Geoscientific
Model Development

Direct numerical simulations of particle-laden density currents with adaptive, discontinuous finite elements

S. D. Parkinson[1]**, J. Hill**[1]**, M. D. Piggott**[1,2]**, and P. A. Allison**[1]

[1]Applied Modelling and Computation Group, Department of Earth Science and Engineering, South Kensington Campus, Imperial College London, London SW7 2AZ, UK
[2]Grantham Institute for Climate Change, South Kensington Campus, Imperial College London, London SW7 2AZ, UK

Correspondence to: S. D. Parkinson (s.parkinson11@imperial.ac.uk)

Abstract. High-resolution direct numerical simulations (DNSs) are an important tool for the detailed analysis of turbidity current dynamics. Models that resolve the vertical structure and turbulence of the flow are typically based upon the Navier–Stokes equations. Two-dimensional simulations are known to produce unrealistic cohesive vortices that are not representative of the real three-dimensional physics. The effect of this phenomena is particularly apparent in the later stages of flow propagation. The ideal solution to this problem is to run the simulation in three dimensions but this is computationally expensive.

This paper presents a novel finite-element (FE) DNS turbidity current model that has been built within Fluidity, an open source, general purpose, computational fluid dynamics code. The model is validated through re-creation of a lock release density current at a Grashof number of 5×10^6 in two and three dimensions. Validation of the model considers the flow energy budget, sedimentation rate, head speed, wall normal velocity profiles and the final deposit. Conservation of energy in particular is found to be a good metric for measuring model performance in capturing the range of dynamics on a range of meshes. FE models scale well over many thousands of processors and do not impose restrictions on domain shape, but they are computationally expensive. The use of adaptive mesh optimisation is shown to reduce the required element count by approximately two orders of magnitude in comparison with fixed, uniform mesh simulations. This leads to a substantial reduction in computational cost. The computational savings and flexibility afforded by adaptivity along with the flexibility of FE methods make this model well suited to simulating turbidity currents in complex domains.

1 Introduction

Density currents, also known as gravity or buoyancy currents, occur when two fluids with different densities meet. The density difference creates a pressure gradient that causes the more dense fluid to intrude beneath the less dense fluid. They occur in both natural and man-made situations, within a wide range of environments, and over a vast range of temporal and spatial scales. When a fluid contains particles in suspension the bulk density of that fluid changes. Density currents that are at least partly driven by a density variation due to suspended particles are termed particle-laden density currents. Examples include pyroclastic flows, dust storms, avalanches and turbidity currents.

A single submarine particle-laden density current can involve $100 \, \mathrm{km}^3$ of sediment (Talling et al., 2007). That is approximately 10 times the annual sediment flux into the ocean from all of the Earth's rivers combined (Talling et al., 2007). They can travel for hundreds of kilometres over the sea bed at speeds of tens of metres a second (Heezen and Ewing, 1952). Turbidity currents, a dilute sub-class of submarine particle-laden density currents where particle–fluid interactions dominate dynamics, can deliver enough destructive power to break telecommunications cables which can have huge financial implications (Heezen and Ewing, 1952). Turbidity currents can have an impact on pollutant dispersal from river outflow (Huang et al., 2005; Bombardelli et al., 2004), industrial discharge (Hallworth and Huppert, 1998; Bonnecaze et al., 1993), oil drilling (Curran et al., 2002) and water-injection dredging (Harris et al., 2002). Turbidity currents are also a key process for the movement of sediment around our planet (Talling et al., 2012). They form a significant

component of the stratigraphic record, and their deposits can form hydrocarbon reservoirs (Kneller and Buckee, 2000; Sequeiros et al., 2009). Having a good understanding of turbidity current behaviour can also allow us to predict and improve water quality in reservoirs by enhancing our understanding of pollutant concentrations (Alavian et al., 1992; Huang et al., 2005) and oxygenation (Best et al., 2005).

Studying turbidity currents is not easy. They occur infrequently and unpredictably at any particular location, and tend to destroy any measurement devices that are positioned to measure their effects. The dynamics of these currents are highly complex, with strong feedback between turbulence and sediment suspension making measurement of the dynamics difficult (Kneller and Buckee, 2000; Serchi et al., 2012). Small-scale laboratory experiments can provide useful insight into the dynamics of these currents, but are limited by scaling issues and the available measurement techniques (Kneller and Buckee, 2000).

High-resolution numerical models have become an important tool for the detailed analysis of particle-laden density current dynamics. Models that resolve the vertical structure of the flow are typically based upon the Navier–Stokes equations. Direct numerical simulations (DNSs) of particle-laden density currents should be carried out in three dimensions as unrealistic cohesive vortices form in two dimensions that have a significant impact on virtually all of the important outputs from these simulations (Cantero et al., 2007; Espath et al., 2014). The scale of particle-laden density currents is often described using the Grashof number. The Grashof number approximates the ratio of buoyant to viscous forces. This is equivalent to the square of the Reynolds number of a flow where the buoyancy velocity is used as the characteristic velocity. DNS modelling of particle-laden density currents has been achieved in three dimensions at moderate Grashof numbers of $\mathcal{O}\left(10^6\right)$ by Necker et al. (2002) and Nasr-Azadani and Meiburg (2011). Espath et al. (2014) have simulated a particle-laden density current at a Grashof number of $\mathcal{O}\left(10^8\right)$. Computational power has limited modelling of higher Grashof number flows. A Grashof number of $\mathcal{O}\left(10^8\right)$ translates into modelling of low volumetric particle concentration 10^{-4}–10^{-3} % flows in water at laboratory scales. Even at these moderate Grashof numbers a fully turbulent flow is obtained (Necker et al., 2002; Espath et al., 2014), and very useful insights have been obtained from simulations of particle-laden currents within this range.

There has been some success in modelling these flows using DNS in two dimensions which makes the problem more computationally tractable (Blanchette et al., 2005; Ooi et al., 2007). However, Espath et al. (2014) showed that the only important diagnostic that can be accurately predicted using a two-dimensional DNS model is the sedimentation rate. Another alternative to three-dimensional DNS is to use turbulence models to handle the small-scale turbulence and only resolve the large-scale motions. Turbulence modelling

is undoubtedly important for extending models that resolve the vertical structure of the flow to more realistic Grashof numbers. Whether it is appropriate to use a turbulence model depends upon the diagnostics that are important in the study. DNS simulations are necessary to perform detailed analysis of turbulent structures, and also for the validation of turbulence models for this application.

DNS models of turbidity currents in three dimensions have generally been formulated using spectral element techniques (Necker et al., 2002) and finite differences (Espath et al., 2014; Nasr-Azadani and Meiburg, 2011). These models are designed to be highly efficient, having structured meshes and, in most cases, high-order methods such that these computationally challenging problems become tractable. However, these computationally optimised methods make simulations in irregularly shaped domains very difficult (Mohd-Yusof, 1998). Natural turbidity currents propagate over complex bathymetries. The interaction of turbidity currents over complex geometries is therefore of obvious interest. The model of Nasr-Azadani and Meiburg (2011) can model turbidity currents in geometries with some complex features using the immersed boundary method, but this method has limitations and is not suitable for all use cases (Mohd-Yusof, 1998).

The finite-element method (FEM) benefits from the ability to easily accommodate complex geometries via the use of flexible unstructured meshes (Donea and Huerta, 2005). Hence, FEM provides an alternative approach to modelling interactions of turbidity currents in complex geometries. However, FEM is significantly more expensive than spectral element techniques (Mohd-Yusof, 1998). This paper presents a novel particle-laden density current model that has been built within Fluidity, an open source, general purpose, multiphase computational fluid dynamics FEM code (Imperial College London AMCG, 2014). This paper simulates a lock release density current at a Grashof number of 5×10^6 in two and three dimensions with a configuration similar to that of Necker et al. (2002). The governing equations are well established and have been validated extensively against experimental data across a range of simulation configurations (Sequeiros et al., 2009; Necker et al., 2002; Espath et al., 2014; Huang et al., 2007; Georgoulas et al., 2010). This paper validates the use of novel computational methods, including unstructured mesh adaptivity and discontinuous finite elements, through convergence analyses and by direct comparison with the results from the previous models of Necker et al. (2002) and Espath et al. (2014), providing a framework for future modelling efforts of this type. It is shown that adaptivity reduces the required element count by approximately two orders of magnitude for this application in comparison with fixed, uniform mesh simulations. The computational savings and flexibility afforded by adaptivity along with the flexibility of FEM make this model well suited to simulating turbidity currents in complex domains.

2 Mathematical model

2.1 Governing equations

A general discussion of particle motion in a non-uniform flow by Maxey and Riley (1983) stated that pressure gradient, added mass, viscous Stokes and augmented drag, and buoyancy forces need to be considered. This work considers flows where the particles are very small in relation to the length scales of motion. It also considers situations where the particle density is significantly larger than that of the surrounding fluid. For example, silica is ≈ 2.6 times the density of water in turbidity currents, and ≈ 2600 times the density of air in dust storms. Based upon these restrictions Maxey and Riley (1983) showed that the dominant forces in the equations of motion for a single particle relative to the fluid in which it is suspended are the viscous Stokes drag and buoyancy terms.

Particle-laden density currents consist of a collection of particles and hence particle collisions must also be considered. The effect of these collisions can be safely ignored by limiting the model to applications where the sediment volume fraction is less than 1 % (Necker et al., 2002). Flocculation of sediment particles can be ignored if the model is further restricted to sediment sizes of sand and larger ($> 64\,\mu\text{m}$) (Maerz et al., 2011).

Cantero et al. (2008) showed that the effects of inertia on particle motion in density currents are insignificant for particles smaller than $250\,\mu\text{m}$ in diameter and hence the model is restricted to sediment diameters below $250\,\mu\text{m}$.

