Anisotropic metric-based mesh adaptation for ice flow modelling in Firedrake

Davor Dundovic[1], Joseph G. Wallwork[2], Stephan C. Kramer[3], Fabien Gillet-Chaulet[4], Regine Hock[1,5], and Matthew D. Piggott[3]

[1]Department of Geosciences, University of Oslo, Oslo, Norway
[2]Institute of Computing for Climate Science, University of Cambridge, Cambridge, UK
[3]Department of Earth Science and Engineering, Imperial College London, London, UK
[4]Univ. Grenoble Alpes, CNRS, INRAE, IRD, Grenoble INP, IGE, 38000 Grenoble, France
[5]Geophysical Institute, University of Alaska Fairbanks, Fairbanks, USA

Correspondence: Davor Dundovic (davord@uio.no)

Abstract.

Glaciological modelling is a computationally challenging task due to its high cost and complexity associated with large spatial- and long time-scale simulations. In this paper, we provide a comprehensive overview of state-of-the-art feature-based anisotropic mesh adaptation methods and demonstrate their effectiveness for time-dependent glaciological modelling using

5 the Python-based Firedrake finite element library. We introduce a novel hybrid time-dependent fixed-point mesh adaptation algorithm that generates a more optimal initial mesh sequence. The algorithm requires approximately 50% fewer iterations in order to reach mesh convergence, while still controlling spatial error and its temporal distribution. We demonstrate the effectiveness of anisotropic mesh adaptation and the novel fixed-point algorithm on a Marine Ice Sheet Model Intercomparison Project (MISMIP+) experiment. We show that we are able to achieve solution accuracy comparable to a uniform 0.5 km

10 resolution mesh simulations by using a sequence of adapted meshes with, on average, 10–30 times fewer vertices, depending on the sensor field used to drive mesh adaptation. Due primarily to the iterative nature of the mesh adaptation process employed, this translates in practice into a 3–6 times lower overall computational cost.

1 Introduction

Despite significant improvements in ice sheet models, the accurate prediction of the contribution of ice sheets to sea-level

15 rise in the 21st century remains uncertain (Aschwanden et al., 2021). A major uncertainty pertains to correctly modelling the dynamics of the grounding line, which is where the ice transitions from being grounded to floating in the ocean (Durand and Pattyn, 2015). Early studies demonstrated that having appropriate refinement of the computational mesh near the grounding line is vital for obtaining reliable numerical results (Schoof, 2007; Vieli and Payne, 2005). This has been further confirmed by the Marine Ice-Sheet Model Intercomparison (MISMIP) experiments, highlighting the mesh resolution's importance regardless

20 of the level of approximation in the force balance equations (Pattyn et al., 2012, 2013). While some studies have proposed methods to mitigate mesh dependency by introducing a smooth transition in the basal stress at the grounding line at the cost of

model accuracy (Leguy et al., 2014), the majority have focused on maintaining model accuracy while reducing computational costs. Different strategies have been developed to achieve the latter, including models with explicit mesh movement to solve for the grounding line position (Vieli and Payne, 2005; Moreno-Parada et al., 2023) or fixed-grid models incorporating sub-

25 grid parameterisation of the grounding line (Pattyn et al., 2006; Gladstone et al., 2012; Seroussi et al., 2014). However, these approaches rely on the hydrostatic criterion to define the grounding line position and thus cannot be readily applied to flow models that solve the complete force balance equations and treat the grounding line position as a contact problem (Gagliardini et al., 2016).

Alternative strategies rely on mesh adaptation methods that select finer mesh resolution in regions of interest (e.g., in the

30 vicinity of the grounding line) and coarser elsewhere to preserve numerical resources. These methods continue to be understudied and underutilised in the glaciological modelling community, while their popularity has steadily grown in wider computational fluid dynamics fields (Alauzet and Loseille, 2016). They are particularly popular in finite element and finite volume modelling due to the ability of these methods to utilise different types of meshes (structured, unstructured, or hybrid), and, through sophisticated error estimation and adaptation algorithms, to enhance computational efficiency and improve solution

35 accuracy. In the context of mesh adaptation, "solution accuracy" refers specifically to minimising discretisation error, i.e., the error that arises from approximating continuous mathematical equations by discrete numerical methods. Other sources of error are not addressed.

Different criteria have been used in the literature to define the regions of interest with desired mesh resolution. The distance to the grounding line, or alternatively distance from flotation, has been the most popular criterion to define areas warranting

40 smaller grid sizes (Durand et al., 2009; Goldberg et al., 2009; Gladstone et al., 2010; Cornford et al., 2013; Gudmundsson et al., 2012; Jouvet and Gräser, 2013; Dos Santos et al., 2019). However, such criteria do not explicitly control the solution error and neglect the error contribution of domain regions away from the grounding line, while refining other regions where this may not be necessary, hence making the simulation less efficient. Distance-from-flotation based criteria have been shown to be inferior to refinement criteria based on an error estimator (Dos Santos et al., 2019). Goldberg et al. (2009) uses the jumps

45 in strain rate at cell boundaries as a generic estimator of the numerical error to define areas that require finer mesh resolution. In their method, the total number of mesh nodes is constant and this generic estimator is not adapted to handle all flow regimes as it tends to increase the resolution at the shear margins, sometimes at the expense of the resolution at the grounding line. Dos Santos et al. (2019) implement a true *a posteriori* error approximation (i.e., the actual numerical error is approximated *after* the solution had been obtained), the Zienkiewich-Zhu error estimator, for the deviatoric stress tensor and ice thickness.

50 They find that using the distance to grounding line and the error estimator criteria separately and in combination produce similar results on a benchmark steady-state problem, but predict that the combined criterion would yield superior performance in simulations involving real ice sheets.

Existing ice sheet models utilising mesh adaptation also differ in the way they cope with grounding line movement in time-dependent problems. Some models use fixed grids, which requires the *a priori* refinement of areas where the grounding line is

55 susceptible to move through, while other models use mesh adaptation techniques that can involve moving meshes (Durand et al., 2009; Goldberg et al., 2009), mesh refinement and/or coarsening (Goldberg et al., 2009; Jouvet and Gräser, 2013; Cornford

et al., 2013) or the entire remeshing of the domain (e.g. the Úa finite-element ice-flow model described in Gudmundsson et al., 2012). However, none of these studies consider the temporal distribution of spatial error associated with time-dependent mesh adaptations.

60   Mesh adaptation strategies have become popular in computational fluid dynamics to capture complex multiscale phenomena such as shock wave propagation (e.g. Frey and Alauzet, 2005; Alauzet et al., 2007). The general aim is to control the accuracy of the solution by adapting the size and shape (or anisotropy) of individual mesh elements. A core challenge is in finding efficient and reliable 'estimators' of the numerical error which are used as the basis with which to define refinement criteria. Frey and Alauzet (2005) propose an estimator based on the interpolation error; the mesh size is then defined by an anisotropic

65   metric map which allows to equidistribute the error on the computational mesh. Such approaches are also known as *feature-based* approaches, since the constructed metric aims to capture the features of solution fields, or derived solution-dependent sensor fields, from which meshes are to be adapted. While not based on a true a posteriori approximation error, it has been found effective in practice to control the numerical error and offers flexibility in combining metrics obtained for different variables. Generation of anisotropic meshes allows us to capture strongly directional processes and geometries even more

70   efficiently. This is desirable in glacier shear margins, for example, where much finer resolution is required across the shear margin than along it. Such methods have been applied in finite element ice flow models to capture the dynamics of fast-flowing outlet glaciers using a metric defined from the observed surface velocity norm, while keeping overall computational costs low by coarsening other regions (Morlighem et al., 2010; Seddik et al., 2012; Gillet-Chaulet et al., 2012). However, those works do not consider the non-linear coupling of the solution and the underlying mesh, which requires an iterative mesh adaptation

75   procedure. Furthermore, they generate only a single mesh that is used throughout the simulation.

The purpose of this paper is twofold. Firstly, we aim to provide a self-contained description of anisotropic metric-based mesh adaptation methods suitable for ice sheet and glacier modelling. Secondly, we demonstrate and evaluate their ability to control solution accuracy while maintaining low computational cost in the context of grounding line dynamics modelling. We build on earlier applications of the method in glaciological modelling, but implement a novel adaptation procedure appropriate

80   for transient simulations, which consists of a spatio-temporal error analysis and a generation of multiple meshes that control the error.

2   Anisotropic metric-based mesh adaptation

Consider a partial differential equation (PDE) problem written in the generic operator form $F(u) = 0$ on a bounded domain Ω with an exact solution $u \in V$. Since the domain is bounded, we can prescribe a spatial discretisation in terms of a *mesh*, $\mathcal{H}$ of

85   Ω, which consists of a finite number of non-overlapping elements and a corresponding number of vertices N_v. The numerical solution $u_h \in V_h$, where $V_h \subset V$ is a discrete subspace of V, is obtained here via finite element methods, with an associated approximation error $u - u_h$. The cost of computing the solution u_h on $\mathcal{H}$ increases as N_v increases. However, while increasing N_v through the uniform refinement of a mesh would generally be expected to result in reduced errors, there is not a monotonic relationship between N_v and error more generally. Uniform refinement is also expected to give a sub-optimal reduction in

90 error as N_v increases. Mesh adaptivity seeks to address this: to more optimally and robustly link a decreasing error with an increasing N_v.

The general mesh adaptation problem considered in this paper can be formulated in an a priori way as follows:

$$\text{Find } \mathcal{H}_{\text{opt}} \text{ such that } \mathbf{E}(\mathcal{H}_{\text{opt}}) = \min_{\mathcal{H}} \mathbf{E}(\mathcal{H}), \tag{1}$$

where $\mathcal{H}_{\text{opt}}$ is the optimal mesh with desired number of vertices N_v that minimises some measure of the approximation error
95 $\mathbf{E}(\mathcal{H})$. The goal of mesh adaptation is therefore to achieve minimal approximation error for a given computational cost, which can be directly related to the number of vertices, N_v, of $\mathcal{H}$. In anisotropic mesh adaptation, the optimal mesh is found iteratively by changing the size, shape, and orientation of its elements.

2.1 Continuous metric framework: 2D overview

In order to formulate the mesh adaptation problem (1) such that it is well-posed and that it can be analysed using familiar
100 methods from variational calculus, Loseille and Alauzet (2011a) propose the *continuous metric framework*. The framework establishes a strong duality between discrete mesh elements and continuous mathematical objects stemming from Riemannian geometry. The key idea in Riemannian geometry is to define the way of measuring distances in a space, which, unlike in Euclidean geometry, may vary between points $\boldsymbol{x} \in \Omega$. The notion of measuring distances arises from the definition of a Riemannian metric $\mathbf{M}(\boldsymbol{x})$. A spatially-varying metric leads to local notions of geometric quantities such as length, volume
105 and angle. This is particularly useful in the context of anisotropic mesh adaptation, as it allows us to prescribe different sizes, orientations and directional stretchings of individual mesh elements in the domain. We refer the reader to Loseille and Alauzet (2011a, b) for a detailed description and analysis of the framework. However, in order to be self-contained, we illustrate how Riemannian metrics can be used to drive mesh adaptation on a two-dimensional domain $\Omega \subset \mathbb{R}^2$.

