Multi-scale hydro-morphodynamic modelling using mesh movement methods

Article

It is advisable to refer to the publisher's version if you intend to cite from the work. See [Guidance on citing](#).

www.reading.ac.uk/centaur

CentAUR

Central Archive at the University of Reading

Reading's research outputs online

ORIGINAL PAPER

Multi-scale hydro-morphodynamic modelling using mesh movement methods

Mariana C. A. Clare[1] (ID) · **Joseph G. Wallwork[1]** · **Stephan C. Kramer[1]** · **Hilary Weller[2]** · **Colin J. Cotter[1]** · **Matthew D. Piggott[1]**

Abstract

Hydro-morphodynamic modelling is an important tool that can be used in the protection of coastal zones. The models can be required to resolve spatial scales ranging from sub-metre to hundreds of kilometres and are computationally expensive. In this work, we apply mesh movement methods to a depth-averaged hydro-morphodynamic model for the first time, in order to tackle both these issues. Mesh movement methods are particularly well-suited to coastal problems as they allow the mesh to move in response to evolving flow and morphology structures. This new capability is demonstrated using test cases that exhibit complex evolving bathymetries and have moving wet-dry interfaces. In order to be able to simulate sediment transport in wet-dry domains, a new conservative discretisation approach has been developed as part of this work, as well as a sediment slide mechanism. For all test cases, we demonstrate how mesh movement methods can be used to reduce discretisation error and computational cost. We also show that the optimum parameter choices in the mesh movement monitor functions are fairly predictable based upon the physical characteristics of the test case, facilitating the use of mesh movement methods on further problems.

Keywords Mesh adaptation · Mesh movement · Sediment transport · Morphology

Mathematics Subject Classification 65M60 · 65M50 · 35Q35 · 86A05

1 Introduction

Over the last few decades, hydro-morphodynamic models have been increasingly used to simulate erosion in both fluvial and coastal zones (see Amoudry and Souza (2011),

✉ Mariana C. A. Clare
    m.clare17@imperial.ac.uk

1   Imperial College London, London, UK

2   University of Reading, Reading, UK

       Springer

Papanicolaou et al. (2008)). Unfortunately, these models are generally computationally expensive, especially as they are often required to simulate very long-term morphological effects with relatively small timestep sizes in order to resolve hydrodynamic features such as waves and tides. These models must also often be run more than once for calibration purposes, as for example in Villaret et al. (2016), Harris et al. (2018) and Clare et al. (2021b), thus further increasing the computational cost. In addition, when applied to coastal regions, hydro-morphodynamic models must resolve problems with complex and fundamentally multi-scale domains. Unstructured mesh models, such as those based on finite element methods (Piggott et al. 2008a, b), are ideally suited to providing the required mesh flexibility, and multiple tools exist for generating multi-scale fixed meshes, which are suitable for a wide range of geophysical applications (for example Avdis et al. (2018)). However, these are insufficient in cases with significant morphology changes as the areas that require finer mesh resolution vary throughout the simulation. Therefore, many fluvial and coastal scenarios are unfeasible to model using standard fixed meshes of appropriate resolution. A solution to this time-dependent multi-scale issue is to use mesh adaptation methods, such as mesh movement and h-refinement, which offer the potential to improve result accuracy and/or reduce computational cost. Any computational cost reduction means that there are more problems that are feasible to model and more calibration runs that can be performed for an equivalent cost.

The focus and main novelty of this work is that we apply mesh movement methods, also known as r-adaptation (Budd et al. 2009), to a hydro-morphodynamic model for the first time. Mesh movement methods work by fixing the mesh topology and dynamically moving the (fixed number of) mesh nodes, thus allowing different regions of the domain to vary between low and high resolution as flow structures pass through them. In most hydro-morphodynamic problems, including the ones considered here, regions warranting the highest mesh resolution move continuously with the flow and using mesh movement thus allows us to track these features of interest. Furthermore, the nature of mesh movement avoids the problem of hanging nodes and means that simple data structures can be used because the number of mesh nodes and their connectivity remains fixed during the simulation. This means that the mesh movement approach presented here in the context of one model can in principle be readily retrofitted to other existing hydro-morphodynamic models. We thus hypothesise that mesh movement will lower the computational cost of using hydro-morphodynamic models for the same or a better accuracy than that achieved using fixed uniform meshes, and that it can do so in a relatively easy to implement manner.

The mesh movement method used in this work is driven by monitor functions, which utilize a concept of 'mesh density' to distribute the mesh across the domain. Given that the bed is of primary interest when modelling erosion, we seek to construct a generalised mesh movement approach that is suitable for most hydro-morphodynamic problems. We make use of monitor functions that focus mesh resolution in regions where the bed is sloped, has high curvature and/or transitions from wet to dry. The relative importance of each of these components is discussed, but we conjecture that the scaling factor that optimises the error reduction will be reasonably predictable from the physical characteristics of the test case.
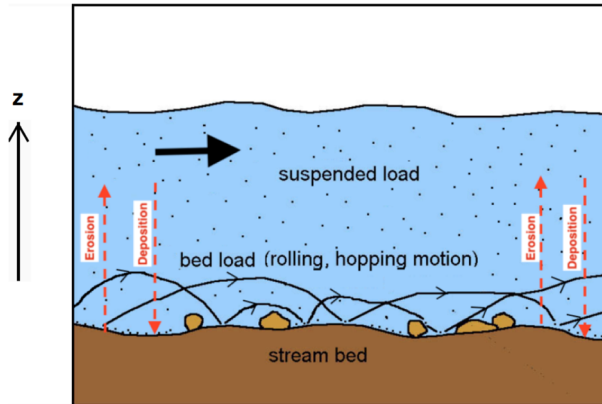
Fig. 1 Diagram of sediment transport processes. Adapted from Geology Cafe (2015)

For our hydro-morphodynamic model, we use the 2D depth-averaged coupled model of Clare et al. (2021b) to simulate both suspended sediment transport and bed-load transport for non-cohesive sediment with bed updates (see Fig. 1). This model is developed within the finite element coastal ocean modelling system *Thetis* (Kärnä et al. 2017), which is well-suited for the implementation of mesh movement methods, as discussed in Sect. 2. Other works have previously applied the alternative mesh adaptation technique, h-refinement, to hydro-morphodynamic models (for example Mayne et al. (2002), Delandmeter (2017), Benkhaldoun et al. (2013)). Note that for h-refinement the mesh topology is more substantially altered with cells locally created or destroyed and mesh connectivity altered (Piggott et al. 2006, 2009), resulting in the disadvantage that a much more complex data structure is required to implement it than that required for mesh movement. Furthermore, our model has key advantages and/or differences to the hydro-morphodynamic models for which h-refinement has been previously applied. In particular, Mayne et al. (2002) shows that h-refinement can accurately capture sharp differences in sediment concentration, but their model simulates cohesive sediments (*i.e.* clay and silt) rather than the non-cohesive sediments, which are present in many fluvial and coastal environments. Similarly, Delandmeter (2017) shows that h-refinement can accurately capture sharp gradients in hydrodynamics, but they do not simulate the bedlevel changes caused by sediment erosion and deposition meaning their results are inaccurate especially in cases, such as the ones we consider, where there is a lot of bed movement. Finally, Benkhaldoun et al. (2013) again shows that h-refinement can accurately capture sharp gradients in sediment concentrations, but their model uses finite volume methods, whereas our model uses a discontinuous Galerkin finite element discretisation, which has several advantages for hydro-morphodynamic problems as discussed in Clare et al. (2021b).

The applications of h-refinement to hydro-morphodynamic models discussed above focus on refining the mesh based on either the hydrodynamics or the sediment concentration. Instead, in this work, we refine the mesh based on the bedlevel as well as potentially other solution fields. Focussing the monitor function on the bedlevel is sensible because this is the variable of most interest in erosion problems; other

variables are generally only of interest in the way in which they pertain to the bed and its evolution. This means that complex characteristics of these other variables do not need to be resolved if they play no role in the bed's evolution. However, the construction of appropriate error metrics that take account of this takes us into the field of goal-oriented mesh adaptation (Becker and Rannacher 1996, 2001) which is beyond the scope of this work.

In order to test our novel hydro-morphodynamic model mesh-movement framework, in this work we consider multi-scale test cases with complex evolving bathymetries, which are difficult to capture accurately on coarse and/or fixed meshes that may incorrectly represent small structures. We also consider test cases with a moving wet-dry interface that can be difficult to capture on a fixed mesh, a frequent scenario in the modelling of coastal zones. Note that to simulate sediment transport in wet-dry domains, we have developed and implemented a new conservative discretisation approach to our model as part of this work, as well as a sediment slide mechanism. Hence, we are able to establish a generalised mesh movement methodology that can be applied to a variety of fluvial and coastal zone test cases.

The remainder of this paper is structured as follows: in Sect. 2 we outline the mesh movement methods used; in Sect. 3 we give a brief outline of the hydro-morphodynamic model including the new conservative discretisation approach; in Sect. 4 we apply our mesh movement methods to a series of test cases and finally in Sect. 5 we conclude this work.

2 Mesh movement

There are a variety of approaches that can be taken to apply mesh movement including (but not limited to): imposing a prescribed mesh velocity (Donea et al. 2017); re-interpreting the mesh as a structure of stiff beams (Farhat et al. 1998); and enforcing mesh transformations using monitor functions (Huang et al. 1994; Budd and Williams 2009). In this work we consider the latter.

In the monitor function based approach, the 'physical' mesh—upon which the prognostic equations are solved—is moved during the time period of simulation, by defining it to be the image of some mapping applied to a *fixed* reference mesh, which determines the way in which the mesh is moved. The mesh's 'density' is prescribed by a user-provided monitor function; a concept first proposed in White (1979). In that work, in the context of a one dimensional PDE, a monitor function based on the arc length of the PDE solution was used. The monitor-based framework enforces the even distribution of the monitor function across mesh elements, meaning that regions with high values have elements with small areas and vice versa. This is typically achieved through the solution of an auxiliary PDE, which describes the mesh movement (such as Huang et al. (1994), Budd and Williams (2009)). As may be expected, the choice of monitor function greatly impacts the geometry of the moved mesh and should therefore be chosen with care.

Because our hydro-morphodynamic model is developed within the coastal ocean model Thetis and built using the automated code generation PDE solver framework, Firedrake, (Rathgeber et al. 2016), we can utilise Firedrake to implement this monitor

function based approach. The Firedrake framework is well suited to this because it can readily solve the complex non-linear PDEs used in this method (see McManus et al. (2017)). Moreover, such an approach has already been implemented in the wider Firedrake framework in McManus et al. (2017), McRae et al. (2018), Wallwork et al. (2020), although note compared to these works, our work has the added complexity of implementing mesh movement with a coupled model.

Within the Firedrake framework, we could combine the hydro-morphodynamic model and mesh movement equation into a single system of equations solved at each timestep in an *Arbitrary Lagrangian-Eulerian (ALE)* type approach (Piggott et al. 2006; Budd and Williams 2009; Donea et al. 2017). However, for a hydro-morphodynamic problem, this approach is unnecessarily computationally expensive because it requires the mesh to be moved at every timestep which may be superfluous for many problems (see Fig. 6, for example). Therefore, we take an alternative approach and only move the mesh after a user-defined number of timesteps. A consequence of this choice is that we can no longer assume we are solving our problem in a moving reference frame, and thus the solution variables must be interpolated between meshes during the PDE time integration loop. Here this interpolation is implemented using *libsupermesh* (Maddison et al. 2017) and performed using a conservative Galerkin projection based approach. Although the simple act of interpolation can introduce additional errors, our conservative Galerkin approach ensures that quantities are conserved in the mesh-to-mesh transfer and it is generally more accurate than the alternative simpler interpolation methods available in *Firedrake*.

2.1 Equidistribution and the Monge-Ampère equation

The approach to mesh movement used in this work is concerned with obtaining a discrete representation of a sufficiently smooth map,

$$\mathbf{x} : \Omega_C \times [0, T] \to \Omega_P, \qquad \mathbf{x}(\mathcal{H}_C, t) = \mathcal{H}_P(t), \qquad t \in [0, T]. \qquad (1)$$

Here $\Omega_C \subset \mathbb{R}^n$ is a *computational* reference domain, which remains fixed and $\mathcal{H}_C$ is an associated computational mesh. The *physical* domain is denoted by $\Omega_P : [0, T] \to \mathbb{R}^n$, is allowed to vary with time and has an associated physical mesh, $\mathcal{H}_P$, upon which our hydro-morphodynamic model is solved. As with the domain, $\mathcal{H}_P = \mathcal{H}_P(t)$ is allowed to vary with time, whereas $\mathcal{H}_C$ remains fixed.

The map (1) has certain constraints imposed on it. In particular, the user-specified *monitor function*, $m : \Omega_P \times [0, T] \to (0, \infty)$ must be *equidistributed*, in the sense that

$$m \det \underline{\mathbf{J}} = \theta, \qquad \text{where} \qquad \theta(t) = \frac{\int_{\Omega_P} m(\mathbf{x}, t) \, dx}{\int_{\Omega_C} d\xi} \quad \text{and} \quad \underline{\mathbf{J}} = \frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \qquad (2)$$

is the Jacobian transform with respect to the computational coordinate $\boldsymbol{\xi} \in \Omega_C$. The normalisation coefficient $\theta : [0, T] \to (0, \infty)$ acts to conserve the domain volume. In this work, we do not consider boundary deformations and hence will always assume that $\Omega_P = \Omega_C =: \Omega$, meaning θ depends only on the monitor function. The mesh

movement problem can thus be stated as follows: *for a given m, find a map,* **x**, *which satisfies (2)*.

Assuming that the equidistribution relation (2) is solved to a sufficiently high accuracy, the determinant term remains positive, because both m and θ are strictly positive. As such, the Jacobian cannot change sign. This prevents mesh tangling (where one or more mesh elements become inverted) because it means that no two vertices of the original mesh can meet, nor can two edges. It also means that elements cannot become degenerate, because their area must remain strictly positive.

