

# Goal-Oriented Metric-Based Mesh Adaptive Tidal Farm Modelling

MARINE 2021

Joseph G. Wallwork<sup>1,\*</sup>, Lucas Mackie<sup>1</sup>, Stephan C. Kramer<sup>1</sup>, Nicolas Barral<sup>2,3</sup>,  
Athanasios Angeloudis<sup>4</sup> and Matthew D. Piggott<sup>1</sup>

<sup>1</sup>Department of Earth Science and Engineering, Imperial College London, London, SW7 2AZ, U.K.

<sup>2</sup>University of Bordeaux, CNRS, Bordeaux INP, IMB, UMR 5251, 33400 Talence, France.

<sup>3</sup>INRIA, IMB, UMR 5251, 33400 Talence, France.

<sup>4</sup>School of Engineering, Institute for Infrastructure & Environment, University of Edinburgh,  
Edinburgh, EH9 3FB, U.K.

\*Corresponding author: Joseph G. Wallwork, j.wallwork16@imperial.ac.uk

## ABSTRACT

The modelling of a tidal array farm is an inherently multi-scale endeavour. It requires the simultaneous resolution of tidal processes across tens or hundreds of kilometres of coastal ocean (including estuaries, or even entire seas), the hydrodynamics in the neighbourhood of the farm (hundreds of metres), the wakes of individual turbines (metres, or tens of metres) and device hydrodynamics (sub-metre). As such, the construction of an accurate, computationally efficient numerical model requires careful consideration of the underlying discretisation.

In this paper, we apply time-dependent mesh adaptation techniques based on the Riemannian metric framework to an idealised tidal array and assess the quality of the resulting approximations. Whilst classical hierarchical mesh adaptation methods modify mesh element/cell size in order to improve resolution locally, the metric-based approach also allows for control of element shape and orientation, which can be especially advantageous for advection-dominated problems. Metrics are normalised in such a way that the resulting discretisation is multi-scale in both space and time as per Alauzet and Olivier (2010). Typically, metrics are constructed from recovered derivatives of solution fields, such as fluid vorticity. Alternatively, metrics may be derived from goal-oriented error estimates, enabling accurate estimation of a diagnostic quantity of interest (QoI). In the context of tidal farm modelling, one clear QoI is the power output. Building upon the idealised steady-state test case considered in Wallwork et al. (2020), which represents turbines using a drag parametrisation in a depth-averaged shallow water model, we demonstrate here that goal-oriented mesh adaptation can be used to obtain an accurate approximation of tidal farm power output using relatively few overall degrees of freedom.

**Keywords:** Riemannian metric; mesh adaptation; adjoint methods; tidal power; Thetis.

## NOMENCLATURE

$b$	Bathymetry [m]
$c_b$	Background drag [-]
$c_t$	Turbine drag [-]
$C_{\mathcal{T}}$	Thrust coefficient associated with turbine $\mathcal{T}$ [-]
$F_{\mathcal{T}}$	Drag force exerted by turbine $\mathcal{T}$ [N]
$g$	Gravitational acceleration [ $9.81 \text{ m s}^{-2}$ ]
$\text{Re}_K$	Mesh Reynolds number associated with element $K$ [-]
$T_{\text{tide}}$	Tidal cycle [s]
$\mathbf{u}$	Horizontal fluid velocity [ $\text{m s}^{-1}, \text{m s}^{-1}$ ]
$U$	Characteristic velocity [ $\text{m s}^{-1}$ ]
$\eta$	Free surface elevation [m]
$\nu$	Horizontal kinematic viscosity [ $\text{m}^2 \text{ s}^{-1}$ ]
$\rho$	Fluid density [ $\text{kg m}^{-3}$ ]
DG	Discontinuous Galerkin
DOFs	Degrees of Freedom
DWR	Dual Weighted Residual
SIPG	Symmetric Interior Penalty Galerkin
SPD	Symmetric Positive-Definite
QoI	Quantity of Interest

## 1 INTRODUCTION

Goal-oriented error estimation and mesh adaptation techniques have the primary objective of minimising the error accrued when evaluating a diagnostic *quantity of interest* (*QoI*). In the context of modelling a tidal farm – a collection of tidal stream turbines designed to extract energy from high velocity tidal currents – this could be the total energy output of the farm, the associated profit or a measure of its environmental impact. Given a particular choice, error estimators such as the dual weighted residual (Becker & Rannacher, 2001) can be used to deduce error indicator fields conveying how much error may be attributed to different components of the discretisation (such as particular finite elements and timesteps). These error indicators ‘localise’ the error contributions and can be used to drive mesh adaptation routines which construct new discretisations better representing the QoI.

In this paper, error indicator data is interpreted by a hybrid *hr*-adaptive mesh adaptation toolkit, *Pragmatic* (Barral et al., 2016), in terms of the Riemannian metric framework. This ‘optimal mesh model’, first introduced in George et al. (1991), defines a Riemannian metric space based on error indicator data and updates the mesh so that its elements are ‘unit’ when viewed in that space, i.e. they have edges of (near) uniform length, and consequently are appropriately multi-scale in physical space. For details on metric-based mesh adaptation, see Loseille and Alauzet (2011a, 2011b), Pain et al. (2001), and Piggott et al. (2009).

The tidal farm modelling framework used in this paper parametrises turbines as patches of increased bottom friction in a depth-averaged 2D shallow water model. This model was chosen because it allows large, multi-scale coastal ocean modelling problems to be represented at a greatly reduced computational cost, compared with a 3D model with the same resolution in the horizontal direction. This is applicable subject to some key assumptions, such as limited vertical accelerations, well-mixed water column and relatively negligible vertical dimensions compared to the horizontal. Goal-oriented metric-based mesh adaptation was previously applied to such a model in the steady-state case in Wallwork et al. (2020). It was found to lead to a reduction in error accrued when estimating array power output, for the same number of degrees of freedom. This paper extends the previous work to the time-dependent, tidally varying case, illustrating how goal-oriented adaptation can be used to

resolve the varying multi-scale hydrodynamics within and around an idealised tidal array, assuming that the 2D hydrodynamics are most important.

The organisation of this paper is as follows. Section 2 describes the depth-averaged tidal turbine modelling framework, including details on the model parameters and discretisation used. Section 3 describes a goal-oriented metric-based mesh adaptation method, which is framed around minimising the error accrued when evaluating the energy output of a tidal farm. The mesh adaptation method of Section 3 is applied to an idealised tidal turbine array test case in Section 4. To the best of the authors’ knowledge, this presents the first application of goal-oriented mesh adaptation methods to a time-dependent tidal farm modelling problem. Finally, conclusions are drawn and future work is proposed in Section 5.