At each point $\boldsymbol{x}$ of the domain, the metric $\mathbf{M}$ is identified with a positive-definite matrix in $\mathbb{R}^{2 \times 2}$. It follows from the spectral
110 theorem that $\mathbf{M}$ admits a diagonalisation $\mathbf{M} = \mathbf{R} \operatorname{diag}(\lambda_1, \lambda_2) \mathbf{R}^\mathsf{T}$, where $\mathbf{R} = \mathbf{R}(\boldsymbol{x})$ is a matrix of orthogonal and normalized eigenvectors of $\mathbf{M}$, and λ_1, λ_2 are the corresponding eigenvalues. By defining metric density ρ and anisotropy quotients r_1, r_2 as

$$\rho = \sqrt{\det \mathbf{M}} = \sqrt{\lambda_1 \lambda_2}, \quad r_1 = \sqrt{\frac{\lambda_2}{\lambda_1}} = r_2^{-1}, \tag{2}$$

where anisotropy quotients measure the "stretching" strength in the directions of the eigenvectors in $\mathbf{R}$, the diagonal decom-
115 position can be written in a form that is more instructive in the context of mesh adaptation:

$$\mathbf{M} = \rho \mathbf{R} \begin{pmatrix} r_1^{-1} & 0 \\ 0 & r_2^{-1} \end{pmatrix} \mathbf{R}^\mathsf{T}. \tag{3}$$

In the continuous metric framework, a metric tensor $\mathbf{M}$ is a continuous analogue of a single mesh element. As such, this formulation makes it apparent that the metric $\mathbf{M}$ contains information on the size of the mesh element (through density ρ), their shape (through quotients r_1, r_2), and their orientation (through eigenvectors in $\mathbf{R}$). In this way, the metric-based approach
120 fully supports anisotropic mesh adaptation.
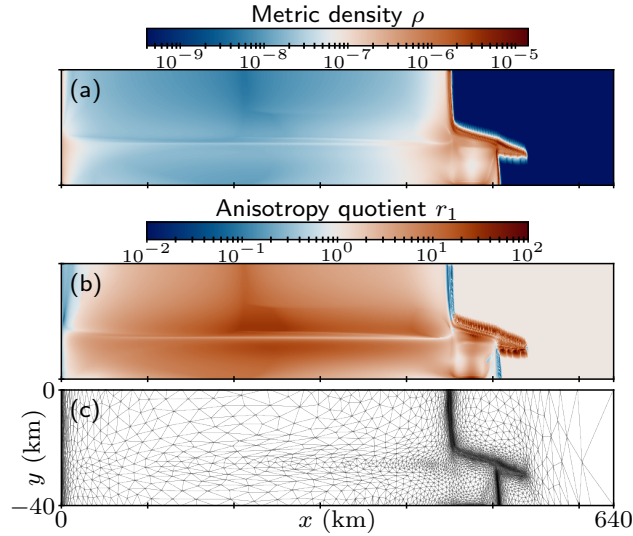
Figure 1. Visualising metric field components: (a) Metric density ρ and (b) anisotropy quotient r_1 of the metric field constructed from the Hessian of the initial basal stress field of MISMIP+ experiments, and (c) the resulting adapted mesh. Adapted element edge lengths were not allowed to be smaller than 1 m, larger than 50 km, and adjacent edge lengths were not allowed to differ by more than 30%. Note that the domain is 16 times longer than it is wide, which is not reflected in the figure for readability.

Analogously to how the metric tensor $\mathbf{M}$ models a single mesh element, the Riemannian metric space $\mathcal{M} = \{\mathbf{M}(\boldsymbol{x})\}_{\boldsymbol{x} \in \Omega}$ models the entire mesh $\mathcal{H}$ of Ω. To that end, the metric complexity

$$\mathcal{C}(\mathcal{M}) = \int_\Omega \rho(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}, \tag{4}$$

represents the integrated amount of resolution of $\mathcal{M}$; i.e., it is the continuous dual of the number of vertices, N_v, of $\mathcal{H}$.

125 As introduced in George et al. (1991), the key idea of anisotropic metric-based mesh adaptation is to generate a mesh which is of unit size element edge lengths when viewed in the prescribed Riemannian metric space. When viewed in the Euclidean metric space, the mesh is then appropriately adapted and anisotropic. Note that the unity criterion is generally impossible to satisfy exactly (consider discretising a rectangular domain with non-overlapping equilateral triangles in Euclidean metric space), so the criterion must be relaxed (for details, see Loseille and Alauzet, 2011a).

130 Following from the diagonalisation of $\mathbf{M}$, the metric is commonly interpreted geometrically by the deformation of a unit circle with a metric map into an ellipse (see Fig. 1.1 in Loseille and Alauzet, 2011a). However, such geometric interpretation is unwieldy for visualising entire metric fields $\mathcal{M}$, and metric density and anisotropy quotient fields may be more appropriate, as shown in Fig. 1. The figure demonstrates how a high metric density ρ prescribes a small adapted mesh element size, and vice versa, while the anisotropy quotient r_1 controls the stretching of mesh elements in the directions of the eigenvectors in $\mathbf{R}$,
135 which appear to be closely aligned with the Cartesian axes.

In cases when the exact solution u is not known, the main challenge lies in finding a reliable estimate of the approximation error $u - u_h$, which is then used in constructing the metric field $\mathcal{M}$. The metric then drives mesh adaptation in a way that reduces the estimated error. In what follows, we consider computing an error estimate based on interpolation error theory.

2.2 Error estimate based on the interpolation error

140 Let us consider the mesh adaptation problem stated in an a priori way in Eq. (1). It can be shown that the approximation error $\|u - u_h\|$ for elliptic problems is bounded by the interpolation error of approximating a continuous field u in V_h:

$$\|u - u_h\| \leq c\|u - \Pi_h u\|, \tag{5}$$

where $\Pi_h u$ is the linear interpolate of u onto V_h and c is a mesh-independent constant (Céa's lemma, e.g. Ciarlet, 1991, 2002). In practice, the approach has been found to hold for hyperbolic problems as well (Frey and Alauzet, 2005). In the context of

145 general adaptation methods, we are interested in how the error scales with polynomial degree (p-adaptation, e.g. Cuzzone et al., 2018) and mesh size (h-adaptation), which we may readily change. In this paper we only consider h-adaptation approaches, but both are active and impactful fields of research.

In particular, it can be shown that the local interpolation error $\|u - u_h\|$ in a single element can be clearly related to the Hessian field of u (see, e.g. Ciarlet, 1991). The result has been readily adopted within metric-based mesh adaptation research

150 due to the (normally) low computational cost involved in computing the Hessian, the directional information contained within it, and the problem-independent nature of Hessian-based interpolation errors (Pain et al., 2001b; Frey and Alauzet, 2005; Piggott et al., 2009; Davies et al., 2011). It has also become a fundamental part of the continuous metric framework, with Loseille and Alauzet (2009) presenting the continuous interpolation error estimation involving a continuous linear interpolate $\pi_{\mathbf{M}} u$. The result was then extended in Alauzet and Olivier (2010), who presented a well-posed continuous formulation of

155 problem (1) that minimises the interpolation error in the L^p norm:

$$\text{Find } \mathcal{M}_{L^p} \text{ such that } \mathcal{E}_{L^p}(\mathcal{M}_{L^p}) = \min_{\mathcal{M}} \left(\int_{\Omega} (u - \pi_{\mathbf{M}} u)^p \, d\boldsymbol{x} \right)^{\frac{1}{p}}, \tag{6}$$

under the constraint that $\mathcal{C}(\mathcal{M}) = N_v$.

For two-dimensional steady-state problems, Loseille and Alauzet (2011b) show that the optimisation problem in Eq. (6) has a unique solution which optimally controls the interpolation error in the L^p norm. The resulting optimal metric is given by

$$160 \quad \mathbf{M}_{L^p} = \mathcal{C}_s \left(\int_{\Omega} (\det |\mathbf{H}|)^{\frac{p}{2p+2}} \, d\boldsymbol{x} \right)^{-1} (\det |\mathbf{H}|)^{\frac{-1}{2p+2}} |\mathbf{H}|, \tag{7}$$

where $\mathcal{C}_s$ is the target metric complexity and $|\mathbf{H}|$ is the absolute value of the Hessian matrix $\mathbf{H} = \mathbf{H}(\boldsymbol{x})$ of u, obtained by taking the absolute value of its eigenvalues in the spectral decomposition. In the numerical demonstrations that follow, we construct the approximation of $\mathbf{H}$ using the robust mixed-L^2 projection recovery technique described in Pain et al. (2001a). The approach ensures a smooth and continuous representation of second-order derivatives.

165   The analogous analysis for time-dependent problems was presented in Alauzet and Olivier (2010), involving a sequence of $N_a \in \mathbb{N}$ meshes $\{\mathcal{H}_i\}_{i=1}^{N_a}$, where each mesh is assigned to one partition of the simulation time interval (see section 2.5). They show that the optimal local metric for a subinterval i is given by

$$\mathbf{M}_{L^p}^i = G_{st} \left(n \left(\det |\mathbf{H}_i| \right)^2 \right)^{-\frac{1}{2p+2}} |\mathbf{H}_i|, \tag{8}$$

where n is the number of timesteps in each subinterval (here constant) and G_{st} is the global space-time normalisation constant

170

$$G_{st} = \mathcal{C}_{st} \left(n^{\frac{p}{p+1}} \sum_{j=1}^{N_a} \int_{\Omega} \left(\det |\mathbf{H}_j| \right)^{\frac{p}{2p+2}} \mathrm{d}\boldsymbol{x} \right)^{-1}, \tag{9}$$

where $\mathcal{C}_{st}$ is the specified space-time metric complexity. The space-time complexity may be interpreted as corresponding to the average number of mesh vertices in the mesh sequence. The number of vertices, however, may vary between meshes, as they are distributed in both space and time among individual meshes in the sequence according to Eq. (8) such that the optimal
175   spatio-temporal error in the L^p norm is obtained.

2.3   Metric intersection

For many applications, adequately capturing the mesh resolution requirements may necessitate information from multiple sources in the domain. For example, it may be desirable to consider all solutions of a multi-variable PDE of interest, or a vector-valued field whose components may instruct a different mesh resolution. Furthermore, a time-dependent field may
180   require significantly different mesh resolution in time in order to adequately resolve its evolution. To this end, individual metrics may be constructed from each such sensor field at multiple timesteps, before they are combined to form a single metric. The combination of metrics is a non-trivial task, and the resulting metric field may vary significantly for different combination approaches and imply different computational cost. Throughout the paper we consider the most common mean of combining metrics: metric intersection.
185   Metric intersection produces a new metric tensor that captures the most restrictive (i.e., finest) resolution requirements from both original metrics. As described in Alauzet et al. (2007), the process involves transforming the two metrics to a common eigenbasis, taking the maximum eigenvalues from each pair of corresponding eigenvalues, and transforming the metric into the original eigenbasis. That is, given two metrics $\mathbf{M}_1$ and $\mathbf{M}_2$ in two-dimensional space, the intersection $\mathbf{M}_{1 \cap 2}$ is defined as

$$\mathbf{M}_{1 \cap 2} = \left(\mathbf{P}^{-1} \right)^{\mathsf{T}} \begin{pmatrix} \max(\lambda_1^1, \lambda_1^2) & \\ & \max(\lambda_2^1, \lambda_2^2) \end{pmatrix} \mathbf{P}^{-1}, \tag{10}$$

190   where $\mathbf{P}$ is the matrix of of normalised eigenvectors $\{\boldsymbol{e}_i\}_{i=1,2}$ of $\mathbf{N} = \mathbf{M}_1^{-1}\mathbf{M}_2$, and $\lambda_i^j = \boldsymbol{e}_i^{\mathsf{T}} \mathbf{M}_j \boldsymbol{e}_i$ are eigenvalues of the original metrics in the common basis. Related to the popular geometric interpretation of metrics with an ellipsoid (see section 2), metric intersection is commonly visualised as producing a metric whose associated ellipsoid is the largest one that can fit within the ellipsoids of the original metrics (see Fig. 2 in Alauzet et al., 2007).
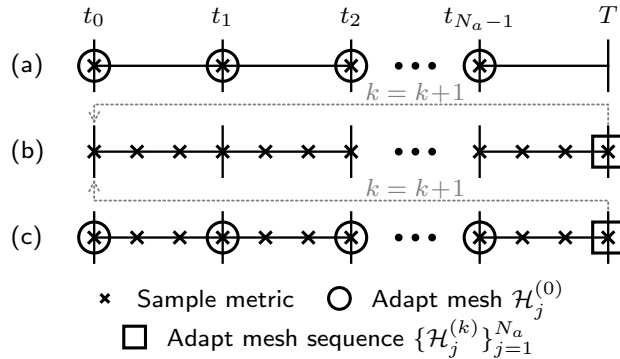
Figure 2. Sketch of (a) classical, (b) global fixed-point, and (c) hybrid time-dependent mesh adaptation algorithms over the simulation time interval $(t_0, T]$ partitioned into N_a equally long subintervals.

2.4 Metric postprocessing

195 After the metric had been computed, it may be desired, or even required, to post-process the metric field before the adapted mesh is generated. This includes prescribing the minimum and maximum element edge lengths, as well as maximum tolerated element anisotropy, by appropriately truncating eigenvalues λ_1, λ_2. Such constraints are particularly useful in order to incorporate existing physical knowledge into the mesh adaptation process, which may not be contained in chosen sensor adaptation fields. Preventing large differences in element sizes and anisotropy between original and adapted meshes may also improve

200 robustness.

Certain choices of sensor fields may lead to metric fields that exhibit a very sharp gradient, as is the case in Fig. 1. Borouchaki et al. (1998) show that such large rates of variation may lead to poor quality of adapted meshes. To that end, they introduce a *metric gradation* routine in order to control its variation and therefore smooth out the metric field.

