Journal of Computational Physics 375 (2018) 1220–1237

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Modelling local scour near structures with combined mesh movement and mesh optimisation

J.M. Nunez Rattia [*], J.R. Percival, S.J. Neethling, M.D. Piggott

Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London, London, SW7 2ZA, United Kingdom

ARTICLE INFO

Article history:
Received 28 February 2018
Received in revised form 10 September 2018
Accepted 12 September 2018
Available online 17 September 2018

Keywords:
Sediment transport
Scour
Turbulence modelling
Finite element method
Mesh adaptivity
Mesh movement

ABSTRACT

This paper develops a new implementation coupling optimisation-based anisotropic mesh adaptivity algorithms to a moving mesh numerical scour model, considering both turbulent suspended and bedload sediment transport. The significant flexibility over mesh structure and resolution, in space and time, that the coupling of these approaches provides makes this framework highly suitable for resolving individual marine structure scales with larger scale ocean dynamics. The use of mesh optimisation addresses the issue of poor mesh quality and/or inappropriate resolution that have compromised existing modelling approaches that apply mesh movement strategies alone, especially in the case of extreme scour. Discontinuous Galerkin finite element-based discretisation methods and a Reynolds Averaged Navier–Stokes-based turbulent modelling approach are used for the hydrodynamic fluid flow. In this work the model is verified in two dimensions for current-dominated scour near a horizontal pipeline. Combined adaptive mesh movement and anisotropic mesh optimisation is found to maintain both the quality and validity of the mesh in response to morphological bed evolution changes, even in the case where it is severely constrained by nearby structures.

1. Introduction

A variety of mathematical and computational models of scour around hydraulic structures, such as offshore pipelines or wind turbine foundations, have been developed over recent years. These seek to describe the coupling between the hydrodynamic and morphological components of the scouring process. Accurate modelling of scour is important since this can lead to the damage and failure of hydraulic and marine structures. Large sums of money are spent in the repair of marine structures as a result of scouring [1]. Hence significant investment is also made in scour protection, guided by predictions of the potential failure mechanisms. Complex numerical models can be used to simulate the (turbulent) flow around the structure, ideally with fully coupled two-way interactions with the morphology of the erodible bed. Seabed morphological models involve a sediment transport description to calculate erosion/accretion processes. An important element of the sediment transport description are formulae for the bedload and suspended transport.

The evolution of the boundary mesh location in response to bed morphodynamics also requires changes to the internal computational mesh. If this is not done, or is not done well, this can lead to poor quality meshes which compromise simulation stability and accuracy. A widely used approach to this problem is to use mesh movement methods to propagate

[*] Corresponding author.
E-mail address: j.nunez-rattia14@imperial.ac.uk (J.M. Nunez Rattia).

https://doi.org/10.1016/j.jcp.2018.09.026

the boundary mesh movement into the domain, in an attempt to maintain mesh quality throughout [2,3]. However, in the case of extreme bed movement and/or where this movement is close to a fixed structure in the domain (which inevitably constrains the mesh and its ideal movement to some degree), it is very difficult to maintain mesh quality with mesh movement algorithms alone.

One of the earliest studies to present a holistic dynamic description of the local scour process at submarine pipelines, taking into account both sediment transport contributions (i.e. bedload and suspended transport), was performed by Brørs [4]. In that approach the computational mesh evolved in time in order to represent the moving bed location due to scour under a single pipeline. A structured computational grid was utilised with each node in the domain moved vertically, maintaining a relative spacing. But this type of method can lead to problems with resolution and mesh characteristics (e.g. orthogonality and skewness), especially in the case of large bed deformations [2,4,5]. With unstructured computational grids, it is relatively straightforward to use variable resolution within the domain, but due to arbitrary connectivity and cell shape it is also arguably more complicated to move the mesh nodes on the bed and in the domain interior. Although the development of moving grid techniques has been generally less well studied than other adaptive mesh techniques, there have been a lot of progress in the past decades, such as partial differential equations (PDE)-based and spring-based analogy methods [6]. From the moving mesh PDE-based approaches, the relatively simple Laplacian smoothing has been a common choice to tackle mesh deformation in response to local sediment scour due to its robustness and ease of implementation with an unstructured grid in a complex domain [7]. As we will argue in this paper (section 5), movement of nodes alone may not be sufficient to maintain an adequate mesh resolution and/or structure, especially for cases with extreme bed deformation.

With the wider development of unstructured grid based methods in computational fluid dynamics (CFD), a broad range of techniques for arbitrary resolution specification and updating of the computational mesh are available. The main motivation for these is to impart both resolution and geometric flexibility as well as to optimise computational efficiency: minimising computational cost for a required accuracy, or maximising accuracy for a given cost. Here a further goal is targeted: the use of this range of techniques to maintain an optimal mesh when subjected to external boundary deformation.

Techniques for updating the mesh include those which: (i) perform local topological operations (e.g. sub-dividing elements, or changing mesh connectivity through the swapping of element faces and edges) to change the size, and potentially the shape, of mesh elements (variously termed adaptive mesh refinement (AMR), *h*-adaptivity and mesh optimisation depending on the specifics of the algorithm [8–10]); (ii) methods which continuously move or redistribute the location of mesh vertices while maintaining mesh connectivity (often termed mesh movement or *r*-adaptivity [6]); and (iii) their combination in so-called *hr*-adaptive methods [11].

In this work we present a new mesh optimisation/movement (or *hr*-adaptive) framework for computational morphodynamics. This includes the use of relatively sophisticated mesh movement algorithms to account for the bed evolution, while utilising mesh optimisation methods in order to: maintain mesh quality under extreme and complex scenarios; vary the total degree of freedom count as the problem complexity evolves; and help track solution features in the wider domain. The framework will be demonstrated on a complex scour problem with a hydraulic structure (i.e. pipeline) close to the bed; the closeness being particularly challenging due to the constraints the structure imposes on the mesh movement, and the significant difference in degree of freedom count ideally required in the gap between structure and bed as the scour progresses at different times into the simulation. Scenarios considered include both live-bed (i.e. the approaching flow continuously transports sediment into a local scour hole) and clear-water (i.e. the approaching flow is clear and does not contain sediment) conditions. Results are benchmarked against laboratory data and prior numerical studies which also considered the same test case. This paper represents a significant extension and improvement over our previous paper [12] which reported on the initial steps in this work.

The remainder of this paper is organized as follows. In section 2, we review the governing mathematical equations which are relevant for the physics of local scour and turbulent flows. In section 3, we present control volume/finite element-based numerical discretisation algorithms for solving coupled computational fluid dynamics with sediment conservation laws. Section 4 describes the adaptivity algorithms we employ, including mesh movement and optimisation-based anisotropic mesh adaptivity. Numerical experiments are carried out in section 5, where validation and benchmarking against physical and numerical experiments are considered.

2. Mathematical equations

This section presents the mathematical formulation for the flow field model and the scour model including the sediment transport equations.

2.1. Flow model

The approach taken here for the hydrodynamics is to consider the single phase solution of the incompressible Reynolds-averaged Navier–Stokes (RANS) equations under the Boussinesq buoyancy approximation. This comprises the continuity equation (incompressibility condition)

$$\nabla \cdot \mathbf{u} = 0, \tag{1}$$

and the momentum equation

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = -\nabla\left(\frac{p}{\rho_0}\right) + \nabla\cdot\left(\nu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)\right) + \frac{\rho}{\rho_0}\mathbf{g}, \tag{2}$$

where $\mathbf{u} = (u, v, w)^T$ is the velocity field in Cartesian coordinates, ρ is the fluid density, p is the total pressure, $\mathbf{g} = (0, 0, -g)^T$ is the acceleration due to gravity, pointing in the negative vertical direction, and $\nu = \nu_0 + \nu_T$ is the sum of the kinematic molecular viscosity ($\nu_0 = 10^{-6}\ m^2/s$) of water and an additional *eddy* viscosity, ν_T, used to represent subgrid-scale turbulent processes. The Boussinesq buoyancy approximation, valid for small density differences $|\rho - \rho_0| \ll \rho_0$, has been invoked in order to allow us to replace ρ by ρ_0 (a reference water density, $\rho_0 = 10^3\ kg/m^3$) in all terms of the momentum equation, except where the density is multiplied by gravity. In the applications considered here the density variations due to suspended sediment concentration dominate all other state variables dependencies, and density (ρ_s is the sediment particle density) the variations are assumed to have a linear dependence on the sediment concentration, c, i.e. our equation of state is assumed to take the form $\rho = \rho_0 + (\rho_s - \rho_0)\,c$.

2.2. Turbulence closure

The standard RANS, two-equation, $k - \epsilon$ turbulence model with near-wall treatment [13] is used in this study to parameterise unresolved mixing length scales through a turbulent or eddy viscosity, ν_T. Two coupled scalar advection–diffusion–reaction equations are solved to determine the magnitude of the turbulent eddy viscosity, defined as

$$\nu_T = C_\mu \frac{k^2}{\epsilon},$$

where $C_\mu = 0.09$ is an empirically chosen constant, k is the turbulent kinetic energy (TKE) and ϵ is the TKE dissipation rate. The two additional prognostic equations, which are also coupled to the momentum equation (2), take the form

$$\frac{\partial k}{\partial t} + \nabla\cdot\left(k\mathbf{u} - \left(\nu_0 + \frac{\nu_T}{\sigma_k}\right)\nabla k\right) = P_k - \epsilon, \tag{3}$$

$$\frac{\partial \epsilon}{\partial t} + \nabla\cdot\left(\epsilon\mathbf{u} - \left(\nu_0 + \frac{\nu_T}{\sigma_\epsilon}\right)\nabla \epsilon\right) = \frac{\epsilon}{k}\left(C_1 P_k - C_2\epsilon\right), \tag{4}$$

where $P_k = \frac{\nu_T}{2}\left(\left\|\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right\|^2\right)$ and ϵ account for the production and dissipation of turbulent kinetic energy, respectively. The values of $\sigma_k = 1.0$, $\sigma_\epsilon = 1.3$, $C_1 = 1.44$ and $C_2 = 1.92$ are further empirically chosen constants.

Equations (1) and (2), together with (3) and (4) need to be solved with appropriate initial and boundary conditions which will be discussed later (section 5).

2.2.1. Boundary conditions for the turbulence model

Near-wall modelling is clearly critical for this application and hence boundary conditions need to be treated carefully. In line with the implementation described in [13], a thin boundary region of fluid of width y is conceptually excluded from the computational domain, with boundary conditions on the velocity, tangential stress and for k and ϵ applied at the interface of the excluded region. Here the normal component of the fluid velocity, $\mathbf{u}\cdot\mathbf{n}$ is set to match that of the moving boundary mesh velocity, $\boldsymbol{V}$, i.e. there is no penetration.