## 2 TIDAL FARM MODELLING

### 2.1 Shallow Water Equations

In this paper, coastal hydrodynamics are modelled using the time-dependent, non-rotational, nonlinear shallow water equations in non-conservative form:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + g \nabla \eta + \frac{c_t + c_b}{H} \|\mathbf{u}\| \mathbf{u} = \nabla \cdot (\nu \nabla \mathbf{u}), \quad \frac{\partial \eta}{\partial t} + \nabla \cdot (H \mathbf{u}) = 0. \quad (1)$$

The prognostic variable tuple  $\mathbf{q} := (\mathbf{u}, \eta)$  is comprised of (depth averaged) fluid velocity,  $\mathbf{u}$ , and free surface elevation,  $\eta$ , and is defined on a two-dimensional spatial domain,  $\Omega \subset \mathbb{R}^2$ , and some time interval. The total water depth,  $H = \eta + b$ , arises from a bathymetry field,  $b$ , which is fixed in time. Bottom friction is represented in the model using a quadratic drag parametrisation. The dimensionless drag coefficients  $c_b$  and  $c_t$  convey the background drag and drag due to the turbines, respectively, the latter of which is described in Subsection 2.5. Numerical experiments in this paper assume a constant background drag,  $c_b = 0.0025$ .

### 2.2 Discretisation

Given a mesh of the domain, spatial discretisation of (1) is achieved using the  $\mathbb{P}1_{DG} - \mathbb{P}1_{DG}$  equal order DG discretisation due to the *Thetis* coastal ocean model (Kärnä et al., 2017). *Thetis* uses the SIPG method described in Hillewaert (2013) for the purposes of representing viscosity. Whilst *Thetis* can be run on triangular or quadrilateral meshes, we opt to use triangular elements in this paper, since the mesh adaptation toolkit we use assumes simplicial meshes. *Thetis* is based on the *Firedrake* Python finite element package (Rathgeber et al., 2016), which automatically generates C code and uses *PETSc* (Balay et al., 2019) to solve linear and nonlinear systems.

Meshes are denoted by the symbol  $\mathcal{H}$  in this work and are interpreted as finite collections of elements,  $K \subset \Omega$ . Time integration is performed using the Crank-Nicolson method with implicitness  $\theta = 0.5$ .

### 2.3 Viscosity

Whilst viscosity is a physical property of a fluid, it also provides a tool to be deployed by the numerical modeller. Treating it consistently with the eddy viscosity approximation, artificially increasing viscosity smooths out turbulent effects and – when used appropriately – can improve the stability of a numerical scheme. By smoothing out turbulent effects, the flow becomes increasingly laminar. This is undesirable in applications which aim to capture vortices in the wake of tidal turbines, since such features are inherently turbulent and therefore cannot be captured without an appropriately small viscosity.

We address the above concerns by adopting a mesh-dependent viscosity – one which is sufficiently small to capture vortices in the wake of turbines, yet sufficiently large elsewhere to ensure stability of the spatial discretisation. Our specification is made via the mesh Reynolds number,  $\text{Re}_K := h_K U / \nu$ , where  $U$  is a characteristic fluid speed and  $h_K$  is the circumradius of an isotropic mesh element  $K \in \mathcal{H}$ . Given a small viscosity,  $\nu_{\text{target}}$ , which we would like to impose near the turbines and a maximum tolerated mesh Reynolds number,  $\text{Re}_{\text{max}}$ , we set

$$\nu_K := \begin{cases} \frac{h_K U}{\text{Re}_{\text{max}}} & \text{Re}_K > \text{Re}_{\text{max}} \\ \nu_{\text{target}} & \text{otherwise} \end{cases}, \quad K \in \mathcal{H}. \quad (2)$$

This element-wise quantity is projected into (vertex-wise)  $\mathbb{P}1$  space, which has an additional smoothing effect that can be beneficial for the shallow water solver. For simplicity, the characteristic speed is chosen to be constant in both time and space in this work, providing an upper bound. Consequently, the mesh Reynolds number is overestimated. This is preferable to underestimation because it puts more emphasis on stability, rather than vortex capture.

## 2.4 Tides

Tidal forcings appear as boundary conditions for the free surface elevation on forced boundary segments  $\Gamma_F \subset \partial\Omega$ . (Impermeable) free-slip conditions are applied on the unforced boundaries,  $\Gamma_{\text{freeslip}} = \partial\Omega \setminus \Gamma_F$ . The use of free-slip conditions is a rather simplistic approach to modelling the dynamics on any coast other than tall cliffs. However, they are sufficient for the purposes of the numerical experiments presented in Section 4. Taken together, we have boundary conditions,

$$\eta = \eta_F \quad \text{on} \quad \Gamma_F, \quad \mathbf{u} \cdot \hat{\mathbf{n}} = 0 \quad \text{on} \quad \Gamma_{\text{freeslip}}, \quad (3)$$

where  $(\mathbf{u}, \eta)$  now denote the fluid velocity and elevation values on the domain boundary. Since we are using a DG discretisation, these values are weakly enforced.

## 2.5 Tidal Turbine Parametrisation

Suppose we have a tidal farm,  $\mathcal{F}$ , interpreted as a finite set of turbines, with each turbine  $\mathcal{T} \in \mathcal{F}$  interpreted as a subset of the spatial domain. Each turbine is meshed explicitly, so that  $\mathcal{T} \subset \mathcal{H}$ . The initial (un-adapted) mesh for the experimental setup presented in Section 4 includes 6 m fine resolution in order to capture the turbine footprints.

With the representation above, the turbines may be modelled using a drag parametrisation which acts within its ‘footprint’. The drag coefficient  $c_t$  in (1) depends on the thrust coefficient  $C_{\mathcal{T}} = C_{\mathcal{T}}(\mathbf{u})$  in each turbine footprint,  $\mathcal{T}$ . The thrust coefficient relates the drag force exerted on the flow by turbine  $\mathcal{T}$  to the area it sweeps in the vertical,  $A^{\text{swept}}$ , via

$$F_{\mathcal{T}}(\mathbf{u}) = \frac{1}{2} \rho C_{\mathcal{T}} A^{\text{swept}} \|\mathbf{u}\| \mathbf{u}. \quad (4)$$

In general, the thrust coefficient is a function of fluid velocity, which may depend on a cut-in speed, for example. However, it is assumed to be constant across all turbines within a farm for the purposes of this paper. The turbine drag coefficient,  $c_t$ , is calculated by scaling the thrust according to the ratio of area swept in the vertical and footprint area,  $A^{\text{footprint}}$ , since it is the area swept by the turbine blades which determines the energy extraction:

$$c_t = \sum_{\mathcal{T} \in \mathcal{F}} \frac{A^{\text{swept}} C_{\mathcal{T}}}{2 A^{\text{footprint}}} \mathbb{1}_{\mathcal{T}}, \quad (5)$$

where  $\mathbb{1}_{\mathcal{T}}$  is an indicator function which is unity within the footprint of turbine  $\mathcal{T}$  and zero elsewhere. It should be remarked that the thrust coefficient as described is based on an upstream velocity. In

order to account for the fact that the velocity at the turbine is depth-averaged velocity, we use the thrust coefficient correction recommended in Kramer and Piggott (2016).