In dimension $n > 1$ the mesh movement problem is ill-posed, meaning additional constraints must be imposed on the map (1). Our hydro-morphodynamic model described in Sect. 3 is in $n = 2$ spatial dimensions and therefore this is indeed the case in this work. Following the optimal transport approach advocated in McRae et al. (2018), Budd and Williams (2009), Budd et al. (2013), Weller et al. (2016), we assume that the map takes the form

$$\mathbf{x}(\boldsymbol{\xi}) = \boldsymbol{\xi} + \nabla\phi_{\boldsymbol{\xi}}(\boldsymbol{\xi}), \tag{3}$$

for some scalar potential $\phi : \Omega_C \times [0, T] \to \Omega_P$. Substituting (3) into (2) yields

$$m \, \det\left(\underline{\mathbf{I}} + \underline{\mathbf{H}}(\phi)\right) = \theta, \tag{4}$$

where $\underline{\mathbf{I}}$ is the identity matrix in $\mathbb{R}^{n \times n}$ and $\underline{\mathbf{H}}(\phi)$ denotes the Hessian of the potential with respect to the computational coordinates. Equation (4) is a nonlinear PDE of *Monge-Ampère* type.

Under an appropriate choice of monitor function, the elliptic PDE (4) has been shown to be capable of producing equidistributed meshes, which admit more accurate solutions of the underlying PDE (in this case the hydro-morphodynamic model) without increasing the number of degrees of freedom or modifying the mesh topology (for example see McRae et al. (2018), Budd and Williams (2009), Budd et al. (2013), Weller et al. (2016)).

A criticism of mesh movement methods that are driven by solutions of Monge-Ampère type equations is that the underpinning theory requires a degree of convexity of domain geometry (see Theorem 4.3 of Budd and Williams (2009)), which is not present in ocean domains bounded by coasts. However, for many realistic coastlines, the use of a wetting-and-drying scheme in the hydrodynamics solver such as the one present in our model (see Sect. 3) means that domain convexity can be ensured, because the domain boundary is no longer constrained to topographic contours and may be chosen as a convex superset of the 'wet' region. It is the convexity of the entire meshed domain (both 'wet' and 'dry' regions) that is important, not just the wet region. Hence, meshing all islands we avoid voids in the (both wet and dry) mesh; similarly for coastal regions.

An alternative mesh movement algorithm, such as an ALE type method in which the mesh moves continuously at every timestep, could be used to drive the wetting-and-drying itself, moving the coastal boundary as appropriate. Of course, the domain would not be convex in this case. Compared with such a method, the mesh movement method described in this work does not directly tackle the wetting-and-drying problem, but does allow for an interior refined region of the overall mesh to resolve and move with the

wet-dry interface. As such, it should be similarly capable to resolve moving coastlines, albeit with redundant nodes in the dry region, which exist to ensure convexity.

It should also be noted that convexity is a sufficient—but not necessary—condition for solvability.

2.2 Implementing mesh movement

As discussed above, in order to implement mesh movement based on (4), the user must choose a computational mesh $\mathscr{H}_C$, an initial physical mesh $\mathscr{H}_P$ with an identical topology (usually chosen to coincide with $\mathscr{H}_C$) and a monitor function m. The choice of monitor function is particularly important (see Sect. 4) because it determines the way in which mesh resolution is to be distributed over the domain.

The mesh movement itself is achieved in an iterative fashion. Each iteration involves the numerical solution of (4) for the scalar potential on the current physical mesh, followed by the transformation of this mesh according to (3). Equation (4) is an elliptic PDE, with two sources of nonlinearity: one from the determinant and another from the product with the monitor function, via the dependence of the physical coordinates on the computational coordinates. Thus, we follow Budd and Williams (2009) and McRae et al. (2018) and parabolise the Monge-Ampère equation

$$-\frac{\partial}{\partial \tau} \Delta \phi = m \, \det \left(\underline{\mathbf{I}} + \underline{\mathbf{H}}(\phi) \right) - \theta, \tag{5}$$

where τ denotes 'pseudotime'. As ϕ approaches a steady state, the derivative term on the left-hand side converges to zero, whereby the solution of (5) tends to the solution of the elliptic form (4). It is proved in Awanou (2015) that the sequence of solutions given by solving (5) converges to the solution of (4), provided the 'pseudotimestep' $\Delta \tau > 0$ is sufficiently small and the initial guess for ϕ is sufficiently close to the solution. A pseudotimestep of $\Delta \tau = 0.1$ is found to be sufficient in this work.

Note, the problems considered in McRae et al. (2018) have periodic boundary conditions, whereas in this work, we consider non-periodic domains. We can distinguish between outer open boundaries and land boundaries, which can always be defined sufficiently far inland that they are always dry. The outer wet boundaries could be defined as a parametric curve, with mesh movement constrained to this curve, although this does not fully resolve the conservation issue. Dry cells do not contribute significantly to mass conservation, so the associated errors are not of large concern. However, we can prescribe zero motion on the outer boundary of the dry cells as no 'wet' dynamics ever reach these locations.

In the preliminary investigations presented here, we use rectangular domains and constrain the movement of boundary nodes to be tangential to the corresponding segment. The extension of our mesh movement algorithm to allow for non-axes-aligned geometry and temporal boundary deformations will be the subject of future work.

To solve (5), still following McRae et al. (2018), we use a mixed finite element method. Let $\Psi \subset H^2(\Omega)$ and $\Sigma \subset L^2(\Omega)^{2\times 2}$ denote the function spaces in which the potential and its Hessian at a given timestep reside. At pseudotimestep k, the

forward Euler scheme is applied to obtain $\phi_{k+1} \in \Psi$ as follows:

$$\langle \nabla \psi, \nabla \phi_{k+1} \rangle_{L^2(\Omega)} = \langle \nabla \psi, \nabla \phi_k \rangle_{L^2(\Omega)} + \Delta\tau \langle \psi, m \, \det(\underline{\mathbf{I}} + \underline{\mathbf{H}}_k) - \theta \rangle_{L^2(\Omega)}, \quad \forall \psi \in \Psi. \tag{6}$$

Its Hessian, $\underline{\mathbf{H}}_{k+1}$, is then recovered via an L^2 projection (Lakkis and Pryer 2013):

$$\langle \underline{\boldsymbol{\sigma}}, \underline{\mathbf{H}}_{k+1} \rangle_{L^2(\Omega)} = -\langle \nabla \cdot \underline{\boldsymbol{\sigma}}, \nabla \phi_{k+1} \rangle_{L^2(\Omega)} + \langle \underline{\boldsymbol{\sigma}} \cdot \widehat{\mathbf{n}}, \nabla \phi_{k+1} \rangle_{L^2(\partial\Omega)}, \quad \forall \underline{\boldsymbol{\sigma}} \in \Sigma. \tag{7}$$

In practice (6) and (7) are solved using finite-dimensional subspaces, $\Psi_h \subset \Psi$ and $\Sigma_h \subset \Sigma$. In this work, we choose both spaces to be piecewise linear and continuous ($\mathbb{P}1$). The Hessian is initialised to zero for the first iteration (assuming that the initial physical mesh coincides with the computational mesh).

Equations (6)–(7) are solved to equilibrium, subject to a relative tolerance tol on the residual of (6) as a stopping criterion. The gradient of the scalar potential is then obtained by L^2 projection, similarly to (7), but for the vector, rather than tensor, case. With the recovered gradient, the mesh coordinates are updated according to (3).

Solving (6)–(7) repeatedly can be computationally expensive, because each solve involves establishing a new map from Ω_C to Ω_P. However, in this work, we employ several strategies to reduce this cost. Firstly, we use the final value of ϕ computed during one iteration as the initial guess for the next. This is an appropriate choice provided that the hydro-morphodynamics have not changed too significantly between the iterations. In Sect. 4, we see that due to the slow nature of morphodynamic changes, this is often the case. These slow changes mean it is be excessive to apply mesh movement at every timestep, especially if the modifications to mesh coordinates are only minor and thus the second way we reduce cost is by choosing an appropriate mesh movement frequency. This can be inferred either by small sensitivity studies or by using test case knowledge to infer the approximate rate at which the bed is moving. Finally, whilst the exact equidistribution provided by (4) is mathematically attractive, it is often unnecessary in practice. Figure 2 shows an example of the trade-off between discretisation error and computational cost with respect to the relative solver tolerance tol used to solve the Monge-Ampère equation. It shows that tolerance values below $\mathcal{O}(10^{-3})$ result in large increases in computational cost for almost no accuracy benefits. The case in this figure is for the migrating trench test case from Sect. 4.1 with a mesh size of 10 mesh elements in the x-direction. Similar patterns were observed for other numbers of mesh elements and thus throughout this work we use a relative solver tolerance of tol $= 10^{-3}$ to solve the parabolised form (5).

3 Hydro-morphodynamic model

Before applying our mesh-movement hydro-morphodynamic framework to some fluvial and coastal zone test cases, we must describe and develop the hydro-morphodynamic model. In this work, we use the hydro-morphodynamic model derived in Clare et al. (2021b). This model is able to simulate both suspended sediment and bedload transport taking into account gravitational and helical flow effects. To maximise stability, the time derivatives of our model equations are approximated using a
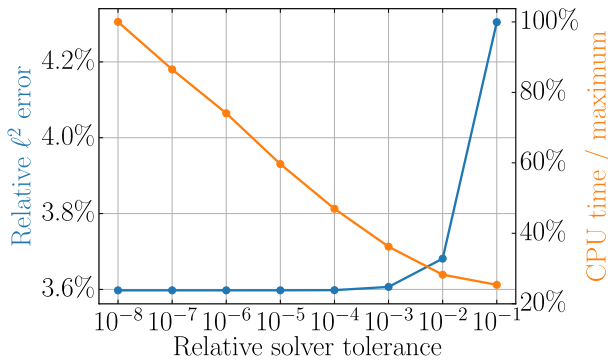
Fig. 2 An example of the trade-off between discretisation error and computational cost with relative solver tolerance for the migrating trench test case. The monitor function (23) has been applied to a mesh with 10 elements in the x-direction in the case $\alpha = 2$, $\beta = 0$. Error is shown as a percentage of the fixed mesh case with 10 mesh elements, whilst CPU time is relative to the adaptive case with $\texttt{tol} = 10^{-8}$

fully-implicit backward Euler timestepping scheme. Full details of the model and its development are provided in Clare et al. (2021b).

3.1 Model equations for a wet-dry domain

So far this model has been used for test cases in which the whole domain is wet. However, in coastal problems, there is often a wet-dry interface that must be simulated. We do this by using the inbuilt wetting-and-drying process in Thetis, which follows the approach detailed in Kärnä et al. (2011): the total water depth, H, is modified using the expression

$$\tilde{H} := \eta - h + f(H), \tag{8}$$

where η is the elevation, h is the bed height, and

$$f(H) := \frac{1}{2}\left(\sqrt{H^2 + \delta^2} - H\right), \tag{9}$$

as shown in Fig. 3.

Here δ is the wetting-and-drying parameter, which Kärnä et al. (2011) recommend is set approximately equal to $d||\nabla h||$ where d is the typical length scale of the mesh. Thus, the hydrodynamic equations in our model become

$$\frac{\partial \tilde{H}}{\partial t} + \nabla \cdot (\tilde{H}\mathbf{U}) = 0, \tag{10}$$

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U} \cdot \nabla \mathbf{U} + \mathbf{g}\nabla \eta = \nu\nabla^2\mathbf{U} - \frac{C_h(\tilde{H})}{\tilde{H}}||\mathbf{U}||\mathbf{U}, \tag{11}$$

where $\mathbf{U}$ is the depth-averaged velocity, $\mathbf{g}$ is gravity, ν the viscosity parameter and C_h the bed friction. Following Funke et al. (2017), to avoid non-differentiable functions,
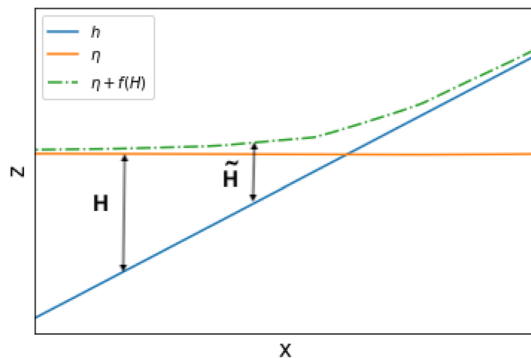
we smoothen the approximation to the norm operator in (11) by adding the numerical parameter ζ to the norm:

$$||\mathbf{U}|| \approx \sqrt{||\mathbf{U}||^2 + \zeta^2}. \tag{12}$$

The value of ζ must be chosen by the user such that it is large enough for the model not to crash due to non-differentiability issues, but not too large that the friction term spuriously affects the velocity. In Funke et al. (2017), ζ is set equal to the wetting-and-drying parameter, but the issue of non-differentiability exists independent of the wetting-and-drying formulation and so here we do not follow this choice. In the test case in Sect. 4.2, we experimented with values of ζ between 0.1 and 1 and found a value of 0.4 is appropriate.

To simulate the suspended sediment transport in the fluid, we use an advection-diffusion equation for the sediment concentration. The non-conservative form

$$\frac{\partial C}{\partial t} + \nabla \cdot (F_{\text{corr}}\mathbf{U}C) = \varepsilon_s \nabla^2 C + \frac{E_b - D_b}{H}, \tag{13}$$

is described in Clare et al. (2021b) where C is the depth-averaged sediment concentration, ε_s the diffusivity coefficient, E_b the erosion flux calculated using Van Rijn (1984), D_b the deposition flux calculated using the formula in Tassi and Villaret (2014), H the depth and F_{corr} a correction factor that accounts for the fact that depth-averaging the product of two variables is not equivalent to multiplying two depth-averaged variables. For full details of the formulae used to calculate E_b, D_b and F_{corr} see Clare et al. (2021b). However, the wetting-and-drying scheme used in this work is known to leak sediment. Thus, in order to ensure sediment is conserved when using this wetting-and-drying scheme, in this work we use the following conservative form,

$$\frac{\partial}{\partial t}(\tilde{H}C) + \nabla \cdot (F_{\text{corr}}\mathbf{U}\tilde{H}C) = \varepsilon_s \nabla^2(\tilde{H}C) + E_b - D_b, \tag{14}$$

where $\tilde{H}$ is the modified water depth given by (8). Instead of solving for C, we solve for the depth-integrated concentration, $\tilde{H}C$, which allows us to use the same finite element formulation that is used for the non-conservative sediment equation. To verify that our conservative scheme has improved the sediment conservation, in "Appendix A" we

consider the simple Thacker test case of oscillations in a paraboloid bowl (Thacker 1981) using both the non-conservative and conservative sediment equations. We find that the normalised mass error is $\mathcal{O}(10^{-2})$ for the non-conservative sediment Eq. (13) but $\mathcal{O}(10^{-12})$ for the conservative sediment Eq. (14), which is close to the numerical precision of the model. Thus, the conservative sediment equation conserves sediment much better than the non-conservative scheme and we can confidently use it to simulate sediment transport in test cases with a wet-dry domain. Note for both the non-conservative and conservative sediment equations, we set the incoming suspended sediment flow rate so that the erosion flux, E_b, equals the deposition flux, D_b, at the upstream boundary. This means that sediment equilibrium is established at the inlet and the bed is not artificially altered at the upstream boundary.