It is further assumed that the tangential component of the stress on the boundary satisfies a relation

$$\mathbf{n}\cdot\boldsymbol{\tau_w} = -\rho_0 u_\tau^2 \frac{\mathbf{u}}{\|\mathbf{u}\|} = -\rho_0 \frac{u_\tau}{\nu_T}\frac{\mathbf{u}}{y_+}, \qquad \text{where} \qquad u_\tau = \max\left\{C_\mu^{0.25}\sqrt{k}, \frac{\|\mathbf{u}\|}{y_+}\right\}. \tag{5}$$

Here $\boldsymbol{\tau_w} = \rho_0\nu\left[\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right]_{z=\eta}$ is the wall shear stress, u_τ is a friction velocity and

$$y_+ = \frac{u_\tau y}{\nu_0},$$

is a nondimensional wall distance. The validity of the definition of u_τ in (5) depends on the criterion that y is chosen at the interface between the viscous sublayer, in which the velocity is assumed to increase linearly away from the wall:

$$\frac{\|\mathbf{u}\|}{u_\tau} = y_+,$$

and the logarithmic turbulent boundary layer, in which it is assumed that

$$\frac{\|\mathbf{u}\|}{u_\tau} = \frac{1}{\kappa}\log y_+ + \beta.$$

Here $\kappa = 0.41$ is the von Kármán constant and $\beta = 5.2$ is another constant for the case of smooth walls. This criterion is enforced by the ansatz that $y_+ = 11.06$, at the bottom of the logarithmic layer range, $11.06 \leq y_+ \leq 300$.

Functionally, the constant y_+ implementation of (5) acts as a constant coefficient quadratic drag boundary condition applied to the momentum equation (2). The empirically determined law of the wall boundary conditions for k and ϵ at the edge of the computational domain are implemented in a weak form as Neumann boundary conditions,

$$\mathbf{n}\cdot\nabla k = 0 \qquad \text{and} \qquad \mathbf{n}\cdot\nabla\epsilon = \frac{u_\tau^5}{\sigma_\epsilon \nu_T y_+}.$$

2.3. Sediment transport model

Simulations of sediment transport coupled with boundary (i.e. bed) evolution obey a fundamental mass balance conservation law (hereafter, the Exner equation) and they are commonly performed under the discretisation of a particular form of the general bed level change equation [14]

$$\frac{\partial \eta}{\partial t} = -\frac{1}{1-n}\left[\nabla\cdot(\mathbf{q_b}+\mathbf{q_s}) + \frac{\partial}{\partial t}\int_{\eta}^{\eta+H} c\,dz\right], \tag{6}$$

where, $\eta(x, y, t)$ represents the bed level, n is the porosity of the bed, and $\mathbf{q_b}$ and $\mathbf{q_s}$ are the bedload rate and suspended rate vectors, respectively. The final term on the right-hand side represents the temporal change in the volume of sediment in the water column of height H, where c denotes sediment concentration.

The total sediment transport is handled by splitting it into bedload and suspended load contributions. The approaches for these two types of sediment transport are detailed below. These sediment transport formulations are limited to compacted (i.e. no void space, $n = 0$) well-graded sands.

2.3.1. Bedload transport

Bedload transport is the part of the sediment transport load taking place at the bed surface resulting in the motion of particles due to rolling, sliding or travelling in small jumps (saltation) along the bed. Many aspects of sediment morphodynamics are governed predominantly by the bedload transport rate, including river erosion, bed form evolution, and potentially the scour around structures such as pipelines or bridge abutments.

Numerous bedload transport formulations appear in the literature and generally consist of semi-empirical relations derived from flume-based experimental data. The bedload transport rate is then calculated as a function of the magnitude of bed/wall shear stress, $\|\boldsymbol{\tau_w}\|$, above its critical value, τ_c. Dimensional analysis then leads to the dimensionless relationships (expressed with a superscript $*$) for the bedload sediment transport rate, $\mathbf{q}^* = f\left(\tau^*, d^*, R^*, Re_p^*\right)$, where,

$$\mathbf{q}^* = \frac{\mathbf{q_b}}{\sqrt{R^* g d_{50}^3}}, \qquad \text{(Einstein number)}$$

$$\tau^* = \frac{\|\boldsymbol{\tau_w}\|}{\rho R^* g d_{50}}, \qquad \text{(Shields number)}$$

$$d^* = d_{50}\left[\frac{R^* g}{\nu_0^2}\right]^{1/3}, \qquad \text{(Dimensionless particle diameter)}$$

$$R^* = \frac{(\rho_s - \rho_0)}{\rho_0}, \qquad \text{(Submerged specific gravity)}$$

$$Re_p^* = \frac{\sqrt{R^* g d_{50}^3}}{\nu_0}. \qquad \text{(Particle Reynolds number)}$$

Generally, granular bed-material will remain still in the bed until the flow initiates incipient motion. This is termed the threshold of motion, with the bed shear stress that causes it termed the critical stress, τ_c. One fitting to experimental data for this critical stress is given by Soulsby [15] as

$$\tau_c^* = \frac{0.30}{1+1.2d^*} + 0.055\left[1 - e^{-0.020d^*}\right].$$

Amongst the many bedload transport formulae proposed in the literature, we make use here of the Engelund & Fredsœ [16] model

$$\mathbf{q}^* = 18.74\left(\tau^* - \tau_c{}^*\right)\left(\sqrt{\tau^*} - 0.7\sqrt{\tau_c^*}\right)\frac{\mathbf{u}}{\|\mathbf{u}\|} \tag{7}$$

(with an applicability range for relatively small particle diameters $0.19\ mm < d_{50} < 0.93\ mm$).

2.3.2. Bed slope control

Most bedload sediment transport formulae are derived and calibrated for rivers with shallow slopes [17]. In excess shear stress scour conditions where a local obstruction (e.g. pipeline or bridge pier) is present, the localised increase in shear stress during the development of scour holes may lead to areas in a model where the local face slope exceeds the angle of repose, ϕ, which is a geotechnical parameter to measure the stability of sediment particles (somewhere between $30^\circ < \phi < 35^\circ$ for loose sands). Numerical algorithms have been proposed to overcome these difficulties in modelling the over-steepening of the bed slope and allow the bed-material to slide down through the adjustment of the bedload flux of sediment in a conservative fashion.

Apsley and Stansby [18] described a flux based algorithm which prescribes an additional *avalanche flux* which is added to the bedload flux, based on the angle of the actual slope, α, and the direction of steepest slope, $\mathbf{b} = \nabla\eta/\|\nabla\eta\|$. This takes the form

$$\mathbf{q}_{\mathrm{aval}} = \begin{cases} (1-n)\,\frac{L^2(\tan(\alpha)-\tan(\phi))}{2\cos(\alpha)\Delta t}\mathbf{b}, & \tan(\alpha) > \tan(\phi)\,, \\ 0, & \text{otherwise,} \end{cases}$$

where L (related to the cell length) and Δt (current timestep) are dimensional values arising from the problem being modelled.

This avalanche flux based approach was used for this study. Other algorithms for sand sliding follow a more geometric approach. For example, Liang et al. [2] described a one-dimensional method which lowers the slope of a cell's facet to yield a new stable slope, but this approach leaves an open issue regarding mass conservation. Another similar idea in Niemann et al. [19] is proposed by setting the new positions of the vertices constituting the sliding face, derived from a function that behaves like the avalanche flux and raises/lowers the bed level according to the mass balance of sediment (i.e. area distribution). This addresses the issue of conservation but it is a method geometrically designed just for one-dimensional slope control.

2.3.3. Suspended sediment transport

Suspended sediment transport involves the particulate material that is carried within the water column. These particles are kept in suspension due to turbulent mixing causing an upward flux which competes with the inherent downward particle settling velocity, u_{si}, due to buoyancy effects.

The sediment concentration, c, is the volume of sediment per total volume of material (fluid and sediment). The suspended sediment concentration can be determined by solving an advection–diffusion equation, with an additional convective term to represent the gravitational effect of the settling velocity:

$$\frac{\partial c}{\partial t} + \nabla\cdot\left[c\left(\mathbf{u} - u_{si}\mathbf{k}\right)\right] = \nabla\cdot(\kappa\nabla c)\,, \qquad \text{where} \qquad \kappa = \kappa_0 + \frac{\nu_T}{S_c^*}. \tag{8}$$

The magnitude of the particle concentration eddy diffusivity κ, is inversely proportional to the Schmidt number, S_c^*, which represents a ratio of the momentum diffusivity (total viscosity) and mass diffusivity (sediment concentration) and is generally chosen to be in the range 0.7–1.0. In the rest of this work $S_c^* = 1$ is assumed.

The settling velocity is obtained from the formula for unhindered sand particles (valid for diameters $d_{50} > 0.1\,mm$) in clear-water [15]:

$$u_{so} = \frac{\nu_0}{d_{50}}\left[\left(10.36^2 + 1.049d^{*3}\right)^{1/2} - 10.36\right].$$

As the sediment concentration rises the settling velocity usually drops, which is essential to account for in the actual suspended sediment concentration effect [20], resulting in the correction $u_{si} = u_{so}\,(1-c)^{2.39}$.

2.3.4. Bed evolution model

Morphological changes to a given bathymetry occur when there is an inequilibrium between deposited and eroded material in a localised region. If the amount of deposited material exceeds the amount of eroded material, it leads to an increase in bed level and vice versa. This mechanism is described by the Exner equation (6) for sediment transport in a simpler form [21]:

$$\frac{\partial\eta}{\partial t} = -\frac{1}{1-n}\left(\nabla_{\mathrm{surf}}\cdot\mathbf{q_b} + (E - D)\right) \tag{9}$$

where, the morphological change is split into contributions from the surface divergence,

$$\nabla_{\mathrm{surf}} = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, 0\right),$$

of the bedload transport and the suspended sediment transport, where E and D represent erosion and deposition respectively.

The terms of the suspended sediment load can be expressed using empirically-derived relations which describe changes in the volume of sediment in the water column together with the suspended sediment load vector (i.e. $\mathbf{q_s}$). The deposition rate,

$$D = u_{si} c_0, \tag{10}$$

is calculated from the settling velocity of the sediment particles, denoted u_{si}, multiplied by the near-the-bed concentration (c_0). The erosion or entrainment rate,

$$E = -\kappa \frac{\partial c}{\partial z},$$

can be expressed in terms of the local turbulent eddy diffusivity (κ) and the vertical concentration gradient, or equivalently the *turbulent* flux of particles up from the bed. Many empirical formulations are available in the literature for estimating the re-entrainment rate of suspended sediment. Some studies [22,23] have performed detailed comparison of formulae and validated these against laboratory results. In this work the model of van Rijn [23] is selected where the entrainment rate can be written in dimensionless form as

$$E^* = \frac{E}{u_{si}} = 0.015 \frac{d_{50} \Psi^{1.5}}{Z_b {d_*}^{0.3}}, \qquad \text{where} \qquad \Psi = \frac{\tau^*}{{\tau_c}^*} - 1, \tag{11}$$

d_{50} is the median particle diameter and Z_b represents a reference level very near the bed close in value to the sand roughness. The reference level assumed was $Z_b = 3d_{90}$, where d_{90} represents the grain diameter of the 90th percentile (90% of sediment is finer).