Given that the turbines are deployed in water of density  $\rho > 0$  (assumed to be the constant  $\rho = 1030 \text{ kg m}^{-3}$  herein), a proxy for the energy output of the tidal farm is given by

$$E(\mathbf{u}, \eta) := \int_{T_{\text{start}}}^{T_{\text{end}}} \int_{\Omega} \rho c_t \|\mathbf{u}(\mathbf{x}, t)\|^3 dx dt = \sum_{\mathcal{T} \in \mathcal{F}} \int_{T_{\text{start}}}^{T_{\text{end}}} \int_{\mathcal{T}} \rho \frac{A^{\text{swept}} C_{\mathcal{T}}}{2A^{\text{footprint}}} \|\mathbf{u}(\mathbf{x}, t)\|^3 dx dt, \quad (6)$$

where  $[T_{\text{start}}, T_{\text{end}}]$  is the time interval we seek to accurately capture energy output over. The integrands of the time integrals on the RHS of (6) (i.e. the space integrals) provide proxies for the power output of the individual turbines.

### 3 GOAL-ORIENTED MESH ADAPTATION

#### 3.1 Error Estimation

The application of goal-oriented mesh adaptation presented in this paper seeks to accurately approximate the energy output of the farm. That is, we seek to minimise  $\delta E := E(\mathbf{q}) - E(\mathbf{q}_h)$ , where  $\mathbf{q} = (\mathbf{u}, \eta)$  is the exact solution of the shallow water problem and  $\mathbf{q}_h = (\mathbf{u}_h, \eta_h)$  is a finite element approximation thereof.

Given the definition of energy output in (6), it is likely that mesh adaptation based on a metric derived from the fluid speed would go some way to provide an accurate energy output approximation. For example, an anisotropic metric may readily be derived from a recovered Hessian of this field. However, it is possible that such a strategy will involve applying increased refinement in regions of the domain far from the tidal farm, where the (curvature of) fluid velocity happens to be large, but that flow and its structure has little or no impact upon the energy output. For example, dynamics downstream of the farm and/or individual turbines if that flow does not impinge on further downstream turbines. This implies a computational inefficiency which we would prefer to avoid.

*Goal-oriented error estimation* provides a means of generating metrics which admit accurate energy output estimates, without wasting resolution unnecessarily on regions of the domain (and intervals in time) where the hydrodynamics are less impactful upon it. It does this using the *adjoint problem* associated with the prognostic equation set – in this case the shallow water equations (1). The adjoint problem depends on the QoI – in this case the energy output – and conveys how its sensitivities propagate across the space-time domain. Let  $\mathbf{q}^*$  denote the solution of the associated adjoint problem and  $\mathbf{q}_h^*$  denote solution of the adjoint of the discrete shallow water model. Note that  $\mathbf{q}_h^*$  lives in the same DG function space as  $\mathbf{q}_h$ .

For the purposes of the following presentation, we restrict attention to the finite element problem solved in a single timestep. The shallow water problem may then be expressed weakly in the form

$$r(\mathbf{q}_h, \mathbf{w}) = 0, \quad \forall \mathbf{w} \in V, \quad (7)$$

where  $r(\cdot, \cdot)$  is referred to as the *weak residual* on the single timestep. Due to the pioneering work of Becker and Rannacher (2001) the energy output error may be approximated to first order by the error estimate

$$\delta E \approx |r(\mathbf{q}_h, \mathbf{q}^* - \mathbf{q}_h^*)|, \quad (8)$$

which is known as the *dual weighted residual (DWR)*. That is, the error accrued in evaluating the energy output may be approximated by a known expression involving the ‘forward’ finite element approximation,  $\mathbf{q}_h$ , and the error in the corresponding ‘adjoint’ approximation,  $\mathbf{q}_h^*$ .

### 3.2 Error Indication

In order to apply mesh adaptation, we need to first deduce local contributions to the global quantity (8), so that a mesh adaptation method can make local modifications to the discretisation, as appropriate. The simplest means of doing so is to consider element-wise contributions. We refer to Wallwork et al. (2020) for details on how this can be performed in the case of shallow water tidal farm model.

Without further treatment, the adjoint error term  $\mathbf{e}^* := \mathbf{q}^* - \mathbf{q}_h^*$  cannot be evaluated directly, since it contains the exact adjoint solution tuple. One approach is to substitute  $\mathbf{q}^*$  with a higher order approximation, such as may be obtained by solving the adjoint problem in a (globally or locally) enriched finite element space, or through application of a technique such as superconvergent patch recovery. An alternative approach is proposed in Becker and Rannacher (2001), which has a much lower computational cost than such methods in general. Its application to time-dependent tracer transport problems was considered in Wallwork et al. (2021). The so-called *difference quotient* method reformulates the error indicator on each element, rather than attempting to approximate the true adjoint solution. The resulting error indicators take the form

$$\begin{aligned} \mathcal{E}_K := & \left( \|\Psi_u(u_h)\|_{L^2(K)} + h_K^{-\frac{1}{2}} \|\psi_u(u_h)\|_{L^2(\partial K)} \right) \|\nabla^2 u_h^*\|_{L^2(K)} \\ & + \left( \|\Psi_v(v_h)\|_{L^2(K)} + h_K^{-\frac{1}{2}} \|\psi_v(v_h)\|_{L^2(\partial K)} \right) \|\nabla^2 v_h^*\|_{L^2(K)} \\ & + \left( \|\Psi_\eta(\eta_h)\|_{L^2(K)} + h_K^{-\frac{1}{2}} \|\psi_\eta(\eta_h)\|_{L^2(\partial K)} \right) \|\nabla^2 \eta_h^*\|_{L^2(K)}, \end{aligned} \quad (9)$$

where  $(u_h, v_h)$  stand for the  $x$ - and  $y$ -components of  $\mathbf{u}_h$ , respectively, and  $\partial K$  denotes the edge set of an element  $K \in \mathcal{H}$  with circumradius  $h_K$ . Here  $(\Psi_u, \Psi_v, \Psi_\eta)$  denote components of the strong residual and  $(\psi_u, \psi_v, \psi_\eta)$  are flux terms due to the integration by parts and the DG discretisation. In practice, the Laplacians of the adjoint solution components are constructed using a recovery method. In this work, the gradient is obtained using an  $L^2$  projection, the finite element derivative of which is an element-wise quantity. Note that the weighting term evaluates flux terms in the *forward* problem at the finite element *adjoint* solution.