To complete our hydro-morphodynamic model, we use the Exner equation, which governs how the suspended sediment and bedload transport affect the bed. This is unaffected by wetting-and-drying and thus has the same form as that given in Clare et al. (2021b):

$$\frac{(1 - p')}{m_f} \frac{\partial z_b}{\partial t} + \nabla \cdot \mathbf{Q}_b = D_b - E_b, \tag{15}$$

where $\mathbf{Q}_b$ is the bedload transport flux, p' the porosity, z_b the bathymetry (also known as the bed profile) and m_f a morphological acceleration factor used to artificially increase the rate of bedlevel changes compared with the underlying hydrodynamics and thus save computational time when simulating long-term morphodynamic change (see Clare et al. (2021b) for more details). Throughout this work, we calculate $\mathbf{Q}_b$ using the Meyer-Peter-Müller formula

$$\mathbf{Q_b} = \phi_s \sqrt{g \left(\frac{\rho_s}{\rho_f} - 1 \right) d_{50}^3} \left(\cos \xi, \sin \xi \right), \tag{16}$$

where ϕ_s is the non-dimensional sediment rate, d_{50} the average sediment size, ρ_s the sediment density, ρ_f the water density and $\cos \xi = \frac{U_1}{\sqrt{U_1^2 + U_2^2}}$ and $\sin \xi = \frac{U_2}{\sqrt{U_1^2 + U_2^2}}$ for $\mathbf{U} = (U_1, U_2)$ (see Clare et al. (2021b) for more details). In practice, the magnitude and direction of $\mathbf{Q_b}$ depends on the gradient of the bed, but this is not reflected in (16). Thus, following standard hydro-morphodynamic model practice (see e.g. Tassi and Villaret (2014)) we also modify this formula using the slope effect corrections described in Clare et al. (2021b), which account for bedload transport at a slower rate if gravity is acting against it and vice versa.

If the bed of interest has complex steep slopes (like in Sect. 4.3) then hydro-morphodynamic models can generate physically unrealistic slopes. For these cases, we follow Apsley and Stansby (2008) and implement a 'sediment slide' mechanism (sometimes called an avalanche mechanism) to our model, which prevents the slope angle exceeding an angle of repose. The mechanism works by adding a component q_{aval} in the direction of the maximum slope, $\widehat{\mathbf{b}}$, to the Exner equation (15), which becomes

$$(1 - p') \frac{dz_b}{dt} + \nabla_h \cdot \left(\mathbf{Q_b} + q_{\mathrm{aval}} \widehat{\mathbf{b}} \right) = D_b - E_b. \tag{17}$$

As we are solving a 2D depth-averaged problem, we ignore the z-component of $\widehat{\mathbf{b}}$ and thus

$$\nabla_h \cdot (q_{\text{aval}}\widehat{\mathbf{b}}) = \nabla_h \cdot \left(-\frac{n_z^2 q_{\text{aval}}}{\sin \lambda}\nabla_h z_b\right) = \nabla_h \cdot (-\gamma \nabla_h z_b), \tag{18}$$

meaning the new Exner equation is

$$(1 - p')\frac{dz_b}{dt} + \nabla_h \cdot (\mathbf{Q_b} - \gamma \nabla_h z_b) = D_b - E_b. \tag{19}$$

Note in (18), λ is the slope angle, n_z is the z-component of the unit normal to the surface, defined by

$$\widehat{n}_z = \frac{1}{\sqrt{1 + |\nabla_h z_b|^2}}, \tag{20}$$

and q_{aval} is

$$q_{\text{aval}} = \begin{cases} 0.5(1 - p')\chi^2 \frac{(\tan \lambda - \tan \phi)}{\cos (\lambda \Delta t)}, & \tan \lambda > \tan \phi, \\ 0, & \text{otherwise,} \end{cases} \tag{21}$$

where ϕ is the angle of repose, p' the porosity and χ the length scale that controls how quickly the sediment is redistributed. In Apsley and Stansby (2008), it is argued that the value of χ does not need to be very precise, and thus following their work we set it equal to the approximate mesh step size if the mesh were uniformly distributed. This mechanism stops the slope angle exceeding ϕ, although it does not model how the sediment slides down the slope.

In summary, the wetting-and-drying hydro-morphodynamic model is given by Eqs. (10), (11), (14) and (15) (or instead (19) if using a sediment slide mechanism).

3.2 Finite element implementation

We solve the hydro-morphodynamic model using the finite element coastal ocean modelling system *Thetis*, which is built using the code-generating framework *Firedrake* (Kärnä et al. 2017). Throughout, we use the implicit backward Euler method as the time-stepping algorithm. Note that due to the complexity and nonlinearity of this combined system the equations are solved alternately, as opposed to as a monolithic system, which also means our model is not fully implicit. This does not adversely affect our results, because the timescale of the morphodynamic changes is much longer than the model timestep. The hydrodynamic and sediment concentration equations in our hydro-morphodynamic model are then solved using a discontinuous Galerkin finite element discretisation (DG) with piecewise linear basis functions. A full description of the finite element formulation for the hydrodynamic equations and the sediment concentration equations can be found in Vouriot et al. (2019) and Clare et al. (2021b) respectively. To solve the Exner equation (15), we use P1 elements with a continuous Galerkin finite element discretisation (CG) rather than DG. Full details of the implementation of the Exner equation (15) in CG and the weak form of the terms are given in Clare et al. (2021b). However, in this work, we also use (19) (a modified version of the Exner equation), which includes the sediment slide mechanism $-\nabla_h \cdot (\gamma \nabla_h z_b)$.

The corresponding term in the weak form of (19) is obtained by multiplying it by the test function, ψ and integrating by parts once to give the following

$$\int_{\Omega} -\nabla_h \cdot (\gamma \nabla_h z_b^{n+1}) = \int_{\Omega} \gamma (\nabla_h \psi) \cdot (\nabla_h z_b^{n+1}) \, dx, \qquad (22)$$

which is added to the weak form of the Exner equation derived in Clare et al. (2021b). Here $n + 1$ indicates the $(n + 1)^{\text{st}}$ timestep (because we solve the Exner equation implicitly) and there is no boundary integral because we assume Neumann conditions of no flux at the domain boundary and using CG means the centered flux values on either side of each interior edge cancel each other out over the whole domain. Note, by using this weak form, we are no longer considering second order derivatives here.

We have thus constructed a full mesh-movement hydro-morphodynamic model framework, where the mesh on which the hydro-morphodynamic model is solved is moved from timestep to timestep using the methods described in Sect. 2. This framework is not currently parallelised, but *Thetis* itself is already parallelised. Note further that mesh movement is not constrained to occur at *every* timestep, meaning we can choose how frequently the mesh is moved. The framework may now be applied to a series of test cases with complex bathymetries and/or wet-dry interfaces.

4 Test cases

4.1 Complex bathymetry test case: migrating trench

As a first test case, we consider the complex bathymetry case of a migrating trench, based on an experiment in Van Rijn (1980). This test case has already been used in Clare et al. (2021b) for validation of our hydro-morphodynamic model configuration in the case of a fixed uniform mesh.

Here, we use the same setup and parameter values as those described for this test case in Clare et al. (2021b). In particular, this means we use the fully wet version of the hydro-morphodynamic model with no sediment slide mechanism to model both suspended and bedload transport with magnitude and angle corrections from the slope effect, and use the Nikuradse friction formula to simulate friction. Additionally, we use a morphological acceleration factor, m_f, of 100 to aid computational efficiency. The only difference between the set-up here and that in Clare et al. (2021b) is the value of the diffusivity coefficient ε_s. Clare et al. (2021b) shows that this test case is sensitive to ε_s and altering the mesh resolution will change the effective numerical diffusivity of the model, which makes assessing performance against experimental data challenging, due in part to the issue of "getting the right answer for the wrong reasons". Thus, we calibrate a value for diffusivity from the experimental data using a uniform fixed mesh with $\Delta x = 0.05$ m (320 mesh elements in the x-direction) in the model. We use a gradient-based optimisation routine to perform optimum parameter estimation and compute the gradients using the adjoint method. More detail on the use of the adjoint framework with the hydro-morphodynamic model can be found in Clare et al. (2021a). We find that the optimum value of the diffusivity coefficient is

$\varepsilon_s = 0.18011$ to five significant figures. Throughout this section, this value is used as the diffusivity parameter.

4.1.1 Fixed uniform meshes

Before applying mesh movement methods, we study how the error varies on a series of fixed meshes varying from 320 mesh elements in the x-direction down to only 8 elements in the x-direction. Throughout this work, the error in the bed profile is calculated at the end of the simulation. The existence of experimental data for this test case means that we can calculate the total error, which is equal to the model error (the error due to using a simplified model to approximate a real world problem), plus discretisation error (the error arising from using a finite mesh to solve the model equations). In this section, the error is calculated using a pointwise ℓ_2 error norm at the location of the data points in the experiment.

Figure 4a shows the total error between the final bed profile from the experimental data and from the model output for a series of fixed uniform meshes. In this figure, the points marked with a star are the results where the number of mesh elements in the x-direction = [10, 20, 40, 80] and the points marked with a square are those where the number of mesh elements in the x-direction = 8. This distinction has been made because for both the star and square points the initial profile of the trench is incorrectly defined on meshes but in a different way. For example, instead of the second slope starting at the correct location of $x = 9.5$ m, for the star points it starts at $x = 9.6$ m, whereas for the square point it starts at 10 m. These differences affect the results on the fixed uniform mesh. For the remaining circle points (number of mesh elements in the x-direction = [32, 64, 160, 320]), the initial profile is correctly defined. This difference in the initial trench profile definition results in fluctuations in the error trend. It is surprising that with 10 elements, the total error is lower than that obtained on finer meshes. This is likely due to this test case's sensitivity to diffusivity and the fact that the effective diffusivity is different at the coarser mesh resolutions. Despite these fluctuations, the general trend is that the value of the error norm begins to plateau around 20 mesh elements and by 32 mesh elements the error has clearly converged to a value of approximately 0.0244 m. This is because, once the initial trench profile is accurately defined on the mesh, as noted in Clare et al. (2021b), the model is then quite insensitive to changes in Δx and Δt and thus the total error in the solution is dominated by the model error rather than the discretisation error from the mesh.

To better understand the discretisation error, we calculate the error norm between the final bed profile for different mesh resolutions compared to the final bed profile obtained with the finest fixed uniform mesh with 320 mesh elements in the x-direction ($\Delta x = 0.05$ m). Figure 4b shows that as the mesh becomes finer the discretisation error decreases with an order of approximately $(\Delta x)^2$. Note, we again separate out visually the number of mesh elements in x-direction = [10, 20, 40, 80] marked with a star, the number of mesh elements in x-direction = 8 marked with a square and the remaining marked with a circle. Like with the total error the sets converge at slightly different rates. Comparing Fig. 4a and 4b shows that, when there are 20 mesh elements or fewer, the total error is dominated by the discretisation error, but after this point the total error begins to be dominated by the model error. This is because above 20
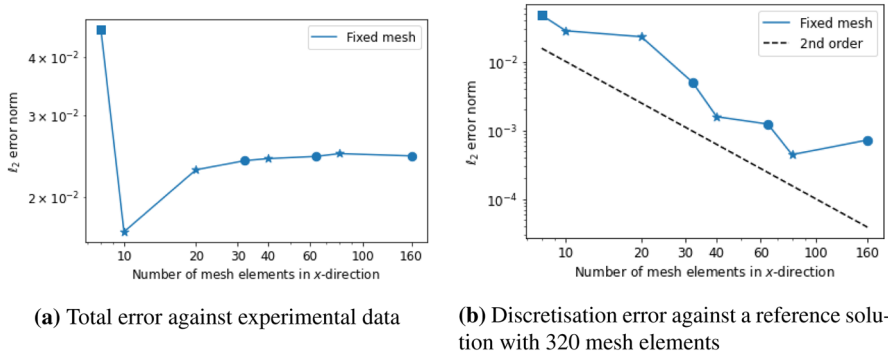
(a) Total error against experimental data

(b) Discretisation error against a reference solution with 320 mesh elements

Fig. 4 Errors in the final bed profile for a series of fixed uniform meshes for the migrating trench test case. The marker shapes distinguish differences in the definition of the initial profile on the mesh: for the circle points the initial trench profile is accurately defined on the mesh; for the star and square points the initial trench profile is ill-defined but in different ways

mesh elements the initial trench profile is either accurately defined or close to being accurately defined even on a fixed uniform mesh, and so the discretisation error is small.

