A dynamic *hp*-adaptive discontinuous Galerkin method for shallow water flows on the sphere with application to a global tsunami simulation.

Accepted for publication in Monthly Weather Review on Septembre 8, 2011

SÉBASTIEN BLAISE *

Institute for Mathematics Applied to Geosciences

National Center for Atmospheric Research, Boulder, CO

currently at: *Université Bordeaux 1, Talence, FRANCE*

AMIK ST-CYR

Institute for Mathematics Applied to Geosciences

National Center for Atmospheric Research, Boulder, CO

currently at: *Royal Dutch Shell, Houston, TX*

Corresponding author address: Sébastien Blaise, Université Bordeaux 1, Institut de Mathématiques de Bordeaux, 351, cours de la Libération, F-33405 Talence cedex, FRANCE

E-mails: `sebastien.blaise@u-bordeaux1.fr`, `Amik.St-Cyr@shell.com`

ABSTRACT

A discontinuous Galerkin model solving the shallow water equations on the sphere is presented. It captures the dynamically varying key aspects of the flows by having the advantageous ability to locally modify the mesh as well as the order of interpolation within each element. The computational load is efficiently distributed amongst processors in parallel using a weighted recursive coordinate bisection strategy. A simple error estimator, based on the discontinuity of the variables at the interfaces between elements, is used to select the elements to be refined or coarsened. The flows are expressed in three-dimensional Cartesian coordinates but tangentially constrained to the sphere by adding a Lagrange multiplier to the system of equations. The model is validated on classical atmospheric test cases and on the simulation of the February 2010 Chilean tsunami propagation. The proposed multiscale strategy is able to reduce the computational time by an order of magnitude on the tsunami simulation, clearly demonstrating its potential towards multi-resolution three-dimensional oceanic and atmospheric applications.

1

1. Introduction

Over the last decades, global climate models have been widely used for the prediction of future climate trends. However, climate change will mostly affect the ecosystems and social and economic well-being at the regional scale (IPCC 1997). Hence, improving our capabilities in regional atmospheric simulations is of critical importance. It is recognized that the climate of the atmosphere is influenced by a disparate range of richly interacting spatiotemporal scales (Wheeler and Kiladis 1999). Indeed, localized flow structures, like hurricanes, may play an important role in obtaining the correct climate signal (Emanuel 2005; Mann and Emanuel 2006). Alas, the climate problem involves multiple physical scales as it couples with the oceans, the carbon cycle, ice sheets and various other processes yielding a difficult multiscale and multi-physics problem.

Current models are incapable of representing the multiscale aspects of the climate system. This is due in part to the physical parameterizations involved. Historically, climate models are spawns of weather models ran on much longer time-scales, at much lower resolution, with physics crudely parameterized (out of sheer necessity) instead of directly simulated (Slingo et al. 2009). Nowadays, with much more powerful computing facilities[1], it is tempting to readdress some of those simplifying assumptions by attempting to run a weather model at weather resolutions on climate time-scales. Still, it is not clear if such an approach will produce viable climate predictions. In order to reach such resolutions and generate a correct climate, various new alternative approaches to physics need to be investigated (see for instance Grabowski (2001); Khairoutdinov and Randall (2001); Khouider et al. (2011)). The goal of these approaches is to reproduce the correct multiscale wave patterns observed in nature directly. Apart from the multicloud approach described in Khouider et al. (2011), the so-called cloud resolving models are extremely costly to run as compared with current parameterizations. Models will require more spatial resolution to eventually benefit of such improved physics. A very first step toward this goal consists of making them extremely scalable, while increasing their spatial resolution, regardless of the physics employed (Dennis et al. 2005a; Wehner 2008; Bhanot et al. 2008; Satoh et al. 2008). While this approach is promising, some flaws have recently been identified (see McClean et al. (2010) for instance). Nevertheless, even if a correct multiscale physics package was available, the remaining problem with such approaches would be mainly computational. Each uniform doubling of the horizontal resolution results into an increase in computational cost of a factor 8, where a factor 2 is stemming from each dimension in space and time. Physics are the computational bottleneck of a climate or weather model and their cost is unfortunately directly proportional to the total number of spatial grid nodes employed. One way to reduce the number of nodes in a model, while preserving the range of spatial scales involved, is to employ fully unstructured dynamically adaptive meshes. Such approaches were proven very successful quite recently on spherical geometries (Burstedde et al. 2010; Wilcox et al. 2010).

This article describes the development of a set of numerical techniques and demonstrates their potential for solving multiscale environmental flows in an efficient fashion. As a first step towards a more realistic three-dimensional model, it focuses on the shallow water equations, presenting the major difficulties found in the horizontal aspects of three-dimensional global geophysical models (Williamson et al. 1992).

---

[1] http://www.top500.org

Designed for unstructured grids, the high-order discontinuous Galerkin (DG) method (Cockburn et al. 2000) is a good candidate to renew the dynamical cores employed in environmental flows models. It is conservative, accurate and well suited for advection-dominated flows (Cockburn and Shu 2001). The DG method enjoys most of the strengths of finite element and finite volume schemes while avoiding most of their weaknesses. The polynomial interpolation used inside each element allows for a high-order representation of the solution. As for finite volume methods, advection schemes take into account the characteristic structure of the equations. Moreover, no degree of freedom is shared between two geometric entities. This high-level of locality considerably simplifies the implementation of the method and contributes to its parallel efficiency. Finally, the mass matrix is block diagonal, and for explicit time-stepping schemes no linear solver is needed. We also observe a growing interest for the DG methods in marine modeling (Aizinger and Dawson 2002; Bernard et al. 2007; Kubatko et al. 2006; Blaise et al. 2010a,b; Comblen et al. 2010). For atmospheric modeling, the high-order capabilities of this scheme are attractive (Nair et al. 2005a,b; Giraldo 2006; Giraldo and Restelli 2008; St-Cyr and Neckels 2009; Nair et al. 2009), and the increasing use of DG follows the trend to replace the global spectral transform methods with local ones (Neale et al. 2010; Dennis et al. 2011).

The combination of the high-order DG method with dynamic refinement allows the resolution to be enhanced statically in the areas of interest for regional climate simulations and dynamically where the unsteady dynamics are more demanding. Hence, the computational power is used more effectively by concentrating the CPU load where it is needed. Dynamic adaptation of the computational mesh to locally modify the resolution, also known as h-adaptivity, has been introduced in the last decade to simulate shallow water flows on the sphere (Remacle et al. 2006; Bernard et al. 2007; Läuter et al. 2007; Weller et al. 2009). While those studies are based on conforming local mesh modifications such as edges splitting, collapsing, swapping and node movements, other approaches consider non-conforming mesh modifications (St-Cyr et al. 2008) or the more classical Berger-Oliger algorithm (Chen et al. 2011). The present work focuses on a *hybrid* Adaptive Mesh Refinement (AMR) method, consisting of recursive non-conforming elements splitting of an initially unstructured mesh. It thus provides a fast and very localized adaptation procedure (Löhner and Baum 1991). Having recourse to a shallow water DG code to simulate multiscale flows, Kubatko et al. (2009) showed the gain of efficiency obtained by using $p-$adaptation. It consists of modifying the local order of interpolation during the simulation. To the authors' knowledge, only one very recent study was published using simultaneously h and p-adaptation to solve shallow water problems (Eskilsson 2010). While the simulations presented by Eskilsson (2010) are simplified and characterized by smooth fields on planar domains, the present work describes the first parallel hp-adaptive simulation of complex realistic flows on the sphere such as tsunamis.

This work has been conducted with a overarching goal of building a flexible software framework for the multiscale simulation of various problems in the climate sciences[2]. Hence, the model described herein is a generic high-order hp capable model for the simulation of various conservation laws. It is flexible enough to test innovative numerical techniques while relying on efficient computational kernel implementations with special attention to parallel

---

[2]A Multiscale Unified Simulation Environment for geoscientific applications, `http://www.muse.ucar.edu`

performance. Solving the shallow water equations is part of this attempt and constitutes the main topic of this paper. While dynamic adaptation is used, the definition of optimal refinement criteria is outside the scope of this work and will only be brievly discussed. The paper is organized in two sections. The model is described in the first section, with its discretization in both space and time as well as a description of how the equations are constrained onto the sphere. A short presentation of the hp adaptation process is shown with details about the parallel load balancing strategy. The second section is dedicated to the assessment of the model. It is validated on test cases accepted by the community. The first test considers a geostrophically balanced flow where the availability of an exact solution permits the computation of the high-order spatial convergence rates. Then follows the simulation of a flow impinging a mountain using the adaptive model which is compared with a high-resolution solution. Finally, the simulation of a realistic global tsunami event is presented.

2. Model description

The model solves the shallow water equations on the sphere using the nodal DG method with dynamic adaptivity. This section is devoted to the description of the model, including the continuous equations, their discretization, the handling of flows on the sphere and the dynamic adaptivity procedure.

a. The shallow water equations

The conservative shallow water equations may be obtained by depth-integrating the Navier-Stokes and continuity equations. A thorough description of the equations and the associated hypotheses can be found in Pedlosky (1986). They consist of a system of equations for the momentum and the evolution of the free-surface:

$$\frac{\partial H\mathbf{u}}{\partial t} = -gH\boldsymbol{\nabla}\eta - \boldsymbol{\nabla}\cdot(H\mathbf{u}\mathbf{u}) - f\boldsymbol{k}\times(H\mathbf{u}) + \frac{\boldsymbol{\tau^s}+\boldsymbol{\tau^b}}{\rho}, \tag{1}$$

$$\frac{\partial \eta}{\partial t} = -\boldsymbol{\nabla}\cdot(H\mathbf{u}), \tag{2}$$

where η, $\mathbf{u}$ and H are the elevation of the free-surface, the depth-averaged horizontal velocity and the total depth. The gravitational acceleration, the Coriolis parameter, the constant density and the surface and bottom stresses are respectively denoted g, f, ρ, $\boldsymbol{\tau^s}$ and $\boldsymbol{\tau^b}$. The vertical unit vector, pointing upwards, is designated by $\boldsymbol{k}$. The elevation gradient term of the momentum equation (1) can be written as a sum of a flux term and a source term, leading to

$$\frac{\partial H\mathbf{u}}{\partial t} = \underbrace{-\frac{g}{2}\boldsymbol{\nabla}\left(H^2 - h^2\right) - \boldsymbol{\nabla}\cdot(H\mathbf{u}\mathbf{u})}_{\text{flux terms}} + \underbrace{g\eta\boldsymbol{\nabla}h - f\boldsymbol{k}\times(H\mathbf{u}) + \frac{\boldsymbol{\tau^s}+\boldsymbol{\tau^b}}{\rho}}_{\text{source terms}}, \tag{3}$$

where $h = H - \eta$ is the depth at rest.

4

b. Weak formulation

For the sake of completeness and to highlight the peculiarities of the described implementation, we provide here the full weak DG finite element formulation for the shallow water equations. The weak formulations of the different equations are derived separately: the momentum equation is first considered, followed by the free-surface equation.