We refer the reader to Dapogny et al. (2014b) for details on the metric gradation routine and edge length truncation imple-

205 mented in the mesh generator Mmg2d, which is used for all results in this paper (see section 2.7).

2.5 Time-dependent mesh adaptation algorithms

Mesh adaptation has been successfully integrated with finite element methods since the 1980s, when the literature almost exclusively focused on applying mesh adaptation to steady simulations (e.g., Peraire et al., 1987). The mesh adaptation procedure involved an iterative process which would typically begin with a coarse uniform mesh. The mesh is then iteratively adapted

210 until the convergence of the steady-state solution is reached, according to a mesh convergence study. In this context, "mesh convergence" refers to the process where successive refinements of the mesh result in progressively smaller changes in the solution, indicating that the solution is approaching a stable state that is independent of further mesh refinement. For details on mesh convergence see, e.g., Loseille and Alauzet (2011a).

Early attempts at mesh adaptation for time-dependent simulations simply employed the above-described algorithm multiple
times throughout the simulation. This approach is referred to as the *classical mesh adaptation algorithm* in Alauzet et al.
(2007) and we refer to it as such throughout the paper. Namely, the mesh is adapted $N_a \in \mathbb{N}$ times throughout the simulation,
at times t_i, $i = 0, \ldots, N_a - 1$, based on the current solution state at that time. In this way, the simulation involves N_a different
meshes rather than a single fixed mesh. Having different meshes at different times during the simulation allows us to prescribe
fine resolution in particular regions of the domain *only* at times when that is needed. After the mesh had been adapted, the
solution and any other spatially-varying fields in the PDE are then transferred onto it and the simulation resumes until the end
simulation time T. The algorithm is visualised in Fig. 2(a).

While suitable for steady simulations, the classical mesh adaptation algorithm exhibits several shortcomings specific to
time-dependent simulations. Firstly, the approach does not control the temporal distribution of error since the metric is only
normalised in space according to Eq. (7), as only the current solution is used to guide the mesh adaptation. Additionally, by
not considering the future evolution of the solution, the adapted mesh may not adequately resolve the solution at subsequent
simulation times. In such a case, the mesh is said to *lag* with respect to the solution. In the context of glaciological modelling,
a common consequence of a lagging mesh is the migration of the grounding line out of the fine resolution region of the mesh
and into the coarse region, where the grounding line dynamics are no longer accurately captured (e.g., IGE-Elmer/Ice in Sun
et al., 2020). While increasing the mesh adaptation frequency would alleviate the lag, doing so may introduce large errors due
to frequent solution transfers between meshes.

To efficiently avoid the mesh lagging behind the solution, the adaptation algorithm requires a prediction of where fine
resolution is likely to be required before the next mesh adaptation. An accurate prediction will ensure that the simulated
phenomena remain well-resolved when reducing the total number of mesh adaptations N_a. Several prediction strategies have
been proposed over the years, such as *metric advection* for advection-dominated problems (Wilson, 2010; Smith et al., 2016),
and more generally-applicable methods, such as the introduction of safety regions around the fine-resolution region of the mesh
(Löhner and Baum, 1992). The latter has become the most common mesh adaptation criterion in marine ice-sheet modelling,
where a fine-resolution region was introduced within a set distance from the grounding line. However, the strategy has several
shortcomings, as described in the introduction.

Here we adopt the *global fixed-point mesh adaptation algorithm* described in Alauzet and Olivier (2011) in order to remedy
the stated issues related to time-dependent problems. Thanks to its iterative procedure, the algorithm is suitable for solving the
non-linear mesh optimisation problem given in Eq. (6). The simulation time interval $(0, T]$ is partitioned into N_a equally long
subintervals

$$(t_{j-1}, \, t_j] = ((j-1) T/N_a, \, j T/N_a], \tag{11}$$

where $j = 1, \ldots, N_a$ and T/N_a is the fixed subinterval length. To each subinterval we assign a mesh $\mathcal{H}_j^{(k)}$, where k is the
fixed-point iteration number. Partitioning the time interval in this way allows us to prescribe fine resolution in regions of the
domain only at subintervals where that is needed. At each iteration k, the PDE problem is first solved over the entire interval
$(0, T]$ on the sequence of meshes $\{\mathcal{H}_j^{(k)}\}_{j=1}^{N_a}$ before they are adapted. Obtained solutions therefore serve as the approximation

of where fine resolution should be introduced in the mesh during the adaptation process. This is achieved by computing the Hessian metrics for given discrete timesteps, combining them in time according to Eq. (10), and finally computing the space-

250 time normalised metrics according to Eq. (8). In such a way we obtain the metric field $\mathcal{M}_j^{(k)}$ associated with each subinterval. Finally, the meshes are adapted to yield the next iteration's adapted mesh sequence $\{\mathcal{H}_j^{(k+1)}\}_{j=1}^{N_a}$ and the simulation is restarted. The fixed-point algorithm iterates until each of the meshes has satisfied prescribed convergence criteria, or a set maximum number of iterations has been reached. A single iteration of the global fixed-point algorithm is visualised in Fig. 2(b).

The global fixed-point adaptation algorithm offers several advantages over the classical mesh adaptation algorithm. Firstly,
255 the algorithm remedies the lagging mesh problem, since the constructed metric fields contain a prediction of the solution evolution over the entire subinterval. Secondly, since the problem is solved over the entire time interval before meshes are adapted, the metrics are normalised according to space-time normalisation in Eq. (8), which allows for the error to be controlled in both space and time through an iterative procedure.

On the other hand, since meshes are adapted at the end of each iteration, each subinterval may start with a worse approxi-
260 mation of the initial state compared to the classical mesh adaptation algorithm, which adapts the subinterval mesh before the PDE is solved over the corresponding subinterval. This may imply significant additional computational cost if the initial mesh sequence does not accurately capture the modelled phenomena, which is often the case for coarse uniform meshes. As a result, mesh convergence may be slow and many fixed-point iterations may be required. To decrease the computational cost of the inaccurate initial fixed-point iterations, Wallwork (2021) ramps the target complexity over the first few iterations. The strategy
265 is equally applicable to mesh adaptation algorithms for steady problems. The approach, however, does not necessarily reduce the number of iterations needed to achieve mesh convergence. Rather, it dedicates relatively more computational resources to later, more accurate, iterations.

Here we propose a combination of the two algorithms described above, which yields a better initial approximation of the optimal mesh sequence at a minimal additional cost. We achieve this by incorporating the classical mesh adaptation algorithm
270 within the first iteration of the global fixed-point algorithm. Since the classical adaptation algorithm generates the adapted mesh sequence $\{\mathcal{H}_j^{(1)}\}_{j=1}^{N_a}$ during the forward solve of the algorithm, the only added computational cost compared to the global fixed-point algorithm is that of adapting the meshes. This added cost is normally small compared to the cost of solving the PDE. Solution fields obtained from the first iteration of the hybrid algorithm are therefore more accurate than those from the global fixed-point algorithm. As in the global fixed-point algorithm, solutions are still regularly sampled throughout the
275 simulation. Metric fields $\{\mathcal{M}_j^{(1)}\}_{j=1}^{N_a}$ are then constructed by combining the sampled solutions from respective subintervals, and the mesh sequence is adapted again at the end of the first iteration. The latter iterations then proceed as in the global fixed-point iteration, without incorporating the classical mesh adaptation algorithm since the mesh sequence is already well approximated. The proposed algorithm is visualised in Fig. 2(c).

2.6 Interpolation between meshes

280 The final step of any mesh adaptation procedure is that of interpolating all spatially varying fields to the newly generated mesh. This is a crucial step in the mesh adaptation process, as the choice of transfer method may significantly affect the accuracy of

the solution. Since the optimal choice of the transfer method largely depends on the specific problem at hand, we discuss it here with respect to glaciological applications.

The mesh adaptation toolkit Animate, described in section 2.7, currently offers three methods for interpolating fields between different meshes: linear interpolation, Galerkin projection, and the bounded variant of the Galerkin projection. The most popularly used interpolation method in the majority of applications is that of linear interpolation. However, linear interpolation does not preserve the integral of interpolated fields and its application to ice flow problems would yield unphysical results due to conservation laws being violated. The use of Galerkin projection is therefore more appropriate since it is a conservative operator, despite implying a larger computational cost. In particular, Galerkin projection may introduce new function extrema, which may be problematic in times of extreme glacier retreat when ice thickness is close to zero. In such cases, the interpolated ice thickness might become negative-valued. In order to prevent that, we use the bounded variant of the Galerkin projection for degree-1 basis (P_1) fields, which prevents the addition of new function extrema (Farrell et al., 2009; Farrell and Maddison, 2011). However, by bounding the interpolated solution, the operation introduces additional numerical diffusion which may diminish the accuracy of the overall simulation.

2.7 Software

Firedrake (Ham et al., 2023) is a Python-based finite element library for the numerical solution of PDEs utilised throughout this paper. It provides a high-level interface for the definition of finite element variational forms using the Unified Form Language (UFL, Alnæs et al., 2014) which are then compiled into efficient low-level code for the solution of the PDEs. Firedrake utilises PETSc for solving linear and non-linear systems, as well as for its underlying mesh concept (Balay et al., 1997, 2023; Lange et al., 2016).

Mesh generation in Firedrake can be performed using built-in classes, reading in meshes generated by external generators (such as gmsh, Geuzaine and Remacle, 2009), or by using Netgen (Schöberl, 1997), an external generator that has been integrated into Firedrake. Recent developments involve the *Animate* library for metric-based anisotropic mesh adaptation (Wallwork et al., 2024a). Animate implements the presented Riemannian metric framework, Hessian recovery methods, as well as metric operations and interfacing with PETSc's Riemannian metric functionality. A separate library, *Goalie*, has been developed for time-dependent mesh adaptation and supports both feature- and goal-oriented adaptation approaches (Wallwork et al., 2024b). Both libraries have undergone development synchronously with the development of the work presented in this paper.

Mesh optimisation is performed by the external library Mmg2d, which is part of the Mmg platform (Balarac et al., 2022; Dobrzynski and Frey, 2008) that has been integrated into PETSc. Mmg2d requires mesh topology and a metric field defined at each mesh node as input, and returns the mesh where element sizes have been adapted to the given metric field. While the work presented in this paper focuses on 2D mesh adaptation, Animate and Goalie also support 3D mesh adaptation through the implementation of the Mmg3d and ParMmg mesh generation toolkits (Dapogny et al., 2014a).
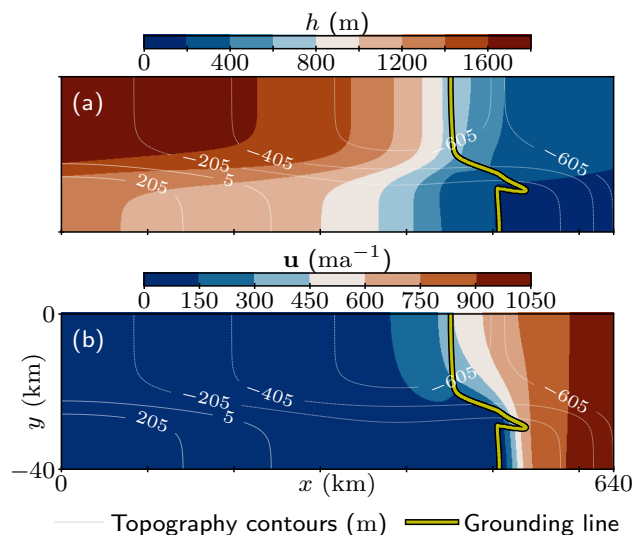
Figure 3. Initial steady-state (a) ice thickness h and (b) ice velocity $\boldsymbol{u}$ of the MISMIP+ experiment with the sliding law of Eq. (12).

3 Numerical demonstrations: MISMIP+

3.1 Experimental setup

315 In this section we demonstrate the above-described anisotropic metric-based mesh adaptation methods on a commonly-used benchmark problem for evaluating glaciological flow models. We use the numerical experiment from the third Marine Ice Sheet Model Intercomparison Project (MISMIP+, Asay-Davis et al., 2016; Cornford et al., 2020) featuring an idealised ice stream, which is an elongated region of fast-flowing ice within an ice sheet. While the ice sheet exhibits negligible basal sliding, the rapid ice stream is dominated by processes at the ice-bed interface (Stokes, 2018). The ice experiences a sudden change of flow

320 regime at the grounding line, where the ice stream flows into the floating ice shelf, which is no longer in contact with the bed topography.

