Journal of Computational Physics 321 (2016) 512–531

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Higher-order conservative interpolation between control-volume meshes: Application to advection and multiphase flow problems with dynamic mesh adaptivity

A. Adam [a,b,*], D. Pavlidis [a], J.R. Percival [a,b], P. Salinas [a,b], Z. Xie [a], F. Fang [a], C.C. Pain [a], A.H. Muggeridge [b], M.D. Jackson [b]

[a] *Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London, Prince Consort Road, London, SW7 2BP, UK* [1]

[b] *Novel Reservoir Modelling and Simulation Group, Department of Earth Science and Engineering, Imperial College London, Prince Consort Road, London, SW7 2BP, UK* [2]

ARTICLE INFO

Article history:
Received 22 January 2016
Received in revised form 9 May 2016
Accepted 27 May 2016
Available online 31 May 2016

Keywords:
Mesh adaptivity
Mesh-to-mesh interpolation
Galerkin projection
Conservative
Interface capturing
Viscous fingering

ABSTRACT

A general, higher-order, conservative and bounded interpolation for the dynamic and adaptive meshing of control-volume fields dual to continuous and discontinuous finite element representations is presented. Existing techniques such as node-wise interpolation are not conservative and do not readily generalise to discontinuous fields, whilst conservative methods such as Grandy interpolation are often too diffusive. The new method uses control-volume Galerkin projection to interpolate between control-volume fields. Bounded solutions are ensured by using a post-interpolation diffusive correction. Example applications of the method to interface capturing during advection and also to the modelling of multiphase porous media flow are presented to demonstrate the generality and robustness of the approach.

© 2016 Published by Elsevier Inc.

1. Introduction

Dynamic, adaptive meshing is often used during numerical simulations of transient fluid flows to improve accuracy [1,2]. The mesh is refined in regions where properties are changing rapidly in space and (usually) coarsened in regions where the properties change more slowly in order to improve accuracy whilst minimising computational effort. The mesh may change every time-step depending on the error metrics used to control refinement and coarsening. This inevitably means that data must be mapped from one mesh to another using an interpolation algorithm [3]. The interpolation must be done efficiently, to minimise computational overheads, and in such a way that the field being re-meshed remains bounded. The method must also be conservative when dealing with fields constrained by a conservation law (e.g. mass).

It is advantageous to model transient flow problems using a Control-Volume Finite Element (CVFEM) formulation as this helps enforce mass conservation. Mesh adaptivity for control-volume fields may be used to better capture shocks and

* Corresponding author at: Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London, Prince Consort Road, London, SW7 2BP, UK.

E-mail address: alexander.adam06@imperial.ac.uk (A. Adam).

[1] http://amcg.ese.imperial.ac.uk.

[2] http://norms.ese.imperial.ac.uk.

http://dx.doi.org/10.1016/j.jcp.2016.05.058
0021-9991/© 2016 Published by Elsevier Inc.

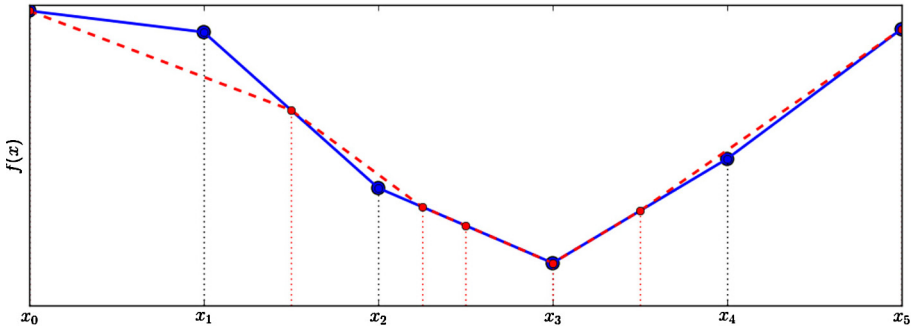


Fig. 1. Node-wise interpolation in 1D. The dotted lines represent element boundaries on the target/donor mesh. The large nodes are donor mesh nodes, shown connected by solid lines representing a linear function interpolating an element. The smaller nodes are the target mesh nodes, and function values there are given by linearly interpolating function values on the donor mesh. The resulting node-wise interpolation onto the target mesh is then represented by the dashed line.

interfaces between fluids as well as material property boundaries in both inertial and porous media flow [4–8]. In the latter example, discontinuous representations of control-volume fields (that is to say when the dual[3] finite element representation is discontinuous) may further help to capture abrupt changes in material properties (e.g. rock permeability) [9,10].