4.1.2 Mesh movement

In order to apply mesh movement to the test case, we must choose an appropriate monitor function. From studying the fixed uniform meshes, we have determined that the main source of discretisation error in this test case is due to the fact that the initial trench profile is not accurately defined on the mesh. Even if a mesh is chosen so the profile is initially accurately defined, as soon as the simulation starts, the bed begins to move so the profile may quickly become ill-defined (hence the differences in total error even for the circle points in Fig. 4a). Furthermore, there are large sections of the trench profile that are flat and are relatively unchanged during the simulation, especially at the inflow. Therefore, a good choice of monitor function for this test case is one that results in increased mesh resolution in regions where the bed gradient and/or curvature is high and reduced mesh resolution where the bed gradient and/or curvature is lower. In this work, we choose the following monitor function because it allows us to control the effect of the first and second order bathymetry derivatives on the mesh movement

$$m(x, y) = 1 + \max_{x,y} \left(\alpha \frac{\left\| \underline{\boldsymbol{H}}(z_b) \right\|_F}{\max\limits_{x,y} \left\| \underline{\boldsymbol{H}}(z_b) \right\|_F}, \ \beta \frac{\left\| \nabla z_b \right\|_2}{\max\limits_{x,y} \left\| \nabla z_b \right\|_2} \right), \tag{23}$$

where $\underline{\boldsymbol{H}}(z_b)$ represents the Hessian of the bathymetry and $||\cdot||_F$ the Frobenius norm. Because we want to capture all the areas where the bed gradient or the curvature is high, our monitor function depends on the maximum of the first and second order derivatives, meaning that a region where only one of the derivatives is high is not given a lower weighting than a region where both derivatives are high. This monitor function
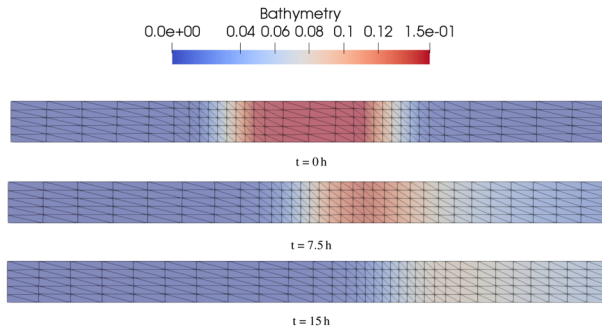
Fig. 5 Snapshots of bathymetry and the underlying mesh for a moving mesh simulation of the migrating trench test case. The monitor function (23) has been applied to a mesh with 32 elements in the x-direction in the case $\alpha = \beta = 3$. Results are shown at three points in time demonstrating mesh movement to capture bed evolution to the right

introduces two user-defined parameters, which control the effect of the underlying bathymetry on mesh movement; α controls the effect of the second order derivative (curvature), whilst β controls the effect of the first-order derivatives (slope). In order to be able to compare these parameters across test cases, we choose to normalise the derivatives in the monitor function. Note that both the Frobenius norm and the 2-norm are taken on an element-by-element basis and then projected into the $\mathbb{P}1$ space rather than being calculated component-wise. This is because we found that computing the norms component-wise results in an insufficiently smooth monitor function, leading to model divergence. Using an element-wise formulation introduces additional numerical diffusion that helps to counteract this.

Figure 5 shows an example of how the mesh is moved using (23) for this particular test case. Note the mesh hardly moves in the y-direction because the bathymetry is uniform in this direction.

With the mesh movement method used in this work, we can choose how frequently the mesh is moved. If the modification to the mesh coordinates proposed by (4) is only minor then solving this equation at every timestep of our hydro-morphodynamic model may prove an unnecessary expense. Because of the slow moving nature of this test case, we can infer that we can set a low mesh movement frequency without having an adverse effect on the results. More precisely, Fig. 6 shows the effect of changing the frequency of the mesh movement. The more timesteps between each mesh movement, the greater the discretisation error, but the lower the computational cost. This is to be expected as mesh movement methods have a non-negligible computational cost. Note that the total number of timesteps in this simulation is 2160 and therefore in the lowest frequency case the mesh is only moved once during the simulation (at the initial time). Significantly, the figure shows that if a large enough number of timesteps per mesh movement is chosen, we can reduce the computational cost below that of using a fixed mesh and still be more accurate. This is because although there is a cost associated with using the mesh movement algorithm, the mesh movement simulations require fewer model iterations to converge and thus when the mesh movement frequency is low, it is possible to achieve a lower overall computational cost when using moving meshes
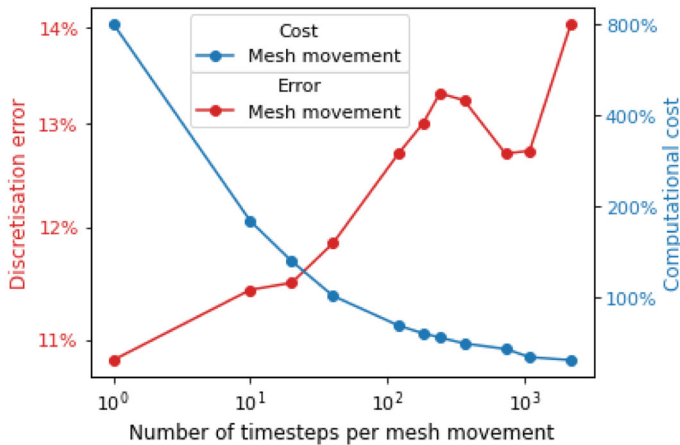
Fig. 6 Trade-off between discretisation error and computational cost due to mesh movement frequency for the migrating trench test case. The monitor function (23) has been applied to a mesh with 32 elements in the x-direction in the case $\alpha = \beta = 3$. Errors and times are percentages relative to the fixed mesh case with 32 elements
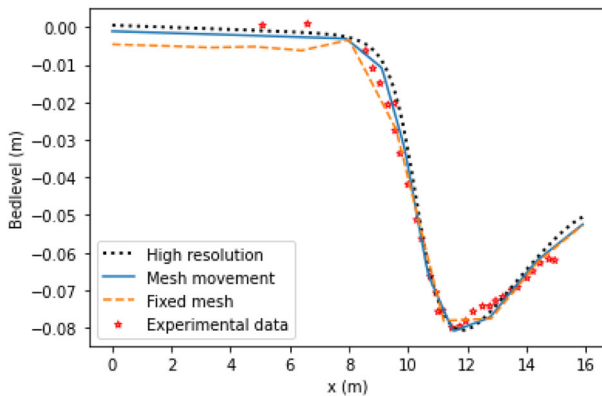

Fig. 7 Comparison of final bedlevels resulting from fixed and moving mesh simulations of the migrating trench test case on a mesh with 32 elements in the x-direction. The moving mesh simulation applies the monitor function (23) with $\alpha = \beta = 3$ every 40 timesteps. The experimental data from Van Rijn (1980) and the final bedlevel due to a high resolution fixed mesh simulation are also shown

compared to fixed meshes. This supports our argument, at least for this simple test case, that mesh movement can not only reduce error but also reduce the computational cost of the simulation. In the remainder of this section, we choose to move the mesh after every 40 timesteps of the hydro-morphodynamic model (equivalent to moving the mesh 54 times during the simulation), as this provides a good balance between computational cost and accuracy. Figure 7 compares the final bedlevel obtained by moving the mesh at this frequency with the bedlevel obtained using a fixed uniform mesh with the same number of elements, the 'true' value obtained using a high resolution mesh with 320 mesh elements in the x-direction and the experimental data from Van Rijn (1980). It

shows the mesh movement solution is much more accurate than the fixed mesh solution relative to the high resolution solution and thus that only moving the mesh every 40 timesteps is sufficient to see notable improvements. Additionally, in most regions the mesh movement solution is a more accurate approximation of the experimental data than the fixed mesh solution, although it should be noted that when comparing results to experimental data, there is an inbuilt model error, which is unaffected by mesh movement.

The mesh movement method used in this work also allows the user to set the values of α and β in (23), which control the mesh movement. In order to determine optimum values of these parameters, we conduct a small sensitivity study, but in future work will seek a gradient-based approach (see Sect. 5). Mesh movement methods have no effect on model error, only on the discretisation error and thus for brevity, we only show the results of the study for discretisation error. Figure 8 shows that for all number of mesh elements, the discretisation error is minimised when α is approximately 3. For the smaller number of elements, the discretisation error is significantly smaller than the discretisation error for the fixed mesh ($\alpha = \beta = 0$) and including the first order derivative of the bathymetry in the monitor function is important (*i.e.* $\beta \neq 0$). However, as the number of elements increases the effect of mesh movement on the discretisation error decreases and the first order derivative becomes less important. There is also a clear distinction in the way the discretisation error varies with α and β above and below 20 mesh elements. This distinction is also seen in Fig. 4, which shows that below 20 mesh elements the discretisation error dominates the model error but above 20 mesh elements the model error dominates the discretisation error. This distinction exists because above 20 mesh elements the initial trench profile is either accurately defined or close to being accurately defined even on a fixed uniform mesh (which is not the case below 20 mesh elements) and so the discretisation error is already very small ($\mathcal{O}(10^{-3})$).

Thus, we can conclude that for this test case a good general parameter choice for the monitor function (23) is $\alpha = 3$ and $\beta = 0$ (for large numbers of mesh elements in the x-direction) and $\alpha = \beta = 3$ (for small numbers of mesh elements in the x-direction). The greater dependence on the second order derivative than the first order derivative is predictable because the regions of the trench that are most difficult to capture are the corners where the second derivative is high.

4.1.3 Introducing a slope in the *y*-direction

The test case considered so far in this section is effectively 1D with little variation in the bathymetry or flow in the y-direction. However, most hydro-morphodynamic problems are effectively 2D in nature and thus we modify the test case by introducing a slope of 0.1 in the y-direction. All setup parameters are kept the same and the initial profile is shown in Fig. 9.

The aim of this modified migrating trench test case is to show the generality of our mesh movement framework and that, if the monitor function parameters are known for a similar simple test case, then a full sensitivity analysis is not required to find the parameters for the more complex case. Before using mesh movement, we first consider a series of fixed uniform meshes, in order to compare their accuracy to the
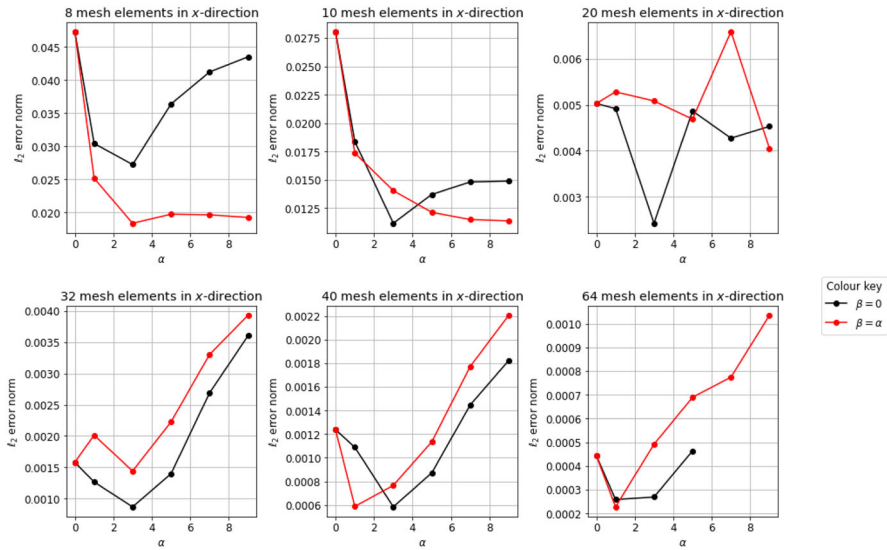
Fig. 8 Discretisation error curves for moving mesh simulations of the migrating trench test case under different values of α and β in monitor function (23)
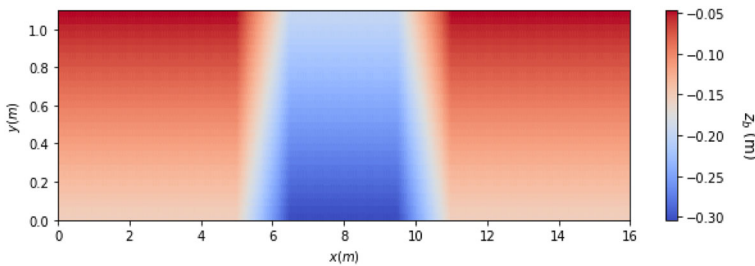

Fig. 9 Initial bed profile for the modified migrating trench test case

accuracy of the mesh movement framework. We choose 8, 10, 20, 32, 40 and 64 mesh elements in the x-direction. As the length in the y-direction is one sixteenth that in the x-direction, we choose the number of mesh elements in the y-direction to be one sixteenth of the number in the x-direction (rounding up to the nearest integer where necessary). Note that for the eight elements in the x-direction we use one element in the y-direction and for ten elements in the x-direction we use two elements in the y-direction, to allow for some mesh movement in the y-direction.

With the extra slope in the y-direction, we no longer have experimental data to compare against but can still analyse the discretisation error by considering results obtained on a fixed uniform mesh of 320 mesh elements in the x-direction ($\Delta x = 0.05$ m) and 20 mesh elements in the y-direction ($\Delta y = 0.055$ m) as a high resolution approximation of 'the truth'. Furthermore, as we no longer have pointwise experimental values, from this point forward, the error is calculated using the L^2 error norm over the whole domain. We note that for fixed uniform meshes as the mesh becomes finer the discreti-

sation error decreases approximately linearly (see Fig. 11b later in the section). Note the truncation error of the finite element representation and the associated discretisation of the non-linear shallow water equations used here is known to be second order (Comblen et al. 2010). However, formal order of convergence of the coupled nonlinear equations including morphodynamics are hard to derive and therefore a reduction in the observed order of convergence is expected.

Following our analysis from the original migrating trench test case, for this modified case, we move the mesh every 40 timesteps. In addition, as we are testing the generality of our mesh movement framework, instead of conducting a full sensitivity analysis to determine the parameters in the monitor function (23), we use parameters of the same magnitude as those found in the original test case. The introduction of an extra slope though, suggests that a different relationship between α and β may be optimal and thus we conduct a small study. Figure 10 shows the results of this study and shows that for small numbers of mesh elements in the x-direction the second order derivative of the bathymetry is the most important, then for 32 mesh elements the first and second order derivatives are of equal importance, and finally for more than 32 mesh elements, the first order derivative becomes the most important. This is predictable from the results from the original test case because for the latter, mesh movement results in a large error reduction for small numbers of mesh elements but when the number of elements is large, it has less of an effect. We infer from this that for the modified test case, for small numbers of mesh elements, the discretisation error is dominated by errors in the x-direction and thus the second order derivative is important (as in Sect. 4.1), but as the number of mesh elements increases the error in the y-direction starts to dominate meaning the first order derivative becomes the most important due to the linear slope in the y-direction.

Figure 10 also shows that using monitor parameters with a greater magnitude (a magnitude of 5) than that used in Sect. 4.1 (a magnitude of 3) results in a greater error reduction. This is due to the fact that for the modified case, the bathymetry is more complex and so more mesh movement is required. Therefore, whilst the magnitude of monitor function parameters obtained from simpler test cases results in error reduction in more complex problems, optimum mesh movement in this more complex case can be achieved by increasing the magnitude of the monitor function parameters.