1) MOMENTUM EQUATION

The Galerkin weak formulation of the momentum equation is obtained by multiplying equation (3) by a set of vectorial test functions $\widehat{\mathbf{u}}$ and integrating over the whole domain Ω:

$$\int_\Omega \widehat{\mathbf{u}} \cdot \frac{\partial H\mathbf{u}}{\partial t} \, \mathrm{d}\Omega = -\int_\Omega \widehat{\mathbf{u}} \cdot \frac{g}{2} \boldsymbol{\nabla} \left(H^2 - h^2\right) \, \mathrm{d}\Omega + \int_\Omega \widehat{\mathbf{u}} \cdot g\eta\boldsymbol{\nabla}h \, \mathrm{d}\Omega$$
$$- \int_\Omega \widehat{\mathbf{u}} \cdot \left(\boldsymbol{\nabla} \cdot (H\mathbf{u}\mathbf{u})\right) \, \mathrm{d}\Omega - \int_\Omega \widehat{\mathbf{u}} \cdot \left(f\boldsymbol{k} \times (H\mathbf{u})\right) \, \mathrm{d}\Omega + \int_\Omega \widehat{\mathbf{u}} \cdot \frac{\boldsymbol{\tau^s} + \boldsymbol{\tau^b}}{\rho} \, \mathrm{d}\Omega. \quad (4)$$

The unknowns as well as the test function belong to a common finite dimensional space, which definition will be detailed later. The domain Ω is then split into N_e elements Ω_e. Assuming now that each test function is nonzero in one element, and zero elsewhere, we can localize (4) obtaining

$$\int_{\Omega_e} \widehat{\mathbf{u}} \cdot \frac{\partial H\mathbf{u}}{\partial t} \, \mathrm{d}\Omega = \underbrace{-\int_{\Omega_e} \widehat{\mathbf{u}} \cdot \frac{g}{2} \boldsymbol{\nabla} \left(H^2 - h^2\right) \, \mathrm{d}\Omega + \int_{\Omega_e} \widehat{\mathbf{u}} \cdot g\eta\boldsymbol{\nabla}h \, \mathrm{d}\Omega}_{\text{elevation gradient} \,=:\, I_E}$$
$$\underbrace{-\int_{\Omega_e} \widehat{\mathbf{u}} \cdot \left(\boldsymbol{\nabla} \cdot (H\mathbf{u}\mathbf{u})\right) \, \mathrm{d}\Omega}_{\text{advection} \,=:\, I_A} \underbrace{-\int_{\Omega_e} \widehat{\mathbf{u}} \cdot \left(f\boldsymbol{k} \times (H\mathbf{u})\right) \, \mathrm{d}\Omega}_{\text{Coriolis} \,=:\, I_C} + \underbrace{\int_{\Omega_e} \widehat{\mathbf{u}} \cdot \frac{\boldsymbol{\tau^s} + \boldsymbol{\tau^b}}{\rho} \, \mathrm{d}\Omega}_{\text{stress} \,=:\, I_S}. \quad (5)$$

The variables H, $\mathbf{u}$ and η are considered discontinuous at the boundaries of each element. The local problems (5) have to be coupled with each other, which in DG methods is obtained by means of numerical fluxes, mimicking what is done for finite volume formulations. To see how numerical fluxes can be introduced in (5), we integrate by parts and the boundary fluxes $\mathcal{F}$ appear at the interfaces between elements. Those fluxes are the only mechanism that allows the information to be passed between the elements. The key ingredient of the weak formulation is the manner in which the unique fluxes $\mathcal{F}^*$ are specified, as the variables are not uniquely defined on those interfaces. Following Nair et al. (2005a), we use the local Lax-Friedrichs numerical flux, which ensures the stability of the method. Considering the unknown $H\mathbf{u}$ to be solved by the equation, the local Lax-Friedrichs unique flux is defined as

$$\mathcal{F}^* = \frac{1}{2} \left(\left(\mathcal{F}^R + \mathcal{F}^L\right) \cdot \mathbf{n} - \lambda \left((H\mathbf{u})^R - (H\mathbf{u})^L\right) \right), \quad (6)$$

where the subscripts R and L correspond to the values of the discontinuous fields respectively at the right and left of the interface. The vector $\mathbf{n}$ is the rightward normal; and $\lambda = \max\left(\sqrt{gH} + \|\mathbf{u}\|\right)$ is the maximum absolute value of the propagation speed of the

information (maximum eigenvalue of the system (2-3)) taken over the two neighboring elements. Introducing the mean $\{S\} = 0.5\left(S^R + S^L\right)$ and jump $[S] = 0.5\left(S^R - S^L\right)$ operators, the Lax-Friedrichs flux (6) can be writen as $\mathcal{F}^* = \{\mathcal{F}\} \cdot \mathbf{n} - \lambda[H\mathbf{u}]$. Each term of the right hand side of (5) is then derived as follows:

- elevation gradient:

$$I_E = \int_{\Omega_e} \boldsymbol{\nabla} \cdot \widehat{\mathbf{u}} \frac{g}{2} \left(H^2 - h^2\right) \ \mathrm{d}\Omega - \int_{\Gamma_e} \widehat{\mathbf{u}} \frac{g}{2} \left(\{H^2\} - h^2\right) \cdot \mathbf{n} \ \mathrm{d}\Gamma$$
$$+ \int_{\Omega_e} \widehat{\mathbf{u}} \cdot g\eta \boldsymbol{\nabla} h \ \mathrm{d}\Omega, \quad (7)$$

- advection:

$$I_A = \int_{\Omega_e} \boldsymbol{\nabla}\widehat{\mathbf{u}} : H\mathbf{u}\mathbf{u} \ \mathrm{d}\Omega - \int_{\Gamma_e} \widehat{\mathbf{u}} \cdot \{H\mathbf{u}\mathbf{u}\} \cdot \mathbf{n} \ \mathrm{d}\Gamma, \quad (8)$$

- Lax-Friedrichs penalisation (expressed in the right hand side):

$$I_P = \int_{\Gamma_e} \widehat{\mathbf{u}} \cdot \lambda[H\mathbf{u}] \ \mathrm{d}\Gamma, \quad (9)$$

where Γ_e denotes the one-dimensional contour of the element Ω_e. The Coriolis and stress terms remain unchanged since no spatial differentiation operator is appearing in their expressions. It is possible to do integration by parts of the elevation gradient and advection terms once again (Hesthaven and Wartburton 2008), leading to

- elevation gradient:

$$I_E = -\int_{\Omega_e} \widehat{\mathbf{u}} \cdot \frac{g}{2} \boldsymbol{\nabla} \left(H^2 - h^2\right) \ \mathrm{d}\Omega - \int_{\Gamma_e} \widehat{\mathbf{u}} \frac{g}{2} \left([H^2] - h^2\right) \cdot \mathbf{n} \ \mathrm{d}\Gamma$$
$$+ \int_{\Omega_e} \widehat{\mathbf{u}} \cdot g\eta \boldsymbol{\nabla} h \ \mathrm{d}\Omega \quad (10)$$

- advection:

$$I_A = -\int_{\Omega_e} \widehat{\mathbf{u}} \cdot \left(\boldsymbol{\nabla} \cdot (H\mathbf{u}\mathbf{u})\right) \ \mathrm{d}\Omega - \int_{\Gamma_e} \widehat{\mathbf{u}} \cdot [H\mathbf{u}\mathbf{u}] \cdot \mathbf{n} \ \mathrm{d}\Gamma. \quad (11)$$

The elevation gradient term (10) contains spatial derivatives of squared variables, which are difficult to compute accurately and can produce oscillations in the numerical solution. For robustness, the gradient is computed under an alternative form obtained by a combination of the two surface integrals of (10). The elevation gradient term is thus evaluated by

$$I_E = -\int_{\Omega_e} \widehat{\mathbf{u}} \cdot gH\boldsymbol{\nabla}\eta \ \mathrm{d}\Omega - \int_{\Gamma_e} \widehat{\mathbf{u}} \frac{g}{2} \left([H^2] - h^2\right) \cdot \mathbf{n} \ \mathrm{d}\Gamma, \quad (12)$$

6

in which the surface term corresponds to its original form (1). Gathering the different terms of the equation together leads to the following formulation for the momentum equation:

$$\int_{\Omega_e} \widehat{\mathbf{u}} \cdot \frac{\partial H\mathbf{u}}{\partial t} \, \mathrm{d}\Omega = \underbrace{- \int_{\Omega_e} \widehat{\mathbf{u}} \cdot gH\boldsymbol{\nabla}\eta \, \mathrm{d}\Omega - \int_{\Gamma_e} \widehat{\mathbf{u}}\frac{g}{2} \left([H^2] - h^2\right) \cdot \mathbf{n} \, \mathrm{d}\Gamma}_{I_E}$$

$$\underbrace{- \int_{\Omega_e} \widehat{\mathbf{u}} \cdot (\boldsymbol{\nabla} \cdot (H\mathbf{u}\mathbf{u})) \, \mathrm{d}\Omega - \int_{\Gamma_e} \widehat{\mathbf{u}} \cdot [H\mathbf{u}\mathbf{u}] \cdot \mathbf{n} \, \mathrm{d}\Gamma}_{I_A}$$

$$\underbrace{- \int_{\Omega_e} \widehat{\mathbf{u}} \cdot (f\boldsymbol{k} \times (H\mathbf{u})) \, \mathrm{d}\Omega}_{I_C} + \underbrace{\int_{\Omega_e} \widehat{\mathbf{u}} \cdot \frac{\boldsymbol{\tau}^s + \boldsymbol{\tau}^b}{\rho} \, \mathrm{d}\Omega}_{I_S} + \underbrace{\int_{\Gamma_e} \widehat{\mathbf{u}} \cdot \lambda[H\mathbf{u}] \, \mathrm{d}\Gamma}_{I_P}. \quad (13)$$

2) FREE-SURFACE EQUATION

Using the same procedure, obtaining the DG formulation for the free-surface equation is straightforward and leads to

$$\int_{\Omega_e} \widehat{\eta}\frac{\partial\eta}{\partial t} \, \mathrm{d}\Omega = - \int_{\Omega_e} \widehat{\eta}\boldsymbol{\nabla} \cdot H\mathbf{u} \, \mathrm{d}\Omega - \int_{\Gamma_e} \widehat{\eta}[H\mathbf{u}] \cdot \mathbf{n} \, \mathrm{d}\Gamma + \int_{\Gamma_e} \widehat{\eta}\lambda[\eta] \, \mathrm{d}\Gamma, \quad (14)$$

where $\widehat{\eta}$ is the scalar test function associated with the equation.

c. Discrete formulation

The discrete formulation is then obtained by replacing the test functions and the solution by the corresponding DG polynomial approximation. The solution is approximated in each element by

$$\begin{pmatrix} H\mathbf{u}^h \\ \eta^h \end{pmatrix} = \begin{pmatrix} \sum_{l=1}^{p} (H\mathbf{u})_l \, \phi_l \\ \sum_{l=1}^{p} \eta_l\phi_l \end{pmatrix}, \quad (15)$$

where p is the number of nodes in the considered element, while ϕ_l are the associated shape functions. Using quadrilateral elements, they are obtained by the tensorial product of Lagrange polynomials (Hesthaven and Wartburton 2008). Following the usual Galerkin procedure, we select discrete components of the test functions belonging to the same space as the polynomial basis functions used to approximate the solution (Karniadakis and Sherwin 2005; Hesthaven and Wartburton 2008). The unknown degrees of freedom associated to the node l are represented by the vector $(H\mathbf{u})_l$ for the transport and the scalar η_l for the free-surface elevation.

The two-dimensional integration rules are derived from the one-dimensional ones (see Deville et al. (2002)). The model is able to use either the Gauss-Legendre (GL) or the Legendre-Gauss-Lobatto (LGL) quadrature rules (Fig. 1). The LGL discretization is faster and simpler, as it does not need any interpolation of the variables at the edges of the

elements when the fluxes are computed. However, it is also less accurate: with p nodes, the GL quadrature rule integrates exactly polynomials of order $(2p-1)$ while the LGL rule only integrates exactly polynomials of order $(2p-3)$. Using under integration with integration by parts once, Kopriva and Gassner (2010) obtained faster simulations for a given error with GL nodes. However, they showed that this conclusion depends on the considered application. In the case of complex flows, the higher order of integration provided by the GL integration rule may be needed to guarantee the absence of aliasing that can trigger oscillations, especially in the presence of additional metric terms associated with the resolution of the equations on the sphere. However, the use of the LGL quadrature nodes in the simulations described in this article did not generate any oscillation and did not increase significantly the error. Hence, we relied on the LGL rule, allowing us to perform faster simulations.

d. Curvilinear transformations

High-order methods need a high-order mapping in order for them to be effective: the high order discretization will benefit from a high order representation of the computational domain. While recent techniques can integrate the two-dimensional shallow water equations on a large class of manifolds (e.g. Bernard et al. (2009)), we focus on a simpler method to operate on the sphere. To this aim, we construct a mapping stemming from the original basis used in the mesh generation process and, for exact surfaces, a projection mapping points on the original mesh to the analytical surface. Next, the Jacobi matrices involved in the transformation are generated numerically by employing the discrete differentiation matrix (in each parametric direction) on the reference element. With the Jacobi matrices, all geometric entities can be recovered.