Whilst efficient techniques already exist in the literature for the interpolation of finite element fields [11,12], as will be explicitly demonstrated in this paper, existing methods of control-volume interpolation produce poor results when applied to discontinuous fields. The interpolation introduces a large error into the solution which can become irreparably distorted, highlighting the need for a new method without these deficiencies.

The paper is structured as follows. In Section 1, existing methods of mesh to mesh interpolation are reviewed outlining their strengths and weaknesses. Having motivated the need for a new method of control-volume interpolation, Section 2 discusses the theory underlying the new method, including the implementation of a bounded solution correction. Section 3 provides three distinct applications of the method to interface capturing problems, finishing with an application to viscous fingering in porous media flow, in order to demonstrate the robustness of the method. The examples demonstrate scenarios where other methods of control-volume interpolation can fail to give acceptable solutions. Finally, some key conclusions drawn from the work are presented in Section 4.

1.1. Node-wise interpolation

The simplest approach to interpolation, referred to here as node-wise interpolation proceeds by setting the values of a field at the nodes on the new (target) mesh to be equal to function values at those physical positions on the old (donor) mesh [13]. This method of interpolation is sometimes known as collocation interpolation [13] or consistent interpolation [11] in the literature. More precisely, if ψ^{new} represents a field on the new mesh, then the node values of this field $\psi_i^{new}(\boldsymbol{p_i})$ are given by expanding in donor mesh basis functions and evaluating at the new node positions $\boldsymbol{p_i}$:

$$\psi_i^{new}(\boldsymbol{p_i}) = \sum_j N_j^{old}(\boldsymbol{p_i})\psi_j^{old} \tag{1}$$

where N_j^{old} are basis functions (finite element or control-volume depending on the nature of the field ψ^{old}) defined on the old mesh. An illustration of the node-wise interpolation procedure for a 1D (linear) finite element field is shown in Fig. 1.

Node-wise interpolation has several drawbacks. In particular, it is not conservative in the sense that integrals (volume or surface) of the resultant field are not in general the same as integrals of the original field when taken over the same region of physical space. In addition, it has been demonstrated that node-wise interpolation does not in general preserve the extrema of the interpolant [12,14] and moreover is unsuitable for application to discontinuous fields that are not point-wise defined without substantial modification to the procedure [15,16]. Finally, for higher-order discretisations, node-wise interpolation does not guarantee bounded solutions.

1.2. Galerkin projection/interpolation

Galerkin interpolation is defined as the interpolation method that is optimal in the L^2 norm [11]. In addition to being conservative, the Galerkin method circumvents many of the above issues that afflict node-wise interpolation and in particular is applicable to both continuous and discontinuous interpolants [3,17]. Consider a field ψ^{old} defined on the old mesh,

[3] The control-volume and finite element meshes are dual to one another in the sense that they are related by topological duality. This transformation maps dimension k objects into dimension $d-k$ objects (d is the total number of dimensions). In 2D, edges are therefore mapped into faces, faces into vertices and vertices into faces. This dual relationship can be seen later in Fig. 3.

and ψ, some interpolation of this field that is defined on the new mesh. The requirement of maximal accuracy in the L^2 norm is that $\frac{\partial}{\partial \psi^i}\|\psi^{old} - \psi\|_{L_2} = 0$ (where the ψ^i are the coefficients of ψ when expanded in target mesh basis functions) or equivalently:

$$\int \frac{\partial}{\partial \psi^i}(\psi^{old} - \psi)^2 dV = 0. \tag{2}$$

Expanding $\psi = \sum_i N_i \psi^i$ then leads to the defining equation of a Galerkin projection:

$$\begin{aligned} &\int N_i^{new}(\psi^{new} - \psi^{old})dV = 0 \\ &\implies \sum_j \int N_i^{new} N_j^{new} \psi_j^{new} dV - \sum_j \int N_i^{new} N_j^{old} \psi_j^{old} dV = 0 \end{aligned} \tag{3}$$