Taking all of the above into account, the acceleration of particles will be the same as that of the containing fluid. Particles will move with a velocity equal to the sum of the fluid velocity, $\tilde{\boldsymbol{u}}$, and a settling velocity, $\tilde{u}_\text{s}\boldsymbol{k}$, where $\sim$ denotes a dimensional value. The settling speed, $\tilde{u}_\text{s}$, is obtained by balancing the Stokes drag and buoyancy forces, and $\boldsymbol{k} = (0, 0, 1)^T$. This means that the evolution of the particle volume fraction, $\tilde{c}$, can be described with a transport equation of the form

$$\frac{\partial \tilde{c}}{\partial \tilde{t}} + (\tilde{\boldsymbol{u}}_i - \tilde{u}_\text{s}\boldsymbol{k}_i)\,\frac{\partial \tilde{c}}{\partial \tilde{\boldsymbol{x}}_i} = \tilde{\kappa}\,\frac{\partial^2 \tilde{c}}{\partial \tilde{\boldsymbol{x}}_i\,\partial \tilde{\boldsymbol{x}}_i}\ , \tag{1}$$

where $\tilde{\kappa}$ is the particle concentration diffusivity. Particle concentrations will never be completely uniform. Individual particles will have a range of masses and settling velocities which lead to slight variations in speed and trajectory. As particles move past one another they interact and cause a self-induced hydrodynamic diffusion (Davis and Hassen, 1988). The magnitude of this diffusivity is generally chosen to be 0.7–1.0 times the ambient fluid viscosity (Necker et al., 2002; Cantero et al., 2008). Higher values help convergence and stability of the solution. It has been shown that this value has little effect on the relevant flow quantities so long as it does not significantly exceed the ambient fluid viscosity (Hartel et al., 2000).

Continuity and momentum balance in this model is governed by the Navier–Stokes equations. The fluid is assumed to be incompressible. Due to the very low volumetric concentrations, the displacement of fluid by the suspended particles can be ignored (Necker et al., 2002). Therefore the velocity field is considered to be divergence free. The Boussinesq approximation is adopted, in which density is considered constant except in the buoyancy term. The isotropic kinematic viscosity, $\tilde{\nu}$, is assumed constant throughout the model. This leads to the following form for the Navier–Stokes equations:

$$\frac{\partial \tilde{\boldsymbol{u}}_i}{\partial \tilde{t}} + \tilde{\boldsymbol{u}}_j\frac{\partial \tilde{\boldsymbol{u}}_i}{\partial \tilde{\boldsymbol{x}}_j} = -\frac{\partial \tilde{p}}{\partial \tilde{\boldsymbol{x}}_i} - \tilde{g}'\boldsymbol{k}_i + \frac{\partial}{\partial \tilde{\boldsymbol{x}}_j}\tilde{\mathbf{S}}_{ij}, \tag{2a}$$

$$\frac{\partial \tilde{\boldsymbol{u}}_i}{\partial \tilde{\boldsymbol{x}}_i} = 0, \tag{2b}$$

where the stress tensor, $\tilde{\mathbf{S}}$, for an incompressible flow is defined as

$$\tilde{\mathbf{S}}_{ij} = \tilde{\nu}\left(\frac{\partial \tilde{\boldsymbol{u}}_i}{\partial \tilde{\boldsymbol{x}}_j} + \frac{\partial \tilde{\boldsymbol{u}}_j}{\partial \tilde{\boldsymbol{x}}_i}\right)\ , \tag{3}$$

and the buoyancy density, $\tilde{g}'$, is a function of the sediment density, ρ_c, the ambient fluid density, ρ_a, the magnitude of the acceleration due to gravity, $\tilde{g}$, and the volumetric concentration of sediment, $\tilde{c}$, with the form

$$\tilde{g}' = \frac{\tilde{\rho}_\text{c} - \tilde{\rho}_\text{a}}{\tilde{\rho}_\text{a}}\,\tilde{g}\tilde{c}. \tag{4}$$

The assumption $\tilde{c} \ll 1$ and $(\tilde{\rho}_\text{c} - \tilde{\rho}_\text{a})/\tilde{\rho}_\text{a} = \mathcal{O}(1)$ justify the use of the Boussinesq approximation.

Particle-laden density currents deposit and/or erode the surface over which they travel. This means that the suspended mass changes with time, i.e. they are non-conservative. They have the potential to accelerate if their mass increases, or decelerate more rapidly due to settling out of the suspended sediment. The dynamics of sediment erosion are complex. Sediment on the bed will affect turbulence in the boundary layer. Larger sediment grains will shield smaller ones and grains may adhere to each other in the bed. The bed shape will also change as sediment is eroded, generating complex topography that may promote or inhibit further sediment erosion. Empirical algorithms have been developed for predicting erosion rates based upon particle parameters and the bed shear stress. Erodible boundaries, Γ^d, are modelled here as a flux boundary condition of the form

$$\boldsymbol{n}_i\tilde{\kappa}\,\frac{\partial \tilde{c}}{\partial \tilde{\boldsymbol{x}}_i} = \tilde{u}_\text{s}\tilde{E} \text{ on } \Gamma^d\ , \tag{5}$$

where $\tilde{E}$ is the non-dimensional erosion rate of sediment into suspension, $\boldsymbol{n}$ is the boundary unit normal vector.

Garcia and Parker (1991) reviewed empirical formulae by Fukushima et al. (1985), Akiyama and Stefan (1985), and Parker et al. (1986) for predicting the erosion of non-cohesive

sediment in finite size ranges. They validated these against laboratory results, and produced an improved formula for sediment erosion. This has been successfully used in work by Huang et al. (2007) and Sequeiros et al. (2009) when modelling sediment-laden density currents and is described as

$$\tilde{E} = \frac{A\tilde{Z}^5}{1 + A\tilde{Z}^5/0.3} \ , \tag{6a}$$

$$\tilde{Z} = \frac{\sqrt{\tilde{\tau}_b}}{\tilde{u}_s} \left(\sqrt{\left(\frac{\tilde{\rho}_c - \tilde{\rho}_a}{\tilde{\rho}_a} \tilde{g}\tilde{d} \right)} \frac{\tilde{d}}{\tilde{\nu}} \right)^{0.6} \ , \tag{6b}$$

where $A = 1.3 \times 10^{-7}$ is a non-dimensional constant and $\tilde{d}$ is the diameter of the suspended sediment. The bed shear stress, $\tilde{\tau}_b$, is defined as

$$\tilde{\tau}_b = \left| \boldsymbol{n}_i \tilde{\mathbf{S}}_{ij} \right| \ . \tag{7}$$

The total flux through an erodible boundary is calculated as

$$\frac{\partial \tilde{\eta}}{\partial \tilde{t}} = \boldsymbol{n}_i \tilde{u}_s \boldsymbol{k}_i \tilde{c}_b - \tilde{u}_s \tilde{E}, \tag{8}$$

where $\tilde{\eta}$ is the depth of the deposited sediment in the bed and $\tilde{c}_b$ is the volumetric concentration of sediment at the sediment bed boundary. $\tilde{E}$ is limited such that it never exceeds $\tilde{\eta}/\Delta t$, where Δt is the period of a time step. As in the work by Necker et al. (2002) and Espath et al. (2014), no adjustment for porosity of the deposit is included.

It is also possible for sediment to be moved along the bed without being entrained into the flow. This process is known as bedload transport. This has not been included in the current work. Sequeiros et al. (2009) stated that suspended sediment is the key factor in the movement of sediment in turbidity currents and that bedload transport can be neglected for currents that do not have a significant fraction of particles larger than 100 μm.

Four parameters are used to non-dimensionalise the equations outlined above:

$$Gr = \left(\frac{\tilde{u}_b \tilde{h}_0}{\tilde{\nu}} \right)^2 \ , \tag{9}$$

$$Sc = \frac{\tilde{\nu}}{\tilde{\kappa}} \ , \tag{10}$$

$$u_s = \frac{\tilde{u}_s}{\tilde{u}_b} \ , \tag{11}$$

$$R_p = \sqrt{\left(\frac{\tilde{\rho}_c - \tilde{\rho}_a}{\tilde{\rho}_a} \tilde{g}\tilde{d} \right)} \frac{\tilde{d}}{\tilde{\nu}} \ , \tag{12}$$

where Gr is the Grashof number, Sc is the Schmidt number, R_p is the particle Reynolds number, and h_0 is a characteristic length scale (half the lock-release depth in this work, as defined in Sect. 3). The buoyancy velocity, $\tilde{u}_b$, is used as the characteristic velocity scale and is defined in terms of the initial buoyancy density, $\tilde{g}_0'$, and the characteristic length h_0, as

$$\tilde{u}_b = \sqrt{\tilde{g}_0' \tilde{h}_0} \ . \tag{13}$$

Equations (1)–(3) and (5)–(8) can now be redefined in non-dimensional form as

$$\frac{\partial c}{\partial t} + (\boldsymbol{u}_i - u_s \boldsymbol{k}_i) \frac{\partial c}{\partial \boldsymbol{x}_i} = \frac{1}{\sqrt{Sc^2 Gr}} \frac{\partial^2 c}{\partial \boldsymbol{x}_i^2} \ , \tag{14}$$

$$\frac{\partial \boldsymbol{u}_i}{\partial t} + \boldsymbol{u}_j \frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{x}_j} = -\frac{\partial p}{\partial \boldsymbol{x}_i} - \boldsymbol{k}_i c + \frac{\partial}{\partial \boldsymbol{x}_j} \mathbf{S}_{ij} \ , \tag{15}$$

$$\frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{x}_i} = 0 \ , \tag{16}$$

$$\boldsymbol{n}_i \frac{1}{\sqrt{Sc^2 Gr}} \frac{\partial c}{\partial \boldsymbol{x}_i} = u_s E \text{ on } \Gamma^d, \tag{17}$$

$$E = \frac{A\,Z^5}{1 + AZ^5/0.3} \ , \tag{18}$$

$$Z = \frac{\sqrt{\tau_b}}{u_s} R_p^{0.6} \ , \tag{19}$$

$$\tau_b = \left| \boldsymbol{n}_i \mathbf{S}_{ij} \right| \ , \tag{20}$$

$$\frac{\partial \eta}{\partial t} = \boldsymbol{n}_i u_s \boldsymbol{k}_i c_b - u_s E, \tag{21}$$

$$\mathbf{S}_{ij} = \frac{1}{\sqrt{Gr}} \left(\frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{x}_j} + \frac{\partial \boldsymbol{u}_j}{\partial \boldsymbol{x}_i} \right) \ . \tag{22}$$

2.2 Discretisation

Fluidity uses the FEM to solve the Navier–Stokes equations. Using finite elements gives great flexibility in element choices and many are available when using Fluidity. A linear discontinuous Galerkin (DG) scheme is used for the discretisation of both the velocity and sediment concentration fields.