In more details, the generation of the mappings from the curvilinear elements on a surface ($\mathbf{x} = (x_1, x_2, x_3)$) to the reference element coordinates $(\xi, \zeta) \in [-1, 1] \times [-1, 1]$ (see Fig. 1) is performed as follows. Considering as an exact surface the sphere, we define the analytical projection $\mathbf{P}(\mathbf{x}) = \mathbf{x}/||\mathbf{x}||$. Suppose the element representation from the mesh generator is bilinear in the three-dimensional coordinates, then, it is possible to write:

$$\hat{\mathbf{x}}(\xi, \zeta) = \mathbf{P}(\sum_{i=1}^{2} \sum_{j=1}^{2} \mathbf{x}_{ij} \ \phi_i^{1D,1}(\xi)\phi_j^{1D,1}(\zeta)), \tag{16}$$

where $\hat{\mathbf{x}}$ is a point lying perfectly on the surface of the sphere and the $\phi^{1D,1}$'s are the usual linear one-dimensional basis functions. The $\mathbf{x}_{ij}$ are the four nodes given by the mesh generator. If the mesh generator could give quadratic elements then nine nodes would be available and the tensor product of the basis functions on the right of (16) would be biquadratic. It is now possible to evaluate $\hat{\mathbf{x}}(\xi, \zeta)$ at any node in the reference element, in particular at $(p + k)^2$ LGL nodes, k being an integer ≥ 1 ($k = 2$ is used in this work). Evaluating at those quadrature nodes, gives a discrete vector $\bar{\mathbf{x}} = (\bar{x}_1, \bar{x}_2, \bar{x}_3)$. Hence, we have the following new representation:

$$\mathbf{x}(\xi, \zeta) = \sum_{i=1}^{p+k} \sum_{j=1}^{p+k} \bar{\mathbf{x}}_{ij} \ \phi_i^{1D,p+k}(\xi)\phi_j^{1D,p+k}(\zeta), \tag{17}$$

where the $\phi^{1D,p+k}$'s are the one-dimensional basis functions of order $p + k$. Notice that it

is possible to create the vector $\bar{\mathbf{x}}$ for each element in the mesh. Then, applying the discrete differentiation matrix, which is associated with the Lagrange basis functions associated to the $p+k$ quadrature points in each direction (Deville et al. 2002), yields the discrete version of the Jacobi matrices: $\partial \bar{x}_m / \partial \xi$, $\partial \bar{x}_m / \partial \zeta$. Each derivative that forms the Jacobi matrix is interpolated from the $(p+k)^2$ points to the original p^2 points. The Jacobi matrices are then used to compute all the geometric information in the model. Finally, notice that it is necessary to use $k \geq 1$ for the design order of the scheme to be p.

e. Temporal discretization

The model is integrated in time using the optimal, three stages, third order strong stability preserving Runge-Kutta scheme (SSP33), also known as the third-order TVD Runge-Kutta scheme (Gottlieb and Shu 1998). To evolve the solution from the time step n to the next one $n+1$, the procedure is composed of three stages:

$$
\begin{aligned}
\mathbf{y}^{(1)} &= \mathbf{y}^n + \Delta t \mathbf{F}(\mathbf{y}^n), \\
\mathbf{y}^{(2)} &= \frac{3}{4}\mathbf{y}^n + \frac{1}{4}\mathbf{y}^{(1)} + \frac{1}{4}\Delta t \mathbf{F}(\mathbf{y}^{(1)}), \\
\mathbf{y}^{n+1} &= \frac{1}{3}\mathbf{y}^n + \frac{2}{3}\mathbf{y}^{(2)} + \frac{2}{3}\Delta t \mathbf{F}(\mathbf{y}^{(2)}),
\end{aligned}
\tag{18}
$$

where $\mathbf{y}^i$ is the vector of all discrete degrees of freedom at the stage i. The discrete right hand side of equations (13-14) using the value of the variables at the stage s is denoted by $\mathbf{F}(\mathbf{y}^s)$.

This method ensures nonlinear stability properties in the numerical solution of hyperbolic partial differential equations with discontinuous solutions (Gottlieb et al. 2011). It preserves strong stability properties of the spatial discretization, which is a great advantage for the simulation of geophysical flows, characterised by a wide range of resolved and unresolved phenomena. Note that many other explicit Runge-Kutta time integrators of different orders are available in the model[3], and others can be easily added by introducing the corresponding array of coefficients.

The marching schemes described here are subject to the CFL condition (Courant-Friedrichs-Lewy), and their time step is controlled by the highest Courant number of the global domain. It is generally associated with the smaller elements of the mesh and can be much larger than the average Courant number because of the variable resolution resulting from the unstructured grid and dynamic adaptation. As a result, the time step used to update the fields in the whole domain is constrained by the size of the smaller element, which is rather inefficient. Although an implicit method is not subject to the stability constraint on the time step, it may be too diffusive for elements characterised by a high Courant number. Hence, for adaptive methods to deliver their full potential, they need to be used in conjunction with other approaches such as local time-stepping (Lörcher et al. 2008). This method still need to be implemented in the code.

---

[3]A Multiscale Unified Simulation Environment available at `http://www.muse.ucar.edu`

f. Solving the equations on the sphere

Solving geophysical flows on the sphere is not a straightforward task: even if the flow is two-dimensional, the velocity field is three-dimensional and must remain tangent to the surface of the sphere. The use of non-cartesian coordinates to formulate the equations, restricting the domain to the surface of the sphere, is tempting but suffers from the existence of singularities at the poles (Mohseni and Colonius 2000).

Discontinuous Galerkin methods were combined with several cartesian techniques to handle the computation of flows on the sphere, such as the cubed sphere (Nair et al. 2005a,b; Dennis et al. 2005b), or local tangent bases (Comblen et al. 2009; Bernard et al. 2009).

In this work, we consider the method proposed by Côté (1988): the equations are solved in a three-dimensional cartesian space, but an additional constraint ensures the fluid particles to remain on the surface of the sphere. It has been successfully used in the DG framework (Giraldo et al. 2002; Giraldo 2006). Although this technique requires the resolution of three equations instead of two for the momentum, it is probably the easiest way to transform a two-dimensional planar model to a model operating on the sphere.

The constraint is enforced at the discrete level to ensure that the discrete solution maintains the flow on the sphere. Consider that water particles located at the nodes at the stage s are at the normalized position $\mathbf{r}$ (normalized coordinates of the nodes). After one Runge-Kutta sub-step (stage $s+1$), those particles will occupy the normalized position $\mathbf{r}^*$. An additional term is added to the discrete momentum equation for each Runge-Kutta stage. It corresponds to a restoring force towards the center of the sphere, with its direction parallel to the semi-implicit position of the fluid particles originating from the nodes (Côté 1988):

$$\frac{(H\mathbf{u})^{s+1} - \alpha_1^s (H\mathbf{u})^s - \alpha_2^s (H\mathbf{u})^n}{\Delta t^s} = \mathbf{F}_m(\mathbf{y}^s) + \mu \frac{\mathbf{r} + \mathbf{r}^*}{2}, \tag{19}$$

$$\mathbf{r}^* = \mathbf{r} + \Delta t^s \, \mathbf{u}^s, \tag{20}$$

where Δt^s corresponds to the Runge-Kutta sub-time step at stage s. In the case of SSP33 (18), those are $\{\Delta t^0, \Delta t^1, \Delta t^2\} = \{1, 1/4, 2/3\}$, while the factors $\{\alpha_1^0, \alpha_1^1, \alpha_1^2\} = \{0, 1/4, 2/3\}$ and $\{\alpha_2^0, \alpha_2^1, \alpha_2^2\} = \{1, 3/4, 1/3\}$. The restriction of the discrete right hand side $\mathbf{F}(\mathbf{y}^s)$ to the momentum equation is denoted $\mathbf{F}_m(\mathbf{y}^s)$. The factor μ needs to be determined such that the updated transport field remains tangent to the sphere. Equation (19) can be written as

$$(H\mathbf{u})^{s+1} = \alpha_1^s (H\mathbf{u})^s + \alpha_2^s (H\mathbf{u})^n + \Delta t^s \, \mathbf{F}_m(\mathbf{y}^s) + \Delta t^s \, \mu \frac{\mathbf{r} + \mathbf{r}^*}{2}. \tag{21}$$

Introducing (20) into (21) leads to

$$(H\mathbf{u})^{s+1} = \alpha_1^s (H\mathbf{u})^s + \alpha_2^s (H\mathbf{u})^n + \Delta t^s \, \mathbf{F}_m(\mathbf{y}^s) + \Delta t^s \, \mu \frac{\mathbf{r} + \mathbf{r} + \Delta t^s \, \mathbf{u}^s}{2}. \tag{22}$$

The transport field must be orthogonal to the position vector at the updated stage, i.e. $(H\mathbf{u})^{s+1} \cdot \mathbf{r}^* = 0$. After multiplying (22) by $\mathbf{r}^*$, using (20) and the orthogonality condition, we obtain

$$0 = (\mathbf{r} + \Delta t^s \, \mathbf{u}^s) \cdot (\alpha_1^s (H\mathbf{u})^s + \alpha_2^s (H\mathbf{u})^n) + \Delta t^s \, \mathbf{r} \cdot \mathbf{F}_m(\mathbf{y}^s)$$

$$+ (\Delta t^s)^2 \mathbf{u}^s \cdot \mathbf{F}_m(\mathbf{y}^s) + \Delta t^s \, \mu \underbrace{\mathbf{r} \cdot \mathbf{r}}_{=1} + \frac{3\mu \, (\Delta t^s)^2}{2} \mathbf{u}^s \cdot \mathbf{r} + \frac{\mu \, (\Delta t^s)^3}{2} \mathbf{u}^s \cdot \mathbf{u}^s, \tag{23}$$

10

which is equivalent to

$$\mu = \frac{-2\left(\mathbf{r} + \Delta t^s\, \mathbf{u}^s\right) \cdot \left(\alpha_1^s (H\mathbf{u})^s + \alpha_2^s (H\mathbf{u})^n + \Delta t^s \mathbf{F}_m(\mathbf{y}^s)\right)}{2\Delta t^s\ + 3\left(\Delta t^s\right)^2 \mathbf{u}^s \cdot \mathbf{r} + \left(\Delta t^s\right)^3 \mathbf{u}^s \cdot \mathbf{u}^s}. \tag{24}$$

Proceeding in this manner ensures that the constraint is treated as an additional implicit term in the equation rather than as an explicit or *a posteriori* correction. Using this method, no initial condition or roundoff error is accumulated. However, the correction term being very small compared with the right hand side, we noticed that simpler methods led to very similar results (e.g. Giraldo (2001)).

g. *Dynamic adaptation*

The main characteristic of the model is its ability to modify the mesh and the order of interpolation inside the elements during runtime, thus concentrating efficiently the computational resources where they are needed in order to capture the key aspects of the flow. In what follows, both the p and $h-$refinement are described. Both refinement strategies are used in conjunction with an error indicator also described therein.

1) h-ADAPTATION

In the model, we choose a hybrid mesh adaptation procedure. It combines an unstructured mesh with a quad-tree refinement strategy within each element. This approach closely follows the work described in Edwards (2002); Stewart and Edwards (2002). The first assumption consists in denoting each element in the initial mesh as the *root* of a quad-tree. A $2:1$ constraint is then enforced between the trees in order to allow for smooth transitions between refined and unrefined regions: two neighboring elements can only be separated by a single level of refinement. This procedure can be pursued until the desired level of refinement is reached (Fig. 2). A similar approach was also used in the spectral element code presented in St-Cyr et al. (2008). The main advantage of such an approach is that it simplifies the construction of the projection operators between the changing meshes. The elements are represented using a high-order basis with each newly created node having to be projected onto the geometry, in this study the surface of the sphere. The resulting mesh is non-conforming, in other words, some element corners face the center of neighboring elements edges. It is not problematic as the neighboring elements only interact through the fluxes between the edges, which can still be computed uniquely as described in Kopriva et al. (2002).

Local interpolation operators are constructed in order to transfer the data between the single element Ω_i and its four child elements Ω_{ij} (Fig. 2). The prolongation (refinement) and restriction (coarsening) operators consist of a L^2 minimization and require the resolution of

$$\int_{\Omega_i} \hat{A}\, A_{\text{coarse}}\ \mathrm{d}\Omega \ \underset{\underset{\text{restriction}}{\longleftarrow}}{\overset{\overset{\text{prolongation}}{\longrightarrow}}{=}}\ \sum_{j=1}^{4} \int_{\Omega_i} \hat{A}\, A_{\text{fine}}\ \mathrm{d}\Omega \tag{25}$$

for each field A approximated by A_{fine} on the refined (child) elements and A_{coarse} on the parent element, $\hat{A}$ being the associated test functions. Notice that the test functions $\hat{A}$, in the case of a prolongation, are the ones on the coarse element while, respectively, in the

11

course of a restriction, the test functions are the ones related to the children elements. The projection (25) is uniquely defined. It conserves locally the integral of the interpolated fields, which is critical for long-term geophysical simulations.