$\forall i \in \{1, 2, \ldots, \mathcal{N}\}$ where $\mathcal{N}$ is the dimension of the function space spanned by the target mesh basis functions N_i and we have redefined the ψ satisfying Eq. (2), the Galerkin projection definition, as $\psi = \psi^{new}$. This equation serves as the starting point for the discussion of the control-volume Galerkin method in the next section. Conservativeness of a standard Galerkin projection follows from Eq. (3) when $N_i^{new} = 1$ (for some i), that is, when the constant function equal to unity is contained amongst the new mesh basis functions.

The challenge in implementing Galerkin projection arises in accurately evaluating the second term in Eq. (3). This term $\sum_j \int N_i^{new} N_j^{old} \psi_j^{old} dV$ contains a mixed mass matrix composed of multiplications of basis functions on the donor and target meshes. The issue that arises in the evaluation of this integral is that the donor mesh basis functions are only guaranteed to be (continuous) polynomials over each element of the donor mesh. Over elements of the target mesh, the basis functions may be discontinuous or piece-wise defined. If the integral of the product of the basis functions is evaluated at Gauss quadrature points on the target mesh, the integral will not be exact, leading to a loss of accuracy and conservativeness in the method.

The solution to this proceeds by constructing a geometry known as the supermesh defined as the intersection of the donor and target meshes. On the elements of the supermesh, the donor mesh basis functions are, by construction, continuous polynomials (since the supermesh is an intersection of donor and target meshes and the donor basis functions are continuous polynomials on the donor mesh), allowing high accuracy evaluation of this integral [11].

In the case of interpolating a field that is defined on a general control-volume mesh, direct construction of a supermesh is challenging. Heuristically this is because in the (3D) finite element case, the supermesh is usually given by the intersection of two tetrahedral meshes which is itself tetrahedral, whilst in the control-volume case it involves the intersection of more complex polyhedra, the classification of which is mathematically challenging. The approach must therefore be adapted in order to easily apply it in the control-volume case.

1.3. Grandy interpolation

The next method of interpolation we consider is Grandy interpolation [18]. It is a conservative approach that is often effective but can be highly diffusive due to its low-order nature. Grandy interpolation proceeds by mapping from donor to target mesh by calculating the volume of intersection of overlapping polyhedra on the donor and target meshes. The method assumes the interpolant to be constant across a donor element and it is this feature that accounts for its diffusive nature. The calculation of the volume of intersection of overlapping polyhedra in this case is an example of the construction that must be done in order to build a control-volume supermesh. The Grandy method may therefore be thought of as an example of performing Galerkin projection to a control-volume field ψ^{old} in Eq. (3) but where the donor field is constant across donor elements. Since the method can be regarded as a special case of a Galerkin projection it is conservative.

2. Control-volume Galerkin interpolation

In this section, the new control-volume interpolation method proposed in this paper is explained in detail. Section 2.1 presents the method as a three step procedure that begins by projecting the control-volume field onto a finite element representation on the donor mesh. The resulting finite element field is then projected to a finite element field on the target mesh by assembling a supermesh. Finally, this field is transformed back into a control-volume representation on the target mesh. Having described the interpolation method, Section 2.2 describes how one can obtain a bounded solution after the interpolation is complete.

2.1. Three projections

Fields defined on control-volume meshes may be interpolated using a three step procedure that is now discussed in detail.