We follow the experiment design and prescribed parameter values as described in Asay-Davis et al. (2016), which places the ice stream in an elongated rectangular domain measuring 640 km in the x-direction and 80 km in the y-direction. The ice is flowing approximately parallel to the x-axis over an idealised bedrock topography. The topography is described by a 6th-order

325 polynomial in the x-direction and an exponential in the y-direction: prescribing an elongated central trough surrounded by steep walls. The bed topography slopes downwards throughout most of the domain, but involves a retrograde slope at around $x = 450$ km where the steady-state ice stream would ground. Beyond the grounding line is the ice shelf that is fed by the upstream flow of ice. The experiment prescribes a no-slip boundary condition at $x = 0$, free-slip conditions at $y = -40$ and $y = 40$ km, and a fixed calving ice front at $x = 640$ km where ice is removed from the domain. The prescribed topography

330 and boundary conditions lead to solutions that are symmetrical about the middle of the domain, at $y = 0$ km. Therefore, we

choose to run the experiment in only half of the domain to preserve computational resources, as several participants in the intercomparison have also done (Cornford et al., 2020).

We focus on the Ice1 group of experiments from the MISMIP+ exercise as it produces more drastic glacier evolution and grounding line migration compared to other experiments in the exercise (Cornford et al., 2020). The Ice1 experiment runs for
335   200 years, where the first 100 years see a retreat of the glacier induced by ice shelf melting. The ice shelf melting is then removed in the last 100 years, when the glacier grows and re-advances. The drastic change of flow regime accompanied with the rapid migration of the grounding line in Ice1 experiment presents an ideal test case for the application of mesh adaptation methods.

3.1.1   Solving equations of glacier flow

340   To solve the equations of glacier flow for ice velocity $\boldsymbol{u}(\boldsymbol{x},t)$ and ice thickness $h(\boldsymbol{x},t)$, we use *icepack*, a Python library built on Firedrake which includes relevant highly customisable glacier flow models (Shapero et al., 2021). As with most current ice sheet models, icepack alternates between solving a diagnostic and a prognostic model at each timestep. The diagnostic model uses ice geometry at the previous timestep to solve the stress balance equations for ice velocity, while the prognostic model evolves ice geometry in time by solving the mass transport equation using the newly-computed ice velocity. All diagnostic
345   icepack models involve Glen's flow law in the non-linear constitutive relation between the stress tensor and the deviatoric strain rate tensor (for details, see Shapero et al., 2021).

Icepack formulates the shallow stream model from a principle of least action, where the action functional consists of terms for viscosity, friction, gravity, and terminus. This is a generalisation of the Shallow Shelf Approximation of the Stokes equations as it involves a bed friction term which is non-zero for grounded ice and zero for floating ice. Therefore, in order to prevent
350   shocks across the grounding line, we gradually reduce friction in its vicinity by using a modified form of the Schoof sliding law derived by Shapero et al. (2021). The basal shear stress $\boldsymbol{\tau}_b$ is zero for floating ice, while for grounded ice it is given by

$$\boldsymbol{\tau}_b = -\frac{\alpha^2 P |\boldsymbol{u}|^{1/3}}{\left[(\alpha^2 P \beta^2)^4 + |\boldsymbol{u}|^{4/3}\right]^{1/4}} \frac{\boldsymbol{u}}{|\boldsymbol{u}|}, \tag{12}$$

where $\alpha^2 = 0.5$, P is the effective pressure at the ice base, $\boldsymbol{u}$ is the horizontal ice velocity, and $\beta^2 = 100\ \mathrm{MPa\,m^{-1/3}\,a^{1/3}}$ is the friction coefficient. The effective pressure P for floating ice is defined as $P = \rho_I g h + \rho_W g d_W$, where ρ_I and ρ_W are
355   densities of ice and water, respectively, g is the gravitational acceleration constant, and d_W is the water depth. Finally, icepack implements the unconditionally stable second-order implicit Lax-Wendroff scheme for the prognostic model (Shapero et al., 2021). The computational cost for solving the diagnostic model is significantly higher than the prognostic model.

The weak form of the shallow stream and mass conservation equations are then discretised by Firedrake using finite element methods. To construct an efficient diagnostic solver, we leverage the formulation of the diagnostic equations as the derivative
360   of a convex action functional, meaning that the Hessian is symmetric and positive-definite (Shapero et al., 2021). We therefore solve linear systems using a direct solver based on Cholesky factorisation (Amestoy et al., 2001, 2006). The critical point secant line search method may be used to solve non-linear systems since the residual equations are the gradient of the diagnostic action
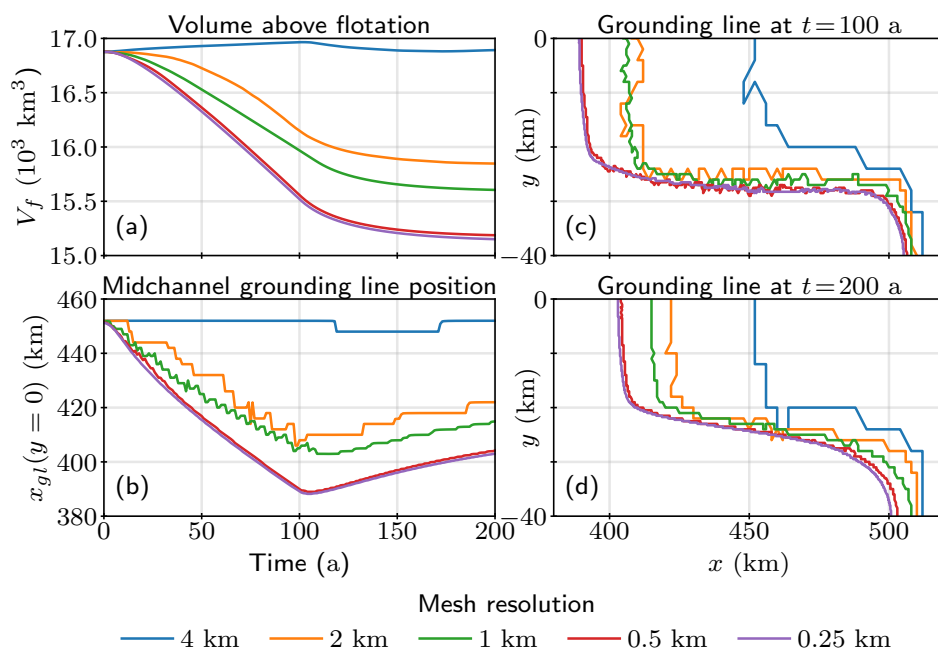
Figure 4. Results of the uniform mesh convergence study for the Ice1 experiment for different mesh resolutions: (a) volume above flotation over time, (b) grounding line position along the midchannel line over time, and (c) grounding line position in the domain at $t = 100$ a and (d) $t = 200$ a.

Table 1. Number of vertices (N_v) and CPU time associated with Ice1 experiment simulations shown in Fig. 4 ran on a single CPU on uniform meshes of varying resolution.

Resolution (km)	N_v	CPU time (min)
4	1771	2.7
2	6741	9.3
1	26 281	36.8
0.5	103 761	200.1
0.25	412 321	951.0

functional. However, we found that the critical point line search sometimes failed to converge with mesh adaptivity, so a more robust backtracking line search was used instead.

3.2 Uniform mesh refinement

In order to study the sensitivity of the solution to mesh resolution and validate the mesh adaptation results, we first run the MISMIP+ Ice1 experiment on a series of uniform structured meshes, with mesh resolutions of 4, 2, 1, 0.5, and 0.25 km.

The Ice1 experiment starts from the initial state computed by running the model forward in time for a long time with constant accumulation rate $a = 0.3 \ \mathrm{ma}^{-1}$ until the sought quasi-steady state is reached. Following Shapero et al. (2021), we efficiently spin up the model by adopting a hierarchical uniform mesh refinement strategy. We begin with a structured uniform coarse mesh of 4 km resolution on which we compute a reasonable approximation of the solution. The mesh is then uniformly refined and the simulation continues. This process is repeated four times, resulting in uniform meshes of 4, 2, 1, 0.5, and ultimately 0.25 km step-sizes in both x- and y-directions. The total spin-up time is 15k years. With each mesh refinement, the difference in ice volume and grounding line position at steady state diminishes. We conclude that the simulation has converged at a uniform mesh resolution of 0.25 km, due to very small differences between the 0.5 km and 0.25 km resolution results at steady-state. This matches convergence studies performed by MISMIP+ participants, who concluded that the 0.5 km mesh resolution is adequate for the experiments (Cornford et al., 2020, supplementary data). The initial quasi-steady state ice thickness h and velocity $\boldsymbol{u}$ are shown in Fig. 3, alongside bed topography contours.

Once the steady-state solution had been computed, we initialise the Ice1 experiment by interpolating the steady-state ice thickness and velocity fields onto uniform resolution meshes. In order to better isolate the differences in results due to different spatial discretisations, for each simulation we use the same timestep size $\Delta t = 1/24 \ a$. Results of the Ice1 experiment on each of the uniform meshes are shown in Fig. 4. Finer resolution simulations exhibit faster ice volume loss and retreat of the grounding line, while the coarsest resolution (4 km) finds a new quasi-steady state. There is again a small difference between the 0.25 and 0.5 km results in most of the temporal evolution. The largest differences appear in the position of the grounding line along the shear margin, where the grounding line position on a 0.5 km mesh fluctuates, and towards the end of the simulation in the grounding line position at the ice ridge near the domain boundary (Fig. 4(d)). Numbers of degrees of freedom and computation times associated with each simulation are given in Table 1, which ranges from 2.7 min for the coarsest resolution, to 951 min for the finest resolution simulation. All simulations were run in serial.

3.3 Mesh adaptation strategy

In the setup of the mesh adaptation experiments we attempt to emulate a more general problem, where the exact or high fidelity solution is generally not available. Therefore, we decide to not utilise any knowledge gained from the uniform mesh resolution simulations in section 3.2. This allows us to judge the presented mesh adaptation algorithm more fairly. The uniform resolution results will only be used for validation, where, in the absence of a closed-form solution, we rely on the 0.25 km resolution results to define reference solutions $\boldsymbol{u}_{\mathrm{ref}}(\boldsymbol{x}, t)$ and $h_{\mathrm{ref}}(\boldsymbol{x}, t)$.

We therefore ignore the gained insight into resolution sensitivity and prescribe minimum and maximum element edge lengths of 1 m and 50 km, respectively. Considering the size of the spatial domain, this effectively means that the elements' size and

shape are unbounded. We use Mmg2d's default maximum metric gradation factor of 1.3, which prevents adjacent element edge lengths to differ more than 30%.

Crucially, when solving time-dependent mesh adaptation problems, we must decide how to split the time domain into N_a
400   distinct subintervals. As discussed in section 2.5, the choice should consider the balance between accurately resolving the ice stream in time, and the added interpolation error and computational cost associated with each mesh adaptation. To that end, we choose to split the simulation time interval $(0, 200 \, \mathrm{a}]$ into 20 equally long subintervals, meaning that meshes are adapted every 10 years of simulation time. A constant timestep $\Delta t = 1/24 \, \mathrm{a}$ is again used.

Similarly, we choose to consider solutions at every 1 year of simulation time in the construction of the metric fields. This
405   provides us with 10 metric fields per subinterval, which is, again, a compromise between accurately representing the evolution of the ice stream and avoiding further computational cost associated with metric computations. The 10 metric fields are combined in time using metric intersection of Eq. (10) to yield a single subinterval metric. This is repeated for each of the 20 subintervals. Afterwards, the 20 subinterval metrics are normalised in space and time according to Eq. (8).

It must be noted that there are certainly other viable choices of partitioning the time interval, which we do not explore here.
410   For example, considering that the thinning of the ice shelf and grounding line retreat is more extreme in the first half of the experiment (Fig. 4), a reasonable choice might involve more frequent adaptations in the first half of the time interval than in the second half. However, as justified earlier, we do not utilise such prior knowledge in the setup of the experiments, and rely on space-time metric normalisation to distribute available degrees of freedom accordingly. Similarly, we always prescribe an equal target complexity to each subinterval metric in the first iteration of the hybrid fixed-point algorithm, which cannot utilise
415   space-time normalisation. This will result in the numbers of vertices, N_v, of each mesh of the mesh sequence $\{\mathcal{H}_j^{(1)}\}_{j=1}^{N_a}$ to be approximately equal.

For interpolating fields between meshes, we use the bounded variant of the Galerkin projection. As discussed in section 2.6, its use is most appropriate in the experiment considered here as it is a conservative operation that does not introduce new maxima in the solution. Since the bounded variant of the Galerkin projection is only implemented for P_1 fields, we use P_1
420   functions to represent both ice velocity and ice thickness over elements.

