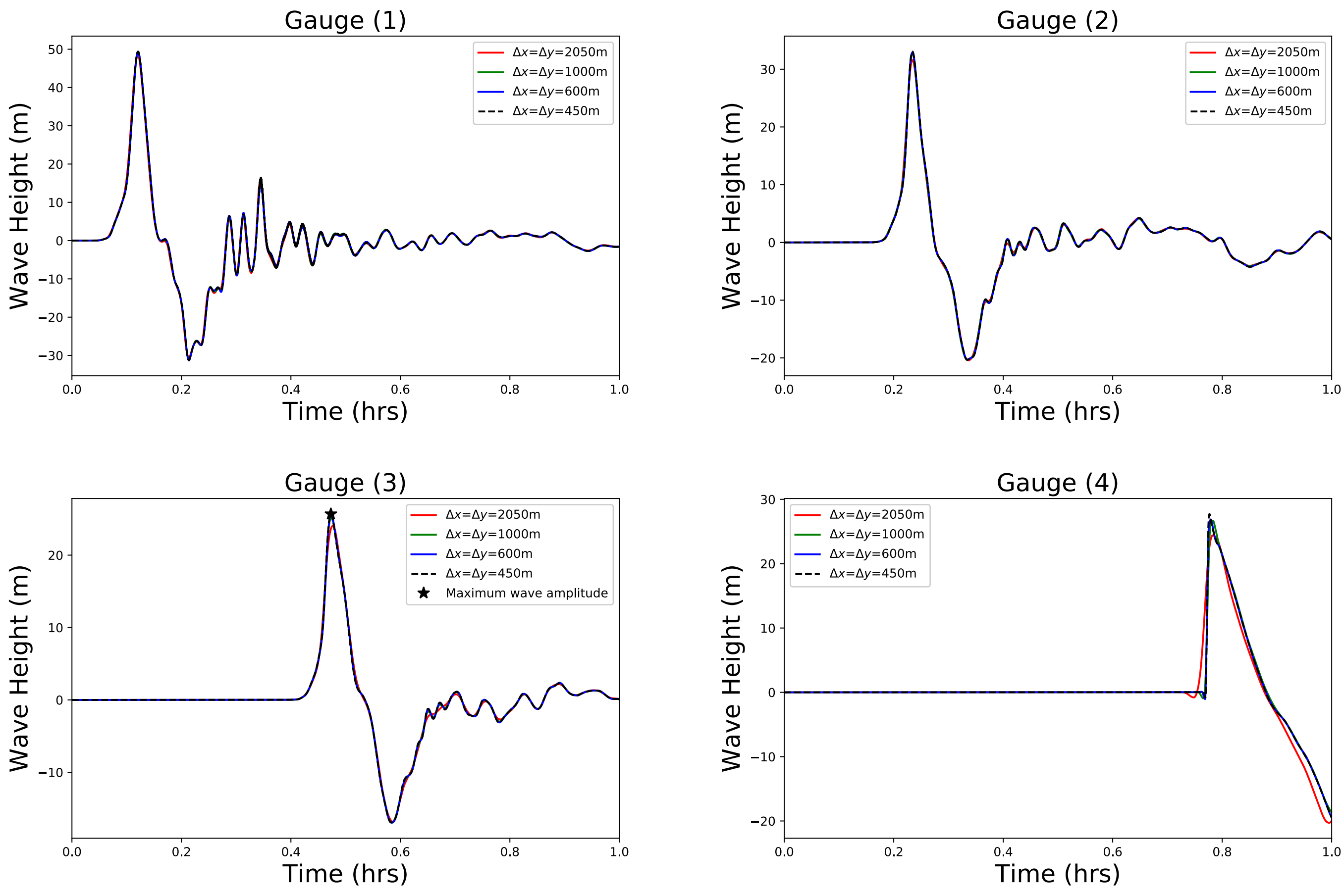


Fig. 16. Results of the numerical simulations at four gauges from left to right on Fig. 15, sorted horizontally starting from the upper left corner. Each plot includes four tests with different spatial resolution: $\Delta x = \Delta y$ =(2050 m - red line, 1000 m - green line, 600 m - blue line, 450 m - black dashed line) run for 1 h. The gauges coordinates (from left to right, from top to bottom) are: a - (350 km, 500 km), b - (400 km, 450 km), c - (500 km, 350 km), and d - (545.8 km, 312.2 km). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
Volna-OP2 performance on a GPU.