There are a number of caveats associated with error indicator (9). Firstly, its sum over all elements consistently overestimates the true QoI error, with the overestimation increasing with mesh size. As such, it is not useful as an error estimator. Nevertheless, its localisation to individual elements provides useful information for mesh adaptation. This is because the metrics constructed from such error estimators (as detailed in the following subsection) are normalised in space, meaning that global scale factors are unimportant. Secondly, two terms which act to symmetrise the viscosity operator in the SIPG method must be dropped when constructing an error indicator of the form (9) from the  $\mathbb{P}1_{DG} - \mathbb{P}1_{DG}$  formulation. Finally, the estimator uses results which have as yet only been proved for elliptic problems. Despite these drawbacks, the results in Section 4 show that difference quotients can still give promising results when applied to the hyperbolic problem investigated here.

### 3.3 The Metric-Based Framework

Given an error indicator field, the metric-based mesh adaptation framework uses the concept of a *Riemannian metric* to dictate the transformations which should be applied to a mesh. In the two-dimensional case considered here a Riemannian metric  $\mathcal{M} = \{\underline{\mathbf{M}}(\mathbf{x})\}_{\mathbf{x} \in \Omega}$  is a tensor field corresponding to a SPD  $2 \times 2$  matrix at each point of the domain. Since the metric is defined point-wise, it gives rise to a Riemannian metric space,  $(\mathbb{R}^2, \mathcal{M})$ .

The core idea of the metric-based mesh adaptation method is to use this Riemannian metric space within the mesher. Adaptation operations are applied to the mesh until it becomes a *unit mesh* when viewed in the Riemannian metric space. That is, each of its elements have edges of unit length when

mapped into the metric space. In general, it is not possible to tessellate a 2D domain with equilateral triangles, so this definition is relaxed slightly to ensure a *quasi-unit mesh*. Note that, whilst the transformed mesh is quite regular, its representation in the (Euclidean) physical space can be rather distorted.

One of the great advantages of metric-based mesh adaptation is that it allows for control of element shape and orientation, as well as size. It is possible to define anisotropic goal-oriented metrics and perform anisotropic goal-oriented mesh adaptation based on scalar error indicators such as (9). For example, the a posteriori anisotropic metric due to Carpio et al. (2013) was applied to a steady-state tidal farm modelling problem in Wallwork et al. (2020). However, anisotropic metrics are not applied to time-dependent problems in this paper, for brevity; they will be used in future work. Instead, the isotropic case is assumed, whereby the metric is the identity matrix scaled by  $\mathcal{E}_K$  in each element. The metric-based mesh adaptation toolkit used herein expects metrics defined in  $\mathbb{P}1$  space (i.e. vertex-wise), so an additional projection step is applied.

For further details on metric-based mesh adaptation applied to steady-state problems, we refer to Loseille and Alauzet (2011a, 2011b) and Pain et al. (2001).

### 3.4 Time-Dependent Case

An additional implementation detail in the time-dependent goal-oriented case is the requirement to solve adjoint equations over a sequence of meshes. We take a fixed-point iteration approach, which involves partitioning the simulated time period into  $N$  subintervals,

$$[T_{\text{start}}, T_{\text{end}}] = \cup_{i=1}^N [t^{(i-1)}, t^{(i)}], \quad T_{\text{start}} = t^{(0)} < t^{(1)} < \dots < t^{(N-1)} < t^{(N)} = T_{\text{end}}. \quad (10)$$

A different mesh is associated with each subinterval, and mesh-to-mesh interpolation is applied to transfer data from one subinterval to the next. The fixed-point iteration approach taken in this paper is based on that outlined in Belme et al. (2012); we refer to that work for details on the algorithm. The main difference in the paper presented here is that it uses goal-oriented metrics derived from an a posteriori error result (8) rather than an a priori one. A notable feature of both implementations is that they use space-time normalisation methods, which allow for the discretisation to be multi-scale in both space and time. In particular, DOFs are distributed across the temporal domain so as to improve the QoI accuracy. This means that the meshes associated with some subintervals may be much finer or coarser than others.

One of the key tunable parameters for metric-based mesh adaptation is the *metric complexity*. For steady-state problems, it is the analogue of the mesh vertex count in (continuous) metric space. For time-dependent problems, it is the continuous analogue of the sum of all mesh vertex counts over all timesteps. By increasing the target metric complexity, we allow for heightened overall mesh resolution. Whilst this typically implies a heightened computational cost, it should also imply a reduction in QoI error (provided that the metric is chosen appropriately).

### 3.5 Software for Mesh Adaptation

In this work, adjoint equations associated with (1) are formulated and solved using the *discrete adjoint* approach. This approach amounts to differentiating through the discretised model, as opposed to the ‘continuous’ model (as in the *continuous adjoint* approach). The discrete adjoint method is made available in Firedrake and Thetis by the *dolfin-adjoint* package (Farrell et al., 2013).

The *Pragmatic* (Barral et al., 2016) anisotropic mesh adaptation toolkit is used to take an input mesh and a metric defined upon it and generate a new mesh. In 2D mode, Pragmatic applies four different types of mesh transformation: vertex insertion, vertex removal, edge swapping and local Laplacian smoothing. The first three modify the mesh topology, whilst the fourth does not.

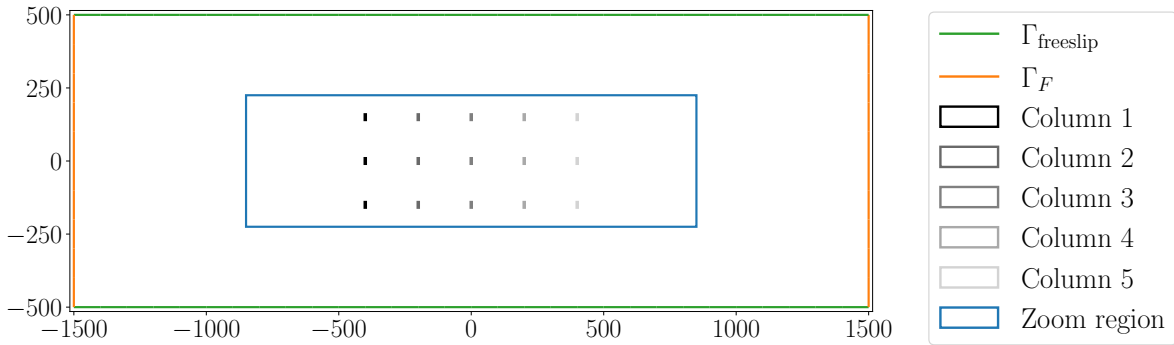


Figure 1: Diagram showing the domain for the tidal array test case, including its (forced and free-slip) boundary segments, five columns of turbines and a zoom region.