It is worth noting that the rate of bed change as expressed in (9) depends on scalar quantities (i.e. D and E), which results in a major simplification for the suspended sediment transport given that the numerical integration of the storage term over the vertical direction (the final term in (6)) is replaced by evaluating the suspended sediment deposition and entrainment rates.

3. Discretisation and numerical solution procedure

The discretisation of the equations described in the previous section are performed here within the control volume/finite element-based model, Fluidity [24], upon unstructured triangular or tetrahedral computational meshes (in this work only results on two-dimensional triangular meshes will be presented).

3.1. Flow model

Equation (2) has been spatially discretised using a $\text{P1}_{DG}\text{P2}$ finite element pair in which the velocity field varies linearly over elements, and may be discontinuous between elements, while pressure has a quadratic variation within and is continuous between elements [25]. In a discontinuous Galerkin (DG) discretisation the shape functions can be selected so that the field variables (and derivatives) are free to be discontinuous between element boundaries. A field that is discretised using a discontinuous function space will have multiple nodal values at element nodes and thus will have more degrees of freedom than a continuous function of the same polynomial order on the same mesh, since these elements do not have repeated nodes. DG methods are locally conservative so they are a popular choice of discretisation for advection-dominated problems as opposed to some continuous Galerkin options such as Petrov–Galerkin-based methods which can lead to excessive numerical diffusion to maintain stability. Also, thanks to their flux representation they perform well on arbitrary meshes and have the convenient properties of producing a block-diagonal mass matrix that can be easily inverted locally for each element, so it does not have to be lumped, thereby resulting in a very efficient computational time.

The use of DG methods for the velocity discretisation requires the choice of flux schemes for both the advective (e.g. an upwind flux approach is used here) and diffusive (e.g. the Bassi–Rebay discretisation method [26] is used here) terms. A mathematical description of the discretised momentum equation is given in section 4.2, where a moving reference frame is assumed.

In a discontinuous Galerkin method of degree $p > 0$, potential numerical oscillations can be expected for the advective part of the discretisation. For P1_{DG} elements, over- and under-shoots in slopes derived after the spatial reconstruction of the inter-element fluxes, $\hat{u}|_{\Gamma_e}$ (where $\hat{u}$ is the convective flux of a scalar quantity transported by a velocity field through the boundary element, Γ_e) can be controlled in order to ensure a bounded solution using slope limiting methods (the vertex-based slope limiter [27] is used here) with the goal of not increasing the total variation of the solution and thus aiding stability and robustness. In the vertex-based limiter, the idea is to bound the admissible slope such that the maximum and minimum centroid values, u_i over the elements surrounding each vertex $\mathbf{x}_i$ are within the bounds of the volume average value, u_c of each element Ω_e:

$$u_i^{\max} = \max\left\{u_c, u_i^{\max}\right\}, \quad u_i^{\min} = \min\left\{u_c, u_i^{\min}\right\}, \tag{12}$$

and this leads to a constrained solution of the form:

$$u_i(\mathbf{x}) = u_c + \alpha_e (\nabla u)_c (\mathbf{x}_i - \mathbf{x}_c), \tag{13}$$

where $(\nabla u)_c$ is the local gradient within the element Ω_e before the slope limiting is applied and α_e is the element-wise correction factor equal to:

$$\min_i = \begin{cases} \min\left\{1, \frac{u_i^{\max} - u_c}{u_i - u_c}\right\}, & \text{if} \quad u_i - u_c > 0, \\ 1, & \text{if} \quad u_i - u_c = 0, \\ \min\left\{1, \frac{u_i^{\min} - u_c}{u_i - u_c}\right\}, & \text{if} \quad u_i - u_c < 0. \end{cases} \tag{14}$$

The procedure used here for solving the discretised system of equations is a pressure correction-based approach. The nonlinear advection term in the momentum equation (2) is dealt with using two Picard iterations to ensure that only linear matrix systems need to be solved. In each iteration first the linearised momentum equation is solved for a preliminary velocity in isolation using a pressure guess. This velocity solution does not satisfy the continuity equation (1) and thus a second pressure correction step is invoked to update the velocity to be solenoidal (divergence-free) and finally the corrected velocity and updated pressure are obtained [28].

The time marching algorithm (θ-scheme) employed here for the flow equations takes an operator-splitting approach in which the diffusive term is solved using an implicit, Crank–Nicolson scheme ($\theta = 0.5$) with an adaptive timestep chosen automatically such that the maximum Courant number over the mesh is always below a user defined value (in this work the value 2 was selected), while the advective term is solved using an explicit forward Euler scheme with a subcycled timestep chosen such that the maximum Courant number for the subcycling is below another user defined value (here 0.2 is used).

3.2. Turbulence model

Equations (3) and (4) are spatially discretised using a flux-limited, control volume-based method [24,9]. A vertex-centred approach on the finite element mesh is employed, with the control volumes around each vertex defined by connecting the mid points of element edges with the centroids of the surrounding elements (in the two-dimensional case). To ensure boundedness of the solution a Sweby limiter [29] is used to combine low and high order approximations to the fluxes across control volume boundaries.

For the temporal discretisation, and the coupling between the κ and ϵ equations as well as momentum, the equations are firstly decoupled, linearised and each equation is solved using the best available values from previous nonlinear Picard iterations (generally two iterations are found to be a good compromise between accuracy and efficiency).

3.3. Sediment transport model

The numerical treatment of the sediment transport model involves a two-step approach in which the results of the suspended sediment (concentration) equation (8) are an input for the Exner equation (9) describing the boundary deformation over an $(N-1)$-dimensional space corresponding to the boundary of the N-dimensional space the flow is computed over. We use the notation Ω_{surf} for this $(N-1)$-dimensional space, formed by projecting out the vertical dimension, and Γ_{surf} for its boundary.

The discretisation procedure used for (8) in this work is very similar to that already described for the two scalar PDEs in the turbulence model. This sediment volume fraction advection–diffusion equation takes the matrix form:

$$\mathcal{M}\frac{c^{n+1} - c^n}{\Delta t} + \mathcal{A}_s\left(\mathbf{u}^{n+\theta_{nl}}, u_{si}\right) c^{n+\theta} + \mathcal{K}_s c^{n+\theta} = 0, \tag{15}$$

where $\mathcal{A}_s\left(\mathbf{u}^{n+\theta_{nl}}, u_{si}\right)$ is the sediment advection matrix, which is a function of the velocity and the particle settling velocity, and $\mathcal{K}_s$ is the sediment diffusion matrix. The nonlinear advection term in the sediment concentration equation is linearised with a nonlinear relaxation parameter, $\theta_{nl} = [0, 1]$ such as $\mathbf{u}^{n+\theta_{nl}} = \theta_{nl}\bar{\mathbf{u}}^{n+1} + (1-\theta_{nl})\,\mathbf{u}^n$, where $\bar{\mathbf{u}}^{n+1}$ is the current best guess of $\mathbf{u}^{n+1}$. For more details of the discretised mathematical model for suspended sediment refer to [30].

As defined by (9), the bed level update is dependent on $\nabla_{\text{surf}} \cdot \mathbf{q_b}$, the surface divergence of the bedload flux, and the erosion and deposition of sediment at the bed boundary. For uniform, steady-state flow and sediment particles of moderate diameter, the variation of $\eta(x, y, t)$ is governed partially by the sediment exchange with the bed-material, but mainly through the divergence of the bedload transport.

Considering bedload transport only and assuming $n = 0$, the relevant portion of the sediment mass balance or Exner equation (9) can be written in the form

$$\frac{\partial \eta}{\partial t} + \nabla_{\text{surf}} \cdot \mathbf{q_b} = 0.$$

This equation provides the rate of bed level change, with the implementation here utilising a Galerkin finite element-based discretisation over the bed surface, using linear trial basis functions equal to test functions and denoted by Ψ_i, for every degree of freedom indexed by i. The weak formulation of this problem then takes the form

$$M_{ij}\frac{\partial \eta_j}{\partial t} = \int\limits_{\Omega_{\text{surf}}} \Psi_i \Psi_j \frac{\partial \eta_j}{\partial t}\, d\Omega_{\text{surf}} = -\left[\int\limits_{\Omega_{\text{surf}}} \mathbf{q_b} \cdot \nabla \Psi_i \, d\Omega_{\text{surf}} + \int\limits_{\Gamma_{\text{surf}}} \mathbf{q_b} \Psi_i \cdot \mathbf{n}\, d\Gamma_{\text{surf}}\right].$$

The literature shows that unrealistic bed profiles can be generated due to numerical instabilities arising from the nonlinear coupling of the flow field and the bed level update (e.g. [18,4,2]). Some studies [31] have illustrated stability issues with this nonlinear hyperbolic (i.e. Exner) equation and concluded that measures need to be taken to avoid numerical instabilities which often manifest at the grid scale through surface wiggles or sawtooth patterns as the surface boundary moves or deforms. One approach to deal with this issue is to add a regularisation term in the matrix-based finite element construction, acting as a diffusion-like filter with a length parameter that can be related to the grid size. This results in the slightly modified *mass matrix*

$$M_{ij} = \int\limits_{\Omega_{\text{surf}}} \left[\left(\Psi_i \Psi_j\right) + \lambda^2 \left(\nabla_{\text{surf}} \Psi_i \cdot \nabla_{\text{surf}} \Psi_j\right)\right] d\Omega_{\text{surf}},$$

where λ is a smoothing length scale, in this work chosen to be twice the length of a typical line segment on the bed boundary. This implementation effectively smooths the bedload transport source term in the Exner equation. This is in contrast to other filtering approaches which explicitly smooth the bed coordinate η itself, introducing a spurious restoring force and potentially mass conservation issues.

The diagnostic field for the net suspended load $(E - D)$ flux is computed as the difference between the erosion, E derived from (11) and deposition, D derived from (10), with the latter computed using the concentration field at the bed (15).

The moving boundary (here the bed) is computed by updating the vertical position of the vertices making up the bottom boundary at each timestep. First, the flow field model is calculated by solving the Navier–Stokes equations with the turbulence model. Then the obtained flow parameters, including the bed/wall shear stress, are employed to solve the bedload and suspended load models allowing a derivation of the boundary displacement or mesh velocity, $\boldsymbol{V}$, at the water-sediment interface:

$$\boldsymbol{V} = \left(0, 0, \frac{\partial \eta}{\partial t}\right), \qquad \text{on} \qquad z = \eta. \tag{16}$$

This in turn triggers a mesh movement algorithm which is applied at every timestep and hence acts continuously, combined with a mesh optimisation step invoked every user-defined number of timesteps (here 15 is used). Both of these mesh update algorithms are described in the following section.