3.4 Results

In this section we demonstrate the ability of a Hessian-based mesh adaptation method to control the solution accuracy in the MISMIP+ Ice1 experiment. We consider constructing metric fields from the Hessian of ice thickness h and ice velocity $\boldsymbol{u}$, their intersection, and the Hessian of basal stress $\boldsymbol{\tau}_b$. The basal stress field is computed from Eq. (12) and the hydrostatic
425   approximation for the normal basal stress, and as such depends on both h and $\boldsymbol{u}$.

3.4.1 Comparing time-dependent mesh adaptation algorithms

We first wish to compare the mesh convergence rate of the global and hybrid fixed-point adaptation algorithms described in section 2.5. To that end, we run experiments for the two approaches. The first iteration of the global fixed-point algorithm utilises an initial mesh sequence $\{\mathcal{H}_j^{(0)}\}_{j=1}^{N_a}$ of $N_a = 20$ uniform $4 \, \mathrm{km}$ resolution meshes, while the hybrid algorithm begins
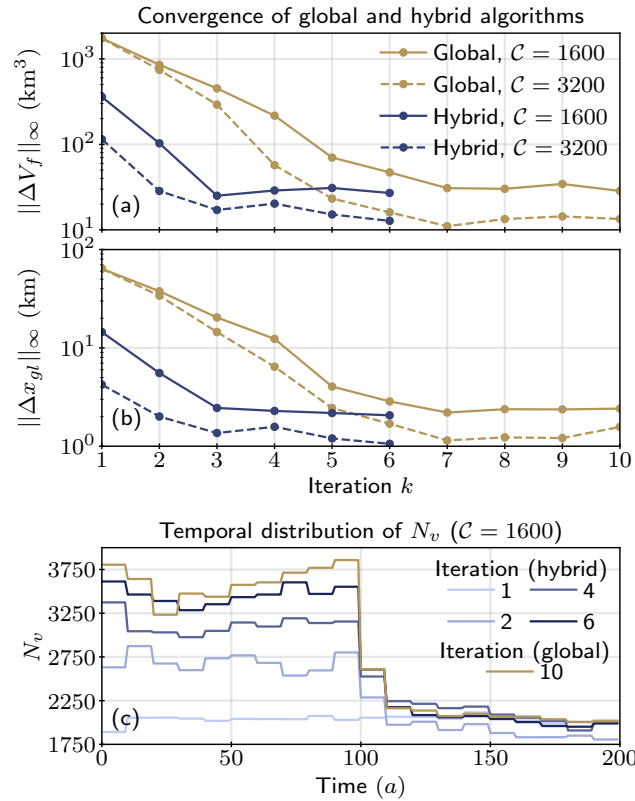
Figure 5. Comparing solution convergence of global and hybrid mesh adaptation algorithms described in section 2.5, with basal stress as the adaptation sensor field and target complexities $\mathcal{C}$ of 1600 and 3200. The resulting maximum deviation from the reference solutions of the (a) volume above flotation and (b) midchannel grounding line position are shown, as well as (c) temporal distribution of N_v for several iterations of the hybrid algorithm and the final iteration of the global fixed-point algorithm for $\mathcal{C} = 1600$.

430 with meshes generated using the classical mesh adaptation algorithm. Meshes are adapted based on the Hessian of the basal stress $\boldsymbol{\tau}_b$, but the same conclusions follow from other choices of adaptation sensor field, as discussed below. In the first iteration of the hybrid fixed-point algorithm, each mesh is adapted at the beginning of the corresponding subinterval. The latter iterations of the two experiments are identical, where the mesh sequence is adapted at the end of the subinterval. We repeat the procedure for two choices of target complexities $\mathcal{C}$: 1600, which is comparable to the number of vertices of the uniform 4 km mesh (see

435 Table 1), and 3200.

The resulting maximum deviation of the volume above flotation $\|\Delta V_f\|_\infty = \max_i |V_f(t_i) - V_{f,\mathrm{ref}}(t_i)|$ and midchannel grounding line position $\|\Delta x_{gl}\|_\infty = \max_i |x_{gl}(y=0, t_i) - x_{gl,\mathrm{ref}}(y=0, t_i)|$ over all simulation years t_i are shown in Fig. 5(a)-(b), where the 0.25 km uniform mesh resolution results are used as reference. In the first iteration of the global fixed-point algorithm, the initial mesh sequence of uniform 4 km resolution is not able to adequately resolve the ice stream dynamics, as

440 shown previously in 3.2. This results in the same high error in the first iteration regardless of the choice of target complexity of
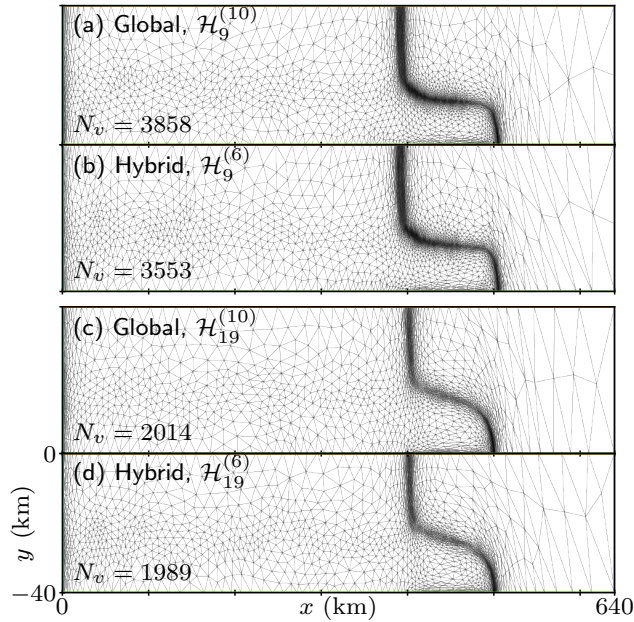
Figure 6. Comparing final adapted meshes obtained from the global fixed-point and hybrid mesh adaptation algorithms for target complexity $C = 1600$. Final adapted meshes $\mathcal{H}_9$ corresponding to the $(90\,\text{a}, 100\,\text{a}]$ subinterval are shown in (a) and (b), and final subinterval meshes $\mathcal{H}_{19}$ corresponding to the $(190\,\text{a}, 200\,\text{a}]$ subinterval are shown in (c) and (d). Meshes adapted using the global algorithm are shown in (a) and (c), while meshes from the hybrid algorithm are shown in (b) and (d).

1600 and 3200, since meshes are only adapted at the end of the iteration. The poor solution accuracy in the first iteration in turn leads to suboptimal metric fields and adapted mesh sequence used in the next iteration. Equal observations follow no matter what choice of sensor field or target complexity is used in constructing the metric, since it is the initial mesh sequence that is far from optimal. However, each next iteration still exhibits smaller deviations, and the mesh sequence converges in about 7

445   iterations to the approximation of the optimal mesh sequence.

In comparison, deviations in the initial iteration of the hybrid fixed-point algorithm are up to an order of magnitude smaller since the ice stream isn't modelled on the inadequate coarse uniform mesh sequence. However, the meshes adapted in the first iteration cannot utilise space-time normalisation, and the number of vertices of each mesh in the mesh sequence $\{\mathcal{H}_j^{(1)}\}_{j=1}^{N_a}$ is therefore approximately equal, as shown in Fig. 5(c). As in the global fixed-point algorithm, the metric fields used to generate

450   the next mesh sequence $\{\mathcal{H}_j^{(2)}\}_{j=1}^{N_a}$ are constructed from solutions sampled from the entire time domain, so the metric fields are able to be normalised in time as well as in space. The adapted mesh sequences $\{\mathcal{H}_j^{(k)}\}_{j=1}^{N_a}$, $k \geq 2$ now contain a higher number of vertices in the first half of the simulation, and fewer in the second half, as shown in Fig. 5(c). This is expected since the grounding line migration is more pronounced in the first half of the simulation, and requires a more widely refined mesh in order for the grounding line to remain within the finely refined mesh region as it migrates in time. Ultimately, the mesh

455   sequence of the hybrid fixed-point algorithm converges to a reasonable approximation of the optimal mesh in only half the

Reference grounding line contours
------ $t = 90$ a ——— $t = 100$ a
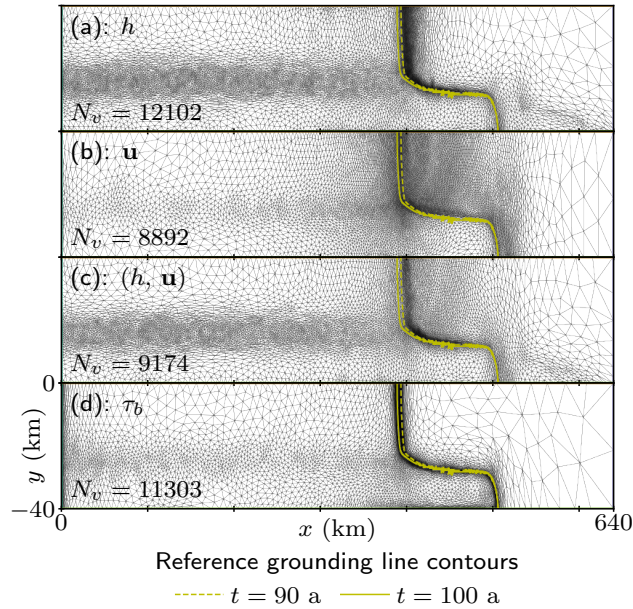
Figure 7. Adapted meshes $\mathcal{H}_9$ associated with the $(90\,\text{a}, 100\,\text{a}]$ subinterval adapted from (a) h, (b) $\boldsymbol{u}$, (c) intersected metric of h and $\boldsymbol{u}$, and (d) $\boldsymbol{\tau}_b$. Shown grounding line contours are computed at the beginning and end of the subinterval from the reference $0.25\,\text{km}$ uniform mesh resolution simulation solutions.

number of iterations as in the global algorithm. As shown in Fig. 6, the global and hybrid mesh adaptation algorithms converge to qualitatively similar adapted meshes $\mathcal{H}_9$ and $\mathcal{H}_{19}$ with consistent vertex counts.

Overall, we have found that a reasonable mesh convergence may be expected in only a few iterations of the hybrid fixed-point algorithm. Namely, most of the effort is done in the first two iterations of the algorithm. Given a reasonable time interval

460 partition and sufficient target complexity, the classical mesh adaptation algorithm embedded within the first iteration generates an adequate adapted mesh sequence $\{\mathcal{H}_j^{(1)}\}_{j=1}^{N_a}$. The mesh sequence $\{\mathcal{H}_j^{(2)}\}_{j=1}^{N_a}$ is then generated at the end of the first iteration from metric fields normalised in both space and time, which allocates additional resolution to subintervals where that is necessary. Solutions obtained in the second iteration are therefore even more accurate, as the error is optimally controlled in space and distributed in time. The mesh sequence generated at the end of the second iteration, $\{\mathcal{H}_j^{(3)}\}_{j=1}^{N_a}$, can therefore be

465 assumed to be a reasonable approximation of the optimal mesh sequence.

As discussed in subsection 3.3, the classical adaptation algorithm utilised in this paper intentionally yields minimal variation in N_v throughout the simulation. The algorithm would highly benefit from a more strategic partitioning of the simulation time interval and a varying target complexity in each subinterval. Doing so would generate an even more optimal initial mesh sequence and potentially further reduce the number of global fixed-point iterations needed to achieve convergence.

3.4.2 Comparing adaptation sensor fields

Finally, we study the mesh convergence and solution accuracy in simulations utilising the hybrid fixed-point adaptation algorithm for different adaptation sensor fields. We repeat the mesh adaptation process for 3 global-fixed point iterations, which implies a total of 4 adaptations of each mesh. As we demonstrated in subsection 3.4.1, this is enough to achieve reasonable mesh convergence. For each choice of sensor field, we prescribe the target metric complexity $\mathcal{C}$ of 800, and repeat the simulation 4 more times such that the target complexity is doubled each time. The final simulation therefore specified the target complexity of $16\mathcal{C} = 12800$. We validate the results of each simulation by computing the relative solution error in L^2 norm based on the reference solutions:

$$\tilde{e}_h = \frac{\|h_{\text{int}} - h_{\text{ref}}\|_{L^2}}{\|h_{\text{ref}}\|_{L^2}}, \tag{13}$$

$$\tilde{e}_{\boldsymbol{u}} = \frac{\||\boldsymbol{u}_{\text{int}}| - |\boldsymbol{u}_{\text{ref}}|\|_{L^2}}{\||\boldsymbol{u}_{\text{ref}}|\|_{L^2}}, \tag{14}$$

where h_{int} and $\boldsymbol{u}_{\text{int}}$ are solution fields interpolated onto the reference mesh and $|\boldsymbol{u}|$ is the velocity magnitude. Adapted meshes $\mathcal{H}_9$ are shown in Fig. 7 for simulations specifying target complexity of $8\mathcal{C}$, with their elements' aspect ratios shown in Fig. 8. The computed solution errors are shown on the left-hand side of Fig. 9, while the right-hand side shows the spatial error distribution at $t = 100$ a on $\mathcal{H}_9$. We show these for $8\mathcal{C}$ complexity since that is when the average solution accuracy of each simulation is closest to one another (Fig. 9(a)-(b)).