Mesh-to-mesh solution transfer is achieved using the conservative interpolation operator provided by *libsupermesh* (Farrell & Maddison, 2011; Maddison et al., 2016).

## 4 NUMERICAL EXPERIMENTS

### 4.1 Idealised Tidal Array

The numerical experiments presented in Wallwork et al. (2020) involve a simple idealised array of two turbines. In those experiments, the problem was solved to steady state, meaning there was no tidal forcing and the flow was actually laminar. However, the setup is useful for illustrating the impact that the locations of the turbines within the array can have upon the total power output. In that paper, the power output was found to be 15% lower when the turbines were aligned in the flow compared with when they were offset by one turbine diameter in opposite directions orthogonal to the background flow. In addition, it was illustrated that the application of goal-oriented metric-based mesh adaptation can lead to more accurate approximations of array power output for similar numbers of DOFs.

In this paper, we consider the extension to the time-dependent case and a larger tidal farm. The test case was originally proposed in Divett et al. (2013) and consists of an array of fifteen turbines arranged in three rows and five columns. That work also demonstrated the importance of array configuration in tidal farm design; a staggered turbine layout was found to extract 54% more energy from the flow than a centred, aligned configuration.<sup>1</sup> This is because a staggered array gives the turbine wakes more opportunity to recover before interacting with another downstream turbine, as well as turbines being able to exploit accelerated bypass flow. Tidal turbines act to extract energy from the flow, so their wakes are effectively momentum deficits. For the version of the tidal array test case underpinning the numerical experiments in this paper, we consider the centred, aligned configuration alone, which implies strongly nonlinear interactions between turbines and wakes.

An initial mesh of the rectangular domain  $\Omega = [-1500, 1500] \times [-500, 500]$  with increased mesh resolution in the array region was generated using *gms*h (Geuzaine & Remacle, 2009). Figure 1 shows the spatial domain, including the forced and free-slip boundary segments, the five columns of turbines and a zoom region used in subsequent plots.

A simple sinusoidal tidal forcing at the Western and Eastern boundaries is used to drive the hydrodynamics. The two forcings are exactly out of phase:

$$\eta_F(x, y, t) := \begin{cases} 0.5 \cos(\omega t) & x = -1500 \\ 0.5 \cos(\omega t + \pi) & x = 1500 \end{cases}, \quad \omega = \frac{2\pi}{T_{\text{tide}}}, \quad T_{\text{tide}} = 1.24 \text{ h}. \quad (11)$$

<sup>1</sup>Note that, since both this study and the steady-state one mentioned above involve idealised test cases, the differences in power/energy output due to array configuration may not be representative of realistic tidal farms.



The resulting boundary conditions are given by substitution of (11) in (3). Note that the tidal period is 10% of the M2 tidal constituent; it was reduced in (Divett et al., 2013) in order to emphasise vorticity.

Figure 2 shows snapshots of the fluid velocity field over a range of time levels over the single tidal cycle  $[0, T_{\text{tide}}]$ , as computed on a high resolution mesh with 184,896 elements and 554,688 DOFs in the  $\mathbb{P}1_{DG} - \mathbb{P}1_{DG}$  function space. Starting from a near zero initial velocity and a linear elevation field which satisfies the boundary conditions, the flow speed increases Westward for the first half of the interval, before reversing to flow Eastward. At time levels  $0.2 T_{\text{tide}}$  and  $0.4 T_{\text{tide}}$ , the West-most column of turbines experiences relatively laminar flow, whilst the flow around the other turbines is much more turbulent. At time levels  $0.6 T_{\text{tide}}$  and  $0.8 T_{\text{tide}}$ , it is the East-most column which experiences relatively laminar flow. The velocity field at  $t = T_{\text{tide}}$  is stored to disk, along with the corresponding elevation. These ‘spun-up’ hydrodynamics are used as initial conditions for subsequent mesh adaptive simulations on the interval  $[T_{\text{tide}}, 1.5 T_{\text{tide}}]$ .