As the node positions do not change during the refinement process, no numerical diffusion is introduced by the interpolation of the fields. The numerical diffusion introduced during mesh coarsening is minimal, resulting from the reduction of the discrete space. Hence, this adaptive mesh technique is usually less diffusive than other procedures based on re-meshing or mesh modifications including nodes displacement, even local (Löhner and Baum 1991). The simple tree structure of the adapted mesh makes the adaptation process very fast, even in parallel (see Burstedde et al. (2010) for a *state-of-the-art* implementation).

2) *p*-ADAPTATION

The adaptation of the order of interpolation simply replaces the set of shape and test functions associated to an element to functions of a different order. Once again, a L^2 projection is used to interpolate the data from a discretization of the element Ω_i using p nodes to a discretization using q nodes:

$$\int_{\Omega_i} \hat{A}^p \sum_{l=1}^{q} (\phi_l^q A_l^q) \ \mathrm{d}\Omega = \int_{\Omega_i} \hat{A}^p \sum_{l=1}^{p} (\phi_l^p A_l^p) \ \mathrm{d}\Omega, \tag{26}$$

where $\hat{A}^p$, ϕ_l^p and A_l^p are the test functions, shape functions and degrees of freedom associated with the discretization using p nodes, while the subscript q corresponds to the q-nodes discretization. As for the mesh adaptation technique, numerical diffusion is only introduced when the order of interpolation is reduced, while an increase in polynomial order does not modify the fields.

Although any order of interpolation from 0 to 15 can be used in the model, elements characterized by an even number of nodes are preferred and will be used in the applications to fully benefit from vectorized instructions[4]. With those instructions, two operations are executed simultaneously, and loops on arrays of even sizes become twice faster. The case of an odd number of nodes (and thus an even polynomial degree) yields complications in the SSE execution. Notice that this can be remedied by introducing padding in the array (i.e. extend the array by 1 and perform computation for both of the last entries).

3) ADAPTATION STRATEGY

The first step of the adaptation procedure is the identification of a set of elements to be refined or coarsened. This decision is based on an error indicator, which can be obtained in multiple ways. In this study, the jump in the values of the variables between each side of the interfaces of the elements is used as an error indicator, as described in Remacle et al. (2005). More details are given in the simulation descriptions.

Our strategy supposes a normal distribution of the error. Such suppositions were successful in other fields (see Remaki and Habashi (2006) for instance) and we perform this in order to avoid a parallel sorting algorithm step. Hence, the mean value μ and standard

---

[4]Streaming SIMD Extensions (SSE) to the x86 architecture.

deviation σ of the error estimator are computed. The refinement criteria is driven by

$$\text{if} \quad \begin{cases} \text{error} \leq \dfrac{\mu - f_u \sigma}{f_p} & \text{unrefine the element,} \\ \text{error} \geq \dfrac{\mu + f_r \sigma}{f_p} & \text{refine the element,} \end{cases} \tag{27}$$

where f_r and f_u can be tuned to improve the procedure. They are both set to 0.2 in this work. The error is computed as the average of the indicator in the considered element. When h-adaptation is considered, $f_p = 1$. For p-adaptation, it is set to $f_p = (p/q)^2$, where p is the current number of nodes in the element, while q is the number of nodes that will characterize the element if the refinement/unrefinement occurs. This is chosen principally because of the clustering of the LGL nodes near the edges of the elements. The closest nodes have a spacing of $h_{LGL} \sim 1/p^2$. Hence, a doubling of the number of LGL nodes corresponds to refining the element in each direction twice based on the smallest spacing (provided that the field is smooth). To avoid strong mesh inhomogeneities that could potentially generate spurious reflections, two neighboring elements can only be separated by one level of mesh refinement. However, no such restriction is enforced for the p refinement or derefinement procedure. Finally, since only time-dependent problems are considered, only one sweep of h refinement and p refinement is performed per adaptation step.

In order to reduce the cost of adaptation, the mesh is not modified every time step, but rather following a user-provided adaptation frequency. This frequency depends on the application considered. No significant difference was observed by adapting the mesh every 100 time steps rather than at each time step. For example, the maximum number of elements being at the wrong resolution (needing refinement/coarsening) because the refinement is not performed at each time step is always under 50 for the tsunami simulation presented in this article. It represents less than 1 percent of the total number of elements. However, to be more general, it may be useful to define a threshold of percentage of elements flagged for adaptation that would indicate when the adaptation procedure needs to be done.

In some situations, the key features of the flow may exit the refined zone, reducing considerably the accuracy of the solution. In those cases, it may be useful to refine in a broader area using a halo (St-Cyr et al. 2008) or more advanced (and more costly) techniques such as metrics advection to refine where the dynamics are complex, but also in the region where the flow will propagate (Wilson 2009).

4) LOAD BALANCING

To all the roots of the trees involved in the base mesh corresponds a barycenter. Using this barycenter, it is possible to employ a recursive coordinate bisection (RCB) to distribute the roots (hence all their leaves) amongst processors. Thus, the smallest load-balanceable entities are the trees associated with the elements in the base mesh. Each root, corresponding to a single element in the base mesh, is weighted by summing the work performed by all of its leaves (the refined active elements). We pick p^2, the variable number of nodes on an element, as a measure of the work associated with a leaf in the tree. Hence, the barycenter associated with the root of a tree is weighted by the total work occurring in its leaves. The weighted RCB developed in Devine et al. (2002); Boman et al. (2007) is then employed to

13

calculate a new properly rebalanced partition[5]. Finally, this new partition list is used to migrate the elements in parallel. Notice that the barycenter of each base element is fixed throughout the simulation and thus only re-computations of the weights are required. This makes the procedure entirely local and fast.

3. Validation

This section focuses on some of the test cases described by Williamson et al. (1992), chosen to validate the behavior of the model (theoretical order of convergence, accuracy for unsteady simulations, adaptation strategy). The section concludes with an application to a realistic tsunami propagation.

a. Global steady state nonlinear zonal geostrophic flow

This steady state configuration is the second test problem described in Williamson et al. (1992). The velocity field corresponds to a solid body rotation along an axis whose angle with the axis of rotation of the Earth is α. In this run, the latter is set to $\alpha = \pi/4$, such that the flow is not aligned with the grid (Fig. 3). The mesh employed is structured with a uniform static resolution. It consists in a cubed sphere (Ronchi et al. 1996; Rančić et al. 1996) made up of one element per cube face. The elements are then recursively split and projected onto the surface of the sphere to obtain finer meshes (Fig. 3).

The fields are initialized in order to establish a geostrophic equilibrium. The initial condition for the zonal and meridional velocities reads

$$
\begin{aligned}
u_z &= u_0 \left(\cos\theta\cos\alpha + \cos\lambda\sin\theta\sin\alpha \right), \\
u_m &= -u_0 \sin\lambda\sin\alpha.
\end{aligned}
\tag{28}
$$

where $u_0 = \frac{2\pi a}{12 \cdot 24 \cdot 3600}$ m s^{-1}. The longitude and latitude are respectively denoted by λ and θ. The mean depth is set by $gh = 2.94 \cdot 10^4$ m^2 s^{-2} with $g = 9.80616$ m s^{-2}. The initial free-surface elevation is given by

$$
\eta = -\frac{1}{g} \left(a\Omega u_0 + \frac{u_0^2}{2} \right) \left(-\cos\lambda\cos\theta\sin\alpha + \sin\theta\cos\alpha \right)^2,
\tag{29}
$$

in which $a = 6.37122 \cdot 10^6$ m denotes the Earth radius and $\Omega = 7.292 \cdot 10^{-5}$ s^{-1} is the rotation rate of the Earth. The vectors and coordinates are expressed in a longitude/latitude basis, and are introduced in the Cartesian framework of the model using appropriate transformations:

$$
\begin{pmatrix} \lambda \\ \theta \end{pmatrix} = \begin{pmatrix} \arctan\left(y/x\right) \\ \arcsin\left(z/a\right) \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} u_z \sin(\lambda) - u_m \cos(\theta)\cos(\lambda) \\ -u_z \cos(\lambda) - u_m \cos(\theta)\sin(\lambda) \\ u_m \sin(\theta) \end{pmatrix}.
\tag{30}
$$

Considering the availability of an analytical solution with continuous derivatives, this test case is convenient to check the numerical convergence of the model. Using elements

---

[5]http://www.cs.sandia.gov/Zoltan

Q_i of order i, the error should theoretically converge to 0 at the rate $i + 1$. This optimal rate of convergence is observed for the different high-order elements tested in this study (Fig. 4). The meshes R_0 to R_3 have been used, corresponding to mesh resolutions at the equator varying from 90° to 11.25°. This mesh resolution is different from the overall spatial resolution, the latter being much higher thanks to the high-order representation within the elements (maximum resolution around 1.5°). For this convergence study, the time step was selected in such a way that the temporal error is negligible with respect to the spatial one. When the spatial discretization order becomes very high, such a requirement would lead to critically too tiny time steps. Therefore, it was not possible to carry on convergence mesh analysis on the finest mesh with Q_5 and Q_7.

b. Zonal flow over an isolated mountain

This is the fifth test case as described in Williamson et al. (1992). The simulation of a flow over an isolated mountain is initialized using the same velocity and free-surface elevation fields as the global steady state nonlinear zonal geostrophic flow (28,29), but using $\alpha = 0$ and $u_0 = 20$ m s^{-1}. The mean depth is modified to include an isolated mountain centered around ($\lambda_c = 3\pi/2$, $\theta_c = \pi/6$):

$$h = h_0 - h_s, \tag{31}$$

where $h_0 = 5960$ m. The mountain height is given by

$$h_s = h_{s0}\left(1 - \frac{r}{R}\right) \tag{32}$$

in which $R = \pi/9$ and $r^2 = \min\left[R^2, (\lambda - \lambda_c)^2 + (\theta - \theta_c)^2\right]$. The maximum height of the mountain is set to $h_{s0} = 2000$ m.

This test case allows a quantitative estimation of the discretization error thanks to the existence of a high resolution solution from the German Weather Service made available online[6]. This solution results from a T426 spectral simulation using the NCAR shallow water spectral model (Jakob-Chien et al. 1995) with a resolution equivalent to about 31 km at the equator.

The simulations were performed with several meshes and orders of interpolations. When dynamic adaptation is used (Fig. 5), the mesh is refined locally where the flow is considered unresolved, in order to reduce the error generated by the model. The refinement criteria for $p-$ and $h-$adaptation correspond to (27). It was previously shown in Bernard et al. (2007) that the jump in the values of the variables between each side of the interfaces of the elements consists in a good representation of the discretization error. Therefore, the latter can be used as an error indicator to identify where the mesh should be refined. Thus, in this simulation, we chose the jump of the free-surface elevation η as the error indicator. Other configurations may be more effective, as for example using a combination of the jumps of different variables or taking into account the smoothness of the fields inside each element. The design of an optimal adaptation strategy has still to be achieved and will be the subject of future work. Simulations were also performed using a threshold on the absolute value of the relative vorticity as an adaptation criteria, as done by St-Cyr et al. (2008).

---

[6]http://icon.enes.org/swm/stswm/node5.html

The mesh is initially refined around the mountain (Fig. 5). Then, it is adapted every 100 time steps according to the distribution of the estimated error. During the first days of simulation, the dynamic refinement mainly occurs in the wake of the mountain where the flow is disturbed. After 10 and 15 days of simulation, the adaptation takes place in various areas of domain that are affected by the complex structure of the flow. The adaptation criteria is the same for both $h-$ and $p-$refinement (27). Hence, the order of interpolation is adapted around the same areas than the mesh.

A very similar distribution of the difference with the solution obtained from the German Weather Service (Fig. 5) is observed in other studies such as Comblen et al. (2009). There is a correlation between the peaks of this difference and the areas where the mesh is refined. However, the mesh is not systematically refined where the difference with the reference solution is high. Indeed, if an error is accumulated and transported, it will not be visible in the jump of the variables at the interfaces between the elements. Refining the mesh would not improve the solution anyway as the error has been generated earlier.