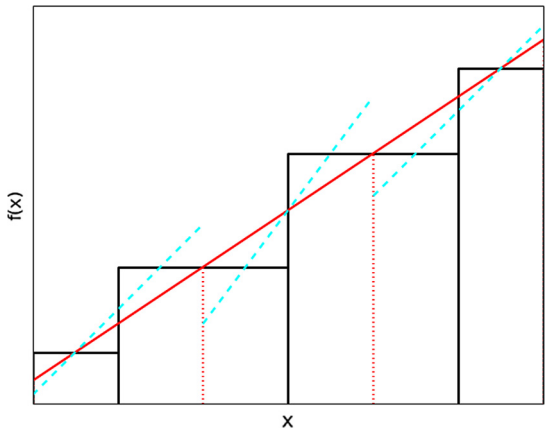


Fig. 2. Illustration of the projection defined by Eq. (4) in 1D. The solid horizontal and vertical lines represent a function defined control-volume wise (constant inside each control-volume) and the dotted lines denote finite element boundaries. The solid diagonal line shows the result of mapping the control-volume function to a continuous Galerkin finite element representation within an element. The dashed lines show the result of mapping the control-volume field to a discontinuous Galerkin finite element representation within each element.

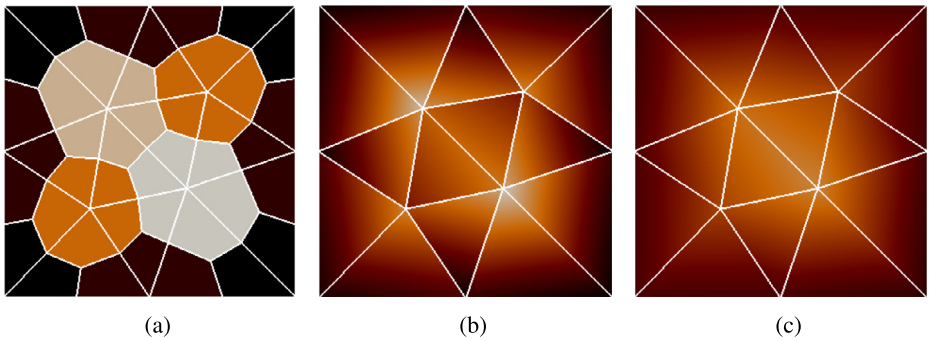


Fig. 3. Illustration of the projection defined by Eq. (4) in 2D. (a) A 2D control-volume function is shown with both element and control-volume meshes overlaid. It is a linear combination of control-volume test functions that are constant across a given control-volume and zero on all others. (b) A projection of the control-volume function in (a) to a discontinuous finite element representation. The finite element solution varies linearly across an element and is discontinuous across element boundaries. (c) Projection of (a) to a continuous finite element representation.

2.1.1. Step one – projection to a finite element representation on the old mesh

First the control-volume (CV) field on the old mesh is mapped via Galerkin projection onto a finite element (FEM) representation within an element which may be either a continuous (CG) or discontinuous (DG) Galerkin representation:

$$\int M^{old}_{CVFEM\,i}(\psi^{old}_{FEM} - \psi^{old}_{CV})dV = 0, \tag{4}$$

where the function $M^{old}_{CVFEM\,i}$ is a test function obtained from the restriction of the finite element shape functions to control-volume i. More precisely, $M_{CVFEM\,i} = N_{FEM\,i}M_{CV\,i}$ (no sum over i) where $N_{FEM\,j}$ are finite element shape functions and $M_{CV\,j}$ are control-volume shape functions. The resulting test function has support on control-volume i and may be continuous or discontinuous between elements. Note that the field after this projection is still defined on the *old* mesh. Expanding using finite element and control-volume basis functions, $\psi^{old}_{FEM} = \sum_j N^{old}_{FEM\,j}\psi^{old}_{FEM\,j}$, and $\psi^{old}_{CV} = \sum_j M^{old}_{CV\,j}\psi^{old}_{CV\,j}$, Eq. (4) can be recast as:

$$\int \sum_j M^{old}_{CVFEM\,i}N^{old}_{FEM\,j}\psi^{old}_{FEM\,j}dV = \int \sum_j M^{old}_{CVFEM\,i}M^{old}_{CV\,j}\psi^{old}_{CV\,j}dV. \tag{5}$$