Figure 7 reveals instructive areas of refinement and coarsening for each constructed metric field. When adapting based on ice thickness, the fine resolution is concentrated in two regions: along the shear margin (the lateral boundary of the ice stream) and in the hinge zone (the region downstream of the grounding line where the ice shelf begins to form and the ice bends and stretches as it adjusts to buoyant forces). This means that the grounding line will easily migrate out of the fine resolution region during glacier retreat, as shown by the grounding line contours of the reference solution in Fig. 7(a). On the other hand, the grounding line will remain in the fine region during glacier advance, as it migrates further into the finely-resolved hinge zone. In comparison, meshes adapted based on ice velocity are mostly refined in the vicinity of the grounding line and in the inner part of the ice shelf, where ice quickly speeds up as it is no longer influenced by friction at the bed. However, the region around the grounding line is not as finely-resolved as in Fig. 7(a); instead, the available resolution is more-or-less equally distributed over a much larger region. As expected, meshes adapted from the intersected metric of the two fields exhibit features of both. However, this also includes a less refined region upstream of the grounding line compared to that in Fig. 7(b), which again leads to the migration of the grounding line into the coarser region of the mesh. Meanwhile, because the basal stress is discontinuous across the grounding line, the metric field computed from its Hessian will prescribe the finest mesh resolution in its vicinity, as shown in Fig. 7(d). This ensures that the grounding line remains in the finely-resolved region of the mesh throughout the subinterval. The region near the inflow boundary is similarly refined, where the basal stress is again discontinuous due to the presence of the velocity term in the denominator in Eq. 12, which is zero at the inflow boundary. However, since basal stress is null in the ice shelf, the region of the mesh corresponding to it is coarsened. Variation in mesh cell sizes in the ice shelf is only due to the metric gradation routine.
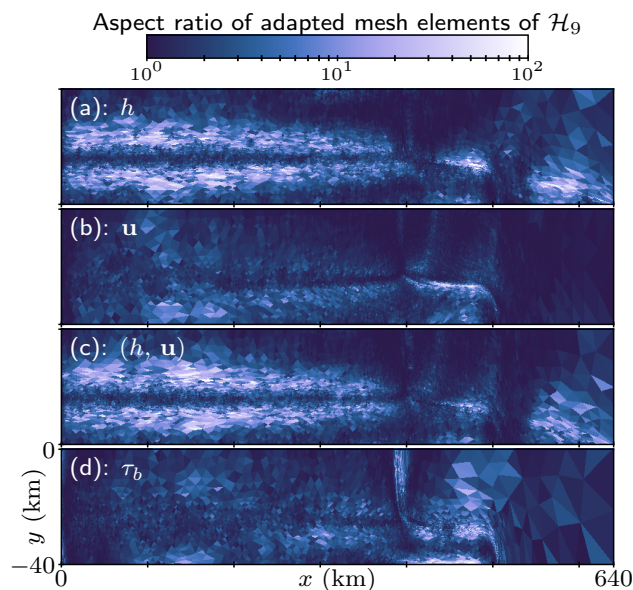
Figure 8. Aspect ratio of elements of adapted meshes $\mathcal{H}_9$ shown in Fig. 7, corresponding to the time subinterval $(90\,\mathrm{a}, 100\,\mathrm{a}]$. The meshes are adapted from the Hessian of (a) h, (b) $\boldsymbol{u}$, (c) intersected metric of h and $\boldsymbol{u}$, and (d) $\boldsymbol{\tau}_b$.

The domain is 16 times longer than it is wide, which is not reflected in Fig. 7 in order to not hinder readability. A measure of cell aspect ratios is therefore shown separately in Fig. 8, where a unit aspect ratio corresponds to a perfectly isotropic cell (i.e.,

505 an equilateral triangle). We observe that meshes whose metrics involve ice thickness contain two bands of highly anisotropic cells in the shear margin with a predominantly x-directional orientation separated by a band of isotropic cells. Since the variation of ice thickness is much sharper across the shear margin than along it, anisotropic elements help to efficiently capture such directional processes along the shear margin by avoiding unnecessary over-refinement in the x-direction. Similarly, in order to capture the discontinuity of the basal stress across the grounding line, meshes adapted based on the basal stress have

510 elements stretched in the direction of the grounding line along its entirety. On the other hand, meshes adapted from ice velocity contain mesh elements with lower anisotropy along the ice stream shear margin, but strongly anisotropic elements in the x-direction along the grounding line flank as the ice drastically speeds up across it.

The above observations directly reflect in the spatial distribution of the solution errors, shown at $t = 100\,\mathrm{a}$ in Fig. 9. While the error in h along the shear margin is lowest on the mesh adapted from h, the same meshes lead to highest errors in the

515 vicinity of the grounding line and in the ice shelf, due to the grounding line migrating out of the fine-resolution region. A more widely refined mesh generated from $\boldsymbol{u}$ is able to most accurately model the sharp velocity changes in the ice shelf, but does not leave enough resolution in the direct vicinity of the grounding line. Overall, the lowest error on both h and $\boldsymbol{u}$ is obtained on the mesh generated from $\boldsymbol{\tau}_b$, which contains enough resolution in the direct vicinity of the grounding line to be able to accurately represent it throughout the 10-year subinterval. However, poorest solution accuracy is obtained along the grounding line flank
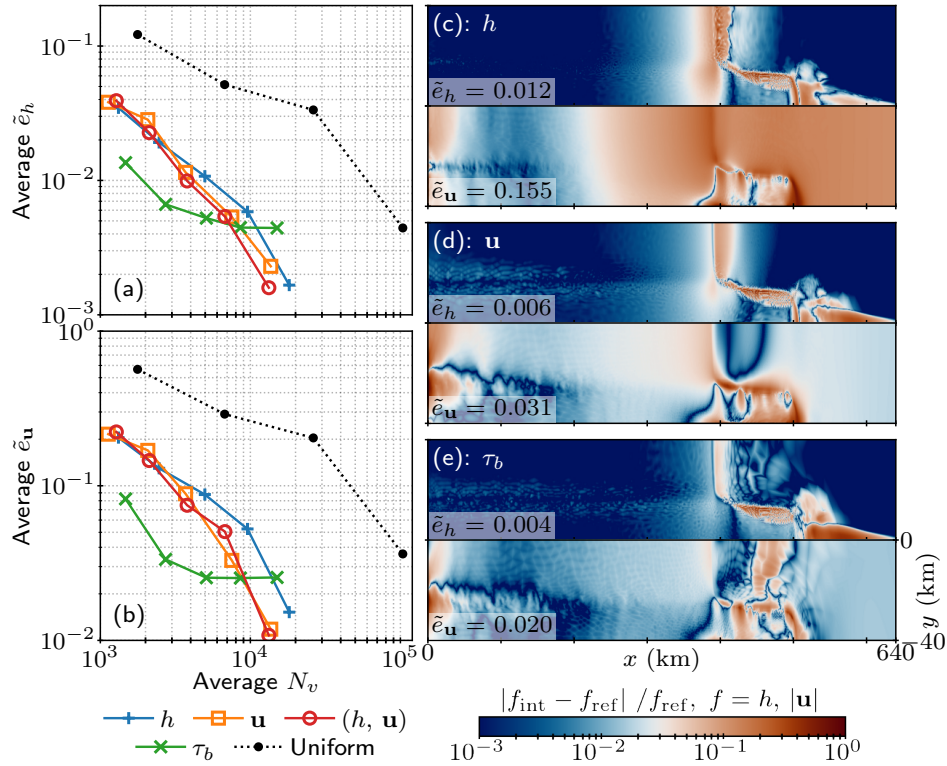
Figure 9. Accuracy of solutions computed on adapted meshes. On the left, solution accuracy averaged over the simulation time interval of modelled (a) ice thickness h and (b) ice velocity $\boldsymbol{u}$ is shown for different sensor fields used in constructing the metric, and for increasing specified target complexity: $\mathcal{C}$, $2\mathcal{C}$, $4\mathcal{C}$, $8\mathcal{C}$, and $16\mathcal{C}$, where $\mathcal{C} = 800$. Solution accuracy of simulations involving uniformly refined meshes are shown for comparison. On the right, spatial distribution of relative errors on h and $\boldsymbol{u}$ is shown at $t = 100$ a for meshes $\mathcal{H}_9$ generated from the Hessian-based metrics of of (c) h, (d) $\boldsymbol{u}$, and (e) $\boldsymbol{\tau}_b$.

520   along the shear margin and in the ice ridge, for all choices of adaptation sensor fields. This was also observed in Fig. 4, where the 0.5 km uniform resolution results most deviate from the reference 0.25 km resolution solutions in these areas.

    Similar conclusions follow by investigating adapted meshes at other subintervals and error distribution at other times, which are then reflected in the time-averaged $\tilde{e}_h$ and $\tilde{e}_{\boldsymbol{u}}$ shown in Fig. 9(a)-(b). At low target complexities, mesh sequences generated from the basal stress $\boldsymbol{\tau}_b$ yield the most accurate solutions since they most accurately represent the evolving grounding line.

525 However, the increase in accuracy gained by further refining such meshes is effectively capped, since the metric computed from $\boldsymbol{\tau}_b$ does not prescribe refinement in the ice shelf. Nonetheless, we obtain the solution accuracy comparable to that of a uniform 0.5 km uniform mesh on a mesh sequence which has nearly 50 times fewer average number of vertices. On the other hand, errors computed on mesh sequences adapted from h, $\boldsymbol{u}$, and their intersection are all similarly high for target complexities of 800 and 1600. Differences begin to be more pronounced for target complexity of 3200, when the metric from $\boldsymbol{u}$ prescribes finer

530 resolution in the vicinity of the grounding line compared to the metric from h. The errors begin to be comparable to that of the
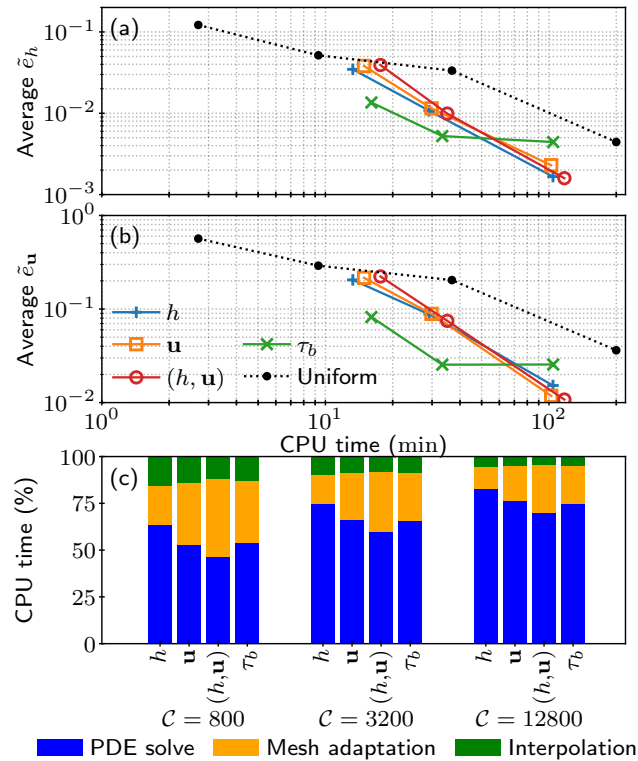
Figure 10. Performance and efficiency comparison of the hybrid mesh adaptation algorithm for different sensor fields and target complexities of 800, 3200 and 12800. Experiments were run on a single CPU and the algorithm terminated after 3 iterations. (a) Average $\tilde{e}_h$ and (b) $\tilde{e}_{\boldsymbol{u}}$ versus CPU time for different adaptation sensor fields: h, $\boldsymbol{u}$, intersection of h and $\boldsymbol{u}$, and $\boldsymbol{\tau}_b$, compared to uniform mesh refinement. (c) Distribution of CPU time across different computational tasks: PDE solve, mesh adaptation (including metric construction), and interpolation.

uniform 0.5 km uniform mesh simulation at the target complexity of 6400. Unlike the meshes adapted from $\boldsymbol{\tau}_b$, the solution error for mesh sequences adapted from h and $\boldsymbol{u}$ keep decreasing for higher target complexities, as shown for the final target complexity of 12800.