Finally, Figs. 11a and 11b present a summary of our discretisation error results for the original and modified cases respectively. Figure 11a shows, for a given number of mesh elements, the minimum discretisation error achieved in our sensitivity study from using mesh movement (optimum parameter set) and the discretisation error achieved by using mesh movement with our generalised parameter set ($\alpha = 3$, $\beta = 3$ for small numbers of mesh elements in the x-direction and $\alpha = 3$, $\beta = 0$ for large numbers of mesh elements). Figure 11b only shows the discretisation error achieved by a generalised parameter set ($\alpha = 5$, $\beta = 0$ for small numbers of mesh elements; $\alpha = 5$, $\beta = 5$ for 32 mesh elements; and $\alpha = 0$, $\beta = 5$ for large numbers of mesh elements) because it was not necessary to conduct a full sensitivity analysis for this test case given that we had already done so for the original case. Moving mesh methods consistently result in a lower error than fixed mesh methods and furthermore our general parameter choice produces very similar error results to the optimum values. Thus, we have shown that mesh movement methods can be used to improve accuracy and reduce computational
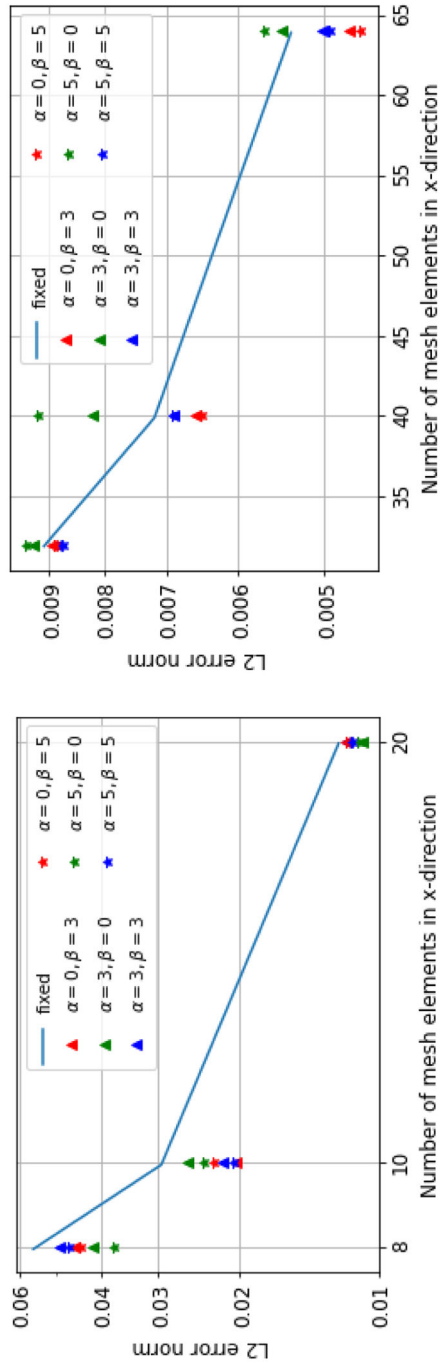
Fig. 10 Discretisation error curves for moving mesh simulations of the modified migrating trench test case under different values of α and β in (23)

(a) Original (Section 4.1.1 and 4.1.2)
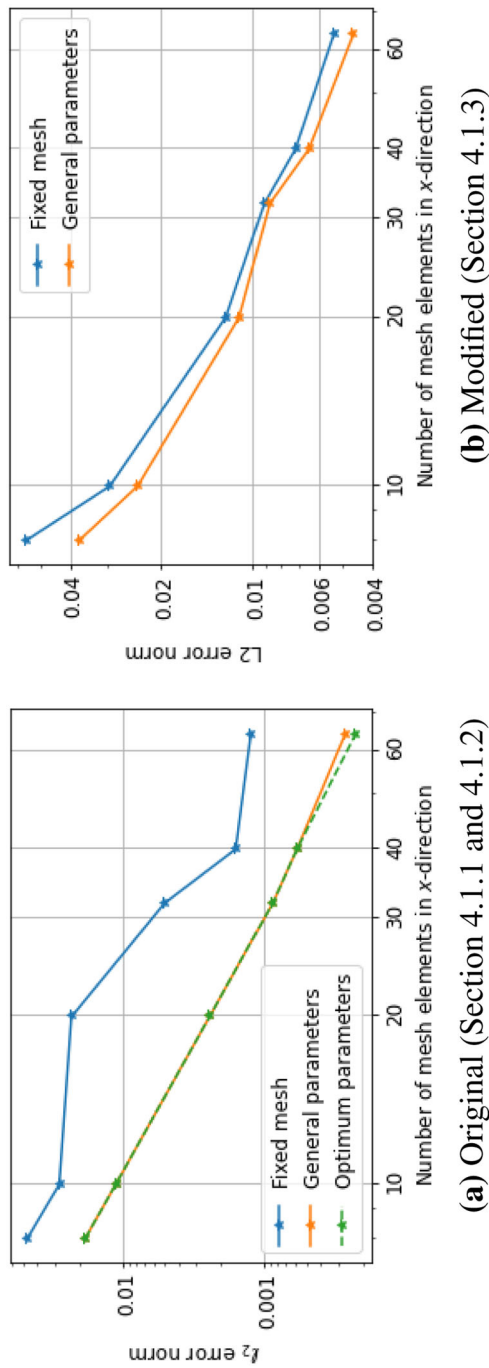
(b) Modified (Section 4.1.3)

Fig. 11 Discretisation error comparison of fixed mesh and moving simulations of the original and modified migrating trench test case. Both general and optimum parameters for the monitor function (23) are considered for the original version. Note that for the original case, the error shown is the pointwise ℓ_2 error norm and for the modified case the error shown is the L2 error norm over the whole domain

cost for quasi-1D and 2D cases with non-trivial bathymetries. We have also been able to use this test case to draw some conclusions about general good parameter choices for (23). In the next section, we test these general optimum parameter choices on a completely different 2D problem to see if they are again optimum or close to optimum choices.

4.2 Wet-dry interface test case: beach profile

We have shown that moving mesh methods can improve accuracy and efficiency for test cases with steep gradients in a fully wet domain. Coastal problems often have a wet-dry interface, for example as a wave or tide moves up a beach. These problems have been historically difficult to solve because of their computational expense, but mesh adaptation methods provide a way to retain accuracy, whilst improving efficiency. Mesh movement methods have been used previously to successfully simulate a wet-dry interface, for example in Zhou et al. (2013). However, to the best of our knowledge they have not previously been used to solve coupled hydro-morphodynamic cases with wet-dry interfaces.

In this section, we consider the test case of a wave running up a beach slope, with the initial bathymetry shown in Fig. 12. The figure also shows an example of the simulated wave surface due to the incoming wave, which is governed by

$$\mathbf{u} = 0.5 \cos{(0.05t)} \quad \text{at } x = 0 \, \text{m}. \tag{24}$$

The total simulated time is $60000 \, \text{s}$ (approximately equivalent to $16 \, \text{h}$) with a morphological acceleration factor of 250, meaning the full wave passes over the beach just over two times.

For this test case, we use a similar hydrodynamic setup to that used for the Balzano test case in Kärnä et al. (2011). Following that work, instead of using the Nikuradse friction formula as in previous test cases, the Manning friction formula is used to determine the bed friction

$$C_h = g \, \frac{n^2}{H^{1/3}}, \tag{25}$$

where n is the Manning drag coefficient set to $0.01 \, \text{sm}^{-1/3}$. As we are using wetting-and-drying, we use the wet-dry hydro-morphodynamic model discussed in Sect. 3.1. The parameters used in the simulation are summarised in Table 1. Note our model cannot currently simulate shoaling and breaking waves and thus a relatively high viscosity value of $1 \, \text{m}^2\text{s}^{-1}$ is used in the hydrodynamics to dissipate energy. This is standard practice, for example Li and Huang (2013) view viscosity as a model calibration parameter for energy dissipation, rather than a physical parameter.

4.2.1 Fixed uniform meshes

To begin, we consider a series of fixed meshes varying from a uniform spacing with 700 mesh elements in the x-direction ($\Delta x = 0.5 \, \text{m}$) to 70 mesh elements ($\Delta x = 5 \, \text{m}$). In order to keep the elements in the fixed mesh roughly uniform, an appropriate Δy
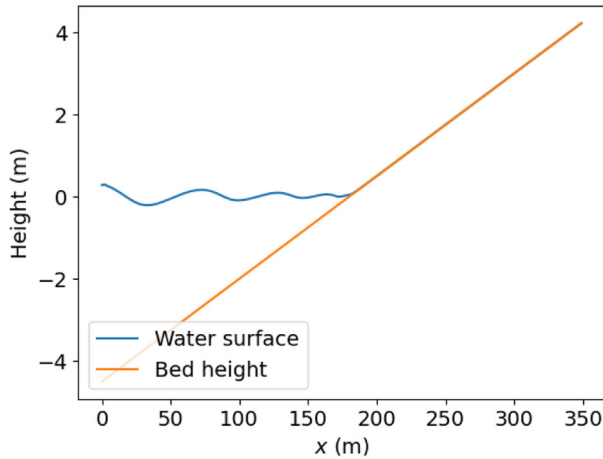
Fig. 12 Initial bathymetry of beach test case with an example of the simulated water surface

Table 1 Parameter values used in the beach test case

Variable name	Variable value
Length in x-direction	350 m
Length in y-direction	10 m
Timestep, dt.	0.5 s
Bed slope gradient	1/40
Morphological simulation time	60,000 s
Morphological acceleration factor	250
Median particle size (d_{50})	1×10^{-4} m
Sediment density (ρ_s)	2650 kg m^{-3}
Water density (ρ_f)	1000 kg m^{-3}
Kinematic viscosity (ν_0)	1 m^{2}s^{-1}
Bed sediment porosity (p')	0.4
Diffusivity (ε_s)	1 m^{2}s^{-1}
Manning friction coefficient (n)	0.01 sm$^{-1/3}$
Wetting-and-drying parameter (δ)	0.05 m

is also chosen. Note that we set $\Delta t = 0.5$ s for all the simulations in this section, in order to ensure that the wave is correctly defined. Because this is a synthetic test case for which there exists no real data, we use the model solution at 700 mesh elements in the x-direction ($\Delta x = 0.5$ m) and 20 mesh elements in the y-direction ($\Delta y = 0.5$ m) as a high resolution approximation of 'the truth' and are thus showing the estimates of the discretisation error. Note for this section the discretisation error is the pointwise ℓ_2 error norm at $y = 5$ m at evenly spaced intervals. Thus, for a fixed uniform mesh Fig. 13 shows that, as the mesh becomes finer, the discretisation error decreases approximately linearly. In Kärnä et al. (2011), they show that, for simple test cases, the wetting and drying scheme detailed in Sect. 3 results in an order of convergence of 1.5. However,
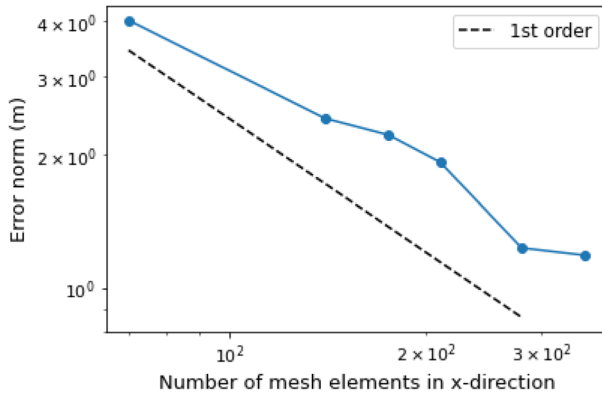
Fig. 13 Discretisation error in a simulation of the beach test case based on a fixed uniform mesh. The reference solution uses a mesh with 700 elements in the x-direction. The discretisation error is the pointwise ℓ_2 error norm

our test case is more complex and includes coupling to morphology changes, and thus a further reduction as observed here is to be expected.

4.2.2 Mesh movement

We apply mesh movement methods using the same monitor function as that used in the previous test cases (23). Figure 14 shows an example of how the mesh is moved using this monitor function for this particular test case. Note that again the mesh hardly moves in the y-direction because the bathymetry is uniform in this direction. To understand the effect of this mesh movement, Fig. 15 compares the final bedlevel obtained by the mesh movement shown in Fig. 14 with the bedlevel obtained using a fixed uniform mesh with the same number of elements and the 'true' value obtained using a high resolution mesh with 700 elements in the x-direction. This shows that using mesh movement allows us to capture the steep bed gradients more accurately than using a fixed uniform mesh of the same resolution.