Eq. (5) may be regarded as a matrix equation $(\mathbf{MN})^{old}\underline{\psi}^{old}_{FEM} = (\mathbf{MM})^{old}\underline{\psi}^{old}_{CV}$, that can be solved for the vector $\underline{\psi}^{old}_{FEM} = \{\psi^{old}_{FEM\,j}\}$ where in component form the matrices are defined as $(\mathbf{MN})^{old}_{ij} = \int M^{old}_{CVFEM\,i}N^{old}_{FEM\,j}dV$ and $(\mathbf{MM})^{old}_{ij} = \int M^{old}_{CVFEM\,i}M^{old}_{CV\,j}dV$. Note that $\mathbf{MN}^{old}$ and $\mathbf{MM}^{old}$ are square matrices since the number of test functions $M^{old}_{CVFEM\,i}, \forall i \in \{1, 2, \ldots, \mathcal{N}\}$ is equal to the number of finite element basis functions $N^{old}_{FEM\,j}, \forall j \in \{1, 2, \ldots, \mathcal{N}\}$ thus resulting in a solvable system of equations. A graphical representation of the projection defined by Eq. (4) is shown in 1D in Fig. 2 and in 2D in Fig. 3.

2.1.2. Step two – projection from old to new mesh by building a supermesh

The resulting FEM representation ψ_{FEM}^{old} interpolating the control-volume field ψ_{CV}^{old} can then be mapped onto the new mesh via a strictly FEM Galerkin projection, greatly simplifying the implementation of this mapping:

$$\int N_{FEM\,i}^{new}(\psi_{FEM}^{new} - \psi_{FEM}^{old})dV = 0. \tag{6}$$

Solving Eq. (6) for ψ_{FEM}^{new} requires computation of the mixed finite element mass matrix $(\mathbf{MM})^{new\text{-}old}{}_{ij} = \int N_{FEM\,i}^{new} N_{FEM\,j}^{old} dV$ which is a complex problem but can be tackled by constructing a supermesh from the intersection of the old and new meshes. This procedure was outlined in Section 2.1, but the reader is referred to [12] for a detailed account of global supermeshing, and [11] for a discussion of local supermeshing and also to [19] for a more detailed account of supermeshing as a whole. The mapping is greatly simplified as it now only requires calculation of intersections of finite element meshes as opposed to control-volume meshes or a mixture of the two.

2.1.3. Step three – projection back to CV representation on the new mesh

Having done this, the resulting FEM representation, ψ_{FEM}^{new}, on the new mesh can then finally be projected back to a CV representation, ψ_{CV}^{new}, on the new mesh:

$$\int M_{CV\ i}^{new}(\psi_{CV}^{new} - \psi_{FEM}^{new})dV = 0. \tag{7}$$

This concludes the three step interpolation procedure. Note that there are advantages to using a DG FEM representation within the mapping rather than a continuous FEM representation as it is more accurate and does not propagate Gibbs oscillations beyond an element as there is no communication between elements. Note also that the complete mapping (Eq. (4)–Eq. (7)) is conservative and does not change the interpolant (that is $\psi_{CV}^{new} = \psi_{CV}^{old}$) if the old and the new mesh are the same. This statement follows by considering (Eq. (4)–Eq. (7)) when the new mesh and old mesh are the same:

$$\int M_{CV\ i}^{new}\psi_{CV}^{new}dV = \int M_{CV\ i}^{new}\psi_{DG}^{new}dV = \int M_{CV\,i}^{old}\psi_{DG}^{old}dV \tag{8}$$

$$= \sum_{k\in CVDG\subset CV_i} \int M_{CVDGk}^{old}\psi_{DG}^{old}dV = \sum_{k\in CVDG\subset CV_i} \int M_{CVDGk}^{old}\psi_{CV}^{old}dV = \int M_{CV\,i}^{old}\psi_{CV}^{old}dV \tag{9}$$

and thus $\int M_{CV\ i}^{new}\psi_{CV}^{new}dV = \int M_{CV\,i}^{old}\psi_{CV}^{old}dV$ or $\psi_{CV\ i}^{new} = \psi_{CV\,i}^{old}$. In Eq. (9), k is an element of the set of CVDG's (discontinuous sub-control-volumes) that are contained inside control-volume CV_i.