A DG discretisation does not enforce continuity across element boundaries. A field that is discretised on a DG function space may therefore have multiple values at element boundaries. It will also have more degrees of freedom than a continuous function of the same order as elements do not share nodes. DG methods are a good choice of discretisation for advection-dominated problems as they produce stable discretisations without the need for stabilisation strategies such

as streamline-upwinding (Peraire and Persson, 2008). DG methods also work well on arbitrary meshes and have the desirable properties of having a block-diagonal mass matrix that can be trivially inverted locally for each element. This allows for certain equations to be solved very efficiently (Bassi and Rebay, 1997).

Use of the DG method requires a choice of flux schemes for both the advective and diffusive terms. A simple upwind flux is chosen for advective terms. A centred flux is used for the diffusive terms. The compact discontinuous Galerkin method (CDG) is used to implement the diffusive terms (Peraire and Persson, 2008). For a detailed discussion of DG discretisations and flux terms see the review paper by Cockburn and Shu (2001).

Advection when using a DG discretisation is not bounded. Undershoots and overshoots can occur that affect the dynamics of a gravity current. Slope limiters are employed to bound the solution although these can be dissipative. Vertex-based slope limiting, as suggested by Kuzmin (2010), is used herein.

For the pressure field a quadratic continuous Galerkin scheme is used. We therefore have a mixed finite element pairing for solving the incompressible Navier–Stokes equations. This element pairing, described by Cotter et al. (2009), has the benefit of satisfying the Ladyženskaja–Babuška–Brezzi (LBB) stability condition and hence needs no stabilisation of the pressure field. Additionally, a higher-order-accurate pressure field means that the pressure gradient term in the momentum equation has the same order of accuracy as the buoyancy forcing term. These two terms dominate in early stages of propagation and hence the ability for these terms to balance is important in determining how the flow evolves.

A Crank–Nicolson time discretisation is used throughout the model which is second-order accurate in time. The coupled system of non-linear equations are solved using two non-linear iterations known as Picard iterations. Within each non-linear iteration the equations are linearised using the best available solution for each variable that is not being solved for. The momentum and conservation equations are solved using a pressure correction scheme.

An adaptive time step is used. This makes the simulation more robust when using a changing multi-scale mesh and also takes advantage of the reducing current velocity over time which allows for much larger time steps towards the end of the simulation. The time step length is based upon obtaining a target Courant number of 2. This is a relatively conservative requirement for the implicit time discretisation used.

There are two exceptions to the use of Crank–Nicolson time discretisation. Slope limiters used with DG discretisations only guarantee a bounded solution in conjunction with an explicit advection scheme. Therefore the sediment transport and momentum equations are solved in two stages. Advection is calculated using explicit subcycles with adaptive time step lengths based upon achieving a Courant number of 0.2 (an order of magnitude smaller than the rest of the model). Diffusion/viscous dissipation is solved using the simulation time step, Δt, and a Crank–Nicolson discretisation. Additionally, the bed shear stress is calculated at the start of each time step and hence the erosion rate is totally explicit. A thorough description of the time discretisation outlined above is available in the Fluidity manual (Imperial College London AMCG, 2014).

2.3 Anisotropic mesh adaptivity

The motivation behind using mesh adaptivity is to optimise the spatial resolution with time such that both the discretisation error and computational cost of a simulation are minimised (Piggott et al., 2005). Adapting the mesh is split into three tasks. The first step is to determine the desired edge lengths, or to form a metric against which element edge lengths can be defined, the second part involves generating a new mesh that better fits these requirements and distributing this mesh amongst the active processors, and the third involves transferring data from the old mesh to the new mesh (see Fig. 1). A brief description of each phase of the process is included below. The reader is referred to the work of Piggott et al. (2008) for more details.

2.3.1 Metric formation

Determining the desired edge lengths for a mesh requires some quantification of the error in the solution due to spatial discretisation. This is difficult to do as there is usually no better estimate of the exact solution than the estimate from the current discretisation. An indirect method of measuring the error is required. Ciarlet (1991) showed that the finite element error can be bounded by the interpolation error for elliptic problems. It is assumed that this also holds for other partial-differential equations. This is deemed a reasonable way of defining an error indicator (Fortin, 2000).

The aim is to minimise the error in fields that are discretised using first-order discretisations. For first-order elements, the interpolation error depends upon the Hessian, $\mathbf{H}$ (the matrix of second-order partial derivatives) (Frey and Alauzet, 2005). The second derivative of such a discretisation is formally zero and hence some method of recovering the Hessian for these fields is needed. In Fluidity a double-lumped Galerkin projection is used to compute the Hessian as described by Pain et al. (2001). This Hessian will contain information about both the magnitude and direction of the curvature of a field and hence can be used to guide generation of anisotropic elements. This is very useful in regions, such as boundary layers, where the solutions vary significantly more rapidly in one direction than in others.

The Hessian is used to form a metric tensor, $\mathbf{M}$, that will guide the mesh optimisation. $\mathbf{M}$ is defined such that the optimal mesh, $\mathcal{M}$, would have edges, v, with unit length when
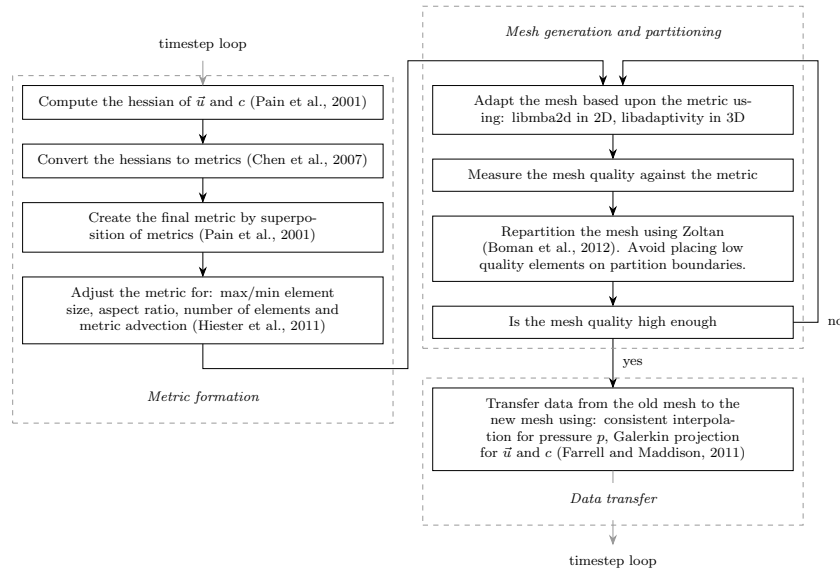
Figure 1. A description of the high-level algorithm involved in adapting the mesh. This algorithm is invoked repeatedly throughout the simulation at a fixed interval specified as a number of time steps.

measured against it:

$$\sqrt{v_i \, \mathbf{M}_{ij} \, v_j} = 1 \quad \forall v \in \mathcal{M} \, . \tag{23}$$

The choice of formulation for $\mathbf{M}$ is therefore fundamental to the way in which the mesh adapts. A formulation suggested by Chen et al. (2007) which controls the L^2 norm of the interpolation error is used:

$$\mathbf{M} = \frac{1}{\varepsilon} \det |\mathbf{H}|^{-\frac{1}{4+n}} \, |\mathbf{H}| \, , \tag{24}$$

where n is the number of dimensions, and ε is the interpolation error bound, a value which is defined by the user. For L^p methods, Loseille and Alauzet (2011) found $p = 2$ to be the optimal value and incorporate more influence from dynamics of smaller magnitudes. Experience has shown this metric formation to be very effective for gravity current simulations (Hiester et al., 2011, 2014).

It is often important to adapt to more than one solution field. When this is the case the final metric is a superposition of the metrics calculated for each individual field (Pain et al., 2001). At this point the metric is also modified to take into account bounds upon the maximum and minimum element size, maximum allowable aspect ratio, edge length gradation, and the number of elements. Additionally, this metric can be advected forward in time providing an estimate of future requirements for the mesh resolution and allowing for more time between adapt operations (Hiester et al., 2011).

2.3.2 Mesh generation and partitioning

The second stage of creating the new mesh is handled by the open-source mesh optimisation library libmba2d in two dimensions, or libadaptivity, another open source library developed alongside Fluidity, in three dimensions. This involves a series of topological and geometrical operations, with the aim of obtaining a mesh with unit edge lengths with respect to the determined metric, see Eq. (23). These operations include node insertion or deletion, edge/face swapping, which preserves the node locations but manipulates edge lengths by changing the configuration of a edge/face between elements, and node movement (Piggott et al., 2009).

The Zoltan library (Boman et al., 2012) is used to partition the mesh in parallel after each adapt iteration. Nodes cannot be adapted at the edge of partitions. After each adapt iteration parameters are passed to the Zoltan library which discourage it from generating partitions through elements that have not been able to adapt. For three-dimensional simulations, a minimum of three adapt iterations are required to allow all elements to adapt and create a mesh that satisfies the metric constraint everywhere. Zoltan's graph re-partitioning algorithm is used to partition the mesh efficiently between adapt iterations. Once a good quality mesh has been obtained the hypergraph partitioning method is used to redistribute the elements amongst the processes.

2.3.3 Data transfer

After creating the new mesh the data are transferred on to it from the previous mesh. For the purposes of describing this step these meshes will be referred to as the target and donor meshes respectively.

Consistent interpolation is used for the transfer of the pressure field from the donor to the target mesh. This field is continuous, and conservation of pressure is not essential.

Consistent interpolation is very cheap and hence is a good choice for data transfer of this field.

All other prognostic fields are discontinuous. Consistent interpolation cannot be used for discontinuous fields as test and trial functions are not continuous across element boundaries. Additionally, consistent interpolation is not conservative and is dissipative. It is important to conserve sediment mass during data transfer. It is also important that dissipation of both velocity and sediment is kept to a minimum. Galerkin projection is used for data transfer of these fields – this is both conservative and non-dissipative (Farrell et al., 2009). This requires the generation of a supermesh, which is the union of both the donor and target mesh. Within each element of the supermesh the test and trial functions for both discretisations are consistent and thus this method is valid for DG discretisations. The construction of a supermesh can be a very expensive operation. Fluidity uses an algorithm developed by Farrell and Maddison (2011) where the supermesh is created locally for each target element. For DG discretisations the Galerkin projection can be carried out entirely locally due to the fact that the mass matrix is block diagonal. This combination greatly increases efficiency.