$\Delta x = \Delta y$ [m]	Number of cells	Run time [s]	Timestep dt [s]	steps	speed [$\frac{\text{cells x timesteps}}{s}$]
2050	492,374	8.4	1.6	2250	131,885,893
1000	2,062,028	57.1	0.8	4500	162,506,585
600	5,744,312	297.0	0.4	9000	174,070,061
450	10,218,002	523.0	0.4	9000	175,835,598

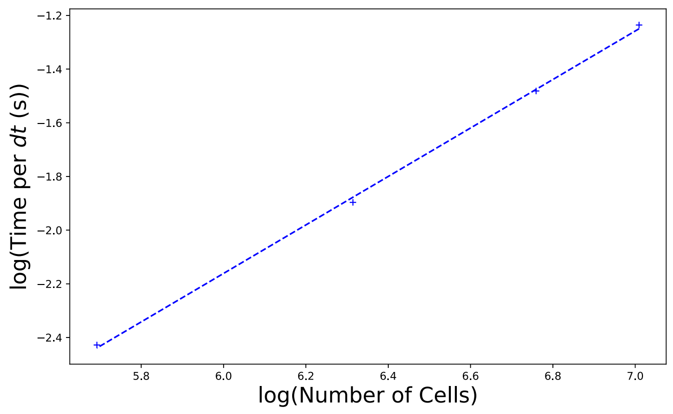


Fig. 18. Time per timestep for the various mesh resolutions with Volna-OP2 on a GPU.

error analysis – Section 4 – and is thus expected. Only the consistency (not the accuracy) of the algorithm is shown by the similar behaviour of the simulations and the convergence towards the solution of the finest mesh (with the theoretically proven convergence rate (Section 3)). The numerical solutions' accuracy is implied through the satisfaction of the boundary conditions and an understanding of the physics behind the solution. An explanation of this point can be found here [45].

It should be noted that for cases which utilize non-uniform mesh resolutions the same error analysis findings will hold true. If the mesh is non-uniform (i.e the characteristic length scale of the cells vary across the domain), the analysis addresses the worst possible scenario and scales all cells by the largest for error computations. Thus, one would expect to see similar behaviour regarding numerical dissipation and dispersion.

Turning to the performance of Volna-OP2 on the GPU, Table 2 and Fig. 18 summarise the runtimes for the various mesh resolutions. As outlined above, the changing bathymetry due to the slide is incorporated into Volna-OP2 by reading in simple text files with the updated bathymetry values. In order to ensure that the slide proceeds at the correct velocity and these files are read in at the correct time stamps, a constraint on the timesteps (dt) must be implemented. For each of the simulations a trial run is carried out with a stable CFL value to find the optimum timestep. As the timestep is dependent on the minimum mesh element size, the simulations with the finer meshes require a smaller timestep and thus a larger number of steps to complete the hour simulation. Despite the different timesteps used, the rate at which the bathymetry is updated is kept constant across all the simulations, with the bathymetry updated every 1.6 s. Defining a speed metric (24), all simulations produce a value greater than 131,885,893 $\frac{\text{cells x timesteps}}{s}$. With Fig. 18 showing the time per time step against number of cells, one can see that we get a linear speed up. Those interested in the scalability of the code on other HPC architectures are referred to [20]. This analysis of the relative errors in Table (1) and computational efficiency informs the user on what resolution will provide an acceptable level of accuracy within a given time constraint.

$$\text{Speed} = \frac{(\text{Number of Cells})(\text{Number of Steps})}{\text{Runtime}}. \tag{24}$$

6. Concluding remarks

Based on the current study we can conclude that Volna-OP2 is a robust and efficient parallel solver for the NSWE. It is based on the finite volume scheme for spatial integration, implementing a MUSCL reconstruction and using the 2nd order Runge-Kutta scheme for integration in time. The scheme is conditionally stable with experimentally confirmed CFL = 1.0. The code can handle complex geometries and simulate real-life cases.