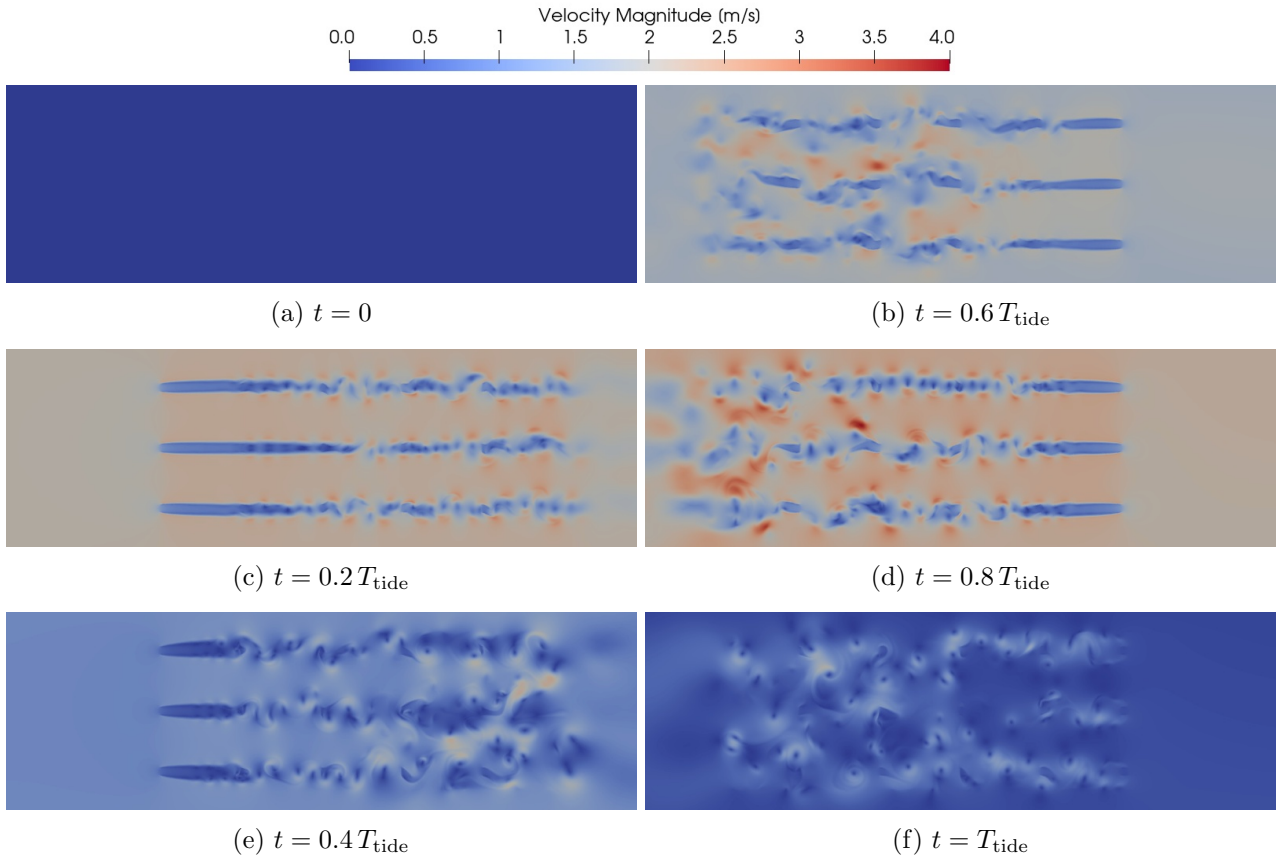


Figure 2: Snapshots of fluid velocity for a high resolution fixed mesh spin-up run for the tidal array test case. The finite element space has 554,688 DOFs and the underlying mesh of the  $3 \text{ km} \times 1 \text{ km}$  domain has 184,896 elements and a maximum aspect ratio of 2.0.

## 4.2 Mesh Adaptation Based on Energy Output

In this subsection, isotropic goal-oriented metrics are applied to the tidal array test case, with the aim of minimising the error in energy output. The simulated interval  $[T_{\text{tide}}, 1.5 T_{\text{tide}}]$  is divided into 40 subintervals of equal length. A target metric complexity of 2,000,000 is imposed. Figure 3 shows snapshots of both fluid velocity and the underlying adapted mesh at a selection of time levels. Minimum and maximum metric magnitudes of 0.01 m and 100 m are imposed across the domain, with a maximum magnitude of 10 m within the turbine footprints.

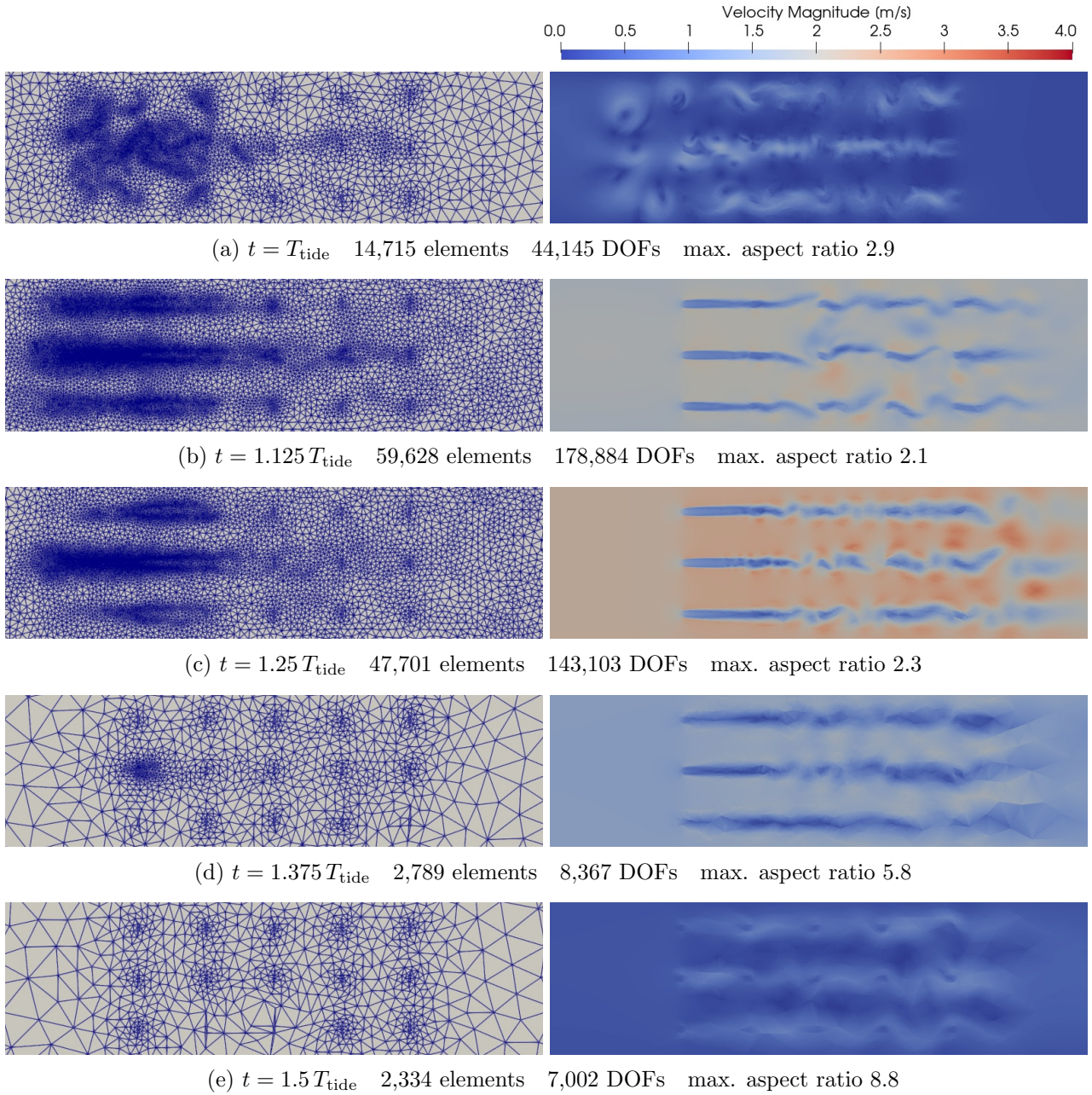


Figure 3: Snapshots of fluid velocity and the underlying mesh for an adaptive simulation of the tidal array test case over the time interval  $[T_{\text{tide}}, 1.5 T_{\text{tide}}]$ . Mesh adaptation is driven by an isotropic metric formulation based on difference quotient DWR indicators of the form (9).