Even though we have been successful in outperforming uniform resolution meshes at a fraction of the total number of
535 vertices, this does not necessarily translate into a successful reduction of total computational time. Time-dependent mesh adaptation routines presented in this paper imply additional costs related to metric computation, mesh generation, and function interpolations between meshes. None of these are present in single, fixed-mesh simulations. Moreover, the procedure is iterative, which involves repeatedly solving the PDE problem and adapting the mesh sequence until the mesh sequence has converged. The final decision behind which adaptation method (if any) to employ should therefore consider the total compu-
540 tation time necessary to achieve a desired solution accuracy. In Fig. 10(a)-(b) we therefore plot the average $\tilde{e}_h$ and $\tilde{e}_{\boldsymbol{u}}$ versus total CPU time for different choices of adaptation sensor field and for target complexities of 800, 3200, and 12800. Thanks to PETSc's and Firedrake's logging infrastructure, we are able to easily extract the time involved in each of the above-mentioned

components, which we show in Fig. 10(c). Over all iterations of the hybrid mesh adaptation algorithm, the PDE problem was solved 4 times, the mesh sequence was adapted 4 times, and ice velocity and thickness fields were each transferred 80 times

545 between subsequent meshes. As in the uniform resolution simulations, all simulations were run in serial.

As shown in Fig. 10(c), solving the PDE problem accounted for approximately only 50% of the computational time for $\mathcal{C} = 800$ and 75% for $\mathcal{C} = 12800$. This indicates that the efficiency of mesh adaptation methods is decreased for meshes with small number of vertices due to the relatively high costs of mesh adaptation and solution interpolation. However, as the solver or mesh complexity increase, the benefits of mesh adaptation become more pronounced. The cost of mesh adaptation is particularly high

550 for vector-valued sensor fields since the Hessian metric of each component must be computed separately, and then intersected to form a single metric. Intersected metric of h and $\boldsymbol{u}$ therefore implies the highest computational cost, since it involves 3 Hessian field computations (one for the scalar field h and one for each component of the vector field $\boldsymbol{u}$) and 2 intersections for each metric (one for intersecting components of $\boldsymbol{u}$ and one for intersecting h and combined components of $\boldsymbol{u}$). As a result, the superior performance of the intersected metric field $(h, \boldsymbol{u})$ observed in Fig. 9(a)-(b) is not reflected in the overall computational

555 efficiency shown in Fig. 10(a)-(b). Meanwhile, adapting the meshes based on $\boldsymbol{\tau}_b$ again clearly outperforms other sensor fields at low target complexities. Such adapted meshes achieve a solution accuracy comparable to the 0.5 km uniform resolution mesh with a sixfold reduction of total computational time. All other sensor fields considered in this section achieve the same accuracy with approximately a threefold reduction of total computational time.

4 Conclusions

560 In this study we have demonstrated the effectiveness of anisotropic metric-based mesh adaptation for ice flow modelling using the Firedrake finite element library. All utilised libraries are Python-based and built on top of Firedrake, which, in combination, provide an intuitive and accessible approach to finite element ice flow modelling and mesh adaptation.

Using the setup of the Ice1 experiment of the Marine Ice Sheet Model Intercomparison Project (MISMIP+), we have shown that feature-based anisotropic mesh adaptation based on the Hessian of the solution-dependent field allows us to efficiently

565 optimise the number of vertices in space and time. Our results confirm the importance of adequately modelling grounding line dynamics, and the necessity of fine mesh resolution in its vicinity. Adapting the mesh sequence based on the basal stress is therefore the most efficient option, since the finest mesh resolution is prescribed along the grounding line at a particular expense of a coarsened ice shelf, where the basal stress is zero. Meshes adapted from ice velocity fields are the most efficient at capturing the ice shelf dynamics, while ice thickness prescribed the most resolution along the shear margin and the ice shelf

570 hinge zone. All choices of the adaptation sensor fields were able to achieve accuracy in ice thickness within a few permille, and the ice velocity magnitude within a few percent of the reference 0.25 km uniform resolution simulation results. Specifically, we show that adapted mesh sequences are able to reach the same solution accuracy as the uniform 0.5 km resolution mesh, but with 10–30 times fewer degrees of freedom. However, this translates into a 3–6 times lower computational cost, primarily due to the iterative procedure of mesh adaptation algorithms.

575   This study attempted to emulate a realistic modelling scenario by not utilising any insight gained from uniform fine resolution simulations, which were run for validation purposes. Similarly, we have not utilised several features of metric-based adaptation in order to precisely control the mesh adaptation process, such as spatially-varying element size constraints. Even so, as in other computational fluid dynamics applications, metric-based mesh adaptation has proved to be a powerful and promising tool for glaciological modelling. The main challenges lie in reducing the interpolation error and further reduction of the associated

580 computational cost of the iterative fixed-point adaptation algorithms.

    We emphasise the importance of the interpolation stage associated with each mesh adaptation. The bounded variant of the projection operator utilised in this paper introduces a potentially significant amount of numerical diffusion, which impairs the overall simulation accuracy. Future Animate development will focus on developing a post-processing routine in order to achieve a minimally-diffusive projection, as described in Farrell et al. (2009).

585   While results here have been obtained at a significantly lower computational cost utilising mesh adaptivity, this study reveals promising possibilities for further time reduction. As shown in section 3.4.2, the main contributor to the computational cost of the mesh adaptation methods employed here is their inherent iterative procedure. To that end, we have demonstrated one possible way of reducing the total number of iterations: by combining the classical mesh adaptation algorithm with the global fixed-point algorithm in order to obtain a more optimal initial mesh sequence. The presented hybrid algorithm necessitated

590 approximately 50% fewer iterations in order to reach mesh convergence than the global fixed-point algorithm alone, while still controlling spatial error and its temporal distribution. As noted in section 3.3, further computational gain may be obtained from a strategic time interval partition, which may reduce the total number of mesh adaptations and solution transfers. Based on these findings, our future work is focused on developing new mesh adaptation schemes inspired by the "on-the-fly" nature of the classical mesh adaptation algorithm, that will further reduce total simulation time. We expect that this will allow for

595 even higher fidelity large- and global-scale glaciological modelling, where each iteration of the fixed-point mesh adaptation algorithm implies significant computational cost.

    Overall, these results support the view that the glaciological modelling community should more widely adopt mesh adaptation, as it has been underutilised despite its significant benefits demonstrated in wider computational fluid dynamics applications.

600 *Code and data availability.* All code and software used in this paper are free and publicly available. Animate and Goalie are available under the MIT license, with the exact versions used in this paper archived on Zenodo: https://doi.org/10.5281/zenodo.11537707 (Wallwork et al., 2024a) and https://doi.org/10.5281/zenodo.13353191 (Wallwork et al., 2024b), respectively. All results in this paper may be reproduced from the source code hosted on Zenodo: https://zenodo.org/doi/10.5281/zenodo.13364993 (Dundovic, 2024). The repository contains scripts used for all presented computations and for generating all figures. Figures were created using Matplotlib (Hunter, 2007) with Scientific colour

605 maps from Crameri (2023).

Author contributions. All authors jointly developed the concept of the study. DD designed the novel hybrid mesh adaptation algorithm, conducted all analyses, ensured reproducibility of results, and created all figures. JW is the original author and main developer of the Animate and Goalie software, to which DD contributed significantly by extending existing features and implementing new ones, with input from JW and SK. MP and RH supervised the work. RH was the Principal Investigator of the study, contributed to the glaciological context of the work, and together with DD secured funding for the research. DD prepared the initial draft, with contributions by FGC in writing the introduction section. All authors, in particular JW and FGC, commented on and contributed to improving the clarity of the manuscript.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. DD was supported by the Norwegian Research Council (Norges Forskningsråd) grant no. 324131, the Norwegian Artificial Intelligence Research Consortium (NORA.ai) and The Alan Turing Institute's Enrichment Scheme. RH was supported by the European Research Council, ERC-AdG grant no. 101096057. The computations presented in this paper were performed on the Norwegian Research and Education Cloud using resources provided by the University of Bergen and the University of Oslo.

The authors thank Daniel Shapero for his input and help on setting up simulations in icepack, and Eleda Johnson for sharing her mesh adaptation experience and for reviewing new contributions to the software.

References

620 Alauzet, F. and Loseille, A.: A decade of progress on anisotropic mesh adaptation for computational fluid dynamics, Computer-Aided Design, 72, 13–39, 2016.

Alauzet, F. and Olivier, G.: An Lp-L∞ space-time anisotropic mesh adaptation strategy for time dependent problems, in: Proceedings of ECCOMAS CFD, pp. 1–19, 2010.

Alauzet, F. and Olivier, G.: Extension of metric-based anisotropic mesh adaptation to time-dependent problems involving moving geometries,
625 in: 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, p. 896, 2011.

Alauzet, F., Frey, P. J., George, P.-L., and Mohammadi, B.: 3D transient fixed point mesh adaptation for time-dependent problems: Application to CFD simulations, Journal of Computational Physics, 222, 592–623, 2007.

Alnæs, M. S., Logg, A., Ølgaard, K. B., Rognes, M. E., and Wells, G. N.: Unified form language: A domain-specific language for weak formulations of partial differential equations, ACM Trans. Math. Softw., 40, https://doi.org/10.1145/2566630, 2014.

630 Amestoy, P. R., Duff, I. S., L'Excellent, J.-Y., and Koster, J.: A fully asynchronous multifrontal solver using distributed dynamic scheduling, SIAM Journal on Matrix Analysis and Applications, 23, 15–41, 2001.

Amestoy, P. R., Guermouche, A., L'Excellent, J.-Y., and Pralet, S.: Hybrid scheduling for the parallel solution of linear systems, Parallel Computing, 32, 136–156, 2006.

Asay-Davis, X. S., Cornford, S. L., Durand, G., Galton-Fenzi, B. K., Gladstone, R. M., Gudmundsson, G. H., Hattermann, T., Holland,
635 D. M., Holland, D., Holland, P. R., et al.: Experimental design for three interrelated marine ice sheet and ocean model intercomparison projects: MISMIP v. 3 (MISMIP+), ISOMIP v. 2 (ISOMIP+) and MISOMIP v. 1 (MISOMIP1), Geoscientific Model Development, 9, 2471–2497, 2016.

Aschwanden, A., Bartholomaus, T. C., Brinkerhoff, D. J., and Truffer, M.: Brief communication: A roadmap towards credible projections of ice sheet contribution to sea level, The Cryosphere, 15, 5705–5715, 2021.

640 Balarac, G., Basile, F., Bénard, P., Bordeu, F., Chapelier, J.-B., Cirrottola, L., Caumon, G., Dapogny, C., Frey, P., Froehly, A., Ghigliotti, G., Laraufie, R., Lartigue, G., Legentil, C., Mercier, R., Moureau, V., Nardoni, C., Pertant, S., and Zakari, M.: Tetrahedral remeshing in the context of large-scale numerical simulation and high performance computing, MathematicS In Action, 11, 129–164, https://doi.org/10.5802/msia.22, 2022.

Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F.: Efficient Management of Parallelism in Object Oriented Numerical Software
645 Libraries, in: Modern Software Tools in Scientific Computing, edited by Arge, E., Bruaset, A. M., and Langtangen, H. P., pp. 163–202, Birkhäuser Press, 1997.

Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W. D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M. G., Kong, F., Kruger, S., May, D. A., McInnes, L. C., Mills, R. T., Mitchell, L., Munson, T., Roman, J. E., Rupp, K., Sanan, P., Sarich, J., Smith, B. F., Zampini, S.,
650 Zhang, H., Zhang, H., and Zhang, J.: PETSc/TAO Users Manual, Tech. Rep. ANL-21/39 - Revision 3.20, Argonne National Laboratory, https://doi.org/10.2172/1968587, 2023.

Borouchaki, H., Hecht, F., and Frey, P. J.: Mesh gradation control, International Journal for Numerical Methods in Engineering, 43, 1143–1165, 1998.

Ciarlet, P.: Basic error estimates for elliptic problems, in: Finite Element Methods (Part 1), vol. 2 of *Handbook of Numerical Analysis*, pp.
655 17–351, Elsevier, https://doi.org/https://doi.org/10.1016/S1570-8659(05)80039-0, 1991.