The error analysis shows that it scales quadratically with refining the spatial mesh. However, reducing the time-step does not have a visible effect on the error. The solution amplitude decays in time, and one can observe both damping and phase shifts. So there is an energy leak from the system, which is expected due to the dissipative Runge-Kutta scheme. However, the error can be minimised by reducing the spatial resolution. The phase error is mostly negative, which corresponds to the phase lag and the artificial wiggles behind the wave front. However, it changes sign occasionally and this leads to the phase lead.

The real case simulations show the efficiency and scalability of the code run on a GPU. The 1 h realistic size tsunami model is simulated in ~ 8.7 min on the finest (450 m resolution, ~ 10.2M cells) mesh using one GPU. However, the simulations have shown that comparable results can be achieved using a coarser mesh and reduced runtime. Therefore the users should be familiar with code pros and cons before setting up their simulation in order to get physically meaningful and numerically accurate results. This acceptable level of accuracy and runtime trade off is an important decision when using Volna-OP2 to perform comprehensive sensitivity analysis tests and uncertainty quantification.

To conclude the paper we want to mention that Volna-OP2 is still under development. Therefore its analysis and benchmark testing play an important role for the code's continuous enhancement and improvement. The code is already an important tool for the tsunami modelling community. However, we hope that our work will help to its wider adoption and will lead to discussion on the most suitable algorithms and software platforms for realistic tsunami modelling and prediction of its effects.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Daniel Giles: Conceptualization, Methodology, Software, Formal analysis, Writing - original draft, Writing - review & editing. **Eugene Kashdan:** Conceptualization, Methodology, Formal analysis, Writing - original draft. **Dimitra M. Salmanidou:** Methodology, Investigation, Writing - original draft, Writing - review & editing. **Serge Guillas:** Supervision, Writing - review & editing. **Frédéric Dias:** Conceptualization, Supervision, Writing - review & editing.

Acknowledgments

The authors wish to acknowledge the DJEI/DES/SFI/HEA Irish Centre for High-End Computing (ICHEC) for the provision of computational resources. Daniel Giles is supported by a Government of Ireland Postgraduate Scholarship from the Irish Research Council (GOIPG/2017/68). Eugene Kashdan would like to acknowledge NVIDIA for the donation, within the framework of the Professor Partnership Programme, of the Tesla K40 GPU which was used for this research. Serge Guillas acknowledges support from the Alan Turing Institute project "Uncertainty Quantification of multi-scale and multiphysics computer models: applications to hazard and climate models" as part of the grant EP/N510129/1 made to the Alan Turing Institute by EPSRC grant no EP/N510129/1.