Due to the difference in time scales governing the hydrodynamics and bed changes, it is standard practice in computational morphodynamics to use a morphological acceleration factor to reduce computational costs. This can be thought of as defining a morphodynamic timestep which is much longer than the flow timestep (often by a factor of 10 to 1000) [2,7]. A morphological acceleration factor (obtained through trial and error in order to control the morphology computation) of 10 is generally used in the early stages of the simulations performed here where bed changes are more rapid, and up to 60 at later stages where changes are more gradual.

4. Mesh adaptivity and parallelisation

4.1. Mesh optimisation/h-adaptivity

The primary objective motivating the use of anisotropic mesh optimisation in this study is to maintain optimal spatial resolution (by which we mean both the shape and size of elements) under external boundary deformation and the associated internal mesh movement. An application of mesh optimisation is relatively costly (perhaps equivalent to several timesteps of the underlying solver, and being dependent on choices over interpolation scheme etc.) and so is generally applied perhaps only every 10–20 timesteps. This is perfectly acceptable here since mesh movement is used every timestep and thus is continuously seeking to maintain a good quality mesh in response to boundary deformation.

Ignoring load balancing and parallelisation which will be discussed below, the application of mesh optimisation can be divided into three steps. The first is to decide what mesh is desired via the generation and use of an appropriate error metric; the second is to actually generate the newly adapted mesh through a series of topological operations applied to the current mesh; and the third is to transfer (or interpolate) all necessary data from the previous mesh to the newly adapted one in order to continue the simulation.

In the first step, to drive mesh optimisation, Fluidity makes use of an explicit Hessian-based approach motivated by interpolation error theory [8]. Local error indicators are derived based on the second derivatives, or Hessian, of one or more scalar flow variables within the domain. For piecewise linear discretisations the interpolation error over an element K can be bounded [32],

$$\|u - \Pi_h u\| \leq C \max_{\mathbf{v}\in E_K} \left\{ \mathbf{v}^T |\mathcal{H}_u| \mathbf{v} \right\}, \tag{17}$$

where Π_h is the $P1$ interpolation operator, $\|\cdot\|$ is a norm acceptable for error control (e.g. the L^∞ norm), $\mathbf{v}$ represents an edge vector in E_K, the set of all edges of K, and C is an $\mathcal{O}(1)$ constant independent of the current mesh and related to the spatial dimension. $\mathcal{H}_u$ is the Hessian of the scalar field u, and the notation $|\mathcal{H}_u|$ indicates that the absolute values of its eigenvalues have been taken (as we are simply interested in the magnitude of errors here).

Hessian-based anisotropic adaptation aims to produce simplex meshes with arbitrarily-oriented shapes following a metric-based stretching. Several methods have been proposed in the literature to compute the Hessian of discrete scalar variables [33]. In Fluidity, a Galerkin projection procedure [34] is used to obtain the Hessian by recovering the gradient of the solution first and then repeating, while making use of mass lumping. The metric tensor is then formed from the recovered Hessian as

$$\mathcal{M}_u = \frac{1}{\epsilon_u} |\mathcal{H}_u|,$$

where ϵ_u is a user-defined weighting. We can then measure the lengths of edges in the mesh with respect to this metric via $\|\mathbf{v}\|_{\mathcal{M}_u} = \sqrt{\mathbf{v}^T \mathcal{M}_u \mathbf{v}}$. If we make the assumption that $C = 1$ then we can interpret ϵ_u as a desired interpolation error in the quantity u. In practice it simply provides a means of controlling the degree of refinement – a smaller ϵ_u leading to a finer mesh.

Other authors have proposed different metric tensor formulations. For example, [35] considered the interpolation error in the L^p norm with the associated metric

$$\mathcal{M}_{L^p} = \frac{1}{\epsilon_u} \det\left(|\mathcal{H}_u|\right)^{\frac{-1}{2p+n}} |\mathcal{H}_u|,$$

where n is the spatial dimension. The inclusion of the determinant of the Hessian as an extra scaling factor results in smaller spatial scales being given more weight in the metric and as a result are better resolved [36]. Following [36] and [37], in this work this form for the metric with $p = 2$ is employed and was found to produce effective results in boundary layer simulations.

Metric tensors can be constructed for separate scalar fields (including the individual components of the velocity vector) and then appropriated combined into a single metric used by the mesh optimisation library. For example, for two metrics derived from two variables, $\mathcal{M}_1$ and $\mathcal{M}_2$, an intersection $\mathcal{M}_1 \cap \mathcal{M}_2$ can be defined such that the desired interpolation error is achieved for all variables [32,34]. In this work, the velocity components are chosen to represent the physics (including boundary layers near surfaces) that will drive the mesh adaptation.

Once the metric formed from the solution fields is constructed, it can be useful to apply additional constraints to the mesh elements such as imposing minimum ($h_{\min}$) and maximum ($h_{\max}$) edge lengths in order to avoid uncontrolled mesh behaviour (e.g. limiting the production of elements larger than the solution domain or smaller than a certain problem scale). This is represented by a modification of the eigenvalues as,

$$\lambda_i = \min\left(\max\left(|\lambda_i|, \frac{1}{h_{\max}^2} \right) \frac{1}{h_{\min}^2} \right). \tag{18}$$

Other bounds such as maximum allowable element aspect ratios, maximum number of elements and gradation over the rate of element size variation can also be introduced [24].

Once the final error metric tensor, $\mathcal{M}$, has been computed it can be used to measure the edge lengths of the elements in the mesh. Metric-based anisotropic mesh adaptation seeks to equidistribute errors such that local mesh modifications (described below) are performed aiming at creating a unit mesh, $\|\mathbf{v}\|_{\mathcal{M}} = 1$ for all edges, i.e. all elements in metric space are isotropic and of unit size. Since the metric tensor is originally formulated from the Hessian matrices of individual solution fields, achieving this goal has the effect of generating (anisotropic) elements in physical space which have small edge lengths in directions and locations where the second derivative of the solution fields are large, and are large in directions and locations where the fields are near linear [10].

The second step involves achieving the goal of constructing a near equilateral and unit sized mesh in metric space. Here a mesh optimisation procedure is employed where a series of topological operations are performed on the mesh to achieve this goal. A mesh quality objective functional is formed which accounts for both the shape and size of elements, as measured with respect to the metric. In this work, a two-dimensional functional is used

$$Q_{\mathcal{M}} = 12\sqrt{3} \frac{|\Delta|_{\mathcal{M}}}{|\partial\Delta|_{\mathcal{M}}^2} F\left(\frac{|\partial\Delta|_{\mathcal{M}}}{3} \right), \quad \text{with,} \tag{19}$$

$$F(x)=\left[\min\left(x,\frac{1}{x}\right)\left(2-\min\left(x,\frac{1}{x}\right)\right)\right]^3,$$

where, $|\Delta|$ represents the area of the element Δ and $|\partial\Delta|$ its perimeter, both measured in the metric $\mathcal{M}$. For more details see [9] and the references therein. A series of topological operations are then applied to the current mesh to minimise this mesh quality functional. In this work, the two-dimensional library ani2d [38] (http://ani2d.sourceforge.net/) is used to perform these mesh operations which include: (i) node creation or edge splitting, (ii) node deletion or edge collapsing, (iii) swapping of common edges between two elements. Note that in three dimensions more operations are possible (e.g. edge to edge, edge to face, and face to edge swapping) [39].

The third and final step involves transferring field data from the old mesh to the new mesh in an interpolation process. The simplest option to transfer data is called consistent interpolation which simply involves evaluating the globally defined finite element solution from the old mesh at the nodal locations of the new mesh. This is the simplest possible approach for any finite element method to regenerate data on a new mesh, but it is non-conservative and is unsuited to discontinuous function spaces such as are used here to represent the velocity field. These issues may be overcome using a conservative Galerkin projection interpolation scheme to map solution fields between meshes. Here this is achieved using a supermeshing based algorithm which triangulates the polygons resulting from the local intersections of the two meshes, thus allowing the integrals in the Galerkin projection to be computed exactly. Further details on the projection and the construction of the supermesh may be found in [40,41]. This approach is used here for the interpolation of the velocity field.

4.2. Mesh movement/r-adaptivity

In a fundamentally different manner to mesh optimisation, mesh movement or r-adaptivity methods locally redistribute the physical locations of vertices of the mesh while maintaining its structural connectivity. In this work this is performed continuously over the course of every timestep and thus the underlying discretisation is formulated within an Arbitrary Lagrangian–Eulerian (ALE) framework [42].

In the Fluidity implementation, this requires that for each simulation timestep, $t_n \to t_{n+1}$, we specify a continuous, piecewise linear vector field, or mesh velocity, $\boldsymbol{V}$, whose action defines the rate of change of the vertices of the elements such that the physical coordinate, $\boldsymbol{x}_i$, of the ith mesh vertex varies as $\boldsymbol{x}_i(t)=\boldsymbol{x}_i(t_n)+(t-t_n)\boldsymbol{V}_i$, with the physical motion of the rest of the mesh following via linearity of the coordinate field. This implicitly defines a (fixed) mesh coordinate, $\boldsymbol{\chi}=\boldsymbol{x}(t_{n+\theta})$, on which the finite element equations are constructed. The time derivative in the inertial terms in the PDEs describing the evolution of the prognostic variables then transform as

$$\left.\frac{\partial a}{\partial t}\right|_{\boldsymbol{\chi}}=\left.\frac{\partial a}{\partial t}\right|_{\boldsymbol{x}}+\sum_i\left.\frac{dx_i}{dt}\right|_{\boldsymbol{\chi}}\frac{\partial a}{\partial x_i}=\left.\frac{\partial a}{\partial t}\right|_{\boldsymbol{x}}+\boldsymbol{V}\cdot\nabla_{\boldsymbol{x}}a.$$