Different simulations for this test case were performed: (1) using both fixed grids and orders of interpolation, (2) having recourse to dynamic adaptation. As expected, the model converges to the reference solution as the resolution is increased (Fig. 6). Increasing the order of interpolation is much more efficient than reducing the grid size by adding elements because the convergence rate increases with respect to the former while the rate remains constant with the latter. Considering the vorticity as an error estimator (St-Cyr et al. 2008) generates a higher global error compared to simulations using the jump of the elevation of the free-surface to identify the areas needing refinement. As pointed out by St-Cyr et al. (2008), an additional level of refinement is required in order for the adaptive simulation to surpass the static one. The resolution characterising the adaptive simulation reaches the resolution of the non-adaptive simulation only in the areas where the dynamic refinement is maximum. It is lower anywhere else in the domain. Thus, it is obvious that an adaptive simulation will always be less accurate than the corresponding non-adaptive simulation based on the maximum level of refinement used by the adaptive one (e.g. the error associated with the simulation $Q_{3-5}R_{2-3}$ is higher than the one obtained by the simulation Q_5R_3). However, an optimal adaptation strategy should greatly increase the accuracy of the adaptive simulation.

This test case, characterized by a large scale flow with a similar complexity over the whole domain, is not multiscale enough to really benefit from mesh adaptation. A tsunami simulation will now be described to illustrate the computational potential of dynamic adaptation for realistic multiscale configurations.

c. Global tsunami simulation

Although idealized test cases are useful to check the desirable properties of a numerical discretization, a more demanding application is required to assess the behavior of a model under realistic conditions, especially its efficiency for the simulation of multiscale problems. In this study, we consider the propagation through the global ocean of the February 27, 2010 tsunami in Chile. The bottom topography of the ocean is very steep, with a water depth varying from 0 m to more than 5000 m in a single element (Fig. 7). This application is thus a good test case to check the robustness of the model when applied to complex realistic configurations.

The bathymetry of the ocean is derived from the ETOPO data (Amante and Eakins 2009). As no wetting-drying scheme is used, a minimum depth of 8 m is considered and the coasts correspond to closed reflective boundaries. There is no open boundary as the computational domain encompasses the global world ocean. The initial condition is computed by assuming that the initial sea surface displacement is equal to the static sea floor uplift caused by an abrupt slip at the Nazca and South American plates interface. This uplift is obtained using Okada's static dislocation formula (Okada 1985) with the coefficients described in Table 1, originating from the University of Bologna Tsunami Research Team[7]. The bottom stress is computed using the Manning formula $\boldsymbol{\tau^b} = -\rho g \frac{n^2}{H^{1/3}}|\mathbf{u}|\mathbf{u}$, with a coefficient $n = 0.03$.

The simulation is computed on an initial mesh built using the GMSH[8] software (Geuzaine and Remacle 2009), with a resolution varying from 5 km up to 1000 km (Fig. 7). The mesh is initially refined using the AMR technique where the initial free-surface elevation gradient exceeds a certain threshold (Fig. 8). The *hp*-adaptation is then performed every 100 time steps using the jumps of the free-surface elevation η at the interface between elements as an error indicator and (27) as adaptation criteria. With two recursive levels of refinement (R_0 to R_2, meaning that each element of the initial mesh can be recursively split twice), the maximum mesh resolution during the simulation is close to 1 km. Due to the high-order shape functions (Q_3 and Q_5 elements, their distribution being driven by the adaptation procedure), the resolution of the discretization is much higher than the mesh resolution and very close to 250m.

As long as the tsunami propagates through the Pacific Ocean, the order of interpolation and the mesh are adapted to precisely track the waves (Fig. 8). The computational power is used effectively by concentrating the load at the front of the wave, where it is needed to accurately resolve the propagation. At the end of the simulation, when the front of the wave is broken by hitting the coasts (Fig. 9), the adaptation procedure focuses on many reflections of the initial wave and the refined areas spread through the domain. Notice that those reflections would be reduced with the inclusion of a wetting-drying (runoff) model.

The elevation of the free-surface has been compared with the DART[9] data from the NOAA Center for Tsunami Research at eight different stations[10]. It is seen that the model estimates accurately the time at which the tsunami reaches the different stations (Fig. 10). The amplitude of the waves is also well predicted, except for the stations 5 and 8. Those stations being located in very shallow areas with an irregular bathymetry and several small islands. The resolution of the model is probably not sufficient to reproduce accurately the height of the waves in those regions.

It can be seen that, after an initialization period, the total number of elements forming the mesh as well as the proportion of higher order elements is rather stable and does not increase without bound (Fig. 11). As a consequence, the evolution of the number of degrees of freedom behaves similarly. The abrupt diminution of the number of elements after about 500 minutes corresponds to the moment when the tsunami leaves the vicinity of the earthquake, where initially the original GMSH mesh was refined. Then, several elements disappear

---

[7] http://labtinti4.df.unibo.it/tsunami

[8] http://www.geuz.org/gmsh

[9] Deep-ocean Assessment and Reporting of Tsunamis.

[10] This data is available for download at http://nctr.pmel.noaa.gov/Dart

simultaneously the moment this area is coarsened.

Statistics involving the computational cost were compared with a corresponding simulation that did not employ dynamic adaptation but was characterized by the same maximum resolution (Table 2). The static simulation was obtained by using the maximum order of interpolation (Q_5) and the maximum level of recursive mesh refinements (R_2) over the whole domain. In contrast, the adaptive simulation uses maximal resolution only where the dynamics are the most demanding. The comparison shows that the number of degrees of freedom can be reduced by a factor of almost ten by using dynamic adaptation. The gain in efficiency, denoted by the simulation time, is slightly inferior. This is due to the cost of the adaptation procedure which includes parallel load balancing. The simulation time related to the configuration without adaptation has been obtained by computing only the first hour of physical time, a complete simulation being too long to run. When no adaptation is involved the time-step is constant, since it is constrained by the gravity waves' maximal speed. Thus, the computational cost of each minute of physical time remains the same throughout the simulation.

4. Conclusions

A discontinuous Galerkin shallow water hp-adaptive model on the sphere has been described extensively. The resolution of the equations on the sphere as well as the adaptation procedure allow for the simulation of multiscale geophysical flows.

The model has been validated on classical shallow water atmospheric test cases, producing the right order of convergence (test case #2) as well as results in agreement with a high resolution reference solution (test case #5). The simulation of the February 2010 Chilean tsunami completed the validation procedure on a realistic configuration by predicting perturbations of the free-surface of the ocean comparable to the DART buoy measurements.

The tsunami configuration demonstrates the potential of dynamic spatial adaptation as a mean to improve the efficiency of multiscale simulations. The highest resolution remains focused on the front of the wave, where the fields are characterized by a sharp distribution. As a consequence, the adaptive tsunami simulation was shown to be almost ten times faster than the statically refined simulation of an equivalent resolution. The computational time being faster than the physical time on a reasonable number of processors, the adaptive model may be integrated in a high resolution tsunami warning process.

It is interesting to notice that, even for the highly unresolved tsunami simulation exhibiting sharp gradients, no filter was needed to keep the model stable. Moreover, no smoothing of the bathymetry was required. There is no additional diffusion apart from the numerical diffusion built in the scheme. Hence, the stabilization provided by the DG method is sufficient to make the model robust, even in the case of difficult configurations.

While a rather simple adaptation strategy was used herein, more advanced adaptation criteria will be investigated. In particular, the decision to refine either the mesh or change the order of interpolation must be improved by using a distinct criteria. The way to obtain the error indicators may also need to be enhanced. For instance by taking into account the field smoothness.

This paper constitutes a proof of concept for the efficient use of hp-adaptation for atmospheric and oceanic simulations. Future work will focus on improved adaptation criteria,

18

three-dimensional extensions, local time-stepping (Lörcher et al. 2008), limiting strategies and physical parameterizations in order to reach the goal of developing an effective multiscale model for environmental flows.

REFERENCES

Aizinger, V. and C. Dawson, 2002: A discontinuous Galerkin method for two-dimensional flow and transport in shallow water. *Advances in Water Resources*, **25**, 67–84, doi:10. 1016/S0309-1708(01)00019-7.

Amante, C. and B. W. Eakins, 2009: Etopo1 1 arc-minute global relief model: Procedures, data sources and analysis. Tech. rep., NOAA Technical Memorandum NESDIS NGDC-24.

Bernard, P.-E., N. Chevaugeon, V. Legat, E. Deleersnijder, and J.-F. Remacle, 2007: High-order *h*-adaptive discontinuous Galerkin methods for ocean modelling. *Ocean Dynamics*, **57**, 109–121.

Bernard, P.-E., J.-F. Remacle, R. Comblen, V. Legat, and K. Hillewaert, 2009: High-order discontinuous Galerkin schemes on general 2d manifolds applied to the shallow water equations. *Journal of Computational Physics*, **228 (17)**, 6514–6535.

Bhanot, G., et al., 2008: Early experiences with the 360tf ibm blue gene/l platform. *International Journal of Computational Methods*, **5**, 237–253.

Blaise, S., R. Comblen, V. Legat, J.-F. Remacle, E. Deleersnijder, and J. Lambrechts, 2010a: A discontinuous finite element baroclinic marine model on unstructured prismatic meshes. Part I: space discretization. *Ocean Dynamics*, **60 (6)**, 1371–1393, doi: 10.1007/s10236-010-0358-3, *Special issue: Multi-scale modelling of coastal, shelf and global ocean dynamics.*

Blaise, S., B. de Brye, A. de Brauwere, E. Deleersnijder, E. J. M. Delhez, and R. Comblen, 2010b: Capturing the residence time boundary layer - application to the Scheldt Estuary. *Ocean Dynamics*, **60**, 535–554, doi:10.1007/s10236-010-0272-8.

Boman, E., et al., 2007: *Zoltan 3.0: Parallel Partitioning, Load-balancing, and Data Management Services; User's Guide.* Albuquerque, NM, Sandia National Laboratories, tech. Report SAND2007-4748W.

Burstedde, C., O. Ghattas, M. Gurnis, T. Isaac, G. Stadler, T. Warburton, and L. Wilcox, 2010: Extreme-Scale AMR. *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, IEEE Computer Society, Washington, DC, USA, 1–12, SC '10.

Chen, C., F. Xiao, and X. Li, 2011: An adaptive multi-moment global model on cubed sphere. *Monthly Weather Review*, doi:10.1175/2010MWR3365.1, in press.

Cockburn, B., G. Karniadakis, and S.-W. Shu, 2000: *Discontinous Galerkin Methods Theory, Computation and Applications*, Lecture Notes in Compuational Science and Engineering, Vol. 11. Springer, Berlin.

Cockburn, B. and C.-W. Shu, 2001: Runge-Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems. *Journal of Scientific Computing*, **16 (3)**, 173–261.

Comblen, R., S. Blaise, V. Legat, J.-F. Remacle, E. Deleersnijder, and J. Lambrechts, 2010: A discontinuous finite element baroclinic marine model on unstructured prismatic meshes. Part II: implicit/explicit time discretization. *Ocean Dynamics*, **60 (6)**, 1395–1414, doi: 10.1007/s10236-010-0357-4, *Special issue: Multi-scale modelling of coastal, shelf and global ocean dynamics.*

Comblen, R., S. Legrand, E. Deleersnijder, and V. Legat, 2009: A finite element method for solving the shallow water equations on the sphere. *Ocean Modelling*, **28**, 12–23, doi: 10.1016/j.ocemod.2008.05.004.

Côté, J., 1988: A Lagrange multiplier approach for the metric terms of semi-Lagrangian models on the sphere. *Quarterly Journal of the Royal Meteorological Society*, **114**, 1347–1352, doi:10.1002/qj.49711448310.

Dennis, J., A. Fournier, W. F. Spotz, A. St-Cyr, M. A. Taylor, S. J. Thomas, and H. Tufo, 2005a: High-Resolution Mesh Convergence Properties and Parallel Efficiency of a Spectral Element Atmospheric Dynamical Core. *International Journal of High Performance Computing Applications*, **19 (3)**, 225–235.

Dennis, J., et al., 2011: CAM-SE: A scalable spectral element dynamical core for the Community Atmosphere Model. *International Journal of High Performance Computing Applications*, submitted.

Dennis, J. M., M. Levy, R. D. Nair, H. M. Tufo, and T. Voran, 2005b: Towards an efficient and scalable discontinuous Galerkin atmospheric model. *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05) - Workshop 13 - Volume 14*, IEEE Computer Society, 7pp.

Deville, M. O., P. F. Fischer, and E. H. Mund, 2002: *High-order methods for incompressible fluid flow*, Cambridge Monographs on Applied and Computational Mathematics, Vol. 9. Cambridge University Press.

Devine, K., E. Boman, R. Heaphy, B. Hendrickson, and C. Vaughan, 2002: Zoltan data management services for parallel dynamic applications. *Computing in Science and Engineering*, **4 (2)**, 90–97.