Where the projection occurs over a surface of the volume mesh (e.g. deposited sediment) projection is carried out in a $(n-1)$-dimensional space. For DG discretisations, all donor mesh surface elements that intersect a target mesh element must be in the same plane as the target mesh element. The Galerkin projection is carried out locally for each target element by rotating the coordinates of the target element and all intersecting donor elements into the $x–y$ plane.

3 Simulation configuration

The classic lock-release setup is used as a test case for the model. This is a well-researched configuration with a range of data against which to validate results (Gladstone et al., 1998; Necker et al., 2002; Cantero et al., 2006; Espath et al., 2014). As defined in Fig. 2, $L_2 = L_3 = 2.0$, which means that $h_0 = 1.0$, $L_{1R} = L_3/2.0$, and $L_1 = 19.0$ which is slightly longer than the final run-out length of the density current considered. The dimensionless parameters are set as

$$Gr = 5.0 \times 10^6, \quad Sc = 1.0,$$
$$u_s = 0.02, \quad R_p = 20.0. \tag{25}$$

As such, the experiment configuration is identical to that of Necker et al. (2002), with the exception of the addition of sediment erosion, and hence a requirement for the definition of R_p, which is defined to lie in the range of R_p for which the erosion algorithm outlined in Eq. (18) is valid, as detailed in Garcia and Parker (1991).

Boundary conditions for velocity are free-slip for all side walls, $u_1 = 0$ at $x = -1$ and $x = 18$, $u_2 = 0$ at $y = 0$, and $y = 2$, and no-slip at the top and bottom of the domain,
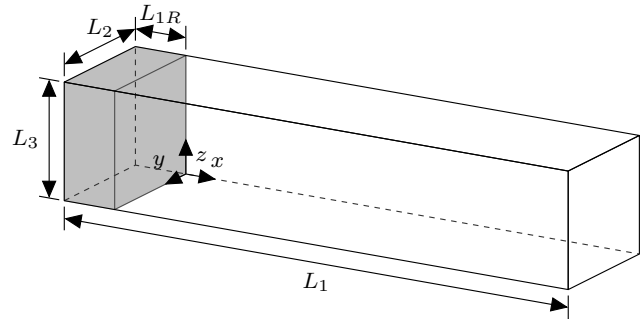

Figure 2. Lock-release simulation domain configuration. The grey region indicates the volume of non-zero sediment concentration at $t = 0$. The coordinate system defines the origin, x_0, y_0, z_0

$u = (0, 0, 0)^T$ at $z = 0$ and $z = 2$. All velocity boundary conditions are applied weakly. Where a velocity component is not set with a Dirichlet condition, a zero Neumann boundary condition is applied. Note that the side wall boundary conditions vary from those of Necker et al. (2002). A free-slip boundary condition should give comparable results to the periodic boundary conditions used in that work.

Boundary conditions for the sediment concentration field are as follows. The erosion boundary condition outlined in Eq. (17) is applied at the bottom surface, $z = 0$. A zero Dirichlet boundary condition is prescribed at $z = 2$ (the top surface of the tank). At all other boundaries $(u + ku_s) \cdot n$ equals zero, hence zero Neumann boundary conditions are applied which enforce zero flux of sediment across these surfaces.

The initial condition for the sediment concentration field in the three-dimensional simulation is similar to that suggested by Hartel et al. (1997) and Cantero et al. (2006). This initial condition is based upon the solution obtained from a purely diffusive problem. Hartel et al. (1997) argued that the problem will be dominated by diffusion for very early stages of the simulation and hence this initial condition is justified as being the condition of the flow a short time period after the initial release. This initial condition includes a perturbation, γ, in a similar way to the work by Cantero et al. (2006). A random perturbation of the initial condition is important to help promote the generation of three-dimensional structures in the flow. Necker et al. (2002) and Espath et al. (2014) use an alternative perturbation of the velocity field for the same purpose. The initial condition for the sediment concentration, including the perturbation, is as follows:

$$c = \frac{1}{2} - \frac{1}{2}\mathrm{erf}\left\{\sqrt[4]{GrSc^2}[x - \gamma]\right\}, \tag{26}$$

with

$$\gamma = \cos\left(\sum_i f(i, x, y, z)\right)\Delta_x, \tag{27}$$

where Δ_x is chosen to be 0.2 and γ is constructed of four sets of waves originating from the four corners of a plane aligned with the lock gate. Each set of waves contain 60 waves with random phases and random wavelengths ranging between 0.02 and ∞. Wave i has amplitude $f(i, x, y, z)$ at positions x, y, z. The perturbation chosen covers a wide range of frequencies so as to not preferentially generate particular wavelengths of oscillations. The mesh is adapted before the first time step to produce a good mesh for this initial condition.

To use adaptivity several controlling parameters need to be defined. The fundamental parameters which define the mesh resolution in the simulation are the interpolation error bounds. The next section describes how values for these parameters were chosen. The time between adapts also requires definitions, and it is necessary to ensure that there is adequate resolution in periods between adapts. Through experimentation it was found that an adapt every five time steps kept the simulation stable towards the beginning of the simulation. At later stages in the simulation an adapt was required every two time steps. The high frequency of mesh adapts was required to limit instabilities in the boundary layer which grow rapidly. Small instabilities that developed did not have any noticeable impact on the important outputs from the simulation. This is discussed further in the following sections. As mentioned in Sect. 2.3.1, metric advection is used to advect the metric, which defines the edge lengths required to meet the interpolation error bounds, forward in time. The metric is conservatively advected through five adapt intervals at each adapt.

4 Choosing appropriate interpolation error bounds

It is possible to define an interpolation error bound for any of the functions in the simulation. In this simulation sediment concentration, velocity and pressure are solved for. It is common practice to adapt to the velocity field for the purposes of resolving the velocity and pressure fields. Good resolution of the sediment concentration field is also required. Hence four interpolation error bounds require definition for the simulation – one for each velocity component, and another for the sediment concentration.

In order to select good values for these parameters a convergence analysis is required. Doing this with three-dimensional models would be prohibitively expensive and hence two-dimensional simulations are used to carry out this convergence analysis. The two-dimensional simulations are defined in the x–z plane. It has been well documented by Necker et al. (2002) and Cantero et al. (2007) that output from two-dimensional simulations of particle-laden density currents do not compare well with three-dimensional simulations. However, two-dimensional simulations are useful for the purpose of understanding the resolution requirements of simulations.

A measure of the quality of a mesh, in terms of the dynamics computed within it, is required. The simulation is of a turbulent flow. Head speed, deposit profile, quantity of suspended sediment and deposition rates are all important outputs from these simulations, but due to the turbulent nature of the flow, which is very sensitive to small changes in the mesh, it is very hard to show convergence of these quantities.

However, one important quantity does show convergence. This is the energy lost due to discretisation, and data transfer errors. DNS simulations resolve all length scales of motion. Convergence analysis will show that the discretisation errors are small enough that they have a negligible impact on the result and that the mesh resolution is fine enough to resolve all of the energy in the flow. The combination of upwind flux terms and slope limiting in the discretisation dissipates energy at scales that the mesh cannot resolve. Additionally, adapting the mesh requires a data transfer operation which will introduce some relatively small errors. By computing the energy budget in the simulation and how this varies over time a value for the energy lost due to discretisation, and data transfer errors, ϵ_d can be obtained. This quantity gives us some indication of how well the scales of motion in the flow are being resolved. Importantly, this value converges as the mesh resolution increases and so gives us a good method of comparing the quality of different mesh configurations. Following the method of Winters et al. (1995), Necker et al. (2002) and Espath et al. (2014), equations for the rates of change of potential energy, E_p, and kinetic energy, E_k, in the system can be derived as follows.

The kinetic energy in the system is

$$E_k = \frac{1}{2} \int_\Omega |\boldsymbol{u}|^2 \, d\Omega . \tag{28}$$

To obtain the time derivative for E_k compute the dot product of the momentum Eq. (15) with $\boldsymbol{u}$ and apply the chain rule to obtain

$$\frac{1}{2} \frac{\partial |\boldsymbol{u}|^2}{\partial t} + \frac{1}{2} \boldsymbol{u}_j \frac{\partial |\boldsymbol{u}|^2}{\partial \boldsymbol{x}_j} = -\boldsymbol{u}_i \frac{\partial p}{\partial \boldsymbol{x}_i} - \boldsymbol{u}_3 c + \boldsymbol{u}_i \frac{\partial \tau_{ij}}{\partial \boldsymbol{x}_j} . \tag{29}$$

Integrating over the domain and integrating by parts, using the continuity Eq. (16) and the knowledge that there are no normal flow boundary conditions on all boundaries, i.e. $\boldsymbol{u}_i \boldsymbol{n}_i = 0$, an equation for the rate of change of E_k is obtained:

$$\frac{\partial E_k}{\partial t} = -\int_\Omega c \boldsymbol{u}_3 \, d\Omega - \int_\Omega \tau_{ij} \frac{\partial \boldsymbol{u}_i}{\partial \boldsymbol{x}_j} \, d\Omega . \tag{30}$$

The potential energy in the system is

$$E_p = \int_\Omega c \boldsymbol{x}_3 \, d\Omega . \tag{31}$$

To obtain the time derivative for this term first multiply the equation for sediment concentration (Eq. 14) by x_3:

$$\frac{\partial c}{\partial t}x_3 + x_3\,(u_i - u_s k_i)\,\frac{\partial c}{\partial x_i} = x_3\,\frac{1}{\sqrt{Sc^2 Gr}}\,\frac{\partial^2 c}{\partial x_i \partial x_i}. \qquad (32)$$

Integrating over the domain, and by parts, using the chain rule and noting that all velocities normal to the wall are zero, an equation for the rate of change of E_p is obtained:

$$\frac{\partial E_p}{\partial t} = \int_\Omega c\,u_3 \mathrm{d}\Omega + u_s \int_\Omega x_3 \frac{\partial c}{\partial x_3}\mathrm{d}\Omega$$

$$+ \frac{1}{\sqrt{Sc^2 Gr}}\left[\int_\Gamma x_3 \frac{\partial c}{\partial x_i} n_i \mathrm{d}\sigma - \int_\Omega \frac{\partial c}{\partial x_3}\mathrm{d}\Omega\right]. \qquad (33)$$