For the momentum equation, application of integration by parts, the Crank–Nicolson timestepping, the Bassi–Rebay diffusion operator and choice of an upwinded numerical flux gives a weak form in Ω being a bounded open domain with boundary Γ representing the general Dirichlet (Γ_D)–Neumann (Γ_N) problem; using tests functions $\boldsymbol{w}$ and $\boldsymbol{\tau}$, respectively, integrating over an element e,

$$\begin{aligned}\sum_e\int_{\Omega^{(e)}}\boldsymbol{w}\cdot\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t}-\nabla_{\boldsymbol{x}}\cdot(\mathbf{u}^*\boldsymbol{w})\cdot\left(\frac{1}{2}\mathbf{u}^{n+1}+\frac{1}{2}\mathbf{u}^n-\boldsymbol{V}\right)dV_{\boldsymbol{\chi}}\\+\int_{\Omega^{(e)}}\nabla_{\boldsymbol{x}}\boldsymbol{w}:\underline{\boldsymbol{\sigma}}^{n+\frac{1}{2}}+\nabla_{\boldsymbol{x}}\cdot\boldsymbol{w}p^{n+\frac{1}{2}}\,dV_{\boldsymbol{\chi}}\\+\oint_{\Gamma_I^e}\boldsymbol{w}\cdot\left\{\left\{\frac{\mathbf{u}^{n+1}+\mathbf{u}^n}{2}\right\}\right\}\mathbf{u}^*\cdot\mathcal{J}^{-1}\cdot\boldsymbol{n}_{\boldsymbol{X}}+\overline{\boldsymbol{w}}\cdot\overline{\underline{\boldsymbol{\sigma}}\cdot\mathcal{J}^{-1}}\cdot\boldsymbol{n}_{\boldsymbol{X}}+dS_{\boldsymbol{\chi}}\\+\oint_{\Gamma_D^e}\mathbf{u}_D\cdot\boldsymbol{w}\mathbf{u}_D\cdot\mathcal{J}^{-1}\cdot\boldsymbol{n}_{\boldsymbol{X}}dS_{\boldsymbol{\chi}}-\oint_{\Gamma_N^e}\mathbf{u}\cdot\boldsymbol{w}\mathbf{u}^*\cdot\mathcal{J}^{-1}\cdot\boldsymbol{n}_{\boldsymbol{X}}+\boldsymbol{w}\cdot\underline{\boldsymbol{\sigma}}dS_{\boldsymbol{\chi}}=0,\end{aligned}$$

where $\mathbf{u}^*$ is the advective velocity guess from the Picard iteration, projected onto a continuous P1 space, and $\underline{\boldsymbol{\sigma}}$ is the P1_{DG} Bassi–Rebay auxiliary stress term satisfying a local finite element equation

$$\int_{\Omega^e}\boldsymbol{\tau}:\underline{\boldsymbol{\sigma}}dV_{\boldsymbol{\chi}}=\oint(\nu+\nu_T)\,\boldsymbol{\tau}:\left(\overline{\mathbf{u}}\boldsymbol{n}+\boldsymbol{n}\overline{\mathbf{u}}\right)dS_{\boldsymbol{\chi}}-\int(\nu+\nu_T)\,\mathbf{u}\nabla\cdot\boldsymbol{\tau}+(\nu+\nu_T)\,(\nabla\cdot\boldsymbol{\tau})^T\,\mathbf{u}dV_{\boldsymbol{\chi}}.$$

Interfacial averaging and upwinding operators in the interior faces subset Γ_I, are defined as

$$\overline{\boldsymbol{a}}=\frac{1}{2}\left(\boldsymbol{a}^-+\boldsymbol{a}^+\right),\quad\{\{\boldsymbol{a}\}\}=\frac{1}{2}\left(\boldsymbol{a}^-+\boldsymbol{a}^+\right)-\frac{\mathbf{u}^*\cdot\boldsymbol{n}}{2\|\mathbf{u}^*\cdot\boldsymbol{n}\|}\left(\boldsymbol{a}^--\boldsymbol{a}^+\right),$$

where a^+ is the value of a variable in the eth element and a^- is the value from the element across the interface. Finally, $\mathcal{J}_{ij} = \frac{\partial x_i}{\partial \chi_j}$ is the Jacobian matrix of the moving mesh transformation at the time level the gradient operator applies.

In this study the mesh velocity of the moving boundary is obtained from the rate of bed level change arising from the Exner equation (9) and propagated into the interior of the mesh, under a specified smoothness criterion, and with the other boundaries held fixed. In this way, the moving mesh method is used to continuously adjust interior node locations in response to bed deformation, while seeking to keep the mesh of high quality and resisting the tangling of elements.

Since the mesh velocity is chosen explicitly, it is possible to choose the mesh coordinate as $\boldsymbol{\chi} = \boldsymbol{x}(t_0) + \frac{1}{2}\Delta t \boldsymbol{V}$, i.e. the physical coordinate at the time level of the right hand side of the PDE. This allows the substitutions $\nabla_{\boldsymbol{x}} = \nabla_{\boldsymbol{\chi}}$, $\mathcal{J} = \mathcal{I}$ in the weak form above, so that the DG method takes essentially the same form as for fixed-mesh problems. A similar method is applied for the control volume discretisation of the tracer quantities (sediment, turbulent kinetic energy and turbulent dissipation). In practice, since the timestepping operator is split and advection solved using an explicit method, this introduces a small discrepancy, of equal order to the use of an explicit timestepping method. Since the bed movement is slow relative to the fluid velocity, we neglect the advection sub-cycling in the wall boundary conditions for the turbulence model.

Previous scour studies, e.g. [2,3,7], have typically applied an approach equivalent to a pure Laplacian smoothing criterion. This simple scheme is appropriate on vertically structured, quadrilateral-based meshes, or to small displacements in which tangling is a minor issue. In this work we demonstrate the advantages of a lineal elasticity approach, specifically a torsional spring-based smoother [43]. This smoother is derived from a discrete analogy to the deformation of an elastic material kept in constant quasi-equilibrium to an externally applied boundary displacement. The individual vertex displacements (and thus the mesh velocity) are assumed to satisfy equations analogous to linear elasticity equations with a fictitious spring attached to each edge connecting two vertices i and j,

$$\mathcal{K}_{ij}\left[\boldsymbol{x}_j(t_{n+1}) - \boldsymbol{x}_j(t_n)\right] = (t_{n+1} - t_n)\mathcal{K}_{ij}\boldsymbol{V}_j = \boldsymbol{0}, \tag{20}$$

along with the boundary condition (16). The lineal torsional stiffness matrix, $\mathcal{K}$, may be partitioned as

$$\mathcal{K} = \sum_{\boldsymbol{k}\in\text{edges}} \mathcal{K}_{\boldsymbol{k}}^{\text{lineal}} + \mathcal{K}_{\boldsymbol{k}}^{\text{torsional}}.$$

Here the lineal term, $\mathcal{K}_{\boldsymbol{k}}^{\text{lineal}}$ applies equal and oppositely directed linear Hookean pseudo-forces at the two vertices sharing an edge, $\boldsymbol{k}$, with a spring constant inversely proportional to the edge length. This term thus resists possible displacements which collapse the edge and acts to equipartition changes in edge length across the whole mesh. Similarly the torsional term, $\mathcal{K}_{\boldsymbol{k}}^{\text{torsional}}$, applies an angular Hookean force on the third node of each of the two elements sharing an edge (assuming we are dealing with two-dimensional triangular elements), with a torsional spring force inversely proportional to the element area, thus resisting displacements which would cause the mesh to tangle by moving that vertex through the edge. The sparse matrix equation (20) for the $\boldsymbol{V}_i$ is solved using the same solvers as the discretised fluid model. This method provides a more robust smoother on unstructured meshes than the Laplacian smoother, at the cost of more expensive assembly.

4.3. Load balancing and parallelisation

In order to parallelise the model (both the discretisation as well as the application of mesh adaptivity), the first step is to divide the whole mesh into a number of sub-domains through a process referred to as domain decomposition; here this is a pre-processing step. After the initial decomposition of the mesh, each Message Passing Interface (MPI) process then reads their local mesh and performs local finite element assembly to form global discretisation matrices which are solved using linear solvers and preconditioners available here through interfacing with the PETSc library (http://www.mcs.anl.gov/petsc/). Since the mesh optimisation step (described above) alters the local degrees of freedom count on each sub-domain, and consequently leads to each MPI process taking a different amount of time to complete impacting on overall solver efficiency, a dynamic load balancing step is a necessity. This involves exchanging elements (or migrating data) between sub-domains and here a parallel graph partitioner algorithm is utilised to achieve this via interfacing with the Zoltan library (http://www.cs.sandia.gov/zoltan/). The load balancing step is additionally used to parallelise a serial mesh optimisation algorithm. Each process optimises the mesh locally with halo regions between sub-domains locked, and hence the overall mesh contains unoptimised elements. Edge weighting is then applied at the graph partitioning step to encourage the load balancing to perturb updated halo regions away from previously locked elements. Local mesh optimisation, and load balancing, steps are then repeated several times resulting in a parallelised mesh adaptivity algorithm as well as a load-balanced mesh where all elements have been considered for optimisation. Further details may be found in [39,44].

5. Numerical results and discussions

In order to demonstrate and validate the modelling approach developed in this work we consider a test case of current-induced scour around a circular object in 2D (which can be thought of as an infinitely long horizontally-lying pipe). This is an important problem since scour can affect the stability of pipelines, and serves as a simple yet representative example of further real-world problems which may involve more complex geometries. It is also a very challenging problem for two reasons: (i) the presence of the circular void representing the structure significantly constrains the mesh and its movement;

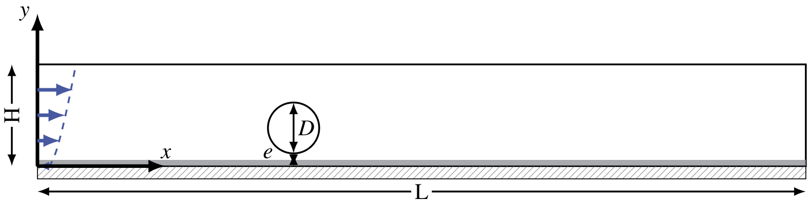


Fig. 1. Schematic of the two-dimensional domain configuration. The grey area indicates a region of infinite sediment excluded from volume calculations. e is the gap between the circle and the bottom erodible boundary.

Table 1
Parameters used for simulations of the horizontal pipe scour test case. These parameters correspond to those used in Liang et al. [2].

Parameter		Experiment $\tau^* = 0.048$	Experiment $\tau^* = 0.098$
Geometry and Flow			
Diameter	D	0.1 m	0.1 m
Maximum gap height	e	10 mm	10 mm
Mean inlet velocity	$\|\mathbf{u}\|$	0.35 m/s	0.5 m/s
Sediment Transport			
Angle of repose	ϕ	30°	30°
Submerged specific gravity	R^*	1.65	1.65
Particle median diameter	d_{50}	0.36 mm	0.36 mm
Particle diameter 90th percentile	d_{90}	0.48 mm	0.48 mm
Unhindered settling velocity	u_{so}	0.055 m/s	0.055 m/s

and (ii) the initial thin clearance between the pipe and the bed leads to high stresses and strong scour, and thus the need for a high degree of mesh movement and also significant differences in the number of degrees of freedom required to accurately model the currents (and hence scouring processes) between the early and late stages of a simulation. This problem also benefits from the availability of experimental data [45] as well as prior modelling studies, e.g. [2].

5.1. Simulation configuration

A two-dimensional initially rectangular flow domain is considered, as shown in Fig. 1. The domain is $L = 30D$ long and $H = 4D$ high, where D is the pipe diameter. The pipe centre was located $10D$ downstream of the inlet boundary. The height of the pipe was such that its bottom just touches the location of a flat bed.