As discussed in the previous sections, we can also choose the mesh movement frequency. This is particularly important in a test case like this one with fast flowing hydrodynamics and therefore we conduct a sensitivity study considering multiple different mesh resolutions. Figure 16 shows that decreasing the number of timesteps per mesh movement almost always decreases the error and always increases the computational cost. For the smaller number of mesh elements considered (70 mesh elements in the x-direction), the error is minimised by using 16 timesteps per mesh movement and increases when the number of timesteps is decreased beyond this point. We hypothesise that this is because when there are fewer numbers of mesh elements, frequent mesh movement can result in numerical instabilities and hence larger errors. By contrast, for both the larger numbers of mesh elements considered, the error is minimised by using 10 timesteps per mesh movement and is proportionally larger using 16 timesteps than at other similar frequencies. Thus, following our observations, in this test case the mesh is moved every 10 timesteps for all mesh resolutions apart from for 70 mesh
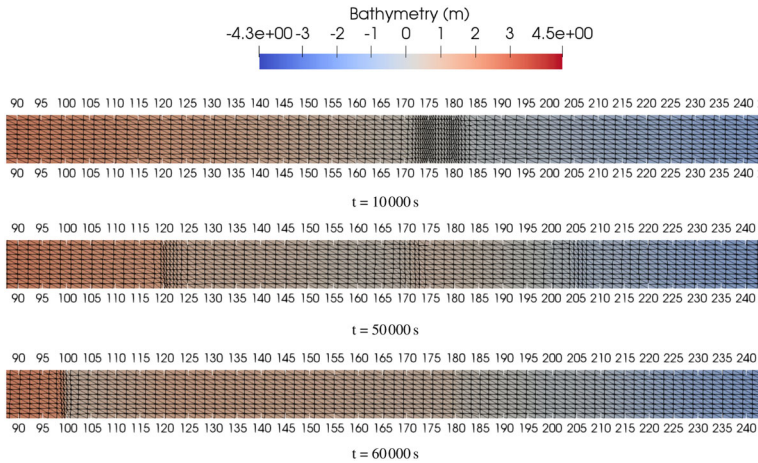
Fig. 14 Snapshots of bathymetry and the underlying mesh for a moving mesh simulation of the beach test case. The monitor function (23) has been applied to a mesh with 175 elements in the x-direction with $\alpha = 5$ and $\beta = 0$. Results are shown at three points in time demonstrating mesh movement to capture bed evolution to the right
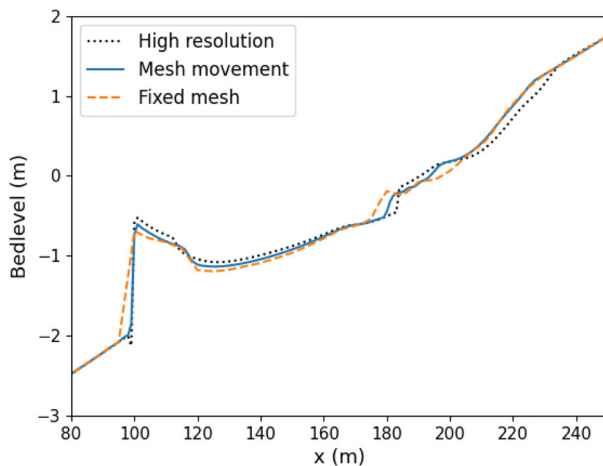

Fig. 15 Comparison of final bedlevel resulting from fixed and moving mesh simulations of the beach test case on a mesh with 175 elements in the x-direction. The moving mesh simulation applies the monitor function (23) with $\alpha = 5$ and $\beta = 0$. The final bedlevel due to a high resolution simulation on a fixed mesh with 700 elements in the x-direction is also shown

elements when it is moved every 16 timesteps. Significantly with this choice of mesh movement frequency we are able to halve the error relative to a fixed uniform mesh of the same resolution.

We can also choose an α and β in (23) to minimise the discretisation error. As the initial bedlevel is smooth (see Fig. 12), the first order derivative of the bedlevel is not important and thus we set $\beta = 0$. Moreover, in the previous section, we found that a parameter magnitude of 3 for simple cases and 5 for more complex cases provides a
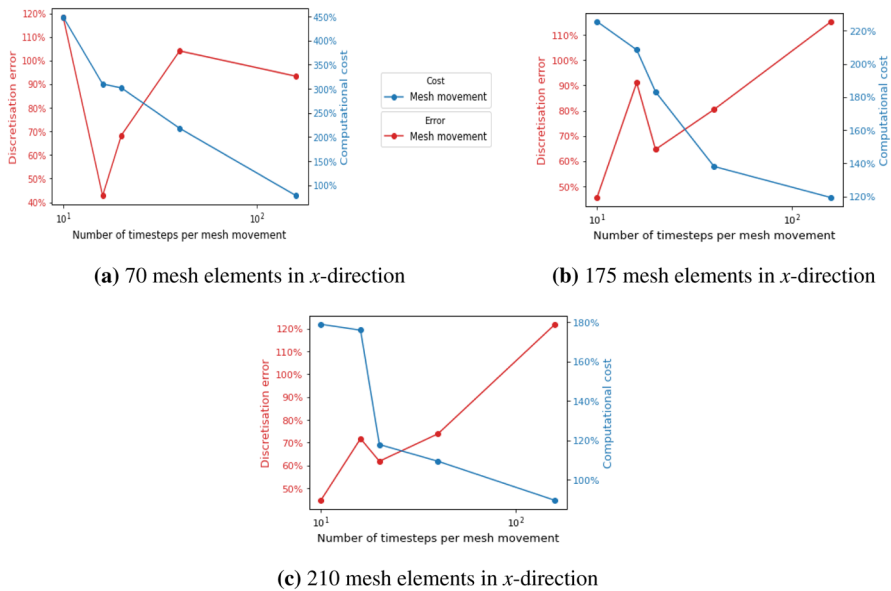
(a) 70 mesh elements in *x*-direction

(b) 175 mesh elements in *x*-direction

(c) 210 mesh elements in *x*-direction

Fig. 16 Trade-off between discretisation error and computational cost due to mesh movement frequency for the beach test case. The monitor function (23) has been applied using $\alpha = 5$ and $\beta = 0$ for all numbers of mesh elements. Errors and times are percentages relative to the fixed mesh case with the same resolution

good general optimisation of the discretisation error and thus we conduct our sensitivity study using similar values.

Figure 17 shows the results of this sensitivity study. Most significantly, using mesh movement it is possible to more than half the error relative to the fixed uniform mesh error ($\alpha = 0$) for almost all mesh resolutions considered. Note that the exception is 280 elements where error reduction is not as large because our model cannot produce a result for $\alpha > 3$ because of Courant number restrictions (recall from Sect. 3.2 that our model is not fully implicit). Additionally, Fig. 17 shows that a general magnitude of 5 corresponds to an either optimum or very close to optimum minimisation in the discretisation error for all mesh resolutions considered. This fits with our physical understanding of the problem because the presence of waves makes the problem more complex thus requiring a stronger mesh movement factor. Figure 18 emphasises the significant error reduction achieved by using mesh movement methods on this problem and importantly shows that for most mesh resolutions the general parameter choice of $\alpha = 5$ and $\beta = 0$ results in similar error results to the optimum parameter values.

We have thus shown that for this relatively complex wetting-and-drying test case, mesh movement methods can more than halve the error at most mesh resolutions compared to a fixed uniform mesh, even when using general parameters that have not been tuned for this specific test case.
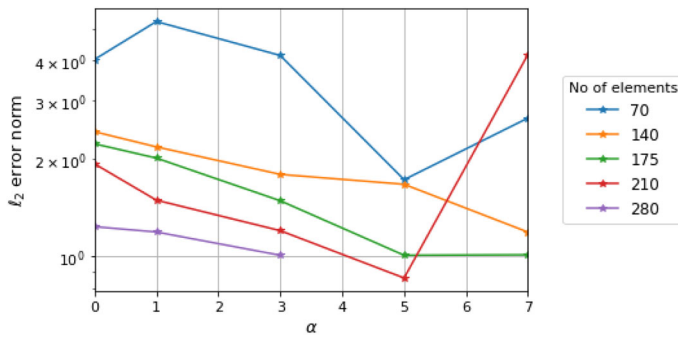
Fig. 17 Discretisation error curves for moving mesh simulations of the beach test case under different values of α in the monitor function (23)
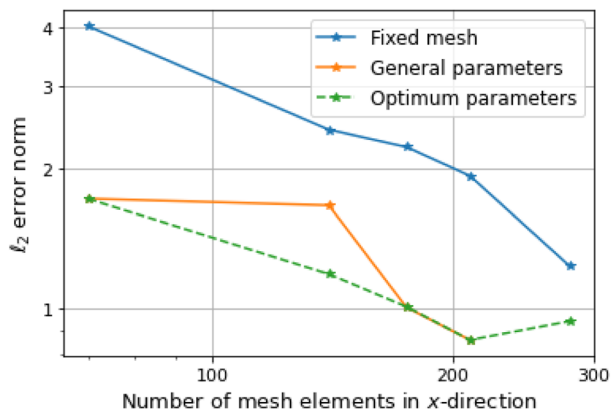

Fig. 18 Discretisation error comparison of fixed mesh and moving simulations of the beach test case. Both general ($\alpha = 5$, $\beta = 0$) and optimum parameters for the monitor function (23) are considered
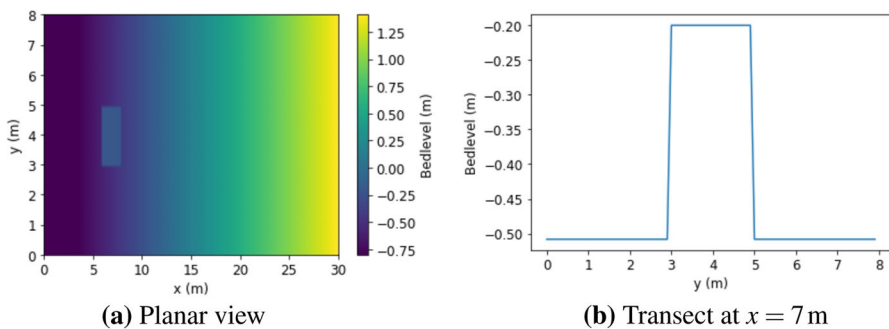

(a) Planar view

(b) Transect at $x = 7\,\mathrm{m}$

Fig. 19 Initial bed profile for the tsunami with obstacle test case. Left: Planar view. Right: Transect at x = 7m

4.3 Complex bathymetry with a wet-dry interface test case: tsunami-like wave with an obstacle

As a final test of our mesh movement framework, we consider an example with both a wet-dry interface and a complex initial bathymetry (Fig. 19). So far in this work, we have mostly considered quasi-1D cases. Thus here we choose a more complex bathymetry of a cube obstacle on a sloping beach, inspired by the 2D test case in Hudson and Sweby (2005). To increase the complexity of this test case, we simulate a series of tsunami-like solitary waves breaking over this beach, inspired by the test case in Kobayashi and Lawrence (2004). It should be noted that the hydrodynamic component of the model described in Sect. 3 is governed by the non-linear shallow water equations, which are not able to exactly represent the propagation of a solitary wave (Barthélemy 2004; Kazhyken et al. 2021). However, because this is a theoretical test case, in this section we are not comparing against experimental data but instead always compare against internally consistent solutions obtained using the shallow water solver and therefore our mesh movement results are still valid. Moreover, in Clare et al. (2021a), we consider the original experimental version of the tsunami-like solitary wave test case in Kobayashi and Lawrence (2004) and find that our model results agree well with experimental data. This suggests that using shallow water equations instead of a dispersive wave model does not result in substantial divergence from the experimental set-up.

Following Kobayashi and Lawrence (2004), we define the incoming wave using the following formula for a positive solitary wave

$$\eta(t) = H_{\text{wave}} \operatorname{sech}^2 \left(\sqrt{\frac{3H_{\text{wave}}}{4h}} \frac{\sqrt{g(H_{\text{wave}} + h)}}{h} (t - t_{\text{max}}) \right) + \eta_{\text{down}}, \qquad (26)$$

where H_{wave} is the average wave height, h the still water depth, t_{max} the arrival time of the wave crest and η_{down} the initial decrease of the elevation at the beginning of the simulation. In the experiment, the solitary wave is generated 8 times with the bed not adjusted after each wave. Given we are not matching with experimental data and for reasons of time, here we only run one wave in our simulation. However, in preliminary tests, we found that we can set a morphological acceleration factor, m_f as high as four without any noticeable differences between the sped-up results and the results with no morphological acceleration. Thus, in this test case, we set m_f equal to four, meaning that this one wave simulation is approximately equivalent to simulating four waves. In addition, in the experiment, the simulation is run for 40 s with $t_{\text{max}} = 23.9$ s for each solitary wave, but for the first 20 s the system is stationary. Thus, we run our model simulation for 20 s with $t_{\text{max}} = 3.9$ s for each solitary wave.

All the parameters used in the simulation are summarised in Table 2 and taken from the experiment in Kobayashi and Lawrence (2004) and also Li and Huang (2013) (which simulated the experiment using Delft3D). Following Li and Huang (2013), we use the Chezy friction formula defined by

$$C_{\text{chezy}} = \frac{g}{n^2} \qquad (27)$$

Table 2 Parameter values used in the tsunami with obstacle test case

Variable name	Variable value
Length in x-direction	30 m
Length in y-direction	8 m
Timestep (dt)	0.025 s
Morphological simulation time	20 s
Morphological acceleration factor	4
Median particle size (d_{50})	1.8×10^{-4} m
Sediment density (ρ_s)	2650 kg m^{-3}
Water density (ρ_f)	1000 kg m^{-3}
Kinematic viscosity (ν)	0.8 m^{2}s^{-1}
Bed sediment porosity (p')	0.4
Diffusivity (ε_s)	1 m^{2}s^{-1}
Chezy friction coefficient (n)	65 m$^{1/2}$s^{-1}
Angle of repose	20°
Wetting-and-drying parameter (δ)	1/60 m
Average waveheight (H_{wave})	0.216 m
Still water depth (h)	0.18 m
Initial elevation decrease (η_{down})	-0.0025 m
Wave arrival time (t_{max})	3.9 s

where n is the Chezy friction parameter. Note that, as in Li and Huang (2013), we do not simulate bedload transport because studies have shown that sediment transport due to tsunami waves mainly occurs due to suspended sediment (Goto et al. 2011). However, recall from Sect. 3.1 that this more complex bathymetry requires the implementation of the sediment slide mechanism.

4.3.1 Fixed uniform meshes

As with the other test cases, we begin by considering a series of fixed meshes with 30, 60, 90, 120 and 150 mesh elements in the x-direction corresponding to 8, 16, 24, 32, 40 mesh elements in the y-direction respectively, meaning the mesh elements are roughly uniform. Because we have combined two test cases to construct this test case, we no longer have experimental data available. Thus, we use the model solution at 600 mesh elements in the x-direction ($\Delta x = 0.05$ m) and 160 mesh elements in the y-direction ($\Delta y = 0.05$ m) as a high resolution approximation of 'the truth'. The discretisation error in this section is the L^2 error over the whole domain.

When we run our hydro-morphodynamic model on these fixed uniform meshes, we find that with only 30 mesh elements in the x-direction, our hydro-morphodynamic model crashes no matter how small a timestep is used. This is because at such a coarse resolution, the model cannot accurately simulate the movement of the tsunami-like wave along the slope and instead unphysical shocks form that cannot be properly resolved. For the other fixed uniform meshes, the discretisation error decreases approx-

imately linearly, as the number of mesh elements increases, the same convergence order as for the wet-dry test case considered in Sect. 4.2.