References

[1] Postacioglu N, Özeren MS. A semi-spectral modelling of landslide tsunamis. Geophys J Int 2008;175(1):1–16.
[2] Behrens J, Dias F. New computational methods in tsunami science. Phil Trans R Soc A 2015;373:20140382.
[3] Titov VV, Gonzalez FI. Implementation and testing of the method of splitting tsunami (MOST) model. NOAA Technical Memorandum ERL PMEL-112, 11 pp UNIDATA; 1997.
[4] Liu PL-F, Woo S-B, Cho YS. Computer programs for tsunami propagation and inundation. Cornell University; 1998.
[5] Gailler A, Hébert H, Loevenbruck A, Hernandez B. Simulation systems for tsunami wave propagation forecasting within the French tsunami warning center. Nat Hazards Earth Syst Sci 2013;13(10):2465.
[6] Zhang YJ, Baptista AM. An efficient and robust tsunami model on unstructured grids. Part I: Inundation benchmarks. Pure Appl Geophys 2008;165(11):2229–48.
[7] Harig S, Chaeroni, Pranowo WS, et al. Tsunami simulations on several scales. Ocean Dyn 2008;58(5):429–40.
[8] Vater S, Behrens J. Well-Balanced Inundation Modeling for Shallow-Water Flows with Discontinuous Galerkin Schemes. In: Finite volumes for complex applications VII-elliptic, parabolic and hyperbolic problems. Springer; 2014. p. 965–73.
[9] Jacobs CT, Piggott MD. Firedrake-Fluids v0.1: numerical modelling of shallow water flows using an automated solution framework. Geosci Model Dev 2015;8(3):533–47.
[10] Berger MJ, George DL, LeVeque RJ, Mandli KT. The GeoClaw software for depth-averaged flows with adaptive refinement. Adv Water Resour 2011;34:1195–206.
[11] Macias J, Castro MJ, Ortega S, Escalante C, Gonzalez-Vida JM. Performance Benchmarking of Tsunami-HySEA Model for NTHMP's Inundation Mapping Activites. Pure Appl Geophys 2017;174:3147–83.
[12] Kennedy AB, Chen Q, Kirby JT, Dalrymple RA. Boussinesq Modeling of Wave Transformation, Breaking, and Runup. I: 1D. J Waterw Port Coastal Ocean Eng 2000;126(1):39–47.
[13] Lynett PJ, Wu T-R, Liu PLF. Modeling wave runup with depth-integrated equations. Coastal Eng 2002;46(2):89–107.
[14] Tavakkol S, Lynett P. Celeris: a GPU-accelerated open source software with a Boussinesq-type wave solver for real-time interactive simulation and visualization. Comput Phys Commun 2017;217:117–27.
[15] Satria MT, Huang B, Hsieh T-J, Chang Y-L, Liang WY. GPU Acceleration of Tsunami Propagation Model. IEEE J Sel Top Appl Earth Obs Remote Sens 2012;5(3):1014–23.
[16] Liang W-Y, Hsieh T-J, Satria MT, Chang Y-L, Fang J-P, Chen C-C, et al. A GPU-Based Simulation of Tsunami Propagation and Inundation. In: Algorithms and architectures for parallel processing: 9th international conference, ICA3PP 2009, Taipei, Taiwan, June 8–11, 2009 proceedings. Springer Berlin Heidelberg; 2009. p. 593–603.
[17] Brodtkorb AR, Hagen TR, Lie K, et al. Simulation and visualization of the Saint-Venant system using GPUs. Comput Vis Sci 2010;13(7):341–53.
[18] Acuña M, Aoki T. Real-time tsunami simulation on multi-node GPU cluster. ACM/IEEE conference on supercomputing; 2009.
[19] Dutykh D, Poncet R, Dias F. The VOLNA code for the numerical modeling of tsunami waves: Generation, propagation and inundation. Eur J Mech – B/Fluids 2011;30(6):598–615.
[20] Reguly IZ, Giles D, Gopinathan D, Quivy L, Beck JH, Giles MB, Guillas S, Dias F. The VOLNA-OP2 tsunami code (version 1.5). Geosci Model Dev 2018;11(11):4621–35.
[21] Poncet R, et al. A Study of the Tsunami Effects of Two Landslides in the St Lawrence Estuary. In: Mosher DC, et al., editors. Submarine Mass Movements and Their Consequences. Springer, Netherlands; 2010. p. 755–64.
[22] Dias F, et al. On the Modelling of Tsunami Generation and Tsunami Inundation. Procedia IUTAM 2014;10:338–55.