2.2. Implementation of bounded solution corrections

Despite having many advantages over other interpolation methods, interpolation of a control volume field by way of Galerkin projection onto a finite element basis will not necessarily result in a solution that is bounded. This issue also arises in node-wise interpolation when the order of the finite element basis functions is quadratic or higher and so is not unique to the Galerkin method. We now outline a procedure whereby given a solution to the above interpolation, ψ_{CV}^{new}, one can obtain a new solution that is bounded.

In order to measure the deviation from boundedness, begin by defining a field $\psi_{CV\ devk}^{new}$ control-volume wise such that at each control-volume k:

$$\psi_{CV\ devk}^{new} = \begin{cases} \psi_{CV\ k}^{new} - \psi_{CV\ maxk}^{new}, & \text{if } \psi_{CV\ k}^{new} > \psi_{CV\ maxk}^{new}; \\ \psi_{CV\ k}^{new} - \psi_{CV\ mink}^{new}, & \text{if } \psi_{CV\ k}^{new} < \psi_{CV\ mink}^{new}; \\ 0, & \text{otherwise}, \end{cases} \tag{10}$$

where boundedness is achieved when $\psi_{CV\ devk}^{new} = 0, \forall k \in \{1, 2, \ldots, \mathcal{M}_{\mathcal{CV}}^{new}\}$. For a control-volume k, if $\psi_{CV\ k}^{new}$ lies strictly within the solution bounds $(\psi_{CV\ mink}^{new}, \psi_{CV\ maxk}^{new})$, the control-volume does not need mass adjustment in order to obtain a bounded solution. We describe such a control-volume as having 'absorptive capacity'.

The strategy to obtain a bounded solution is then to introduce diffusion to the deviation field in order to spread non-zero deviation into neighbouring control-volumes. Iterating this procedure should then lead to a state where all control-volumes are at absorptive capacity. This process must be done in such a way that conservativeness is maintained. This diffusion algorithm in now described in greater detail.

At each iteration, a new vector $\underline{\psi}_{CV\ alt}^{new} = (\psi_{CV\ alt1}^{new}, \psi_{CV\ alt2}^{new}, \ldots, \psi_{CV\ alt\mathcal{M}_{\mathcal{CV}}}^{new})^T$ is solved for that satisfies:

$$\mathbf{M}_L \underline{\psi}_{CV\ alt}^{new} = \mathbf{M}_R \underline{\psi}_{CV\ dev}^{new}. \tag{11}$$

This operation is trivial since by construction, the mass matrix $\mathbf{M}_L$ is diagonal (see below). Since the distributed mass matrix $\mathbf{M}_R$ is not diagonally dominant (see below) a relaxation coefficient $w = 0.5$ is introduced such that:

$$\underline{\psi}^{new}_{CV\ alt} \leftarrow w\underline{\psi}^{new}_{CV\ alt} + (1-w)\underline{\psi}^{new}_{CV\ dev}. \tag{12}$$

The interpolant vector $\underline{\psi}^{new}_{CV}$ is then modified using:

$$\underline{\psi}^{new}_{CV} \leftarrow \underline{\psi}^{new}_{CV} - \underline{\psi}^{new}_{CV\ dev} + \underline{\psi}^{new}_{CV\ alt}. \tag{13}$$

Note that the above operations have no effect on the integral of ψ^{new}_{CV} and hence conservativeness. The mass matrices in Eq. (11) are defined as:

$$\mathbf{M}_{Rij} = \begin{cases} \frac{m_{CV\,i}}{m_{SUR\,j}} m_{CV\,j}, & \text{if } j \in neig\{i\}; \\ 0, & \text{otherwise}, \end{cases} \tag{14}$$

and

$$\mathbf{M}_{Lij} = \begin{cases} m_{CV\,i}, & \text{if } i = j; \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

where $m_{CV\,i}$ is the volume of control-volume i and $m_{SUR\,i}$ contains the sum of the volumes of the surrounding control-volumes:

$$m_{SUR\,i} = \sum_{k \in neig\{i\}} m_{CV\,k} \tag{16}$$

and $neig\{i\}$ labels the control-volumes that share a face with control-volume i, including itself.