When the pressure gradient in the bed-material between the upstream and downstream zones around a bottom-lying pipe reaches a critical value for seepage failure, tunnel erosion may initiate and the current velocity in the gap then increases rapidly [46]. As with similar modelling studies, e.g. [2], this onset phase was skipped here because the physics of the seepage flow within the porous soil underneath the pipeline is not accounted for in this modelling approach. The starting bottom boundary location/shape was therefore adjusted to yield a small initial clearance of size e between pipe and bed. Following Liang et al. [2] the initial bed location was given an initial negative half cycle sinusoidal shaped small perturbation to yield a localised gap between the bottom of the pipe and the bed of amplitude $e = 0.1D$.

In terms of initial and boundary conditions, the simulated flow is assumed to be a hydrostatic pressure-balanced rigid lid flow (i.e. no free surface is present in the model). At the inlet (left-hand side) boundary, a logarithmic velocity profile with a mean inlet velocity was applied (commonly developed in channel geometries) and the turbulent initial quantities were approximated from CFD formulae used for free-stream conditions and based on the turbulence intensity ($I = 5\%$) and the turbulent length scale l, estimated as a small percentage of the channel height (i.e. $l = 0.07H$). At the outlet boundary the pressure perturbation is set to zero. The bottom erodible boundary and the circle/pipe surface are defined with a standard turbulence wall functions reproducing the behaviour of the boundary layer (see section 2.2.1). For sediment concentration boundary conditions the flux of sediment through all surfaces, except the bottom, is set to zero while for the bottom boundary the suspended sediment flux is derived as the net upward normal flux such that $\mathbf{q_s} \cdot \mathbf{n} = E - D$. Within the bottom boundary, the sediment concentration is assumed to be uniform and the volume of sediment is infinitely available such that the bed is always erodible, i.e. the simulations here cannot hit bedrock.

5.2. Scour near horizontal pipe

The numerical simulations were compared to the laboratory experimental results of Mao [45]. While the set-up of this problem is uniform in the cross-flume direction, some three-dimensional effects will of course still be present in reality. However, following the approach taken successfully by a number of authors studying this problem numerically, we also utilise a two-dimensional approach. An overview of the system parameters is given in Table 1. The simulations conducted here correspond to the experiments with conditions for a Shields parameter of $\tau^* = 0.048$ (clear-water scour) and a Shields

Table 2
Parameters used for mesh adaptivity.

Adaptivity Method	Options summary
h-adaptivity	The maximum and minimum edge lengths were specified with values of 0.04 *m* (corresponding to the initial element characteristic length) and 1×10^{-3} *m* respectively. The number of timesteps between optimisation of the mesh was chosen to be 15 as it was found (through experimentation) as the best compromise between cost and robustness. The weights for the error metric construction (e_{f_i} in section 4), chosen to provide good results in terms of overall mesh quality were: $\epsilon_{\mathbf{u1}} = 0.035\ ms^{-1}$, $\epsilon_{\mathbf{u2}} = 0.0035\ ms^{-1}$ for experiment $\tau^* = 0.048$ and $\epsilon_{\mathbf{u1}} = 0.05\ ms^{-1}$, $\epsilon_{\mathbf{u2}} = 0.005\ ms^{-1}$ for experiment $\tau^* = 0.098$ where $u1$ and $u2$ represent the velocity components.
r-adaptivity	A lineal-torsional-spring method was used driven entirely by bed deformation.

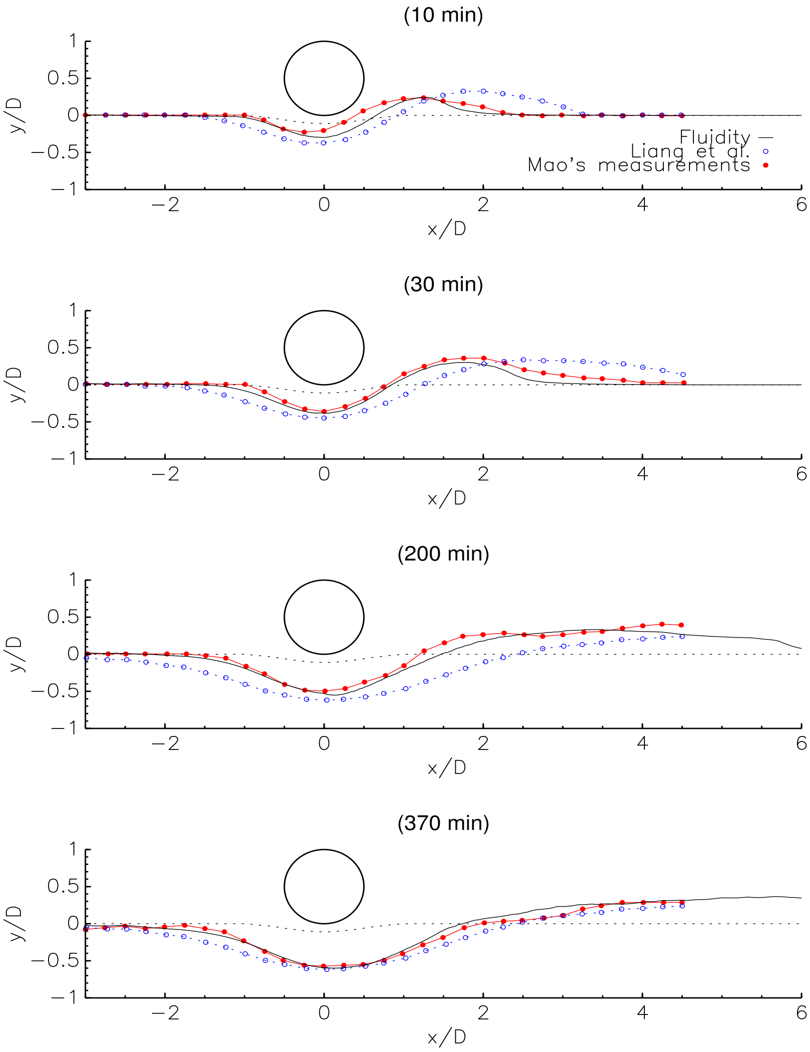


Fig. 2. Bed evolution at different times for the Fluidity model using experimental data from case $\tau^* = 0.048$. Also shown are the corresponding experimental data from Mao [45] and numerical results from Liang et al. [2]. (*a*) Profile I at $t = 10$ *min*, (*b*) Profile II at $t = 30$ *min*, (*c*) Profile III at $t = 200$ *min*, (*d*) Profile IV at $t = 370$ *min*. The dashed line shows the initial bed profile.

parameter of $\tau^* = 0.098$ (live-bed scour). With regards the numerical parameters for mesh adaptivity, the simulations were performed with an unstructured mesh of triangular elements with the parameters as given in Table 2. All computations were conducted in parallel using 16 processor cores.

Based upon laboratory experiments, and corroborated by consistent numerical results, during the early stages of scour the erosion rate is very high (due to the tunnel erosion), so that the scour depth reached ∼ 4 *cm* after 1.5 minutes for the $\tau^* = 0.098$ case and approximately 2.5 *cm* after 10 minutes for the $\tau^* = 0.048$ case. A higher Shields parameter means that the scour hole reaches an equilibrium state faster. The equilibrium stage for the $\tau^* = 0.098$ case began to develop at approximately 30 minutes of simulation time, with a maximum scour depth of ∼ 8 *cm*. Figs. 2 and 3 illustrate experimental