At  $t = T_{\text{tide}}$ , the spun-up hydrodynamics are interpolated as an ‘initial condition’. Accordingly, the mesh associated with the first subinterval is adapted so that these hydrodynamics may be accurately captured. The goal-oriented metric does not attempt to capture the entire field, however: only the upstream hydrodynamics are held to be of importance; little mesh resolution is used surrounding the fifth (i.e. East-most) columns of turbines. The nonlinear interactions between turbines and the fact that turbines act to remove momentum from the flow means that the first (i.e. West-most) column of turbines extracts the most energy when the flow is Eastward. This can be seen in Subfigure 4a, which shows the power output of each turbine column under an extension of the high resolution fixed mesh simulation of Subsection 4.1 for another half tidal cycle. Turbine columns to the East do occasionally extract similar amounts of energy, but they generally extract much less and the levels fluctuate due to the turbulence. Moreover, it is not the instantaneous power output values that are of interest, but the areas under the curves in Figure 4. That the darkest (West-most) curve has by far the largest area indicates why the mesh adaptation algorithm gives so much precedence to the first column. Given that the first column is by far the most important and that we have an advection-dominated problem, dynamics downstream of this column are held to be far less important than the dynamics surrounding and upstream of those turbines. Turbines in the second column generate the least power because the flow conditions in the wake of the first turbine column are close to laminar, meaning that they experience a significant momentum deficit for much of the simulation. That the first column generates the most power and the second column generates the least power is consistent with results reported in Divett et al. (2013).

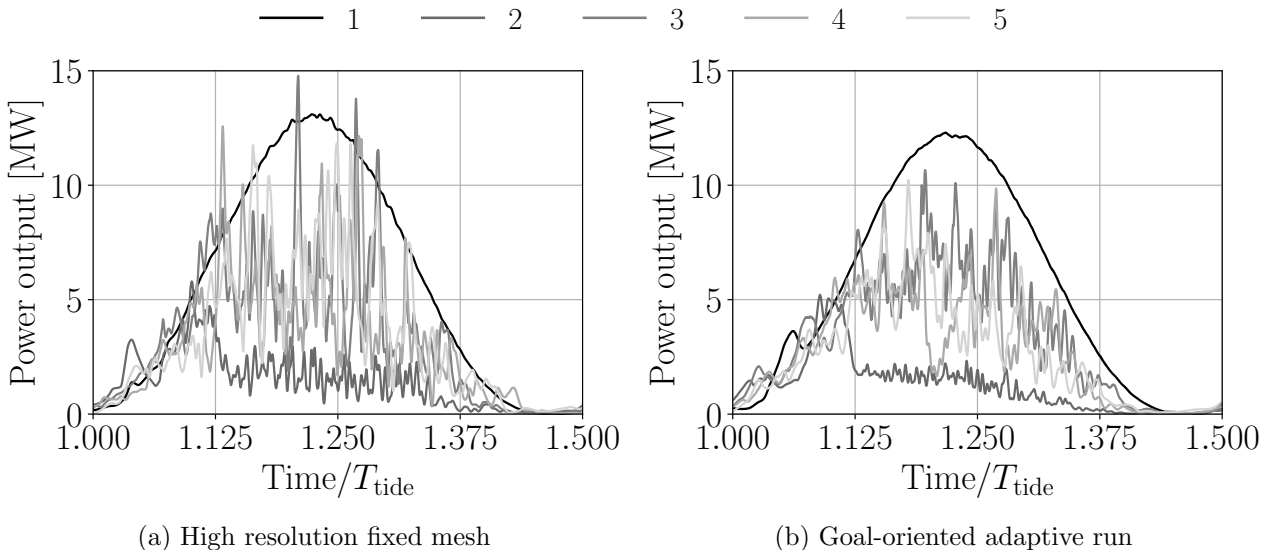


Figure 4: Power output as a function of time, separated by turbine column in the array. The values in Subfigure 4a were computed using the same high resolution fixed mesh as in Subsection 4.1, whereas those in Subfigure 4b are due to the isotropic goal-oriented mesh adaptation method used in this section.

Similar mesh structures as at  $t = T_{\text{tide}}$  are apparent in the snapshots at  $t = 1.125 T_{\text{tide}}$  and  $t = 1.25 T_{\text{tide}}$ . Interestingly, little mesh resolution is deployed for the purposes of capturing vortices. In fact, most of the mesh resolution is used to capture the relatively laminar flow conditions found West of the second turbine column.

The arrow of time implies that dynamics at a particular time are only important from then onwards. At the start of the simulation, the dynamics can potentially impact the entire solution trajectory, whereas towards the end of the simulation the impact is much more limited. This explains why so much more resolution is deployed at  $t = T_{\text{tide}}$  than at  $t = 1.5 T_{\text{tide}}$ , despite the velocity (and hence the contribution to the energy output) being near-zero in both cases.

Figure 4 suggests that the power curve is slightly out of phase with the tidal forcing, with the outputs of all columns becoming near-zero before the tide turns. This is likely due to the significant drag forces inherent in the tidal farm. As a consequence, the power outputs of all columns are significantly larger at  $t = 1.125 T_{\text{tide}}$  than at  $t = 1.375 T_{\text{tide}}$ . Combined with the above argument about the arrow of time, this explains why the mesh used at time  $t = 1.375 T_{\text{tide}}$  is so much coarser than that at  $t = 1.125 T_{\text{tide}}$ . At  $t = 1.375 T_{\text{tide}}$  and  $t = 1.5 T_{\text{tide}}$ , the pre-specified maximum metric magnitude is being imposed within the turbine footprints.

Consider now the columnar power output curves shown in Subfigure 4b, which are due to the goal-oriented mesh adaptation simulation. Despite the fact that even the adapted mesh with the highest element count stated (59,628 in Subfigure 3b) has less than a third of the high resolution fixed mesh element count, the power output curve for the first column agrees well with that shown in Subfigure 4a. The general trend of the second column’s power output curve is also well represented, even though it contributes the least. This is partly because its upstream conditions are well resolved and partly because its variability is fairly small. The power curves for the remaining turbine columns are not well captured. This is for a number of reasons. Firstly, the power curves of these columns are much more difficult to capture than the first two columns because of the turbulent conditions there, of which the high variability in those curves is a symptom. Secondly, the target complexity is relatively small for a problem of this size; if it were increased then it is likely that the mesh adaptation algorithm would deploy more DOFs for the purpose of capturing columns 3-5. Note that the variation in columns 3-5 is reduced in Subfigure 4b, compared with Subfigure 4a. This is likely due to numerical diffusion introduced by the coarse mesh resolution in that region.

To conclude this subsection, we propose further avenues for investigation based on the numerical experiments presented above. Firstly, goal-oriented mesh adaptation is only applied over a single flood tide. In future work, it would be interesting to see how the adaptation algorithm acts to deploy mesh resolution over a sequence of flood and ebb tides. It would also be interesting to see the different ways in which mesh resolution is deployed for alternative configurations of the same farm, such as those considered in Divett et al. (2013). It was revealed in that paper that, unlike the centred and aligned configuration considered here, the second column of a staggered array generates the *most* energy, due to accelerated bypass flow. As such, it seems reasonable to expect that the goal-oriented meshes would look quite different. Given that the  $x$ -component of the flow conditions is much more significant than the  $y$ -component, it is plausible that incorporating anisotropy into the metric would help to more effectively resolve the dynamics. Finally, the above results only consider one fixed mesh resolution level and one target metric complexity. Consideration of multiple values for each of these parameters would allow for more detailed convergence analysis.