An equation for the transfer of energy from E_k and E_p to and from internal energy and heat, and also lost due to the settling of particles can be obtained by combining Eqs. (30) and (33). This equation will not hold for an under-resolved mesh. Energy dissipation that occurs at scales below the grid resolution will be dissipated through application of slope limiting. An additional term, ϵ_d, is therefore included to balance the equation and represent the dissipation due to numerical errors which yields

$$\frac{\partial\left(E_p + E_k\right)}{\partial t} = -\epsilon - \epsilon_s - \epsilon_d, \qquad (34)$$

where

$$\epsilon = \int_\Omega \tau_{ij}\frac{\partial u_i}{\partial x_j}\mathrm{d}\Omega, \qquad (35)$$

and

$$\epsilon_s = \frac{1}{\sqrt{Sc^2 Gr}}\left[\int_\Omega \frac{\partial c}{\partial x_3}\mathrm{d}\Omega - \int_\Gamma x_3 \frac{\partial c}{\partial x_i}n_i\,\mathrm{d}\sigma\right]$$

$$- u_s \int_\Omega x_3 \frac{\partial c}{\partial x_3}\mathrm{d}\Omega. \qquad (36)$$

In order to compare overall mesh quality, ϵ_d is integrated over time to give the single quantity

$$E_D = \int_0^t |\epsilon_d(\tau)|\,\mathrm{d}\tau. \qquad (37)$$

E_D is computed for a set of two-dimensional simulations forming a parameter sweep of values for the interpolation error bounds for the two components of velocity and sediment concentration with values of 4×10^{-3}, $4\times10^{-2.5}$ and 4×10^{-2}. This leads to a total of 27 simulations. The range of values used in the parameter sweep were determined from
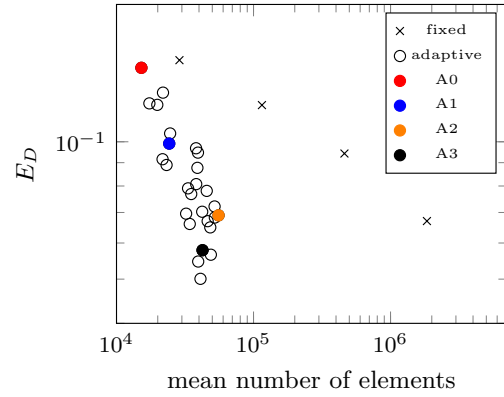

Figure 3. Time-integrated energy conservation error, E_D, against the mean number of elements for a range of two-dimensional simulations using fixed and adaptive meshes. Adaptive simulations represent a parameter sweep of interpolation error bounds with values 4×10^{-3}, $4\times10^{-2.5}$ and 4×10^{-2} for velocity and concentration. Fixed meshes are on uniform triangular grids with edge lengths 5×10^{-2}, 2.5×10^{-2}, 1.25×10^{-2} and 6.25×10^{-3}. Four adaptive simulations are highlighted. The settings for these simulations are detailed in Table 1.

sensitivity analyses performed prior to this. For the purposes of comparing results against fixed mesh simulations the above quantity was also computed for a range of regular, structured triangular mesh simulations with edge lengths 5×10^{-2}, 2.5×10^{-2}, 1.25×10^{-2} and 6.25×10^{-3}, resulting in 2.88×10^4, 1.15×10^5, 4.6×10^5 and 1.84×10^6 elements respectively. The adaptive mesh simulations converge at a higher order than the fixed mesh simulations in relation to the mean number of elements, $\overline{N}_e$, used in the simulation (Fig. 3). The number of elements in adaptive mesh simulations, N_e, varies with time (Table 1). The difference between the maximum and minimum number of elements increases superlinearly as the interpolation error bounds tighten. The number of elements in the domain is a function of the interpolation error bounds, the dynamics of the flow, which vary significantly with time, and also mesh resolution, creating a non-linearity in this relationship, and also the bound set for the maximum element size. The largest relative difference between the maximum and minimum number of elements occurs in adaptive simulation A2 where the maximum is $\approx 130\%$ of the mean, and the minimum is $\approx 40\%$ of the mean. The distribution of element counts throughout a simulation is skewed. An increase in the number of elements implies that element size has decreased. This in turn implies that the length of time steps has decreased, leading to more time steps being required at periods during which there are a large number of elements.

Importantly, Fig. 3 shows that, at the highest resolutions, a saving of more than one order of magnitude is obtained in the mean number of elements required in the simulation. In three dimensions the saving are likely to be even greater.

Table 1. Minimum, maximum and mean number of elements, $\overline{N}_e$, and interpolation error bounds for sediment concentration, ε_c, the x component of velocity, ε_{u_1}, and the y component of velocity, ε_{u_2}, for selected adaptive two-dimensional simulations from the interpolation error bound parameter sweep.

id	ε_c	ε_{u_1}	ε_{u_2}	$\min(N_e)$	$\max(N_e)$	$\overline{N}_e$
A0	4×10^{-2}	$4 \times 10^{-2.0}$	$4 \times 10^{-2.0}$	12 924	15 000	14 017
A1	$4 \times 10^{-2.5}$	$4 \times 10^{-2.5}$	$4 \times 10^{-2.5}$	16 025	31 083	24 900
A2	4×10^{-3}	$4 \times 10^{-3.0}$	$4 \times 10^{-3.0}$	20 515	71 055	55 031
A3	4×10^{-3}	$4 \times 10^{-2.5}$	$4 \times 10^{-2.5}$	19 094	63 767	41 827

Integrating Eq. (38) over time,

$$E_p + E_k = \int_0^t -\epsilon(\tau) - \epsilon_s(\tau) - \epsilon_d(\tau)\, d\tau$$

$$= -E - E_s - E_d. \tag{38}$$

Figure 4 shows how the above quantities vary over the period of the simulation for adaptivity options A3, as detailed in Table 1. Values are compared against two-dimensional results (Espath et al., 2014). There is very good agreement for E, but there is a notable variation between the values for E_s in this work and that of Espath et al. (2014). This is because of the zero Dirichlet boundary condition for sediment concentration at the top of the tank in this work whereas Espath et al. (2014) have a zero flux condition. At very early stages of the simulation the Dirichlet condition results in a flux of sediment through the top of the domain. The overall impact on the simulation is a loss of sediment mass of < 1 % and a total energy loss of $\approx$ 3 %. The zero flux condition is preferable but is not implemented in Fluidity for this discretisation. A future aim will be to implement this boundary condition. Generally there is good agreement for E_k and E_p. In two-dimensional simulations strong coherent vortices form that contain and transport large quantities of the suspended sediment. These vortices play an important role in the transfer of energy between E_k and E_p. Because of the chaotic nature of the creation and propagation of the vortices, there will always be variations in the values of E_k, E_p and to some extent E between simulations.

Another important aspect of the flow is the boundary layer at no-slip boundaries. This feature of the flow requires very small elements in the wall normal direction to resolve the boundary layer properly. Convergence on a solution is quickly obtained for the boundary layer using an adaptive mesh (Fig. 5). All but the most coarse adaptive simulation configurations have converged on to a solution. The fixed mesh configurations show a similar level of convergence for the two highest-resolution simulations. Anisotropic mesh adaptivity is particularly useful for resolving features such as boundary layers which require high resolution in one direction compared to others.

Using the evidence outlined above, the chosen interpolation error bounds were those of adaptive simulation A3. This
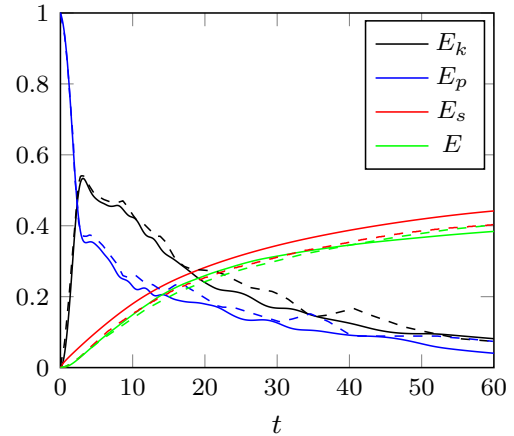

Figure 4. Energy budget evolution with time for simulation with adaptivity options A3 (solid lines —) compared against 2-D results from Espath et al. (2014) (dashed lines - - -). Values are normalised by the initial potential energy, E_T.

simulation had a well-resolved boundary layer and good conservation of energy. This simulation had interpolation error bounds of $4 \times 10^{-2.5}$ for both velocity components. An assumption is made that this error bound will also be suitable to use for the third velocity dimension in the three-dimensional simulation.

Figure 6 shows a qualitative comparison of results from high (and low) resolution fixed mesh simulations, and an adaptive mesh simulation, A3. The low resolution fixed mesh simulation has approximately the same number of elements as the adaptive simulation but is clearly not resolving the turbulent structures as well as the adaptive mesh simulation. Results from the high resolution fixed mesh and the adaptive mesh simulations are very similar. There are some variations in the vortices that are generated and these variations grow with time. Within the scope of this investigation no two simulations ever produced identical results. Very small variations in spatial discretisation lead to small variations at early stages in the simulation which propagate and lead on to larger variations downstream.
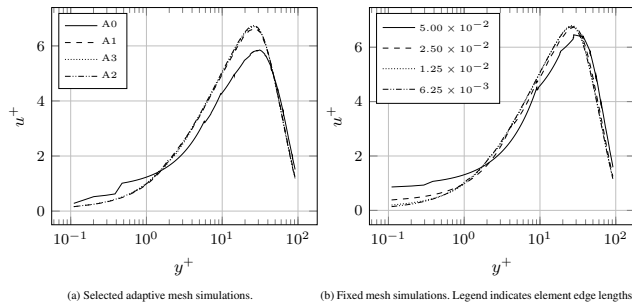
(a) Selected adaptive mesh simulations.　　(b) Fixed mesh simulations. Legend indicates element edge lengths.

Figure 5. Wall normal velocity profile at the location of the nose of the gravity current at $t = 7.5$ for the fixed mesh simulations and selected adaptive simulations. Note that A2 is a higher-resolution simulation than A3.

Figure 6. Heat map of sediment concentration at $t = 4$ and $t = 12$ for the highest-resolution fixed mesh simulation, the adaptive simulation with configuration A3, and a fixed mesh simulation with a similar number of elements to the adaptive simulation.