4.3.2 Mesh movement

This test case is more complex than the ones considered previously because there are two regions of potentially complex bathymetry - the first due to the block and the second due to sediment transport as the wave breaks at the wet-dry interface. Thus, instead of using the same monitor function as in previous test cases (23), we add a component that tracks the wet-dry interface so the new monitor function is

$$
m(x, y) = 1 + \mu \left(\max_{x,y} \left(\alpha \frac{\left\| \underline{\boldsymbol{H}}(z_b) \right\|_F}{\max_{x,y} \left\| \underline{\boldsymbol{H}}(z_b) \right\|_F}, \ \beta \frac{\left\| \nabla z_b \right\|_2}{\max_{x,y} \left\| \nabla z_b \right\|_2} \right) + \frac{\lambda}{\cosh(b_\lambda (\eta - z_b))^2} \right), \quad (28)
$$

where η is the elevation, z_b the bed level, b_λ controls the width of the wet-dry interface tracker and μ, α, β and λ are all user-defined parameters. Note, we set b_λ equal to 1 for all numbers of mesh elements, apart from for the smallest number of mesh elements when we set b_λ equal to 5 to ensure that, despite the small number of elements, the wet-dry tracker still has an effect. Here we choose to keep the wet-dry tracker outside of the expression maximising the first and second order bathymetry derivatives, because the bathymetry gradient and wet-dry interface are independent of each other and so should be separate contributions in the monitor function.

Figure 20 shows an example of how the mesh moves using this monitor function with 120 mesh elements in the x-direction. To better show the mesh movement and bed evolution for this test case with this number of mesh elements, the figure shows the mesh only and the 3D view of the bathymetry separately. The first and second order bathymetry derivatives in (28) cause the mesh to deform around the edges of the block whilst the wet-dry interface monitor tracks the movement of the wave up and down the slope. To illustrate this tracking movement, the wet-dry interface is plotted as a thick black line on Fig. 20.

As with the previous test cases, we can choose how frequently the mesh is moved. Figure 21 shows that increasing the frequency of mesh movement decreases the discretisation error and increases the computational cost. Whilst the cost of using mesh movement is always greater than the cost of using a fixed uniform mesh, mesh movement methods always result in an error that is less than half the size of the error from the fixed uniform mesh. The accuracy improvement seems to plateau at around 20 timesteps per mesh movement and therefore in the remainder of this section, this is the frequency with which we move the mesh. As the simulation time is $20\,\text{s}$ and the timestep is $dt = 0.025\,\text{s}$, this is equivalent to moving the mesh 40 times during the simulation. Figure 22 compares the final bedlevel obtained by moving the mesh at this frequency to the bedlevel obtained using a fixed uniform mesh with the same number of elements and the 'true' value obtained using a high resolution mesh with 600 elements in the x-direction. It shows the mesh movement solution is much more accurate than the fixed mesh solution and thus that moving the mesh at this frequency is appropriate for this new more complex monitor function (28).
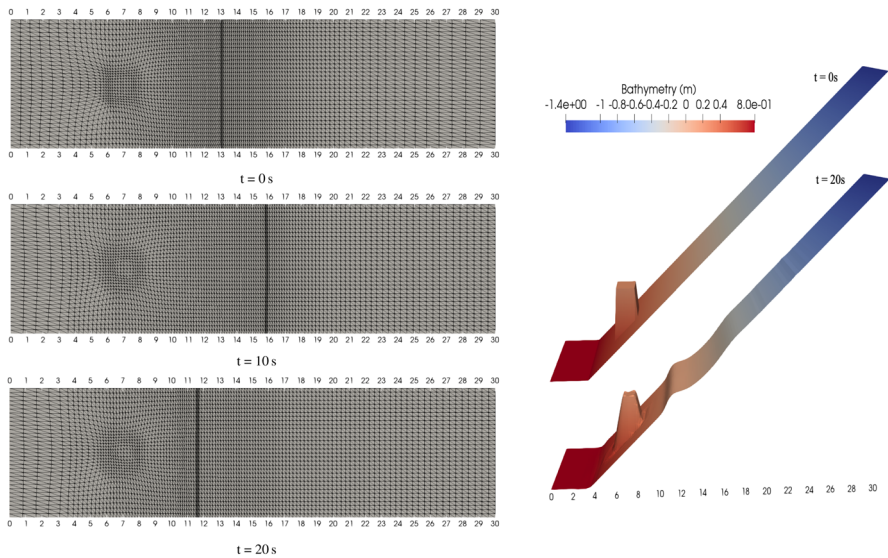
Fig. 20 Mesh movement using (28) for the tsunami with obstacle test case with 120 mesh elements in the x-direction and $\mu = 15$ and $\alpha = \beta = \lambda = 1$ (*i.e.* an equal contribution from the first and second order bathymetry derivatives and the wet-dry interface tracker). Results are shown with only the mesh at three points in time with the wet-dry interface shown as a thick black line (LEFT) and in 3D form at two points in time (RIGHT) demonstrating mesh movement to capture bed evolution to the right
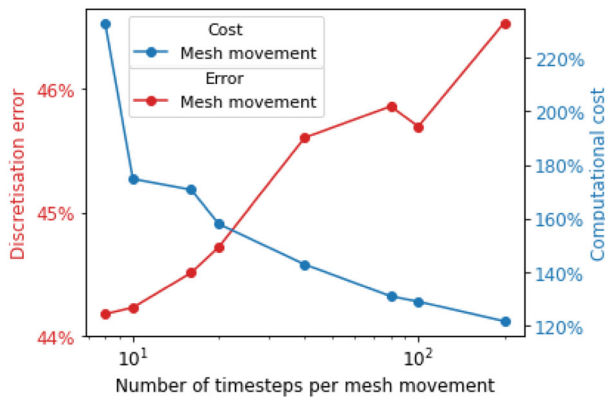
Fig. 21 Trade-off between discretisation error and computational cost due to mesh movement frequency for the tsunami with obstacle test case. The monitor function (28) has been applied to a mesh with 60 elements in the x-direction in the case $\mu = 7$, $\alpha = 0$, $\beta = \lambda = 1$. Errors and times are percentages relative to the fixed mesh case with 60 elements

We can also choose α, β and λ to minimise the discretisation error. In previous test cases in this work, we find that a value equivalent to $\mu \approx 5$ provides a good general optimisation of the discretisation error and thus we conduct our sensitivity study using similar values. Figure 23 shows the results of this sensitivity study and shows that again a magnitude of 5 provides a good general optimisation of the discretisation error. In
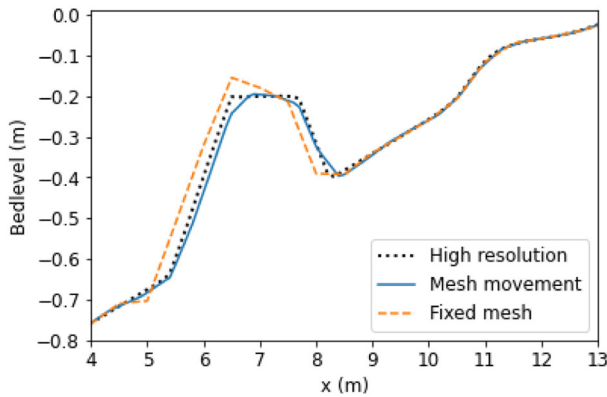
Fig. 22 Comparison of final bedlevels resulting from fixed and moving mesh simulations of the tsunami with obstacle test case on a mesh with 60 elements in the x-direction. The moving mesh simulation applies the monitor function (28) with $\mu = 7$, $\alpha = 0$, $\beta = \lambda = 1$ every 40 timesteps. The final bedlevel due to a high resolution simulation on a fixed mesh with 600 elements in the x-direction is also shown

all cases, the error is reduced by using mesh movement methods compared to the fixed uniform mesh ($\mu = 0$). Note that for 30 mesh elements in the x-direction, the fixed uniform mesh model crashes, which is why no error is plotted for $\mu = 0$ on this subfigure.

The figure also shows the effect on the discretisation error of different relationships between α, β and λ in the monitor function (28). The largest mesh movement errors occur almost always when the monitor function only includes the wet-dry interface tracker ($\lambda = 1, \alpha = \beta = 0$). This is understandable given that with this monitor function, the obstacle in the wave-approach is not well-captured. However, in almost all cases the inclusion of the wet-dry interface tracker with some combination of the first and second order derivative of the bathymetry results in a decrease in the discretisation error relative to the tracker not being present. In fact, it is only with the inclusion of the wet-dry interface tracker that our model can properly resolve the wave movement on the coarse mesh with only 30 elements in the x-direction. This highlights that appropriate mesh movement can not only decrease computational cost and improve accuracy, but also improve model stability and justifies the use of the more complex monitor function (28) in this test case. A good general choice for the relationship between α, β and λ is $\alpha = \beta = \lambda = 1$, *i.e.* an equal weighting between the first and second order derivatives of the bathymetry and the wet-dry interface tracker. This makes physical sense because the bathymetry derivatives are necessary to capture the obstacle correctly and the interface tracker is necessary to capture the erosion and deposition caused by the incoming wave. Using this general parameter choice, we can more than halve our model error for the same number of mesh elements when compared to a fixed uniform mesh, which is a notable result.

We have thus shown that a good general parameter choice for this test case is $\mu = 5$ and $\alpha = \beta = \lambda = 1$, which is the same magnitude as the complex test cases considered in Sect. 4.1.3 and 4.2 and the same order of magnitude as the simple test case in Sect. 4.1. Using these general parameters, in Fig. 24 we consider the relationship
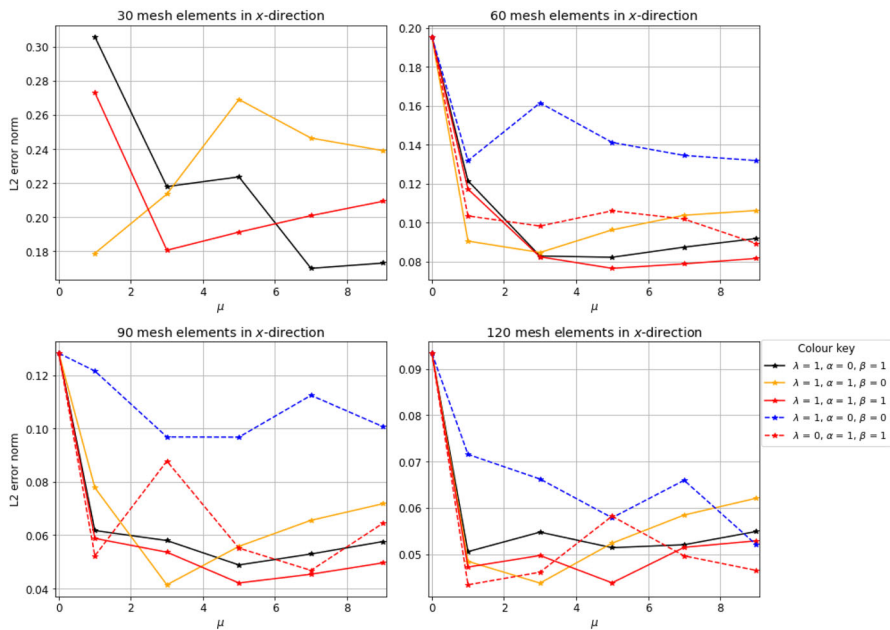
Fig. 23 Discretisation error curves for moving mesh simulations of the tsunami with obstacle test case under different values of α, β and λ in the monitor function (28)

of simulation accuracy versus computational cost. The figure shows that using mesh movement methods results in both a significant improvement in accuracy and a significant reduction in computational cost, even when general parameters are used rather than optimum parameters. (Note the optimum parameters plotted here are the parameters that provide the smallest error and are not necessarily the fastest simulations, hence why the general parameters perform better in the cost-to-accuracy ratio than the 'optimum' parameters). In many cases, it is possible to halve the discretisation error for the same computational cost, a notable improvement. This is a particularly good result if we wish to run this test case more than once for calibration purposes. In addition, using mesh movement methods reduces the number of mesh elements to required to achieve a good accuracy, which means that the memory costs of our simulation are reduced when using adjoint methods through *pyadjoint* (see Clare et al. (2021a)).

Therefore, we have shown that for test cases with relatively complex bathymetries and a wet-dry interface, we can not only in many cases more than halve the error for the same number of mesh elements but also reduce computational cost and improve model stability, even when using general parameters that have not been tuned for this specific test case, a noteworthy result.
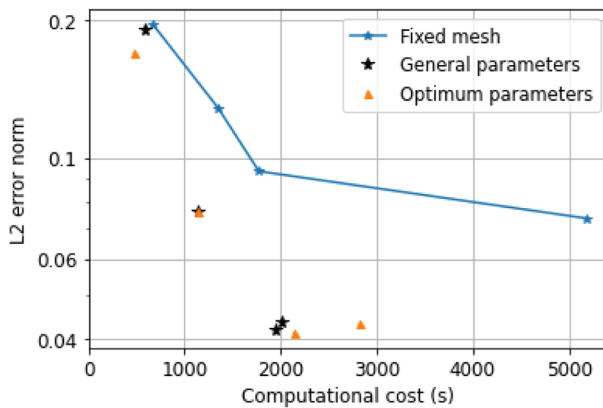
Fig. 24 Computational cost vs discretisation error for fixed mesh and mesh movement methods with general and optimum parameters for the tsunami with obstacle test case. Note the different points correspond to different numbers of mesh elements

5 Conclusion

In this work we have implemented a mesh movement scheme as part of a hydro-morphodynamic model for the first time. We have shown that these mesh movement methods can be used to improve accuracy and decrease the computational cost of hydro-morphodynamic test cases with complex bathymetries and/or moving wet-dry interfaces. Moreover, in certain cases we have demonstrated that mesh movement can also improve model stability. A highlight is that these improvements are particularly significant with test cases with wet-dry interfaces, which many coastal zone applications include. For both the coastal zone test cases considered in this work, using mesh movement methods results in an error that is less than half the size of the error from the fixed uniform mesh with the same number of elements. This in turn leads to a reduction in computational and memory cost, and in future work will allow us both to simulate more complex wet-dry test cases accurately and efficiently, and to more readily conduct model calibration as more simulations can be performed for an equivalent cost.