One concern with the iterative scheme we have presented above is that it substantially adjusts control volumes that are already bounded. This problem can be lessened by the following modification. When iterating over control-volumes, for a given control-volume k, we identify the neighbouring control-volumes to k with the maximum positive and the minimum negative defect adjustment needed and mass is exchanged exclusively between these until no further mass adjustment can be made for k. One then proceeds to control-volume $k+1$. More precisely:

1) Form $\psi_{CV}{}^{new}_{dev\ i}$ and $\psi_{CV}{}^{new}_{dev\ j}$ from Eq. (10).
2) For control-volumes i and j set:

$$\psi^{new}_{alt\ i} \leftarrow \psi^{new}_{dev\ i}, \quad \psi^{new}_{alt\ j} \leftarrow \psi^{new}_{dev\ j}. \tag{17}$$

3) If $\psi^{new}_{alt\ i}\psi^{new}_{alt\ j} < 0$

$$\text{then} \begin{cases} \psi^{new}_{alt\ i} \leftarrow \frac{(\psi^{new}_{alt\ i} m_{CV\,i} + \psi^{new}_{alt\ j} m_{CV\,j})}{m_{CV\,i}} \text{ and } \psi^{new}_{alt\ j} \leftarrow 0, & \text{if } \psi^{new}_{alt\ i} m_{CV\,i} > \psi^{new}_{alt\ j} m_{CV\,j}; \\ \psi^{new}_{alt\ j} \leftarrow \frac{(\psi^{new}_{alt\ i} m_{CV\,i} + \psi^{new}_{alt\ j} m_{CV\,j})}{m_{CV\,j}} \text{ and } \psi^{new}_{alt\ i} \leftarrow 0, & \text{otherwise}, \end{cases} \tag{18}$$

$$\text{and} \quad \psi^{new}_{CV\ i} \leftarrow \psi^{new}_{CV\ i} - \psi^{new}_{CV\ dev\,i} + \psi^{new}_{CV\ alt\,i}, \tag{19}$$

$$\text{and} \quad \psi^{new}_{CV\ j} \leftarrow \psi^{new}_{CV\ j} - \psi^{new}_{CV\ dev\,j} + \psi^{new}_{CV\ alt\,j}. \tag{20}$$

The mass adjustment associated with step 3 above takes the control-volume with the minimal needed mass adjustment, that is control-volume j if $\psi^{new}_{alt\ i} m_{CV\,i} > \psi^{new}_{alt\ j} m_{CV\,j}$ and control-volume i otherwise, and adds its mass deficiency (negative or positive mass) to the other control-volume. The effect of this is to eliminate the donor control-volume's mass deficiency and reduce the mass deficiency of the receiving control-volume so that either $\psi^{new}_{alt\ i} \leftarrow 0$ or $\psi^{new}_{alt\ j} \leftarrow 0$.

In practice, the algorithm above (Eq. (17)–Eq. (20)) may not be able to eliminate all mass deficiencies and it is therefore used together with the mass distribution approach (Eq. (11)–Eq. (13)). By iteratively solving Eq. (17)–Eq. (20) until no further adjustment can be made and then solving Eq. (11)–Eq. (13), and repeating this two-stage process until the maximum control-volume field adjustment is small (e.g. 10^{-6} in the examples in this paper), one can obtain a bounded solution after interpolation.