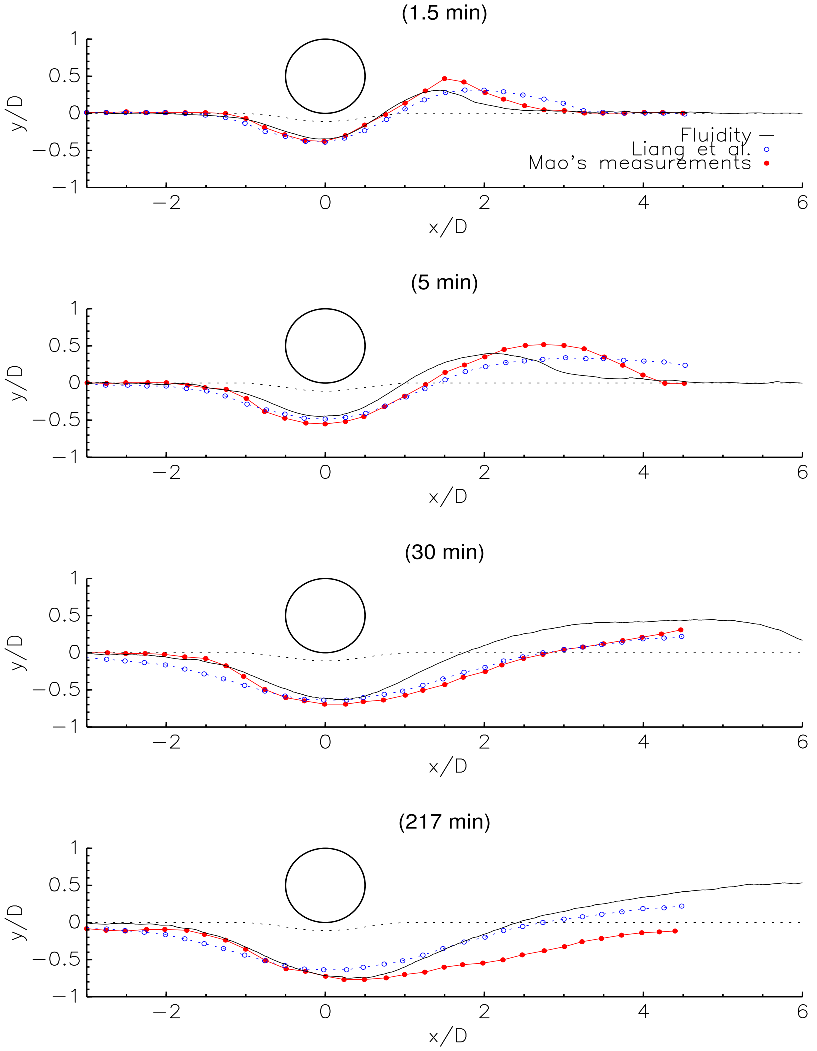


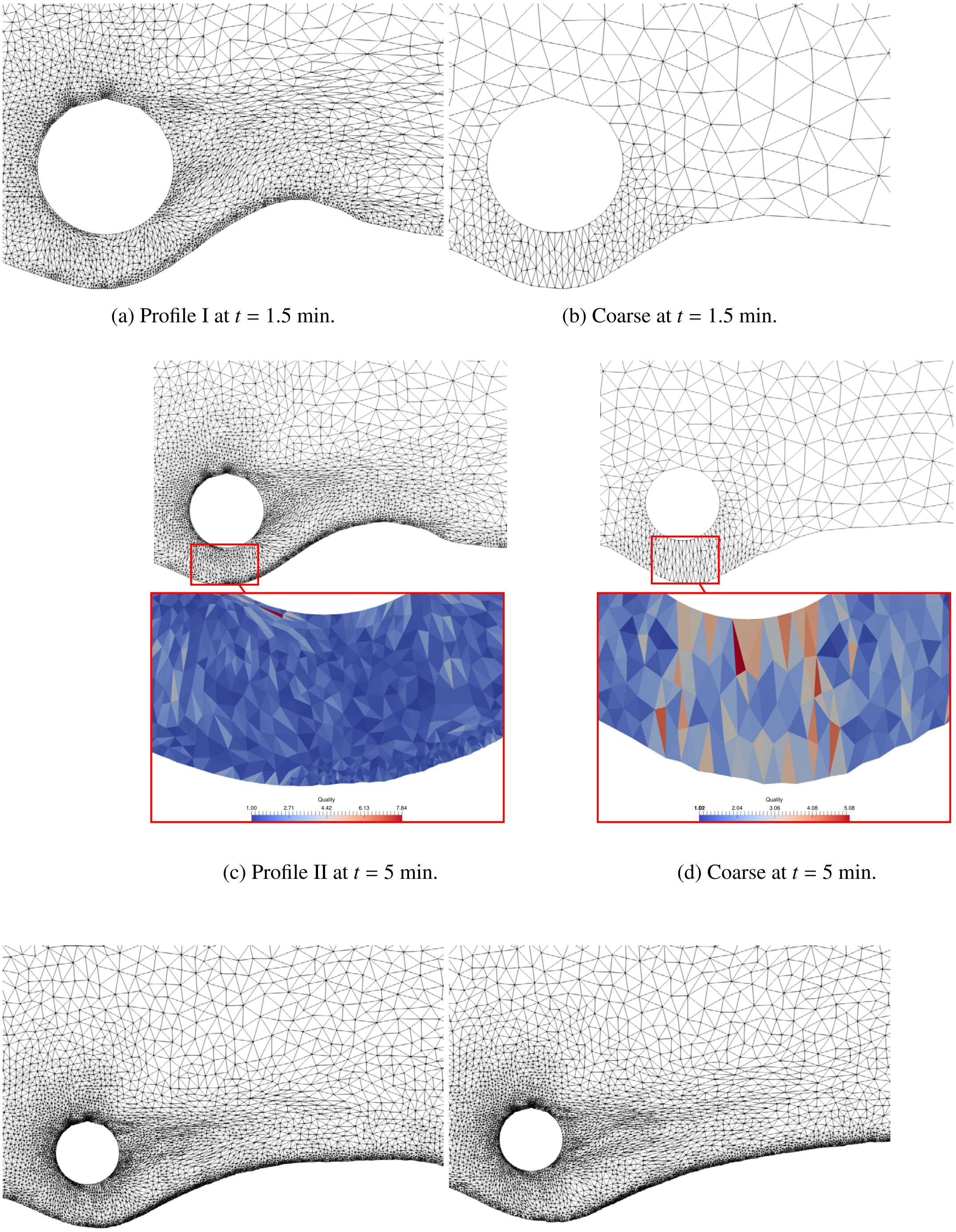
Fig. 3. Bed evolution at different times for the Fluidity model using experimental data from case $\tau^* = 0.098$. Also shown are the corresponding experimental data from Mao [45] and numerical results from Liang et al. [2]. (a) Profile I at $t = 1.5$ min, (b) Profile II at $t = 5$ min, (c) Profile III at $t = 30$ min, (d) Profile IV at $t = 217$ min. The dashed line shows the initial bed profile.

bed profiles around the pipe at different times, along with the numerical results of Liang et al. [2] (for their medium mesh k–ϵ based results which they found to represent a good compromise between accuracy and efficiency).

For the Fluidity results in Fig. 2, which corresponds to the case with $\tau^* = 0.048$, it is observed that at the very early stages of scour (frame (a) – profile I), the scour depth is slightly deeper than that observed experimentally. This could be a result of the assumptions made with regards to the initial bed profile employed to emulate the tunnel erosion stage (and also noting that the Liang et al. [2] results this set-up is based upon show something similar, yet more extreme). The seepage flow phase early in the experimental setup would have led to a slight mismatch between the timings of the experimental and numerical results, with the experimental profiles potentially lagging the numerical ones which are effectively initiated at the end of the seepage phase. It should be noted that this could also be a result on an under-prediction of sediment transport rates in our model.

In frame (b) – profile II, after the tunnel erosion has created a scour hole underneath the pipe, the scour rate decreases as the gap grows. This subsequent stage is referred to as lee-wake erosion, where vortex shedding from the pipe controls the scour. Experimentally, it was observed that during this phase more scour will occur on the upstream side. Our model is able to predict a faster upstream erosion and also a characteristic bedform/dune formation due to accretion of sediment in the downstream section.

By frame (c) – profile III, the scour hole has deepened and the flow below the pipe has slowed, and the vortices have weakened and finally vanished. A consequence of this is that the mean velocity in the gap and the bed shear stress reach a quasi-equilibrium and remain approximately constant; linked to this, no major changes in the bed profiles are observed apart from a slow downstream migration of the dune. A good match is observed in the scour depth with respect to the



(a) Profile I at t = 1.5 min.

(b) Coarse at t = 1.5 min.

(c) Profile II at t = 5 min.

(d) Coarse at t = 5 min.

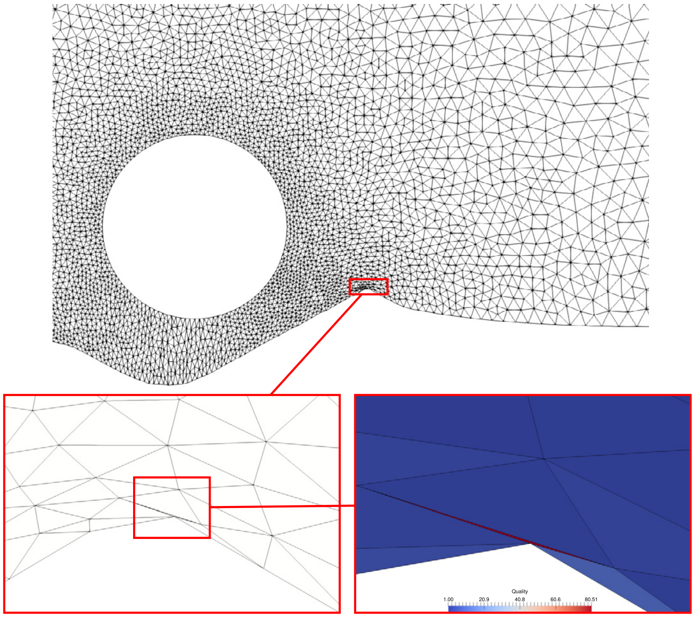
(e) Profile III at t = 30 min.

(f) Profile IV at t = 217 min.

Fig. 4. Mesh visualisations of the adaptive mesh simulation (profiles I–IV), including the mesh quality results of the fixed coarse mesh simulation for the $\tau^* = 0.098$ case. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

other numerical model and the experimental result, and this agreement is maintained until the end of the simulation. Frame (*d*) – profile IV shows both models agreeing very well with the experimental result both in terms of maximum scour depth below the pipe as well as the good downstream profile agreement.

For results with $\tau^* = 0.098$, shown in Fig. 3, it can be seen that most of the scour occurs rapidly in the earliest times of the experiment. Our model results show fairly good agreement with the experimental results throughout the initial



(a) Fine at $t = 1.5$ min.

Fig. 5. Mesh visualisations and mesh quality diagnostic for a mesh movement only simulation at $t = 1.5$ *min* for the $\tau^* = 0.098$ case demonstrating element collapse and near inversion.

tunnel erosion phase and the subsequent lee-wake erosion phase, noted by a good profile match in frames (a) and (b). However, the downstream accretion seems to be over-predicted at the later simulation times. This can be a consequence of the formation of bedforms, which shield the flow and encourage sediment deposition. In the first three frames it is hard to argue which of the two numerical results agree more closely with the experiments. However, at the final time (frame (d)) the Fluidity result shows a better scour depth agreement than the other numerical result, while there are similarities in the two models' disagreement with the downstream experimental profile.

Fig. 4 shows the progression of the mesh as the bed profile evolves for the case with $\tau^* = 0.098$ (n.b. very similar images and discussion would be obtained for the $\tau^* = 0.048$ case) in which changes in the adapted mesh due to the optimisation process are shown (frames (a), (c), (e) and (f)). Also displayed are some frames ((b) and (d)) showing the evolution of a coarse mesh undergoing mesh movement alone (i.e. no mesh optimisation and so fixed degree of freedom count and element connectivity). Mesh quality is compared for the cases with and without mesh optimisation in frames (c) and (d). It can be observed that while the early time mesh in frame (b) looks reasonable, the elements have become poorly shaped and sized by the time of frame (d). In can also be observed that in the early time frame (b), the poor quality mesh (as coarser mesh resolution was initialised in the fixed mesh downstream of the pipe and this can not evolve as the simulation develops) is impacting on the numerical result in terms of the lack of the formation of appropriate downstream bedform profiles. The mesh quality measure plotted is the edge ratio (the ratio of the longest edge length to the shortest) and provides information similar to the element aspect ratio. As observed in frames (c) and (e) and (f), the use of a mesh optimisation approach results in elements that are generally more appropriately shaped and sized. Completely opposite to the mesh movement only result, individual elements are only stretched anisotropically in the direction of flow – as would be expected from the mesh optimisation algorithm and the fact that we form the metric based upon the velocity field – thus providing an optimal approach for the resolution of boundary/shear layers and fronts. In frame (c) the majority of elements in the gap are actually close to isotropic; here we are not seeing anisotropic elements within the bottom boundary layer as might be expected. This is a result of the fact that the bed is not flat, while we are using (isoparametric) elements with flat edges. This results in a constraint on the ability of the mesh optimisation operations to strictly achieve what the error metric is requesting of a fully optimised mesh, e.g. it is unable to remove nodes without modifying the representation of the boundary geometry.

Fig. 5 shows a zoomed in view of a squashed element from a mesh movement only simulation with a finer refinement than was shown above. A thin sliver element can be seen to have formed due to the extreme bed deformation and the simulation is very close to failing. Although the lineal-torsional smoothing method used here offers very robust performance for external moving boundaries, the addition of a mesh optimisation process clearly helps to avoid both the excessive stretching and inversion of the near boundary elements, which would otherwise compromise mesh quality and thus impact the accuracy and robustness of the solver.

Bed slope instabilities were satisfactorily controlled with the avalanche flux (sand-sliding-type) model that prevents the local boundary slopes significantly exceeding the angle of repose or maximum friction angle normally reported for loose sands.