5 The benefits of using mesh adaptivity

In three dimensions, adaptivity is essential to compute this simulation using finite elements with Fluidity. A fixed, and regular tetrahedral grid would have required more than 1×10^9 elements which would have led to an unachievable run time and unmanageable post-processing and visualisation demands. By using adaptivity the number of required elements has been reduced to a maximum of approximately 1×10^7, at least a two order of magnitude reduction, making all aspects of the simulation manageable.

To resolve a comparable flow Espath et al. (2014) used $\approx 6 \times 10^7$ degrees of freedom. A Fluidity simulation with $\mathcal{O}(10^9)$ discontinuous elements has $\mathcal{O}(10^9)$ degrees of freedom. The finite-difference method used by Espath et al. (2014) uses fewer degrees of freedom as it employs a high-order finite-difference discretisation which increases the accuracy of the solution.

Figure 7 shows how the number of elements in the simulations varied over time. Throughout the simulation the number of processor cores that were used was varied between 36 and 512 to keep the number of elements per core in the region of 20 000. The initial drop in number of elements at the start of the simulation is due to the fact that conservative interpolation bounds were used to generate the mesh from the initial conditions. Following this initial drop, during early periods of the simulation the flow is transitioning from a laminar to a turbulent flow. Throughout this period $\boldsymbol{u}$ becomes more and more complex and hence the number of elements required to resolve the flow increase rapidly. The highest amount of elements required is at a $t \approx 10$ at which point the flow has developed into a fully three-dimensional, highly turbulent flow. Beyond this point the number of elements required for the simulation steadily decreases as energy is dissipated from the flow. Additionally, c gradually diffuses such that the curvature of the field decreases and fewer elements are required to resolve the field. Note the drop in elements at $t \approx 25$ of approximately 12.5 %. This coincides with the reduction in the number of time steps between adapts from
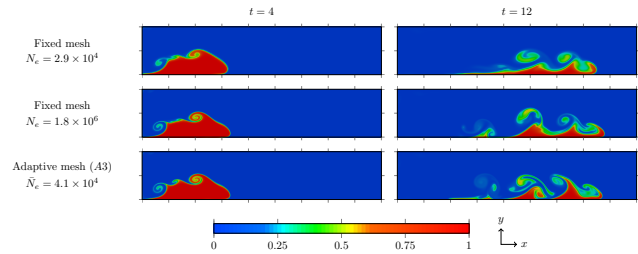
5 to 2 as mentioned in Sect. 3. A reduction in adapt interval means that the metric is not advected so far, and hence fewer elements are required. Also note the noise in the number of elements in the simulation. This is reduced by adapting more regularly and is due to the adapt routine responding to small instabilities in the boundary layer. Future work will look into removing these instabilities. This may require mesh adapts after every time step.

Figure 8 shows the adapted mesh over a subdomain in the region of the current head at two times, $t = 3.5$ and $t = 4$. The images are generated from the three-dimensional simulation, and are taken from a plane at $y = 0$. The cut plane is chosen to be at the edge of the domain as a good two-dimensional representation of the mesh can be obtained at boundaries where all element surfaces are parallel to one another. These images demonstrate how the mesh adapts to the concentration field c, and velocity field, $\boldsymbol{u}$. The images also show how the mesh changes over a short period of the simulation. The change between $t = 3.5$ and $t = 4$ is dramatic. Very few, if any elements, within this view are consistent. The images clearly display how anisotropic elements are generated along the density interface and the boundary layer where the curvature of the solution is highly anisotropic.

Figure 9 shows the distribution of element sizes across a plane of the domain at $y = 1$ at times $t = 8$ and $t = 20$. Three images for each time show the dimensions of the elements in the x, y and z directions. High resolution in the z, or wall normal direction, can be seen on no-slip boundaries in proximity to the wall. Much larger relative resolutions are found in the same regions for the x and y directions. This is enabled by the use of anisotropic mesh adaptivity which keeps the number of elements to a minimum, whilst still resolving this important feature of the flow. These images also highlight the cost of resolving the wall boundary layer with a high density of small elements, and hence a large proportion of the elements found in the near-wall region. The high wall resolution in the z direction is present on all no-slip surfaces and extends beyond the front of the gravity current head. In the x direction the highest resolution is generally found at the front of the gravity current head. This is caused
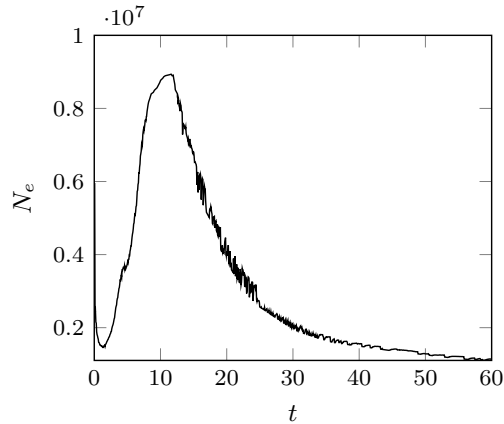
Figure 7. Number of elements in the three-dimensional simulation as a function of time.

Figure 8. Images showing concentration c, velocity magnitude $|\boldsymbol{u}|$, and the adapted mesh at $t = 3.5$ **(a)** and $t = 4$ **(b)** over the subdomain, $3.5 < x < 3.75$, $z < 1.25$ on a y normal plane at $y = 0$.

Figure 9. Heat map indicating the size of the elements in x, y and z across a plane at $y = 1$ for a subset of the domain ($-1 < x < 12$) and times $t = 8$ (left) and $t = 20$ (right). Note that the domain extends to $x = 18$. The region $12 < x < 18$ had no significant regions with element sizes smaller than 0.1.

by a sharp interface in c at the front of the head. There is also high resolution around vortices and at the left-hand wall where there is a strong recirculation of the overlying fluid. The increase in resolution at the left-hand wall is present at both $t = 8$ and at $t = 20$. The same increase in resolution is not present at the right-hand wall at either $t = 8$ or $t = 20$ (not shown in Fig. 9). This is because the fluid has more space to recirculate in front of the current than behind it in both cases such that the curvature of the velocity field, and thus the mesh resolution, is less. In the y direction the largest region of high resolution is in the gravity current body just behind the head of the current due to recirculating flow in this region and hence high curvature in c and $\boldsymbol{u}_2$. The distribution of element sizes in this region varies rapidly due to the three-dimensional turbulent structure of the flow. There is a large difference in the resolution between $t = 8$ and $t = 20$. Within the current the resolution has generally decreased by $t = 20$ but a significantly larger proportion of the domain is below the maximum element size. Generally, the elements appear to be smaller in the z direction. This implies that the interpolation error bound may be proportionally tighter on $\boldsymbol{u}$ in this direction, and could be reduced a little to bring the resolution in line with the other directions.

In Sect. 4 it was shown that an adaptive simulation with the interpolation error bounds used here compared well with the finest fixed mesh simulation (6.25×10^{-3}). It is no surprise that, in the most computationally demanding regions of the flow, Fig. 9 shows that the smallest element edge lengths in the adaptive simulation match well with the fixed mesh edge length.

Adaptivity does of course come at a cost. The mean time required for a parallel adapt operation throughout this simulation, including mesh adaptation, data transfer, mesh partitioning and data migration, was 110 s. This can be compared to a mean time required to compute a time step in parallel of 67 s. Therefore, when adapting every five time steps, approximately 1/4 of the simulation time is spent adapting
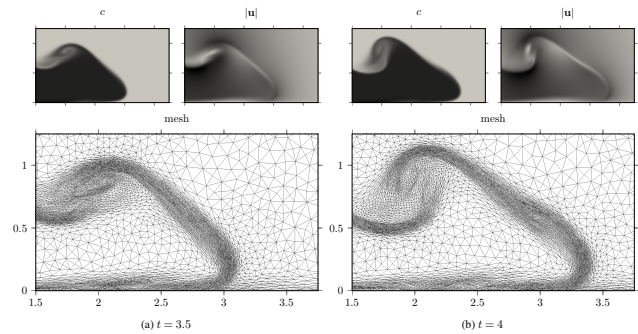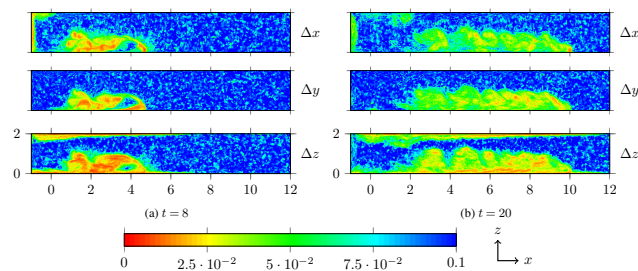
the mesh, or the run time is increased by 33 % compared to a fixed mesh simulation using the same number of elements. When adapting every two time steps approximately 1/2 of the simulation is spent in the adapt stage. The mesh optimisation algorithm used provides the most flexibility for mesh refinement, and hence will produce a highly optimised mesh, but it is potentially more expensive than other adaptivity algorithms. A high percentage of the total simulation time is spent in the adapt phase and hence it may be worth considering cheaper alternatives based upon hierarchical refinement for future models. Regardless of this, the benefits of reducing the number of elements by two orders of magnitude far outweigh the cost of adaptivity. The simulation required approximately 100 000 processor hours. Over 500 cores this equates to just under a week of run time. Assuming a linear increase in run time with number of elements, a fixed mesh simulation would have taken at least an order of magnitude longer and would have been nearly impossible to post-process.

The run time of a simulation is dependent upon many parameters. This run time includes the time required for many online diagnostics and the writing of data to disk. When using adaptivity the frequency of adapts is an additional key parameter that governs a trade-off between total processing hours and time to completion. Increasing the time between
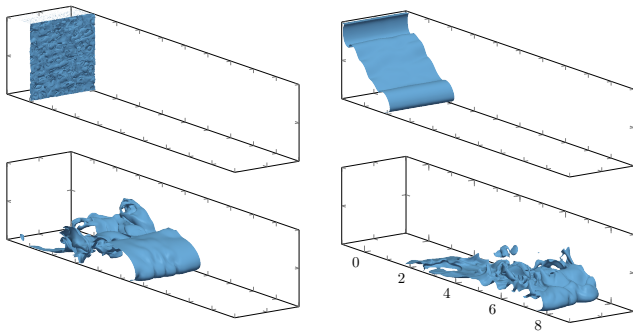
Figure 10. Propagation of density interface over time. This figure shows a contour at a concentration 0.25 at times 0, 2, 8 and 14.