For the mesh movement method considered, we present a monitor function for which the scaling factor that optimises the error reduction is fairly predictable from the physical characteristics of the test case. In particular, we have found that for 2D cases, and 1D cases with steep gradients, optimum mesh parameters have a magnitude of approximately 5. This will facilitate using this monitor function on further problems. In this work, we have used small scale sensitivity studies to obtain these optimum mesh parameters, but in future work, we will use the more rigorous approach of the adjoint framework within Firedrake to allow us to determine the optimum values for the scaling parameters. Section 4.1 showed that our hydro-morphodynamic model can be

combined with the adjoint framework to determine an optimum diffusivity coefficient, and thus this is a promising line for follow-up research.

Declarations

Code availability The relevant code for the moving mesh hydro-morphodynamic model presented in this work has been stored at at https://doi.org/10.5281/zenodo.5155816.

A Appendix: Sediment conservation check

As discussed in Sect. 3.1, the wetting-and-drying scheme used in our model is known to leak sediment and thus we implement a conservative sediment scheme. To verify that our conservative scheme has improved the sediment conservation, we consider the simple Thacker test case of oscillations in a paraboloid bowl with diameter 430,620 m (Thacker 1981). The hydrodynamic version of this test case is presented in Balzano (1998) and we refer the reader here for more details. The free surface is initially a paraboloid of revolution with a depth of approximately 50 m that oscillates inside the bowl with no forcing, but does not leave the bowl: the problem is closed hydrodynamically. We introduce a Gaussian blob of sediment in the wet part of the domain defined by the expression

$$C(t_0) = \begin{cases} 100 \exp^{-(x^2+y^2)/100000} & H \geq 0 \\ 0 & H < 0 \end{cases} \tag{A.1}$$

and run the simulation for 6 h. The principal differences between the conservative and non-conservative schemes are due to the advection term and therefore to avoid unnecessary complications we set the diffusion, erosion and deposition terms to zero. At each timestep, t_n, we calculate the normalised mass error using the following formula

$$\frac{\int C(t_n)\tilde{H}(t_n)\,dx - \int C(t_0)\tilde{H}(t_0)\,dx}{\int C(t_0)\tilde{H}(t_0)\,dx}. \tag{A.2}$$
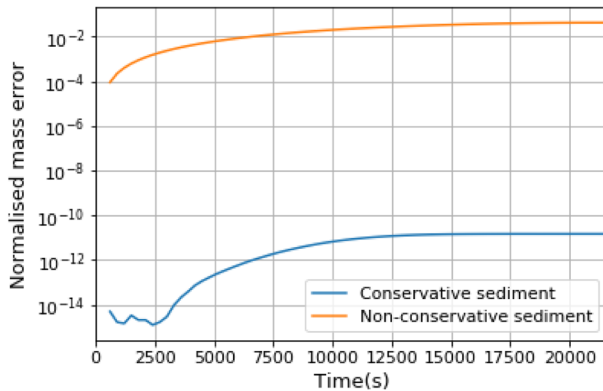
Fig. 25 Sediment mass conservation error when using wetting-and-drying with non-conservative sediment Eq. (13) and with conservative sediment Eq. (14)

Figure 25 shows that the normalised mass error is much lower for the conservative sediment Eq. (14) but $\mathcal{O}(10^{-12})$ than for the non-conservative sediment Eq. (13) and thus that the conservative sediment equation conserves sediment at a much better rate than the non-conservative scheme.

References

Amoudry, LO., Souza, AJ.: Deterministic coastal morphological and sediment transport modeling: A review and discussion. Reviews of Geophysics 49(2) (2011)

Apsley, D.D., Stansby, P.K.: Bed-load sediment transport on large slopes: model formulation and implementation within a RANS solver. J. Hydraul. Eng. **134**(10), 1440–1451 (2008)

Avdis, A., Candy, A.S., Hill, J., Kramer, S.C., Piggott, M.D.: Efficient unstructured mesh generation for marine renewable energy applications. Renew. Energy **116**, 842–856 (2018)

Awanou, G.: Quadratic mixed finite element approximations of the Monge-Ampère equation in 2d. Calcolo **52**(4), 503–518 (2015)

Balzano, A.: Evaluation of methods for numerical simulation of wetting and drying in shallow water flow models. Coastal Eng. **34**(1–2), 83–107 (1998). https://doi.org/10.1016/S0378-3839(98)00015-5

Barthélemy, E.: Nonlinear shallow water theories for coastal waves. Surv. Geophys. **25**(3–4), 315–337 (2004)

Becker, R., Rannacher, R.: A feed-back approach to error control in finite element methods: Basic analysis and examples. Citeseer (1996)

Becker, R., Rannacher, R.: An optimal control approach to a posteriori error estimation in finite element methods. Acta Numerica **10**, 1–102 (2001)

Benkhaldoun, F., El Mahi, I., Sari, S., Seaid, M.: An unstructured finite volume method for coupled models of suspended sediment and bed-load transport in shallow water flows. Int. J. Numer. Methods Fluids **72**(9), 967–993 (2013). https://doi.org/10.1002/d.3771

Budd, C.J., Williams, J.: Moving mesh generation using the parabolic Monge-Ampère equation. SIAM J. Sci. Comput. **31**(5), 3438–3465 (2009)

Budd, C.J., Huang, W., Russell, R.D.: Adaptivity with moving grids. Acta Numerica (2009). https://doi.org/10.1017/S0962492906400015

Budd, C.J., Cullen, M.J.P., Walsh, E.J.: Monge-Ampére based moving mesh methods for numerical weather prediction, with applications to the eady problem. J. Comput. Phys. **236**, 247–270 (2013)

Clare, M.C.A., Kramer, S.C., Cotter, C.J., Piggott, M.D.: Calibration, inversion and sensitivity analysis for hydro-morphodynamic models through the application of adjoint methods (2021a) https://doi.org/10.31223/X5F327

Clare, M.C.A., Percival, J.R., Angeloudis, A., Cotter, C.J., Piggott, M.D.: Hydro-morphodynamics 2D modelling using a discontinuous Galerkin discretisation. Comput. Geosci. **146**, 104658 (2021)

Comblen, R., Lambrechts, J., Remacle, J.F., Legat, V.: Practical evaluation of five partly discontinuous finite element pairs for the non-conservative shallow water equations. Int. J. Numer. Methods Fluids **63**(6), 701–724 (2010)

Delandmeter, P.: Discontinuous Galerkin finite element modelling of geophysical and environmental flows. PhD thesis, Universite catholique de Louvain (2017)

Donea, J., Huerta, A., Ponthot, J.P., Rodríguez-Ferran, A.: Arbitrary Lagrangian–Eulerian methods. Encyclopedia of Computational Mechanics Second Edition pp 1–23 (2017)

Farhat, C., Degand, C., Koobus, B., Lesoinne, M.: Torsional springs for two-dimensional dynamic unstructured fluid meshes. Comput. Methods Appl. Mech. Eng. **163**(1–4), 231–245 (1998)

Funke, S., Farrell, P., Piggott, M.: Reconstructing wave profiles from inundation data. Comput. Methods Appl. Mech. Eng. **322**, 167–186 (2017)

Geology Cafe: Chapter 11 - rivers, streams, and water underground. http://geologycafe.com/class/chapter11.html, accessed on 2020-09-11 (2015)

Goto, K., Takahashi, J., Oie, T., Imamura, F.: Remarkable bathymetric change in the nearshore zone by the 2004 Indian Ocean tsunami: Kirinda Harbor, Sri Lanka. Geomorphology **127**(1–2), 107–116 (2011)

Harris, D.L., Rovere, A., Casella, E., Power, H., Canavesio, R., Collin, A., Pomeroy, A., Webster, J.M., Parravicini, V.: Coral reef structural complexity provides important coastal protection from waves under rising sea levels. Sci. Adv. **4**(2), eaao4350 (2018)

Huang, W., Ren, Y., Russell, R.D.: Moving mesh partial differential equations (MMPDEs) based on the equidistribution principle. SIAM J. Numer. Anal. **31**(3), 709–730 (1994)

Hudson, J., Sweby, P.K.: A high-resolution scheme for the equations governing 2D bed-load sediment transport. Int. J. Numer. Methods Fluids **47**(10–11), 1085–1091 (2005)

Kärnä, T., de Brye, B., Gourgue, O., Lambrechts, J., Comblen, R., Legat, V., Deleersnijder, E.: A fully implicit wetting-drying method for DG-FEM shallow water models, with an application to the Scheldt Estuary. Comput. Methods Appl. Mech. Eng. **200**(5–8), 509–524 (2011)

Kärnä, T., Kramer, S.C., Mitchell, L., Ham, D.A., Piggott, M.D., Baptista, A.M.: Thetis coastal ocean model: discontinuous Galerkin discretization for the three-dimensional hydrostatic equations. arXiv preprint arXiv:1711.08552 (2017)

Kazhyken, K., Videman, J., Dawson, C.: Discontinuous Galerkin methods for a dispersive wave hydro-sediment-morphodynamic model. Comput. Methods Appl. Mech. Eng. (2021). https://doi.org/10.1016/j.cma.2021.1136842010.06167

Kobayashi, N., Lawrence, A.R.: Cross-shore sediment transport under breaking solitary waves. J. Geophys. Res.: Oceans (2004). https://doi.org/10.1029/2003jc002084

Lakkis, O., Pryer, T.: A finite element method for nonlinear elliptic problems. SIAM J. Sci. Comput. **35**(4), A2025–A2045 (2013)

Li, L., Huang, Z.: Modeling the change of beach profile under tsunami waves: a comparison of selected sediment transport models. J. Earthq. Tsunami (2013). https://doi.org/10.1142/S1793431113500012

Maddison, J.R., Panourgias, I., Farrell, PE.: libsupermesh version 1.0. Technical report, University of Edinburgh (2017)

Mayne, D.A., Usmani, A.S., Crapper, M.: An Adaptive Finite Element Solution for Cohesive Sediment Transport. In: Proceedings in Marine Science, vol 5, Elsevier, pp 627–641, https://doi.org/10.1016/S1568-2692(02)80044-4 (2002)

McManus, T.M., Percival, J.R., Yeager, B.A., Barral, N., Gorman, G.J., Piggott, M.D.: Moving mesh methods in Fluidity and Firedrake. Technical Report July, Imperial College London, (2017) https://doi.org/10.13140/RG.2.2.27670.24648

McRae, A.T.T., Cotter, C.J., Budd, C.J.: Optimal-transport-based mesh adaptivity on the plane and sphere using finite elements. SIAM J. Sci. Comput. **40**(2), 1121–1148 (2018). https://doi.org/10.1137/16M1109515

Papanicolaou, A.T.N., Elhakeem, M., Krallis, G., Prakash, S., Edinger, J.: Sediment transport modeling review - current and future developments. J. Hydraul. Eng. **134**(1), 1–14 (2008)

Piggott, M.D., Pain, C.C., Gorman, G.J., Power, P.W., Goddard, A.J.H.: h, r, and hr adaptivity with applications in numerical ocean modelling. Ocean Model. **10**, 95–113 (2006). https://doi.org/10.1016/j.ocemod.2004.07.007

Piggott, M.D., Gorman, G.J., Pain, C.C., Allison, P.A., Candy, A.S., Martin, B.T., Wells, M.R.: A new computational framework for multi-scale ocean modelling based on adapting unstructured meshes. Int. J. Numer. Methods Fluids **56**(8), 1003–1015 (2008)

Piggott, M.D., Pain, C.C., Gorman, G.J., Marshall, D.P., Killworth, P.D.: Unstructured Adaptive Meshes for Ocean Modeling. In: Hasumi, H. (ed.) Hecht MW, pp 383–408. Ocean Modeling in an Eddying Regime, American Geophysical Union, USA (2008)

Piggott, M.D., Farrell, P.E., Wilson, C.R., Gorman, G.J., Pain, C.C.: Anisotropic mesh adaptivity for multi-scale ocean modelling. Philos. Trans. Royal Soc. A: Math., Phys. Eng. Sci. **367**(1907), 4591–4611 (2009). https://doi.org/10.1098/rsta.2009.0155

Rathgeber, F., Ham, D.A., Mitchell, L., Lange, M., Luporini, F., McRae, A.T., Bercea, G.T., Markall, G.R., Kelly, P.H.: Firedrake: automating the finite element method by composing abstractions. ACM Trans. Math. Softw. (TOMS) **43**(3), 1–27 (2016)

Tassi, P., Villaret, C.: Sisyphe v6.3 User's Manual. EDF R&D, Chatou, France (2014)

Thacker, W.C.: Some exact solutions to the nonlinear shallow-water wave equations. J. Fluid Mech. **107**, 499–508 (1981)

Van Rijn, L.C.: Storm Surge Barrier Oosterschelde-Computation of Siltation in Dredged Trenches: Semi-empirical Model for the Flow in Dredged Trenches. Deltares, Delft, The Netherlands (1980)

Van Rijn, L.C.: Sediment Transport, Part II: Suspended Load Transport. J. Hydraul. Eng. **110**(11), 1613–1641 (1984)

Villaret, C., Kopmann, R., Wyncoll, D., Riehme, J., Merkel, U., Naumann, U.: First-order uncertainty analysis using Algorithmic Differentiation of morphodynamic models. Comput. Geosci. **90**, 144–151 (2016)

Vouriot, C.V.M., Angeloudis, A., Kramer, S.C., Piggott, M.D.: Fate of large-scale vortices in idealized tidal lagoons. Environ. Fluid Mech. **19**, 329–348 (2019)

Wallwork, J.G., Barral, N., Kramer, S.C., Ham, D.A., Piggott, M.D.: Goal-oriented error estimation and mesh adaptation for shallow water modelling. Springer Nat. Appl. Sci. **2**, 1053–1063 (2020). https://doi.org/10.1007/s42452-020-2745-9

Weller, H., Browne, P., Budd, C., Cullen, M.: Mesh adaptation on the sphere using optimal transport and the numerical solution of a Monge-Ampère type equation. J. Comput. Phys. **308**, 102–123 (2016)

White, A.B., Jr.: On selection of equidistributing meshes for two-point boundary-value problems. SIAM J. Numer. Anal. **16**(3), 472–502 (1979)

Zhou, F., Chen, G., Huang, Y., Yang, J.Z., Feng, H.: An adaptive moving finite volume scheme for modeling flood inundation over dry and complex topography. Water Resour. Res. **49**(4), 1914–1928 (2013). https://doi.org/10.1002/wrcr.20179