Edwards, H. C., 2002: SIERRA framework version 3: Core services theory and design. Tech. Rep. SAND2002-3616, Sandia National Laboratories.

Emanuel, K., 2005: Increasing destructiveness of tropical cyclones over the past 30 years. *Nature*, **436**, 686–688.

Eskilsson, C., 2010: An hp-adaptive discontinuous Galerkin method for shallow water flows. *International Journal for Numerical Methods in Fluids*, doi:10.1002/fld.2434, in press.

Geuzaine, C. and J.-F. Remacle, 2009: Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, **11**, 1309–1331.

Giraldo, F. X., 2001: A spectral element shallow water model on spherical geodesic grids. *International Journal for Numerical Methods in Fluids*, **35 (8)**, 869–901.

Giraldo, F. X., 2006: High-order triangle-based discontinuous Galerkin methods for hyperbolic equations on a rotating sphere. *Journal of Computational Physics*, **214 (2)**, 447–465.

Giraldo, F. X., J. S. Hesthaven, and T. Warburton, 2002: Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations. *Journal of Computational Physics*, **181**, 499–525.

Giraldo, F. X. and M. Restelli, 2008: A study of spectral element and discontinuous Galerkin methods for the navier-stokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases. *Journal of Computational Physics*, **227**, 3849–3877, doi:10.1016/j.jcp.2007.12.009.

Gottlieb, S., D. Ketcheson, and C.-W. Shu, 2011: *Strong Stability Preserving Runge-Kutta and Multistep Time Discretizations*. World Scientific Publishing Company.

Gottlieb, S. and C.-W. Shu, 1998: Total variation diminishing Runge-Kutta schemes. *Mathematics of Computation*, **67**, 73–85, doi:10.1090/S0025-5718-98-00913-2.

Grabowski, W. W., 2001: Coupling cloud processes with the large-scale dynamics using the cloud resolving convection parameterization (CRCP). *Journal of Atmospheric Sciences*, **(58)**, 978–997.

Hesthaven, J. S. and T. Wartburton, 2008: *Nodal Discontinuous Galerkin Methods. Algorithms, Analysis and Applications*, Texts in Applied Mathematics, Vol. 54. Springer, Berlin.

IPCC, 1997: Special Report. The Regional Impacts of Climate Change: An Assessment of Vulnerability.

Jakob-Chien, R., J. J. Hack, and D. L. Williamson, 1995: Spectral transform solutions to the shallow water test set. *Journal of Computational Physics*, **119 (1)**, 164–187.

Karniadakis, G. E. and S. J. Sherwin, 2005: *Spectral/hp Element Methods for Computational Fluid Dynamics (Numerical Mathematics and Scientific Computation)*. Oxford University Press, USA.

Khairoutdinov, M. F. and D. A. Randall, 2001: A cloud resolving model as a cloud parameterization in the NCAR community climate system model: Preliminary results. *Geophysical Research Letters*, **28 (18)**, 3617–3620.

Khouider, B., A. St-Cyr, A. J. Majda, and J. Tribbia, 2011: MJO and convectively coupled waves in a coarse resolution GCM with a simple multicloud parametrization. *Journal of the Atmospheric Sciences*, doi:10.1175/2010JAS3443.1, in press.

Kopriva, D. A. and G. Gassner, 2010: On the quadrature and weak form choices in collocation type discontinuous Galerkin spectral element methods. *Journal of Scientific Computing*, **44**, 136–155.

Kopriva, D. A., S. L. Woodruff, and M. Y. Hussaini, 2002: Computation of electromagnetic scattering with a non-conforming discontinuous spectral element method. *International Journal for Numerical Methods in Engineering*, **53 (1)**, 105–122.

Kubatko, E. J., S. Bunya, C. Dawson, and J. J. Westerink, 2009: Dynamic p-adaptive Runge-Kutta discontinuous Galerkin methods for the shallow water equations. *Computer Methods in Applied Mechanics and Engineering*, **198 (21-26)**, 1766–1774.

Kubatko, E. J., J. J. Westerink, and C. Dawson, 2006: *hp* discontinuous Galerkin methods for advection dominated problems in shallow water flows. *Computer Methods in Applied Mechanics and Engineering*, **196**, 437–451, doi:10.1016/j.cma.2006.05.002.

Läuter, M., D. Handorf, N. Rakowsky, J. Behrens, S. Frickenhaus, M. Best, K. Dethloff, and W. Hiller, 2007: A parallel adaptive barotropic model of the atmosphere. *Journal of Computational Physics*, **223 (2)**, 609 – 628.

Löhner, R. and J. Baum, 1991: Numerical simulation of time-dependent 3-d flows using adaptive unstructured grids. *Advances in the Free-Lagrange Method Including Contributions on Adaptive Gridding and the Smooth Particle Hydrodynamics Method*, H. Trease, M. Fritts, and W. Crowley, Eds., Springer Berlin / Heidelberg, Lecture Notes in Physics, Vol. 395, 47–56.

Lörcher, F., G. Gassner, and C. Munz, 2008: An explicit discontinuous Galerkin scheme with local time-stepping for general unsteady diffusion equations. *Journal of Computational Physics*, **227**, 5649–5670.

Mann, M. E. and K. A. Emanuel, 2006: Atlantic hurricane trends linked to climate change. *Eos, Transactions American Geophysical Union*, **87 (24)**, 223–244.

McClean, J. L., et al., 2010: A prototype two-decade fully-coupled fine-resolution CCSM simulation. Tech. Rep. ANL/MCS-P1766-0610, Argonne National Laboratories.

Mohseni, K. and T. Colonius, 2000: Numerical treatment of polar coordinate singularities. *Journal of Computational Physics*, **157**, 787–795.

Nair, R., H.-W. Choi, and H. Tufo, 2009: Computational aspects of a scalable high-order discontinuous Galerkin atmospheric dynamical core. *Computers & Fluids*, **38 (2)**, 309 – 319.

Nair, R. D., S. J. Thomas, and R. D. Loft, 2005a: A discontinuous Galerkin global shallow water model. *Monthly Weather Review*, **133**, 876–888, doi:10.1175/MWR2903.1.

Nair, R. D., S. J. Thomas, and R. D. Loft, 2005b: A discontinuous Galerkin transport scheme on the cubed sphere. *Monthly Weather Review*, **133 (4)**, 814–828.

Neale, R. B., et al., 2010: Description of the NCAR community atmosphere model (CAM 4.0). Tech. rep., NCAR.

Okada, Y., 1985: Surface deformation due to shear and tensile faults in a half-space. *Bulletin of the Seismological Society of America*, **75**, 1135–1154.

Pedlosky, J., 1986: *Geophysical Fluid Dynamics, 2nd Edition.* Springer.

Rančić, M., R. J. Purser, and F. Messinger, 1996: A global shallow-water model using an expanded spherical cube: Gnomonic versus conformal coordinates. *Quarterly Journal of the Royal Meteorological Society*, **122 (532)**, 959–982.

Remacle, J.-F., S. S. Frazao, L. Xiangrong, and M. S. Shephard, 2006: An adaptive discretization of shallow-water equations based on discontinuous Galerkin methods. *International Journal for Numerical Methods in Fluids*, **52 (8)**, 903–92.

Remacle, J.-F., X. Li, M. S. Shephard, and J. E. Flaherty, 2005: Anisotropic adaptive simulation of transient flows using discontinuous Galerkin methods. *International Journal for Numerical Methods in Engineering*, **62 (7)**, 899–923.

Remaki, L. and W. G. Habashi, 2006: 3-d mesh adaptation on multiple weak discontinuities and boundary layers. *SIAM Journal of Scientific Computing*, **28**, 1379–1397, doi:10.1137/S1064827503429879.

Ronchi, C., R. Iacono, and P. S. Paolucci, 1996: The "Cubed Sphere": A new method for the solution of partial differential equations in spherical geometry. *Journal of Computational Physics*, **124 (1)**, 93–114.

Satoh, M., T. Matsuno, H. Tomita, H. Miura, T. Nasuno, and S. Iga, 2008: Nonhydrostatic icosahedral atmospheric model (NICAM) for global cloud resolving simulations. *Journal of Computational Physics*, **227**, 3486–3514, doi:10.1016/j.jcp.2007.02.006.

Slingo, J., et al., 2009: Developing the next-generation climate system models: challenges and achievements. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **367 (1890)**, 815–831.

St-Cyr, A., C. Jablonowski, J. M. Dennis, H. M. Tufo, and S. J. Thomas, 2008: A comparison of two shallow-water models with nonconforming adaptive grids. *Monthly Weather Review*, **136**, 1898–1922.

St-Cyr, A. and D. Neckels, 2009: A fully implicit jacobian-free high-order discontinuous Galerkin mesoscale flow solver. *Proceedings of the 9th International Conference on Computational Science*, Springer-Verlag, Berlin, Heidelberg, 243–252, ICCS 2009.

Stewart, J. R. and H. C. Edwards, 2002: SIERRA framework version 3: $h-$adaptivity design and use. Tech. Rep. SAND2002-4016, Sandia National Laboratories.

Wehner, M., 2008: Towards Ultra-High Resolution Models of Climate and Weather. *International Journal of High Performance Computing Applications*, **22 (2)**, 149–156.

Weller, H., H. G. Weller, and A. Fournier, 2009: Voronoi, delaunay, and block-structured mesh refinement for solution of the shallow-water equations on the sphere. *Monthly Weather Review*, **137 (12)**, 4208–4224, doi:10.1175/2009MWR2917.1.

Wheeler, M. and G. N. Kiladis, 1999: Convectively-coupled equatorial waves: Analysis of clouds and temperature in the wavenumber-frequency domain. *Journal of Atmospheric Sciences*, **(56)**, 374–399.

Wilcox, L. C., G. Stadler, C. Burstedde, and O. Ghattas, 2010: A high-order discontinuous Galerkin method for wave propagation through coupled elastic-acoustic media. *Journal of Computational Physics*, **229**, 9373–9396.

Williamson, D. L., J. B. Drake, J. J. Hack, R. Jakob, and P. N. Swarztrauber, 1992: A standard test set for numerical approximations to the shallow water equations in spherical geometry. *Journal of Computational Physics*, **102 (1)**, 211–224.

Wilson, C., 2009: Modelling multiple-material flows on adaptive unstructured meshes. Ph.D. thesis, Imperial College London.

List of Tables

Strike	11°
Dip	25°
Rake	98°
Slip	18.7 m
Length	483000 m
Width	100000 m
Depth	1350 m
Longitude	-73.59°
Longitude	-35.87°

TABLE 1. Seismic parameters used to derive the earthquake initial uplift with the formula from Okada (1985).

	Adaptation	No adaptation	Ratio
Elements **(mean)**	6627 $\rightarrow$ 23208 **(18693)**	88368	**(4.7)**
Nodes per element **(mean)**	4x4 $\rightarrow$ 6x6 **(4.2x4.2)**	6x6	**(1.4x1.4)**
Degrees of freedom **(mean)**	125752 $\rightarrow$ 433152 **(344210)**	3181248	**(9.2)**
Simulation time [s]	52860	426500	8.1

TABLE 2. Comparison of the cost statistics between the simulation with dynamic adaptation and the corresponding simulation with fixed mesh and order of interpolation, characterized by the same maximum resolution. The ratio is an indicator of the gain of efficiency attributable to the adaptation procedure. The simulation times are obtained using 64 processors on a Cray XT5m for a simulation covering a physical time of 2000 minutes.

List of Figures

FIG. 1. Example of Gauss-Legendre (GL) and Legendre-Gauss-Lobatto (LGL) quadrature nodes for a quadrilateral reference element with 4x4 nodes.

Fig. 2. Adaptive mesh refinement process. Prolongation/restriction operators are needed to locally transfer the data between the different grids.