[23] Giles D, McConnell B, Dias F. Modelling with Volna-OP2—Towards Tsunami Threat Reduction for the Irish Coastline. Geosciences 2020;10:226.
[24] Gopinathan D, Venugopal M, Roy D, Rajendran K, Guillas S, Dias F. Uncertainties in the 2004 Sumatra–Andaman source through nonlinear stochastic inversion of tsunami waves. Proc R Soc A 2017;473:20170353.
[25] Stefanakis TS, et al. Can small islands protect nearby coasts from tsunamis? An active experimental design approach. Proc R Soc A 2014;470:20140575.
[26] Beck J, Guillas S. Sequential Design with Mutual Information for Computer Experiments (MICE): Emulation of a Tsunami Model. SIAM/ASA J Uncertain Quantif 2016;4(1):739–66.
[27] Guillas S, Sarri A, Day SJ, Liu X, Dias F. Functional emulation of high resolution tsunami modelling over Cascadia. Ann Appl Stat 2018;12(4):2023–53.
[28] Salmanidou DM, Guillas S, Georgiopoulou A, Dias F. Statistical emulation of landslide-induced tsunamis at the Rockall Bank, NE Atlantic. Proc R Soc A 2017;473:20170026.
[29] Liu X, Guillas S. Dimension Reduction for Gaussian Process Emulation: An Application to the Influence of Bathymetry on Tsunami Heights. SIAM/ASA J Uncertain Quantif 2017;5(1):787–812.
[30] Popinet S. A quadtree-adaptive multigrid solver for the Serre-Green-Naghdi equations. J Comput Phys 2015;302:336–58.
[31] Glimsdal S, Pedersen GK, Harbitz CB, Løvholt F. Dispersion of tsunamis: does it really matter? Nat Hazards Earth Syst Sci 2013;13:1507–26.
[32] Mudalige GR, Giles MB, Reguly I, Bertolli C, Kelly PHJ. OP2: An active library framework for solving unstructured mesh-based applications on multi-core and many-core architectures. 2012 Innovative Parallel Computing (InPar). San Jose, CA; 2012. p. 1–12. doi:10.1109/InPar.2012.6339594.
[33] Brocchini M, et al. An efficient solver for nearshore flows based on the WAF method. Coastal Eng 2001;43(2):105–29.
[34] Harten A, Lax PD, van Leer B. On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws. SIAM Rev 1983;25(1):35–61.
[35] Zhou JG, et al. Numerical solutions of the shallow water equations with discontinuous bed topography. Internat J Numer Methods Fluids 2002;38:769–88.
[36] Howen PJVD, Sommeijer BP. Explicit Runge–Kutta (–Nyström) Methods with Reduced Phase Errors for Computing Oscillating Solutions. SIAM J Numer Anal 1987;24(3):595–617.
[37] Barth TJ, Jespersen DC. The design and application of upwind schemes on unstructured meshes. 27th Aerospace Sciences Meeting; 1989.
[38] Thacker WC. Some exact solutions to the nonlinear shallow water wave equations. J Fluid Mech 1981;107:499–508.
[39] Delestre O, et al. SWASHES: a compilation of shallow-water analytic solutions for hydraulic and environmental studies. Int J Numer Methods Fluids 2013;72:269–300.
[40] Lax PD, Richtmyer RD. Survey of the stability of linear finite difference equations. Commun Pure Appl Math 1956;9:267–93.
[41] Burwell D, Tolkova E, Chawla A. Diffusion and dispersion characterization of a numerical tsunami model. Ocean Model 2007;19:10–30.
[42] Salmanidou DM, Georgiopoulou A, Guillas S, Dias F. Rheological considerations for the modelling of submarine sliding at the Rockall Bank, NE Atlantic Ocean. Phys Fluids 2018;30:030705.
[43] Georgiopoulou A, Shannon PM, Sacchetti F, Haughton PD, Benetti S. Basement-controlled multiple slope collapses, Rockall Bank Slide Complex, NE Atlantic. Mar Geol 2013;336:198–214.
[44] Kelfoun K, Druitt TH. Numerical modeling of the emplacement of Socompa rock avalanche, Chile. J Geophys Res 2005;110.
[45] LeVeque RJ. Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems. Society for Industrial and Applied Mathematics; 2007.