6. Conclusions

This paper presents the implementation and validation of a control volume/finite element-based model utilising *hr*-adaptive mesh techniques for the simulation of scour around hydraulic structures in steady currents. The inclusion of *r*-adaptivity or moving mesh strategies, such as the lineal-torsional-spring smoothing technique employed here, permits the adjustment of the mesh in an efficient manner for simulations with evolving boundaries. To address several issues associated with poor mesh quality or inappropriate resolution at locations and times within a scouring simulation, an additional *h*-adaptivity or mesh optimisation step was successfully utilised. This work thus represents an important step forward in the ability to robustly and accurately simulate extreme scour in the presence of structures using grid based models. Future work will further demonstrate this approach for more complex and three-dimensional structures.

Amongst the open issues to be considered in the future development of this model is the choice of alternative turbulence modelling approaches, such as large eddy simulations (LES), which may be better suited to problems where knowledge of extreme and fluctuating currents, and hence bed shear stresses, may be important. A potentially critical aspect for the test case considered in this work is the interaction between the vortex shedding from the pipe and the bed, which could produce an inaccurate prediction of scour profiles at the early stage of the process. In addition, the treatment of the flow near boundaries is of extreme importance and thus the selection of appropriate wall functions or other approaches to boundary layer representation need to be considered in more detail.

Finally, future work is also expected to extend the bedload sediment transport formulae to include the gravitational influence of the critical bed shear stress, τ_c on large bed slopes. This effect on a sloping bottom will lead to an increase in bedload transport rate in the downslope direction, and a reduction in the upslope direction. A correction factor can be applied to both the magnitude and direction of the sediment transport rate to account for this.

Acknowledgements

The authors would like to kindly acknowledge the UK's Engineering and Physical Sciences Research Council (EPSRC) project EP/L000407/1 for funding which supported this work. Work on the implementation of mesh movement algorithms in Fluidity was funded under ARCHER eCSE project eCSE06-001, where we thank Tim McManus and Ben Yeager in particular for their support. We also acknowledge the support received from Imperial College London's Research Computing Service.

References

[1] B.M. Sumer, Coastal and offshore scour/erosion issues – recent advances, in: Proceedings Fourth International Conference on Scour and Erosion (ICSE2008), vol. K6, 2008, pp. 85–94.

[2] D. Liang, L. Cheng, F. Li, Numerical modeling of flow and scour below a pipeline in currents. Part II: scour simulation, J. Coast. Eng. 52 (2005) 43–62.

[3] L. Zhao, B. Guo, X. Bai, W. Zhang, T. Li, J.J.R. Williams, Finite element modelling of local scour below a pipeline under steady currents, Int. J. Comput. Fluid Dyn. 30 (2016) 1–6.

[4] B. Brørs, Numerical modeling of flow and scour at pipelines, J. Hydraul. Eng. 125 (5) (1999) 511–523.

[5] B.E. Larsen, D.R. Fuhrman, B.M. Sumer, Simulation of wave-plus-current scour beneath submarine pipelines, J. Waterw. Port Coast. Ocean Eng. 142 (5) (2016) 04016003.

[6] C.J. Budd, W. Huang, R.D. Russell, Adaptivity with moving grids, Acta Numer. 18 (2009) 111–241.

[7] X. Liu, M.H. Garcia, Three-dimensional numerical model with free water surface and mesh deformation for local sediment scour, J. Waterw. Port Coast. Ocean Eng. 134 (2008) 203–217.

[8] P.J. Frey, F. Alauzet, Anisotropic mesh adaptation for CFD computations, Comput. Methods Appl. Mech. Eng. 194 (2005) 5068–5082.

[9] M.D. Piggott, P.E. Farrell, C.R. Wilson, G.J. Gorman, C.C. Pain, Anisotropic mesh adaptivity for multi-scale ocean modelling, Philos. Trans. R. Soc. Lond. A, Math. Phys. Eng. Sci. 367 (2009) 4591–4611.

[10] M.D. Piggott, G.J. Gorman, C.C. Pain, P.A. Allison, A.S. Candy, B.T. Martin, M.R. Wells, A new computational framework for multi-scale ocean modelling based on adapting unstructured meshes, Int. J. Numer. Methods Fluids 56 (2008) 1003–1015.

[11] M.D. Piggott, C.C. Pain, G.J. Gorman, P.W. Power, A.J.H. Goddard, *h*, *r*, and *hr* adaptivity with applications in numerical ocean modelling, Ocean Model. 10 (2005) 95–113.

[12] J.M. Nunez Rattia, J.R. Percival, B. Yeager, S.J. Neetlhling, M.D. Piggott, Numerical simulation of scour below pipelines using flexible mesh methods, in: Proceedings Eight International Conference on Scour and Erosion (ICSE2016), 2016, pp. 101–108.

[13] D. Kuzmin, O. Mierka, On the implementation of the k-ϵ turbulence model in incompressible flow solvers based on a finite element discretization, Int. J. Comput. Sci. Math. 1 (2007) 193–206.

[14] C. Paola, V.R. Voller, A generalized Exner equation for sediment mass balance, J. Geophys. Res. 110 (2005) F04014.

[15] R. Soulsby, Dynamics of Marine Sands, Thomas, Telford, 1997.

[16] F. Engelund, J. Fredsøe, A sediment transport model for straight alluvial channels, Nordic Hydrol. 125 (5) (1976) 293–306.

[17] W. Graf, Fluvial Hydraulics: Flow and Transport Processes in Channels of Simple Geometry, McGraw-Hill, 1998.

[18] D.D. Apsley, P.K. Stansby, Bed-load sediment transport on large slopes: model formulation and implementation within a RANS solver, J. Hydraul. Eng. 10 (2008) 1440–1451.

[19] S.L. Niemann, J. Fredsøe, N.G. Jacobsen, Sand dunes in steady flow at low Froude numbers, J. Hydraul. Eng. 137 (2011) 5–14.

[20] J.F. Richardson, W.N. Zaki, Sedimentation and fluidisation. Part 1, Trans. Inst. Chem. Eng. 32 (1954) 35–53.

[21] J. Fredsøe, On the development of dunes in erodible channels, J. Fluid Mech. 64 (1974) 1–16.

[22] M. Garcia, G. Parker, Entrainment of bed sediment into suspension, J. Hydraul. Eng. 117 (1991) 414–435.

[23] L.C. van Rijn, Sediment transport, Part II: suspended load transport, J. Hydraul. Eng. 110 (1984) 1613–1641.

[24] Applied Modelling and Computational Group, Fluidity Manual Version 4.1.12., Department of Earth Science and Engineering, Imperial College London, 2015.

[25] C.J. Cotter, D.A. Ham, C.C. Pain, A mixed discontinuous/continuous finite element pair for shallow-water ocean modelling, Ocean Model. 26 (2009) 86–90.

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

[27] D. Kuzmin, A vertex-based hierarchical slope limiter for p-adaptive discontinuous Galerkin methods, J. Comput. Appl. Math. 233 (2010) 3077–3085.

[28] P.M. Gresho, S.T. Chan, Solving the incompressible Navier–Stokes equations using consistent mass and a pressure Poisson equation, in: ASME Symposium on Recent Development in CFD, Chicago 95, 1988, pp. 51–73.

[29] P.K. Sweby, High-resolution schemes using flux limiters for hyperbolic conservation-laws, SIAM J. Numer. Anal. 21 (1984) 995–1011.

[30] S.D. Parkinson, J. Hill, M.D. Piggott, P.A. Allison, Direct numerical simulations of particle-laden density currents with adaptive, discontinuous finite elements, J. Geosci. Model Dev. 7 (2014) 1945–1960.

[31] H.K. Johnson, J.A. Zyserman, Controlling spatial oscillations in bed level update schemes, J. Coast. Eng. 46 (2002) 109–126.

[32] M.J. Castro-Diaz, F. Hecht, B. Mohammadi, O. Pironneau, Anisotropic unstructured mesh adaption for flow simulations, Int. J. Numer. Methods Fluids 25 (1997) 475–491.

[33] G.C. Buscaglia, E.A. Dari, Anisotropic mesh optimization and its application in adaptivity, Int. J. Numer. Methods Eng. 40 (1997) 4119–4136.

[34] C. Pain, A. Umpleby, C. de Oliveira, A. Goddard, Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations, Comput. Methods Appl. Mech. Eng. 190 (2001) 3771–3796.

[35] A. Loseille, F. Alauzet, Optimal 3d highly anisotropic mesh adaptation based on the continuous mesh framework, in: B.W. Clark (Ed.), Proceedings of the 18th International Meshing Roundtable, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 575–594.

[36] A. Loseille, F. Alauzet, Continuous mesh framework part i: well-posed continuous interpolation error, SIAM J. Numer. Anal. 49 (2011) 38–60.

[37] H.R. Hiester, M.D. Piggott, P.E. Farrell, P.A. Allison, Assessment of spurious mixing in adaptive mesh simulations of the two-dimensional lock-exchange, Ocean Model. 73 (2014) 30–44.

[38] Y. Vassilevski, K. Lipnikov, An adaptive algorithm for quasioptimal mesh generation, Zh. Vychisl. Mat. Mat. Fiz. 39 (1999) 1468–1486.

[39] M.D. Piggott, C.C. Pain, G.J. Gorman, D.P. Marshall, P.D. Killworth, Unstructured adaptive meshes for ocean modeling, in: H. Hasumi, M.W. Hecht (Eds.), Ocean Modeling in an Eddying Regime, AGU, 2008, pp. 383–408.

[40] P.E. Farrell, M.D. Piggott, C.C. Pain, G.J. Gorman, C.R. Wilson, Conservative interpolation between unstructured meshes via supermesh construction, Comput. Methods Appl. Mech. Eng. 198 (2009) 2632–2642.

[41] P.E. Farrell, J.R. Maddison, Conservative interpolation between volume meshes by local Galerkin projection, Comput. Methods Appl. Mech. Eng. 200 (1–4) (2011) 89–100.

[42] J. Donea, A. Huerta, J.P. Ponthot, Arbitrary Lagrangian Eulerian methods, in: Encyclopedia of Computational Mechanics, Wiley & Sons, 2004.

[43] C. Farhat, C. Degand, B. Koobus, M. Lesoinne, Torsional springs for two-dimensional dynamic unstructured fluid meshes, Comput. Methods Appl. Mech. Eng. 163 (1998) 231–245.

[44] G.J. Gorman, C.C. Pain, M.D. Piggott, A.P. Umpleby, P.E. Farrell, J.R. Maddison, Interleaved parallel tetrahedral mesh optimisation and load-balancing, in: P. Bouillard, P. Diez (Eds.), Adaptive Modeling and Simulation (ADMOS), 2009, pp. 101–104.

[45] Y. Mao, The Interaction Between a Pipeline and an Erodible Bed, Ph.D. thesis, Technical University of Denmark, Lyngby, Denmark, 1986.

[46] B.M. Sumer, J. Fredsøe, The Mechanics of Scour in the Marine Environment, World Scientific, 2002.