## 5 CONCLUSIONS

In this paper, we have applied a goal-oriented metric-based mesh adaptation method to the problem of simulating time-dependent hydrodynamics in an idealised tidal farm test case. This is the first time goal-oriented mesh adaptation has been used in the context of time-dependent tidal farm modelling. In Section 3, an accurate energy output assessment of the farm is sought via goal-oriented error estimation techniques. Section 4 demonstrates how the goal-oriented adaptation algorithm deploys mesh resolution over the spatio-temporal domain in order for this aim to be achieved, whilst retaining relatively few degrees of freedom in the adapted meshes.

In future work, we will apply goal-oriented metric-based mesh adaptation framework to more complex problems such as proposed tidal power infrastructure projects, with spatially varying bathymetry data and realistic tidal forcings. In turn, the mesh adaptation capability can be used to accelerate design optimisation calculations (Culley et al., 2016; Funke et al., 2014).

## ACKNOWLEDGEMENTS

This work received funds from the Engineering and Physical Sciences Research Council (grant numbers EP/L016613/1 and EP/R029423/1).

## REFERENCES

- Alauzet, F., & Olivier, G. (2010). An  $L^p$ - $L^\infty$  space-time anisotropic mesh adaptation strategy for time dependent problems. *Proceedings of ECCOMAS CFD*.
- Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W. D., Karpeyev, D., Kaushik, D., Knepley, M. G., May, D. A., McInnes, L. C., Mills, R. T., Munson, T., Rupp, K., Sanan, P., ... Zhang, H. (2019). PETSc Web page. <https://www.mcs.anl.gov/petsc>
- Barral, N., Knepley, M. G., Lange, M., Piggott, M. D., & Gorman, G. J. (2016). Anisotropic mesh adaptation in Firedrake with PETSc DMPlex. *arXiv preprint arXiv:1610.09874*.
- Becker, R., & Rannacher, R. (2001). An optimal control approach to a posteriori error estimation in finite element methods. *Acta numerica*, 10, 1–102.
- Belme, A., Dervieux, A., & Alauzet, F. (2012). Time accurate anisotropic goal-oriented mesh adaptation for unsteady flows. *Journal of Computational Physics*, 231(19), 6323–6348.
- Carpio, J., Prieto, J. L., & Bermejo, R. (2013). Anisotropic “goal-oriented” mesh adaptivity for elliptic problems. *SIAM J Sci Comput*, 35(2), A861–A885.
- Culley, D., Funke, S., Kramer, S., & Piggott, M. (2016). Integration of cost modelling within the micro-siting design optimisation of tidal turbine arrays. *Renewable Energy*, 85, 215–227. <https://doi.org/10.1016/j.renene.2015.06.13>
- Divett, T., Vennell, R., & Stevens, C. (2013). Optimization of multiple turbine arrays in a channel with tidally reversing flow by numerical modelling with adaptive mesh. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1985), 20120251.
- Farrell, P. E., Ham, D. A., Funke, S. W., & Rognes, M. E. (2013). Automated derivation of the adjoint of high-level transient finite element programs. *SIAM Journal on Scientific Computing*, 35(4), C369–C393.
- Farrell, P. E., & Maddison, J. R. (2011). Conservative interpolation between volume meshes by local Galerkin projection. *Computer Methods in Applied Mechanics and Engineering*, 200(1-4), 89–100. <https://doi.org/10.1016/j.cma.2010.07.015>
- Funke, S. W., Farrell, P. E., & Piggott, M. D. (2014). Tidal turbine array optimisation using the adjoint approach. *Renewable Energy*, 63, 657–673.
- George, P. L., Hecht, F., & Vallet, M. G. (1991). Creation of internal points in Voronoi’s type method. control adaptation. *Advances in engineering software and workstations*, 13(5-6), 303–312.
- Geuzaine, C., & Remacle, J.-F. (2009). Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11), 1309–1331.
- Hillewaert, K. (2013). *Development of the discontinuous Galerkin method for high-resolution, large scale CFD and acoustics in industrial geometries* (Doctoral dissertation). Presses univ. de Louvain.
- Kärnä, T., Kramer, S. C., Mitchell, L., Ham, D. A., Piggott, M. D., & Baptista, A. M. (2017). Thetis coastal ocean model: Discontinuous Galerkin discretization for the three-dimensional hydrostatic equations. *arXiv preprint arXiv:1711.08552*.
- Kramer, S. C., & Piggott, M. D. (2016). A correction to the enhanced bottom drag parameterisation of tidal turbines. *Renewable Energy*, 92, 385–396.
- Loseille, A., & Alauzet, F. (2011a). Continuous mesh framework part I: Well-posed continuous interpolation error. *SIAM Journal on Numerical Analysis*, 49(1), 38–60.
- Loseille, A., & Alauzet, F. (2011b). Continuous mesh framework part II: Validations and applications. *SIAM Journal on Numerical Analysis*, 49(1), 61–86.

- Maddison, J. R., Farrell, P. E., & Panourglas, I. P. (2016). *Parallel supermeshing for multimesh modelling* (tech. rep. eCSE03-08). ARCHER. <https://www.archer.co.uk/community/eCSE/eCSE03-08/eCSE03-08-TechnicalReport.pdf>
- Pain, C. C., Umpleby, A. P., De Oliveira, C. R. E., & Goddard, A. J. H. (2001). Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations. *Computer Methods in Applied Mechanics and Engineering*, *190*(29-30), 3771–3796.
- Piggott, M. D., Farrell, P. E., Wilson, C. R., Gorman, G. J., & Pain, C. C. (2009). Anisotropic mesh adaptivity for multi-scale ocean modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *367*(1907), 4591–4611.
- Rathgeber, F., Ham, D. A., Mitchell, L., Lange, M., Luporini, F., McRae, A. T. T., Bercea, G.-T., Markall, G. R., & Kelly, P. H. J. (2016). Firedrake: Automating the finite element method by composing abstractions. *ACM Transactions on Mathematical Software (TOMS)*, *43*(3), 1–27.
- Wallwork, J. G., Barral, N., Ham, D. A., & Piggott, M. D. (2021). Goal-oriented error estimation and mesh adaptation for tracer transport modelling. *EarthArXiv Preprint*. <https://doi.org/10.31223/X56021>
- Wallwork, J. G., Barral, N., Kramer, S. C., Ham, D. A., & Piggott, M. D. (2020). Goal-oriented error estimation and mesh adaptation for shallow water modelling. *Springer Nature Applied Sciences*, *2*, 1053–1063. <https://doi.org/10.1007/s42452-020-2745-9>