Figure 11. Position of density current head against simulation time for the Fluidity simulation and the simulations of Necker et al. (2002) and Espath et al. (2014).

adapts requires a larger amount of elements to ensure that there is adequate resolution throughout the period between adapts. This will increase the number of processing hours required to solve the problem. However, with more elements, the problem can be split amongst more cores whilst keeping the minimum number of elements per core constant. The time to completion is then likely to reduce as fewer adapt operations are required. This parameter can be varied dependent upon what is important to the scientist. Size of output, time to completion, the cost of processor time, and the size of available computers must all be taken in to account. Another adaptive simulation with similar properties, but slightly varied parameters, could require significantly more, or less total processing hours.

6   Results

Figure 10 shows how the density current propagates along the tank in three dimensions. The perturbation in the initial condition for the concentration field is shown in the image relating to $t = 0$. This initial condition creates the initial three-dimensional instabilities required to generate a realistic density current. By $t = 8$ this flow is fully turbulent and three-dimensional. This is in agreement with other models (Necker et al., 2002; Espath et al., 2014). The well-known structures of lobes and clefts are present at the front of the density current from this point onward.

Figure 11 shows how the head position varies with time throughout the simulation. This is computed as the maximum x value, averaged across the width of the domain, obtained from a sediment concentration contour at $c = 0.01$. Lines are plotted showing the agreement with other models. The results compare well with Espath et al. (2014), whose model predicted a head position slightly in front of the model by Necker et al. (2002). Espath et al. (2014) noted the importance of the initial condition to the development of the flow. The initial conditions used in this work are slightly different to those of Espath et al. (2014) and Necker et al. (2002) and hence complete agreement is not expected.
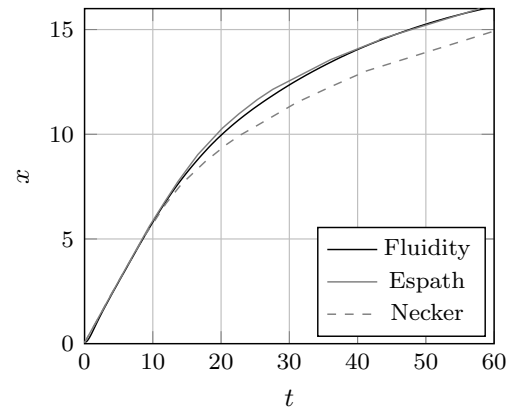
Figure 12 shows the spatially integrated deposition rates over the course of the simulation. Again the deposition rate shows good agreement with other published values. As noted by Necker et al. (2002), the deposition rate increases at a rate proportional to $t^{0.5}$ until approximately $t = 14$, at which point there is a sharp change and the deposition rate begins to drop rapidly at a rate proportional to $t^{-2.5}$. A key difference between the results from this work and those of the other models is the presence of erosion in this simulation. The deposition rates from this simulation are higher than those of Necker et al. (2002) and Espath et al. (2014). Noting that the vertical fluid velocities are small near the bed due to the no-slip boundary condition, and that eroded sediment will be settling, the majority of eroded sediment will almost immediately be deposited, and will never be fully entrained back into the flow. This will lead to an increased deposition rate compared to a simulation without erosion. By making the assumption that all eroded sediment is immediately deposited, a modified deposit rate can be calculated for the Fluidity simulation with the effect of erosion removed. As shown in Fig. 10, this modified deposition rate shows much better agreement with the results of both Necker et al. (2002) and Espath et al. (2014), leading to the conclusion that it is the inclusion of erosion in the simulation that led to the higher deposition rate.

An important diagnostic for applications is the final deposit profile from a particle-laden density current. Figure 13 shows the span-wise averaged deposit profile from the three-dimensional Fluidity simulation compared against those of previous modellers and also from the experiments of De Rooij and Dalziel (2001). A good match is observed in the peak deposit height of $\eta \approx 0.12$ at $x_1 \approx 4$ between all of the models and the experimental results.

There is a notable variation in deposit depths near the lock gate. All models show a smaller deposit depth in this region when compared to the experimental results. The reason for
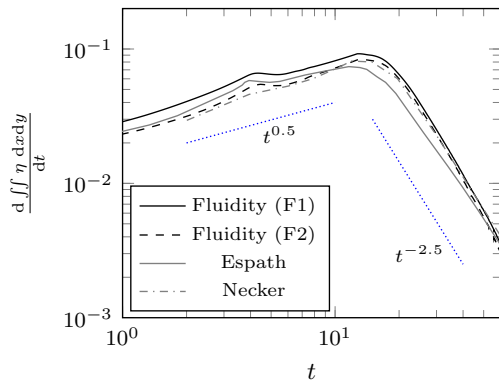
Figure 12. Deposition rates for the three-dimensional simulation. Deposition rates from Fluidity (F1), Fluidity with a modified deposition rate where the erosion rate has been removed (F2), Espath et al. (2014), and Necker et al. (2002).
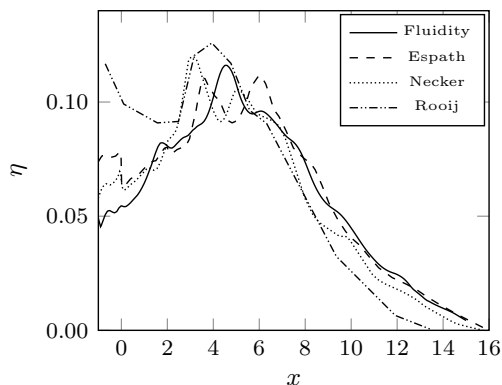

Figure 13. Span-wise averaged deposit profile from the three-dimensional simulation at $t = 60$. Comparisons are made against numerical results of Necker et al. (2002) and Espath et al. (2014) and experimental results of De Rooij and Dalziel (2001).

this is unclear and explanations can only be speculative. One potential cause may be that the sediment in the experimental setup had already begun to settle before the lock gate was released. This may also help to explain the slightly shorter run-out distance resulting from a reduced initial potential energy. Alternatively, there may be processes occurring in the laboratory that are not accurately captured by the computational models.

The results from Fluidity are further from the measured results than the other models in this region. The inclusion of an erosion algorithm is the likely cause of this. The experimental measurements show larger deposits than all of the models upstream, and smaller deposits downstream. Erosional processes will predominately decrease upstream deposits and increase downstream deposits, and hence would increase this discrepancy if applied to any of the models. In addition to this, the erosion algorithm is not configured correctly to match the De Rooij and Dalziel (2001) experiment: $R_p \approx 1$ for the De Rooij and Dalziel (2001) experiment, in

comparison to $R_p \approx 20$ in the Fluidity simulation. This will result in significantly more erosion in the simulation than is likely to have occurred in the experiment.

7 Conclusions

This paper presents validation of a novel three-dimensional finite-element model for simulating particle-laden density currents. The model is validated by assessing the convergence of key variables in two-dimensional simulations and by comparison with results from previous DNS three-dimensional simulations by Necker et al. (2002) and Espath et al. (2014). It has been shown that by using adaptive meshing the number of required elements in these simulations can be reduced by between one and two orders of magnitude. This makes DNS modelling of particle-laden density currents at moderate Grashof numbers an achievable goal using finite elements.

In addition, simulations within complex domains can be achieved fairly trivially using the flexibility afforded by unstructured finite elements. Future work will study flow of turbidity currents along circular channels and across breaks in slope, and will help answer outstanding questions about the dynamics of flows in these situations. Using mesh adaptivity also makes modelling in very large domains achievable. Large regions of the domain where very little is happening come at very little cost. This will enable modelling turbidity currents in deep water where the dynamics are not dominated by the bore created by the overlying fluid. It may also allow for simulations of turbidity current along sinuous channels with over-spilling as well as numerous other similar scenarios.

The cost of these simulations is still very high. It may be possible to further reduce the cost of these simulations, whilst retaining important three-dimensional dynamics, using large eddy simulation (LES). Future work will focus on implementation and testing of LES for this simulation with the aim of reducing the simulation cost.

References

Akiyama, J. and Stefan, H.: Turbidity current with erosion and deposition, J. Hydraul. Eng.-ASCE, 111, 1473–1496, 1985.

Alavian, V., Jirka, G., Denton, R., Johnson, M., Stefan, H., and Members, A.: Density currents entering lakes and reservoirs, J. Hydraul. Eng.-ASCE, 118, 1464–1489, doi:10.1061/(ASCE)0733-9429(1992)118:11(1464), 1992.

Bassi, F. and Rebay, S.: A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, J. Comput. Phys., 131, 267–279, 1997.

Best, J., Kotaschuk, R., Peakall, J., Villard, P., and Franklin, M.: Whole flow field dynamics and velocity pulsing within natural sediment-laden underflows, Geology, 33, 765–768, doi:10.1130/G21516.1, 2005.

Blanchette, F., Strauss, M., Meiburg, E., Kneller, B., and Glinsky, M.: High-resolution numerical simulations of resuspending gravity currents: conditions for self-sustainment, J. Geophys. Res., 110, C12022, doi:10.1029/2005JC002927, 2005.

Boman, E. G., Catalyurek, U. V., Chevalier, C., and Devine, K. D.: The Zoltan and Isorropia parallel toolkits for combinatorial scientific computing: partitioning, ordering, and colorin, Scientific Programming, 20, 129–150, 2012.

Bombardelli, F., Cantero, M., Buscaglia, G., and Garcia, M.: Comparative study of convergence of CFD commercial codes when simulating dense underflows, Mecánica Computacional, 23, 1187–1199, 2004.

Bonnecaze, R., Huppert, H., and Lister, J.: Particle-driven gravity currents, J. Fluid Mech., 250, 339–339, 1993.

Cantero, M., Balachandar, S., Garcia, M., and Ferry, J.: Direct numerical simulations of planar and cylindrical density currents, J. App. Mech., 73, 923–930, doi:10.1115/1.2173671, 2006.

Cantero, M., Lee, J., Balachandar, S., and Garcia, M.: On the front velocity of gravity currents, J. Fluid Mech., 586, 1–39, doi:10.1017/S0022112007005769, 2007.