FIG. 3. The high-order meshes used for the different simulations are obtained by splitting recursively the elements of the initial cube mesh R_0 and projecting them onto the surface of the sphere. MIDDLE: Initial depth (colors) and transport field (arrows) for the test case #2 from Williamson et al. (1992).
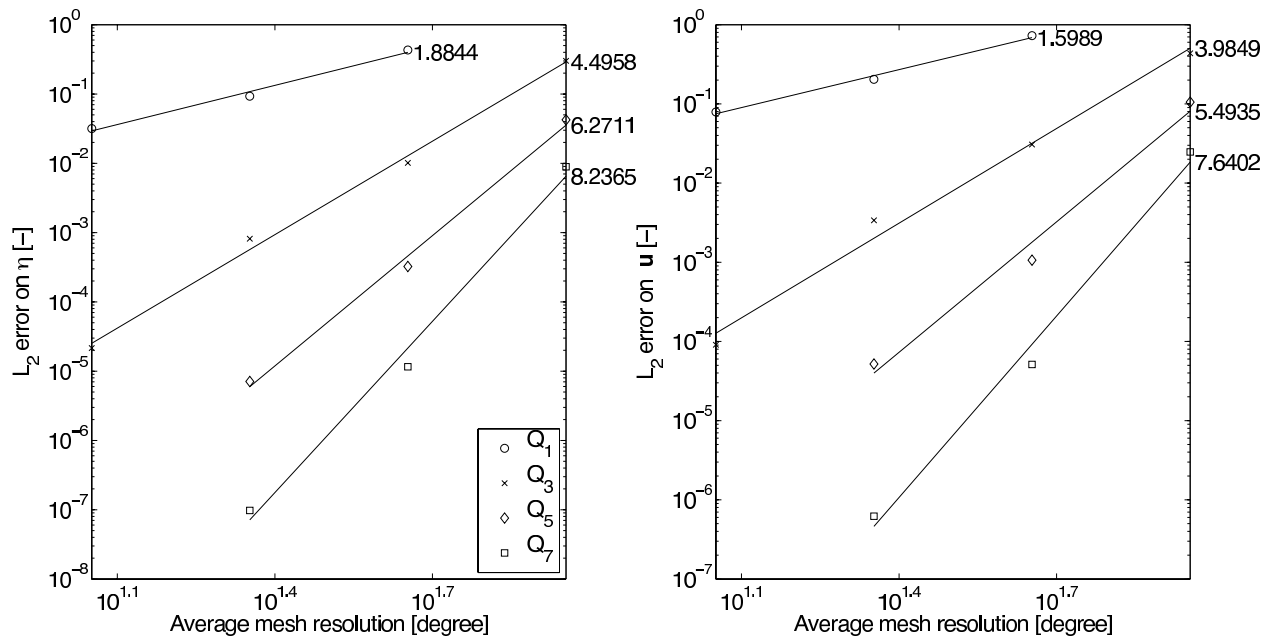
FIG. 4. Convergence of the L_2 error for the test case #2 from Williamson et al. (1992), after five days of simulation. The mean rate of convergence is indicated for each order of interpolation.
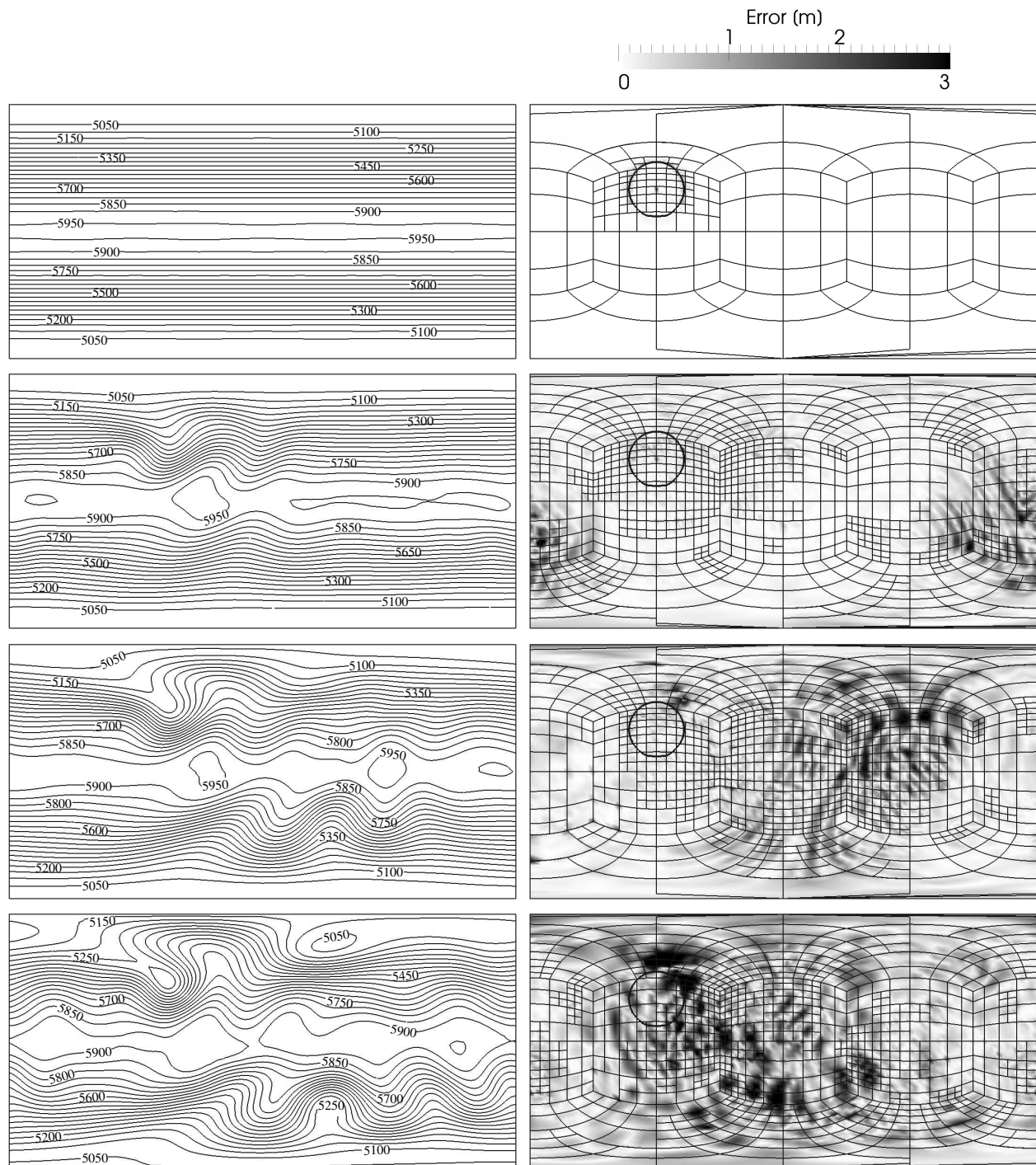
FIG. 5. Snapshots of the simulation for the test case #5 from Williamson et al. (1992) after 0, 5, 10 and 15 days (from top to bottom). Two levels of dynamic mesh refinement are considered (R_{2-4}), with two different orders of interpolation (Q_{3-5}). The Cartesian domain has been mapped to longitude/latitude coordinates. LEFT: depth contour lines from 5050 m to 5950 m with an interval of 50 m. RIGHT: absolute error on the depth $\|H - H_{ref}\|$, using the solution from the German Weather Service as the reference H_{ref}. The black lines represent the dynamically adapted mesh while the circle indicates the contour of the mountain.
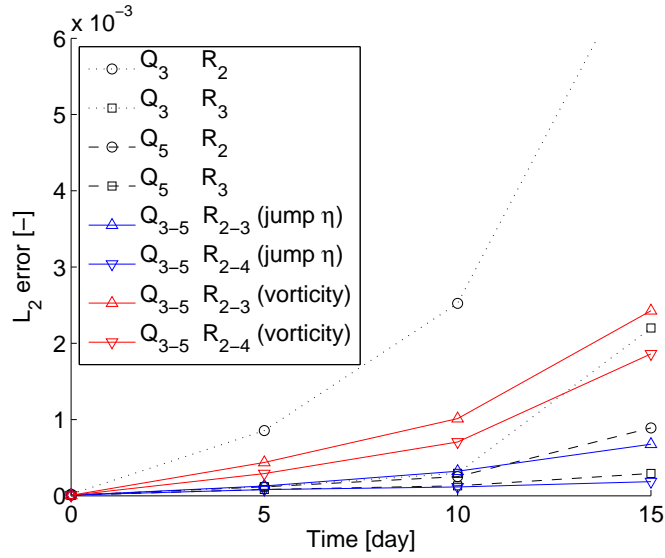
FIG. 6. Evolution of the normalized L_2 error for the test case #5 from Williamson et al. (1992), using the solution from the German Weather Service as the reference. Different meshes and orders of interpolation are considered. The dotted and dashed lines correspond to fixed orders of interpolation and meshes, while the plain lines are obtained using dynamic adaptation. The error estimator is either the jump of the free-surface elevation η at the interface between elements (blue) or the absolute value of the relative vorticity (red).
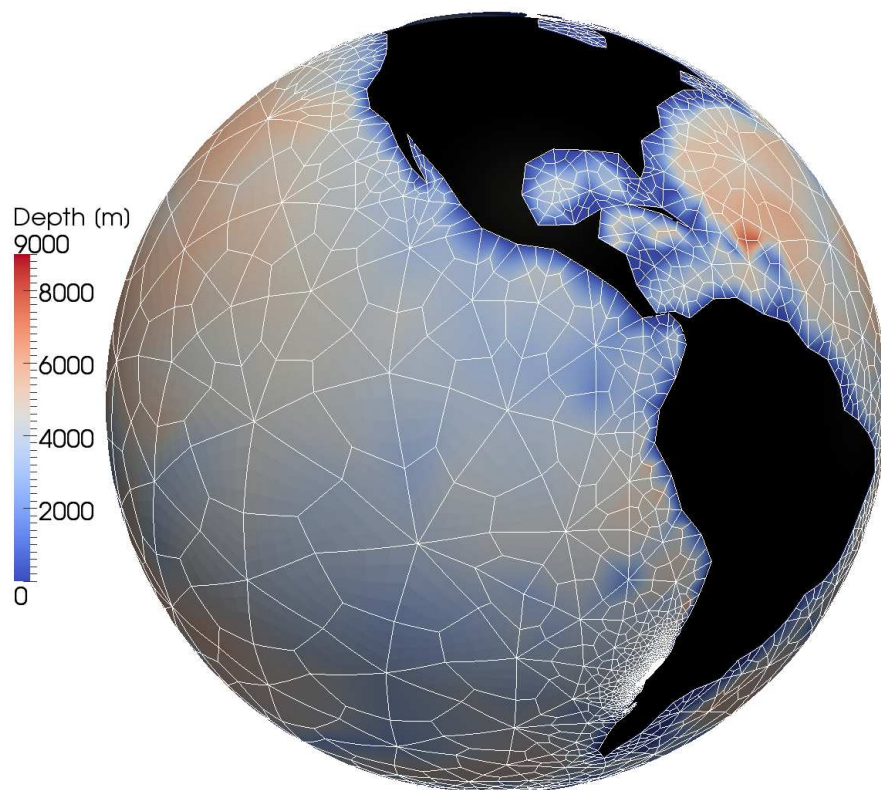
FIG. 7. Water depth at rest and initial mesh before dynamic adaptation, used for the tsunami simulation.
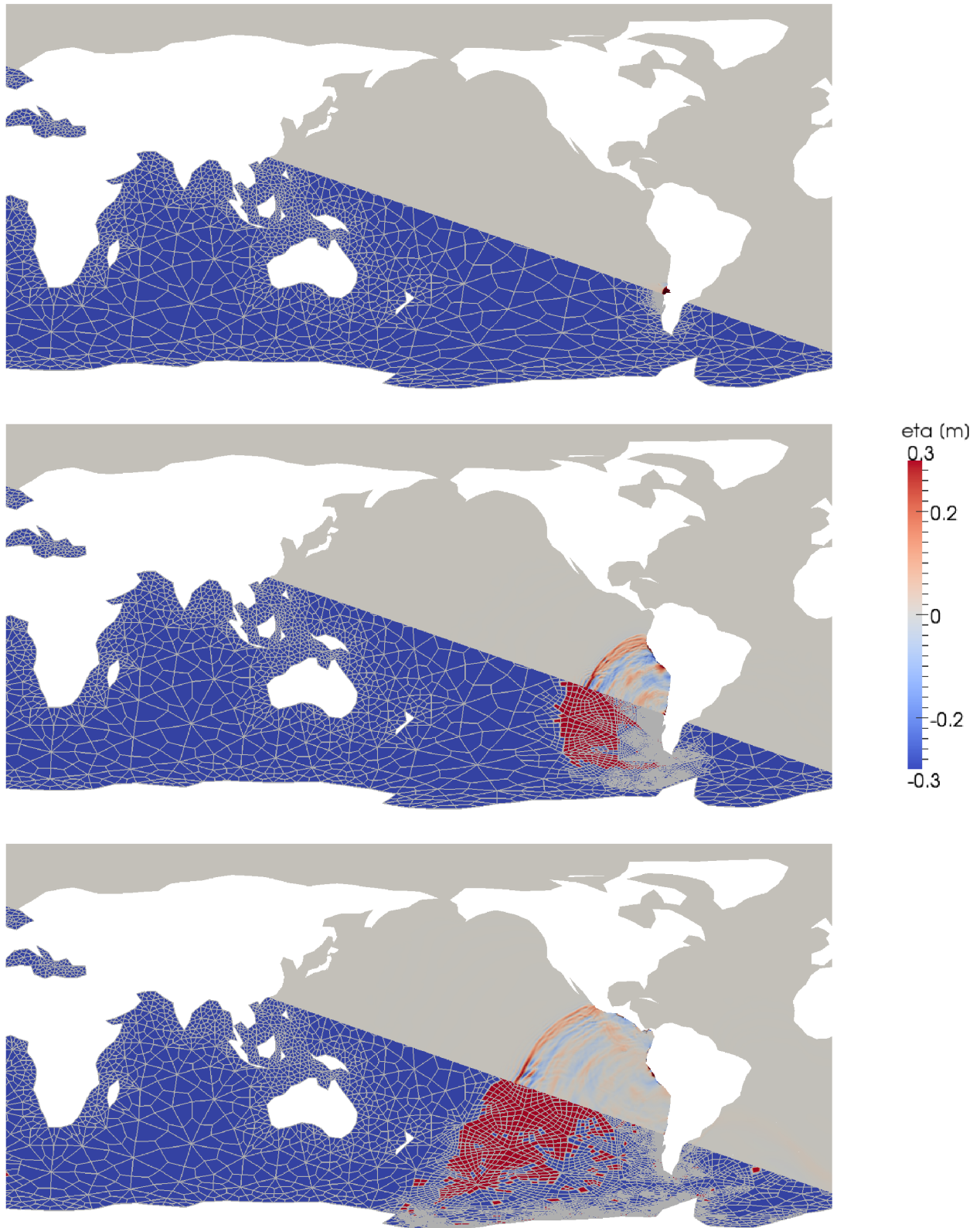
FIG. 8. Propagation of the wave after 0h, 5h and 10h. UPPER PART: Free-surface elevation. LOWER PART: State of the mesh with order of interpolation (blue=Q_3, red=Q_5). The Cartesian domain has been mapped to longitude/latitude coordinates.
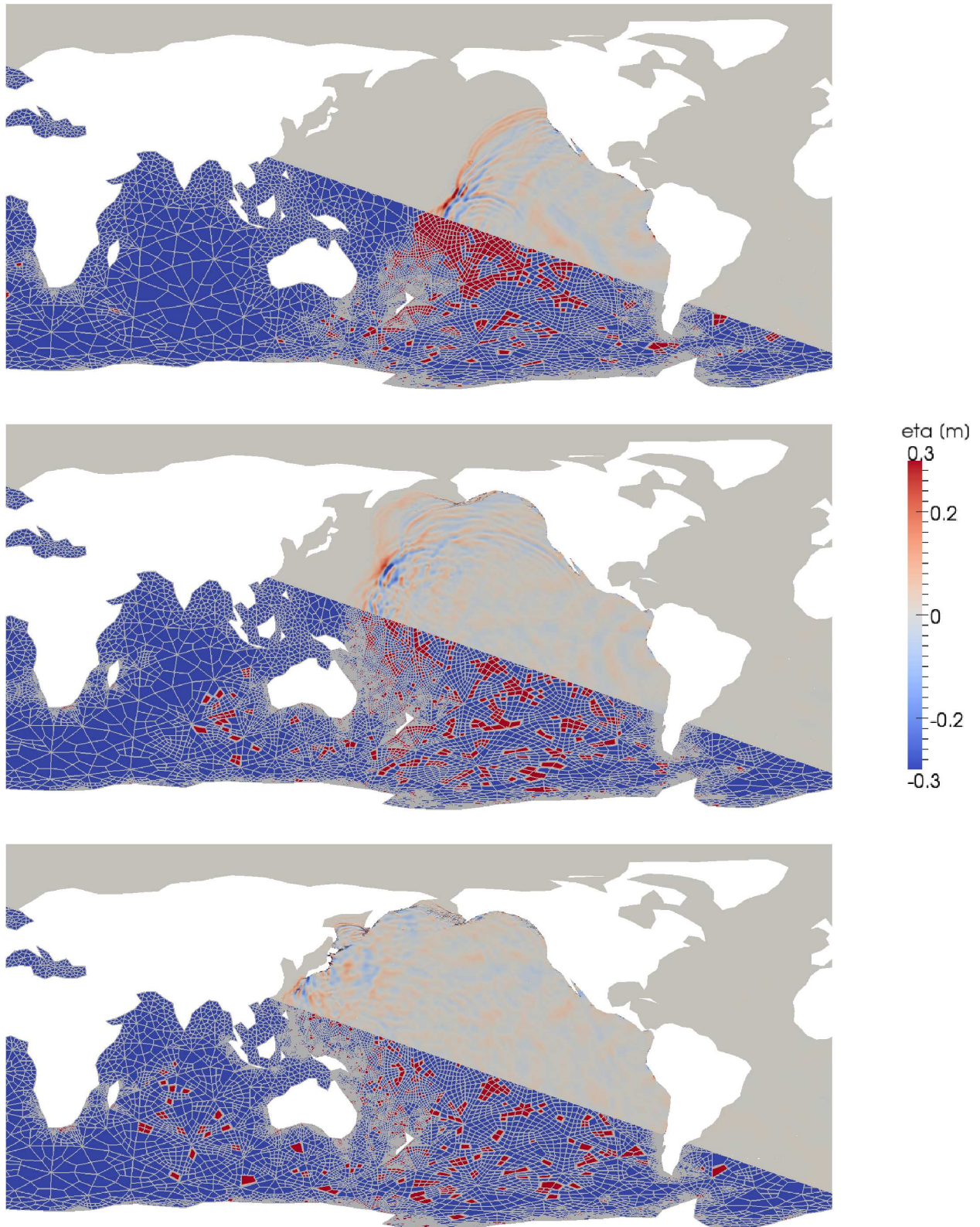
39