Ciarlet, P. G.: The finite element method for elliptic problems, SIAM, 2002.

Cornford, S. L., Martin, D. F., Graves, D. T., Ranken, D. F., Le Brocq, A. M., Gladstone, R. M., Payne, A. J., Ng, E. G., and Lipscomb, W. H.: Adaptive mesh, finite volume modeling of marine ice sheets, Journal of Computational Physics, 232, 529–549, 2013.

Cornford, S. L., Seroussi, H., Asay-Davis, X. S., Gudmundsson, G. H., Arthern, R., Borstad, C., Christmann, J., Dias dos Santos, T., Feld-
660   mann, J., Goldberg, D., et al.: Results of the third marine ice sheet model intercomparison project (MISMIP+), The Cryosphere, 14, 2283–2301, 2020.

Crameri, F.: Scientific colour maps, https://doi.org/10.5281/zenodo.8409685, 2023.

Cuzzone, J. K., Morlighem, M., Larour, E., Schlegel, N., and Seroussi, H.: Implementation of higher-order vertical finite elements in ISSM v4. 13 for improved ice sheet flow modeling over paleoclimate timescales, Geoscientific Model Development, 11, 1683–1694, 2018.

665   Dapogny, C., Dobrzynski, C., and Frey, P.: Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems, Journal of Computational Physics, 262, 358–378, https://doi.org/10.1016/j.jcp.2014.01.005, 2014a.

Dapogny, C., Dobrzynski, C., and Frey, P.: Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems, Journal of computational physics, 262, 358–378, 2014b.

Davies, D. R., Wilson, C. R., and Kramer, S. C.: Fluidity: A fully unstructured anisotropic adaptive mesh computational modeling framework
670   for geodynamics, Geochemistry, Geophysics, Geosystems, 12, 2011.

Dobrzynski, C. and Frey, P.: Anisotropic Delaunay Mesh Adaptation for Unsteady Simulations, in: Proceedings of the 17th International Meshing Roundtable, edited by Garimella, R. V., pp. 177–194, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

Dos Santos, T. D., Morlighem, M., Seroussi, H., Devloo, P. R. B., and Simões, J. C.: Implementation and performance of adaptive mesh refinement in the Ice Sheet System Model (ISSM v4. 14), Geoscientific Model Development, 12, 215–232, 2019.

675   Dundovic, D.: ddundo/mismip-adapt-paper: Code for Anisotropic metric-based mesh adaptation for ice flow modelling in Firedrake paper, https://doi.org/10.5281/zenodo.13364994, 2024.

Durand, G. and Pattyn, F.: Reducing uncertainties in projections of Antarctic ice mass loss, The Cryosphere, 9, 2043–2055, 2015.

Durand, G., Gagliardini, O., Zwinger, T., Le Meur, E., and Hindmarsh, R. C.: Full Stokes modeling of marine ice sheets: influence of the grid size, Annals of Glaciology, 50, 109–114, 2009.

680   Farrell, P. and Maddison, J.: Conservative interpolation between volume meshes by local Galerkin projection, Computer Methods in Applied Mechanics and Engineering, 200, 89–100, 2011.

Farrell, P. E., Piggott, M. D., Pain, C. C., Gorman, G. J., and Wilson, C. R.: Conservative interpolation between unstructured meshes via supermesh construction, Computer methods in applied mechanics and engineering, 198, 2632–2642, 2009.

Frey, P.-J. and Alauzet, F.: Anisotropic mesh adaptation for CFD computations, Computer methods in applied mechanics and engineering,
685   194, 5068–5082, 2005.

Gagliardini, O., Brondex, J., Gillet-Chaulet, F., Tavard, L., Peyaud, V., and Durand, G.: Brief communication: Impact of mesh resolution for MISMIP and MISMIP3d experiments using Elmer/Ice, The Cryosphere, 10, 307–312, 2016.

George, P. L., Hecht, F., and Vallet, M. G.: Creation of internal points in Voronoi's type method. Control adaptation, Advances in engineering software and workstations, 13, 303–312, 1991.

690   Geuzaine, C. and Remacle, J.-F.: Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities, International Journal for Numerical Methods in Engineering, 79, 1309–1331, 2009.

Gillet-Chaulet, F., Gagliardini, O., Seddik, H., Nodet, M., Durand, G., Ritz, C., Zwinger, T., Greve, R., and Vaughan, D. G.: Greenland ice sheet contribution to sea-level rise from a new-generation ice-sheet model, The Cryosphere, 6, 1561–1576, 2012.

Gladstone, R., Lee, V., Vieli, A., and Payne, A.: Grounding line migration in an adaptive mesh ice sheet model, Journal of Geophysical Research: Earth Surface, 115, 2010.

Gladstone, R. M., Payne, A. J., and Cornford, S. L.: Resolution requirements for grounding-line modelling: sensitivity to basal drag and ice-shelf buttressing, Annals of glaciology, 53, 97–105, 2012.

Goldberg, D., Holland, D., and Schoof, C.: Grounding line movement and ice shelf buttressing in marine ice sheets, Journal of Geophysical Research: Earth Surface, 114, 2009.

Gudmundsson, G., Krug, J., Durand, G., Favier, L., and Gagliardini, O.: The stability of grounding lines on retrograde slopes, The Cryosphere, 6, 1497–1505, 2012.

Ham, D. A., Kelly, P. H. J., Mitchell, L., Cotter, C. J., Kirby, R. C., Sagiyama, K., Bouziani, N., Vorderwuelbecke, S., Gregory, T. J., Betteridge, J., Shapero, D. R., Nixon-Hill, R. W., Ward, C. J., Farrell, P. E., Brubeck, P. D., Marsden, I., Gibson, T. H., Homolya, M., Sun, T., McRae, A. T. T., Luporini, F., Gregory, A., Lange, M., Funke, S. W., Rathgeber, F., Bercea, G.-T., and Markall, G. R.: Firedrake User Manual, Imperial College London and University of Oxford and Baylor University and University of Washington, first edition edn., https://doi.org/10.25561/104839, 2023.

Hunter, J. D.: Matplotlib: A 2D graphics environment, Computing in Science & Engineering, 9, 90–95, https://doi.org/10.1109/MCSE.2007.55, 2007.

Jouvet, G. and Gräser, C.: An adaptive Newton multigrid method for a model of marine ice sheets, Journal of Computational Physics, 252, 419–437, 2013.

Lange, M., Mitchell, L., Knepley, M. G., and Gorman, G. J.: Efficient mesh management in Firedrake using PETSc-DMPlex, SIAM Journal on Scientific Computing, 38, S143–S155, https://doi.org/10.1137/15M1026092, 2016.

Leguy, G., Asay-Davis, X., and Lipscomb, W.: Parameterization of basal friction near grounding lines in a one-dimensional ice sheet model, The Cryosphere, 8, 1239–1259, 2014.

Löhner, R. and Baum, J. D.: Adaptive h-refinement on 3D unstructured grids for transient problems, International Journal for Numerical Methods in Fluids, 14, 1407–1419, 1992.

Loseille, A. and Alauzet, F.: Continuous mesh model and well-posed continuous interpolation error estimation, Ph.D. thesis, INRIA, 2009.

Loseille, A. and Alauzet, F.: Continuous mesh framework part I: well-posed continuous interpolation error, SIAM Journal on Numerical Analysis, 49, 38–60, 2011a.

Loseille, A. and Alauzet, F.: Continuous mesh framework part II: validations and applications, SIAM Journal on Numerical Analysis, 49, 61–86, 2011b.

Moreno-Parada, D., Robinson, A., Montoya, M., and Alvarez-Solas, J.: Description and validation of the ice sheet model Nix v1.0, EGU-sphere [preprint], 2023, 1–33, https://doi.org/10.5194/egusphere-2023-2690, 2023.

Morlighem, M., Rignot, E., Seroussi, H., Larour, E., Ben Dhia, H., and Aubry, D.: Spatial patterns of basal drag inferred using control methods from a full-Stokes and simpler models for Pine Island Glacier, West Antarctica, Geophysical Research Letters, 37, https://doi.org/https://doi.org/10.1029/2010GL043853, 2010.

Pain, C., Umpleby, A., de Oliveira, C., and Goddard, A.: Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations, Computer Methods in Applied Mechanics and Engineering, 190, 3771–3796, https://doi.org/https://doi.org/10.1016/S0045-7825(00)00294-2, 2001a.

Pain, C., Umpleby, A., De Oliveira, C., and Goddard, A.: Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations, Computer Methods in Applied Mechanics and Engineering, 190, 3771–3796, 2001b.

Pattyn, F., Huyghe, A., De Brabander, S., and De Smedt, B.: Role of transition zones in marine ice sheet dynamics, Journal of Geophysical Research: Earth Surface, 111, 2006.

Pattyn, F., Schoof, C., Perichon, L., Hindmarsh, R., Bueler, E., De Fleurian, B., Durand, G., Gagliardini, O., Gladstone, R., Goldberg, D., et al.: Results of the marine ice sheet model intercomparison project, MISMIP, The Cryosphere Discussions, 6, 267–308, 2012.

Pattyn, F., Perichon, L., Durand, G., Favier, L., Gagliardini, O., Hindmarsh, R. C., Zwinger, T., Albrecht, T., Cornford, S., Docquier, D., et al.: Grounding-line migration in plan-view marine ice-sheet models: results of the ice2sea MISMIP3d intercomparison, Journal of Glaciology, 59, 410–422, 2013.

Peraire, J., Vahdati, M., Morgan, K., and Zienkiewicz, O. C.: Adaptive remeshing for compressible flow computations, Journal of computational physics, 72, 449–466, 1987.

Piggott, M. D., Farrell, P. E., Wilson, C. R., Gorman, G. J., and Pain, C. C.: Anisotropic mesh adaptivity for multi-scale ocean modelling, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 367, 4591–4611, https://doi.org/10.1098/rsta.2009.0155, 2009.

Schöberl, J.: NETGEN An advancing front 2D/3D-mesh generator based on abstract rules, Computing and visualization in science, 1, 41–52, 1997.

Schoof, C.: Ice sheet grounding line dynamics: Steady states, stability, and hysteresis, Journal of Geophysical Research: Earth Surface, 112, 2007.

Seddik, H., Greve, R., Zwinger, T., Gillet-Chaulet, F., and Gagliardini, O.: Simulations of the Greenland ice sheet 100 years into the future with the full Stokes model Elmer/Ice, Journal of Glaciology, 58, 427–440, 2012.

Seroussi, H., Morlighem, M., Larour, E., Rignot, E., and Khazendar, A.: Hydrostatic grounding line parameterization in ice sheet models, The Cryosphere, 8, 2075–2087, 2014.

Shapero, D. R., Badgeley, J. A., Hoffman, A. O., and Joughin, I. R.: icepack: A new glacier flow modeling package in Python, version 1.0, Geoscientific Model Development, 14, 4593–4616, 2021.

Smith, R. C., Hill, J., Collins, G. S., Piggott, M. D., Kramer, S. C., Parkinson, S. D., and Wilson, C.: Comparing approaches for numerical modelling of tsunami generation by deformable submarine slides, Ocean Modelling, 100, 125–140, 2016.

Stokes, C. R.: Geomorphology under ice streams: moving from form to process, Earth Surface Processes and Landforms, 43, 85–123, 2018.

Sun, S., Pattyn, F., Simon, E. G., Albrecht, T., Cornford, S., Calov, R., Dumas, C., Gillet-Chaulet, F., Goelzer, H., Golledge, N. R., et al.: Antarctic ice sheet response to sudden and sustained ice-shelf collapse (ABUMIP), Journal of Glaciology, 66, 891–904, 2020.

Vieli, A. and Payne, A. J.: Assessing the ability of numerical ice sheet models to simulate grounding line migration, Journal of Geophysical Research: Earth Surface, 110, 2005.

Wallwork, J. G.: Mesh adaptation and adjoint methods for finite element coastal ocean modelling, Ph.D. thesis, Imperial College London, 2021.

Wallwork, J. G., Dundovic, D., Johnson, E., and Kramer, S. C.: Animate, https://doi.org/10.5281/zenodo.11537707, 2024a.

Wallwork, J. G., Dundovic, D., Johnson, E., and Kramer, S. C.: Goalie, https://doi.org/10.5281/zenodo.13353191, 2024b.

Wilson, C.: Modelling multiple-material flows on adaptive unstructured meshes, Ph.D. thesis, Imperial College London, 2010.