Cantero, M. I., Garcia, M. H., and Balachandar, S.: Effect of particle inertia on the dynamics of depositional particulate density currents, Comput. Geosci., 34, 1307–1318, doi:10.1016/j.cageo.2008.02.002, 2008.

Chen, L., Sun, P., and Xu, J.: Optimal anisotropic meshes for minimizing interpolation errors in L^p-norm, Math. Comput., 76, 179–204, 2007.

Ciarlet, P. G.: Basic error estimates for elliptic problems, in: Handbook of Numerical Analysis, 2, 17–351, 1991.

Cockburn, B. and Shu, C.-W.: Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput., 16, 173–261, 2001.

Cotter, C. J., Ham, D. A., Pain, C. C., and Reich, S.: LBB stability of a mixed Galerkin finite element pair for fluid flow simulations, J. Comput. Phys., 228, 336–348, doi:10.1016/j.jcp.2008.09.014, 2009.

Curran, K., Hill, P., and Milligan, T.: The role of particle aggregation in size-dependent deposition of drill mud, Cont. Shelf Res., 22, 405–416, doi:10.1016/S0278-4343(01)00082-6, 2002.

Davis, R. H. and Hassen, M. A.: Spreading of the interface at the top of a slightly polydisperse sedimenting suspension, J. Fluid Mech., 196, 107–134, 1988.

De Rooij, F. and Dalziel, S. B.: Particulate Gravity Currents, Vol. 31 of Special Publication International Association of Sedimentologists, chap. Time- and Space-Resolved Measurements of Deposition under Turbidity Currents, edited by: McCaffrey, W., Kneller, B. C., and Peakall, J., 207–215, Blackwell Publishing Ltd., doi:10.1002/9781444304275.ch15, 2001.

Donea, J. and Huerta, A.: Finite Element Methods for Flow Problems, John Wiley & Sons, Ltd, doi:10.1002/0470013826.fmatter, 2005.

Espath, L., Pinto, L., Laizet, S., and Silvestrini, J.: Two- and three-dimensional direct numerical simulation of particle-laden gravity currents, Comput. Geosci., 63, 9–16, doi:10.1016/j.cageo.2013.10.006, 2014.

Farrell, P. and Maddison, J.: Conservative interpolation between volume meshes by local Galerkin projection, Comput. Method. Appl. M., 200, 89–100, 2011.

Farrell, P., Piggott, M., Pain, C., Gorman, G., and Wilson, C.: Conservative interpolation between unstructured meshes via supermesh construction, Comput. Method. Appl. M., 198, 2632–2642, 2009.

Fortin, M.: Estimation a posteriori et adaptation de maillages, Revue Européenne des Éléments Finis, Rev. Eur. Élém. Finis, 9, 369–504, 2000.

Frey, P.-J. and Alauzet, F.: Anisotropic mesh adaptation for CFD computations, Comp. Method. Appl. M., 194, 5068–5082, 2005.

Fukushima, Y., Parker, G., and Pantin, H.: Prediction of ignitive turbidity currents in Scripps Submarine Canyon, Mar. Geol., 67, 55–81, 1985.

Garcia, M. and Parker, G.: Entrainment of bed sediment into suspension, J. Hydraul. Eng.-ASCE, 117, 414–435, 1991.

Georgoulas, A., Angelidis, P., Panagiotidis, T., and Kotsovinos, N.: 3D numerical modelling of turbidity currents, Environ. Fluid Mech., 10, 603–635, doi:10.1007/s10652-010-9182-z, 2010.

Gladstone, C., Phillips, J., and Sparks, R.: Experiments on bidisperse, constant-volume gravity currents: propagation and sediment deposition, Sedimentology, 45, 833–843, 1998.

Hallworth, M. and Huppert, H.: Abrupt transitions in high-concentration, particle-driven gravity currents, Phys. Fluids, 10, 1083–1087, 1998.

Harris, T., Hogg, A., and Huppert, H.: Polydisperse particle-driven gravity currents, J. Fluid Mech., 472, 333–371, doi:10.1017/S0022112002002379, 2002.

Hartel, C., Kleiser, L., Michaud, M., and Stein, C.: A direct numerical simulation approach to the study of intrusion fronts, J. Eng. Math., 32, 103–120, 1997.

Hartel, C., Meiburg, E., and Necker, F.: Analysis and direct numerical simulation of the flow at a gravity-current head. Part 1. Flow topology and front speed for slip and no-slip boundaries, J. Fluid Mech., 418, 189–212, 2000.

Heezen, B. C. and Ewing, W. M.: Turbidity currents and submarine slumps, and the 1929 Grand Banks [Newfoundland] earthquake, Am. J. Sci., 250, 849–873, 1952.

Hiester, H., Piggott, M., and Allison, P.: The impact of mesh adaptivity on the gravity current front speed in a two-dimensional lock-exchange, Ocean Model., 38, 1–21, doi:10.1016/j.ocemod.2011.01.003, 2011.

Hiester, H., Piggott, M., Farrell, P., and Allison, P.: Assessment of spurious mixing in adaptive mesh simulations of the two-dimensional lock-exchange, Ocean Model., 73, 30–44, 2014.

Huang, H., Imran, J., and Pirmez, C.: Numerical model of turbidity currents with a deforming bottom boundary, J. Hy-

draul. Eng.-ASCE, 131, 283–293, doi:10.1061/(ASCE)0733-9429(2005)131:4(283), 2005.

Huang, H., Imran, J., and Pirmez, C.: Numerical modeling of poorly sorted depositional turbidity currents, J. Geophys. Res., 112, 1–15, doi:10.1029/2006JC003778, 2007.

Imperial College London AMCG: Fluidity manual v4.1.11, doi:10.6084/m9.figshare.995912, 2014.

Kneller, B. and Buckee, C.: The structure and fluid mechanics of turbidity currents: a review of some recent studies and their geological implications, Sedimentology, 47, 62–94, doi:10.1046/j.1365-3091.2000.047s1062.x, 2000.

Kuzmin, D.: A vertex-based hierarchical slope limiter for *p*-adaptive discontinuous Galerkin methods, J. Comput. Appl. Math., 233, 3077–3085, doi:10.1016/j.cam.2009.05.028, 2010.

Loseille, A. and Alauzet, F.: Continuous mesh framework Part II: Validations and applications, SIAM J. Numer. Anal., 49, 61–86, 2011.

Maerz, J., Verney, R., Wirtz, K., and Feudel, U.: Modeling flocculation processes: Intercomparison of a size class-based model and a distribution-based model, Cont. Shelf Res., 31, S84–S93, doi:10.1016/j.csr.2010.05.011, 2011.

Maxey, M. R. and Riley, J. J.: Equation of motion for a small rigid sphere in a nonuniform flow, Phys. Fluids, 26, 883–889, 1983.

Mohd-Yusof, J.: Development of immersed boundary methods for complex geometries, Center for Turbulence Research, Annual Research Briefs, 325–326, 1998.

Nasr-Azadani, M. M. and Meiburg, E.: TURBINS: An immersed boundary, Navier–Stokes code for the simulation of gravity and turbidity currents interacting with complex topographies, Comput. Fluids, 45, 14–28, 2011.

Necker, F., Hartel, C., Kleiser, L., and Meiburg, E.: High-resolution simulations of particle-driven gravity currents, Int. J. Multiphas. Flow, 28, 279–300, doi:10.1016/S0301-9322(01)00065-9, 2002.

Ooi, S., Constantinescu, G., and Weber, L.: 2D large-Eddy simulation of lock-exchange gravity current flows at high Grashof numbers, J. Hydraul. Eng.-ASCE, 133, 1037–1047, 2007.

Pain, C., Umpleby, A., De Oliveira, C., and Goddard, A.: Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations, Comput. Method. Appl. M., 190, 3771–3796, 2001.

Parker, G., Fukushima, Y., and Pantin, H.: Self-accelerating turbidity currents, J. Fluid Mech., 171, 145–81, 1986.

Peraire, J. and Persson, P.: The compact discontinuous Galerkin (CDG) method for elliptic problems, SIAM J. Sci. Comput., 30, 1806–1824, 2008.

Piggott, M. D., Pain, C. C., Gorman, G. J., Power, P. W., and Goddard, A. J. H.: *h*, *r*, and *hr* adaptivity with applications in numerical ocean modelling, Ocean Model., 10, 95–113, doi:10.1016/j.ocemod.2004.07.007, 2005.

Piggott, M. D., Pain, C. C., Gorman, G. J., Marshall, D. P., and Killworth, P. D.: Unstructured adaptive meshes for ocean modeling, Geoph. Monog. Series, 177, 383–408, 2008.

Piggott, M. D., Farrell, P. E., Wilson, C. R., Gorman, G. J., and Pain, C. C.: Anisotropic mesh adaptivity for multi-scale ocean modelling, Philos. T. Roy. Soc. A, 367, 4591–4611, 2009.

Sequeiros, O., Cantelli, A., Viparelli, E., White, J., Garcia, M., and Parker, G.: Modeling turbidity currents with nonuniform sediment and reverse buoyancy, Water Resour. Res., 45, 1–28, doi:10.1029/2008WR007422, 2009.

Serchi, F. G., Peakall, J., Ingham, D., and Burns, A.: A numerical study of the triggering mechanism of a lock-release density current, Eur. J. Mech B-Fluid., 33, 25–39, doi:10.1016/j.euromechflu.2011.12.004, 2012.

Talling, P., Wynn, R., Masson, D., Frenz, M., Cronin, B., Schiebel, R., Akhmetzhanov, A., Dallmeier-Tiessen, S., Benetti, S., Weaver, P., Georgiopoulou, A., Zuhlsdorff, C., and Amy, L.: Onset of submarine debris flow deposition far from original giant landslide, Nature, 450, 541–544, doi:10.1038/nature06313, 2007.

Talling, P. J., Masson, D. G., Sumner, E. J., and Malgesini, G.: Subaqueous sediment density flows: Depositional processes and deposit types, Sedimentology, 59, 1937–2003, doi:10.1111/j.1365-3091.2012.01353.x, 2012.

Winters, K. B., Lombard, P. N., Riley, J. J., and D'Asaro, E. A.: Available potential energy and mixing in density-stratified fluids, J. Fluid Mech., 289, 115–128, 1995.