FIG. 9. Propagation of the wave after 15h, 20h and 25h. UPPER PART: Free-surface elevation. LOWER PART: State of the mesh with order of interpolation (blue=Q_3, red=Q_5). The Cartesian domain has been mapped to longitude/latitude coordinates.
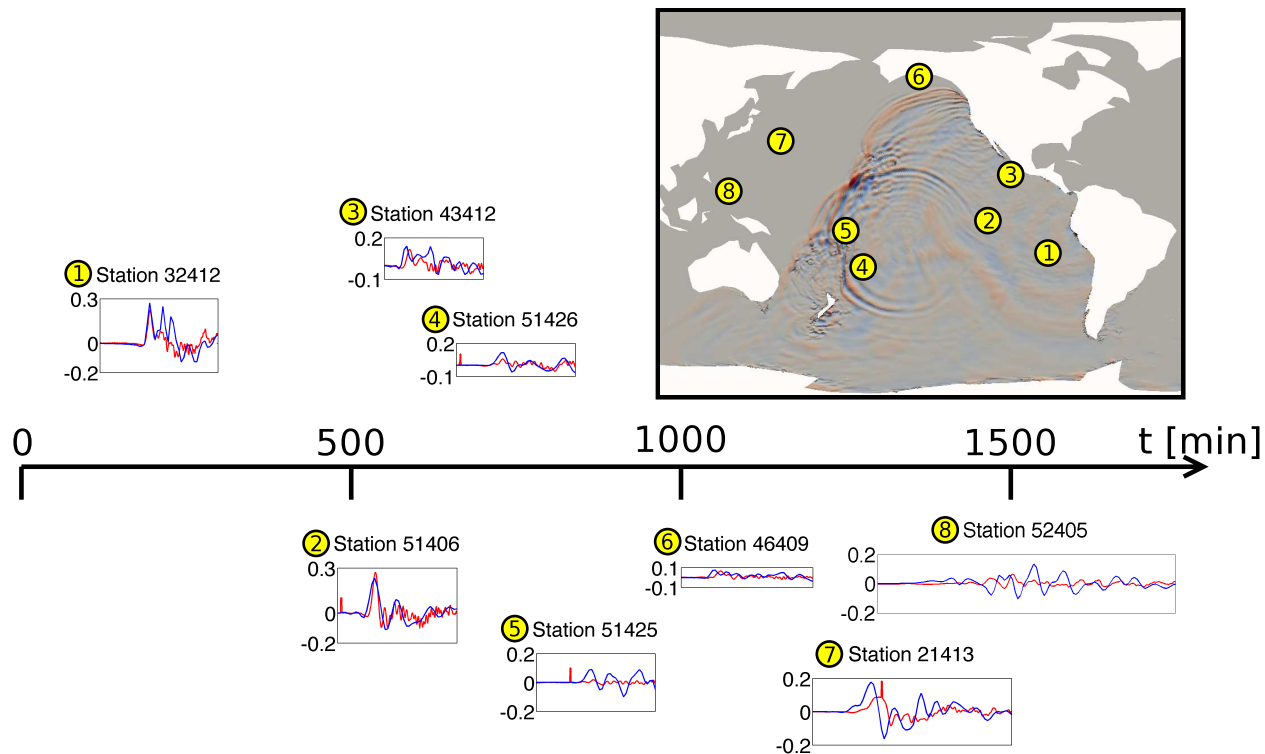
FIG. 10. Free-surface elevation at different DART stations: model data (blue) and DART data (red). The different plot boxes are aligned with the timeline to indicate the time at which the tsunami reaches the different stations (t=0 at the moment of the initial earthquake).
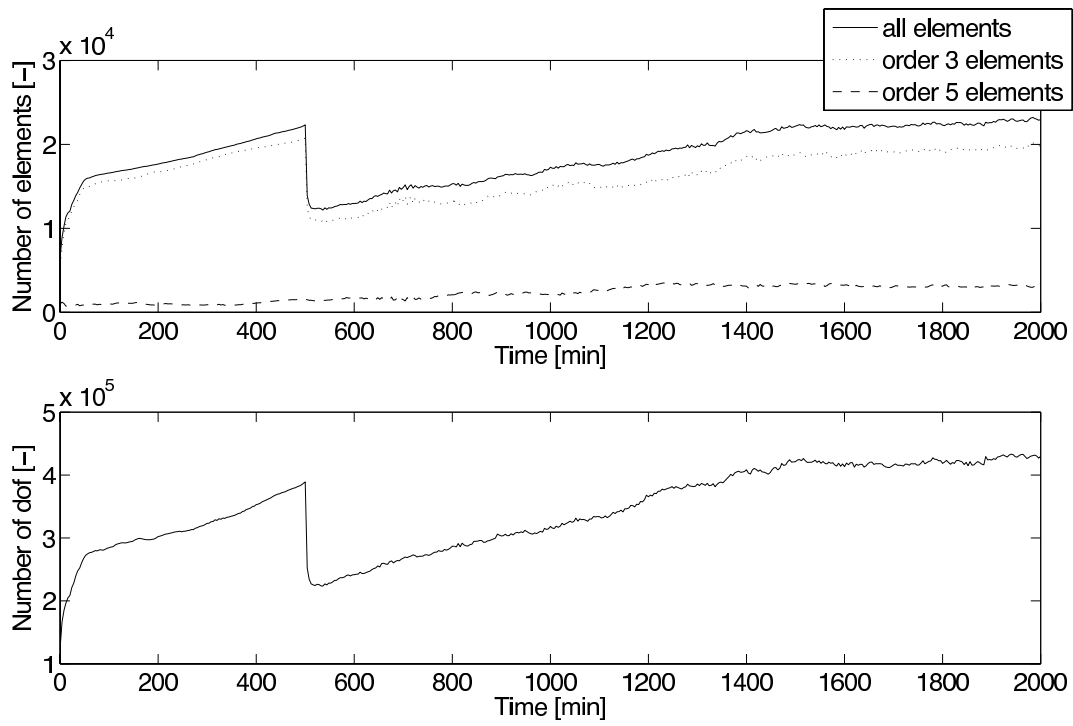
FIG. 11. UP: Evolution of the total number of elements during the adaptive tsunami simulation, with a separation for each order of interpolation. DOWN: Evolution of the total number of degrees of freedom (dof).