3. Applications

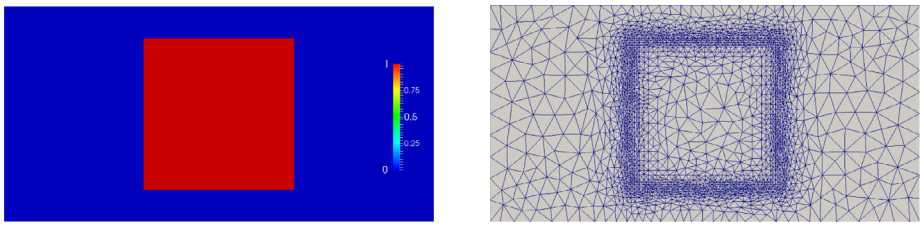
3.1. Interpolation during mesh refinement

Having discussed the theoretical background underlying the proposed new method of control-volume interpolation, a series of applications of the method are now considered.

The first three examples are to advection problems. We consider the dynamics of a scalar field ψ that is passively transported with the (prescribed) bulk velocity $\mathbf{u}$ of the underlying fluid according to the advection equation:

Table 1
Parameters used in the static scalar simulation described in the main text. The variable t is physical time. All measures of distance in this paper may be regarded as dimensionless (hence the absence of units in the specification of domain size and minimum/maximum edge-length).

Size	Initial/final resolution	Min edge-length	Max edge-length
2.0×1.0	5400/44400	0.01–(0.009)t	0.1



(a) (b)

Fig. 4. (a) Initial scalar field distribution. The field is non-zero in a square region of length 0.7 and positioned at the centre of the computational domain. (b) Initial mesh. The mesh is refined along the edges of the square in order to capture the interface.

$$\frac{\partial \psi}{\partial t} + \nabla \cdot (\psi \mathbf{u}) = 0. \tag{21}$$

The mesh adapts to the curvature of the scalar field ψ which is represented control-volume wise and the behaviour of different methods of interpolation used to map this scalar field between meshes are compared. The scalar field ψ is defined on a control-volume mesh that is dual to an n'th order finite element pressure mesh that may be discontinuous or continuous. The pressure representation will be referred to using the notation PnDG in the former case and PnCG in the latter case. The value of n used, as well as the mesh continuity will be stated in each example in turn. The finite element pressure representation is used to calculate the higher-order fluxes for advection as detailed in [4,6].

As a first example, a case where the bulk fluid velocity $\mathbf{u} = 0$ is examined. Consider a 2D scalar field $\psi(x, y)$ that is non-zero on a square region and zero elsewhere as shown in Fig. 4. Despite the lack of any fluid motion, the mesh is made to change over the course of the simulation by decreasing the minimum element edge-length for mesh adaptivity with time (see Table 1 for its functional form). The mesh therefore gradually refines along the square interfaces of the problem and the scalar field must be interpolated at every time-step. With no change in minimum element edge-length there would of course be essentially no interpolation to do in this example. Note also that this simulation involves no advective transport and hence constitutes a pure test of the interpolation method itself and in particular of the conservativeness of the method.

The results after adapting the mesh 100 times using node-wise interpolation and the Grandy method are compared with those obtained by the control-volume Galerkin method presented in this paper. The numerical parameters used in the simulation are specified in Table 1. The resolution specifies the approximate number of elements in the initial mesh as well as in the final mesh. Note that here and in all examples, the final resolutions differ for the three interpolation methods and a single representative value is quoted. The minimum/maximum element edge-lengths bound the element sizes that enter the metric tensor in adaptivity. They control the smallest and largest element sizes permitted in a mesh after an adapt and hence effectively control the mesh resolution for a fixed interpolation error bound and gradation settings (the scaling ratio between adjacent elements). A concise overview of pertinent features of mesh adaptivity can be found in [19]. The minimum/maximum edge-lengths are fixed to be the same for the three interpolation methods. As discussed previously, the minimum element edge-length decreases as a function of time to a minimum value of 0.001.

Using the notation introduced at the start of this section, a P1DG pressure discretisation was used in this example. A diffusive correction as outlined in Section 2.2 was also applied after interpolation to keep the solution fields within physical bounds. The same correction is applied to both node-wise and control-volume Galerkin methods, but is not required in the Grandy case because this is automatically bounded.

The initial state is shown in Fig. 4 and the results of the three interpolation methods after 100 mesh adapts are shown in Fig. 5. It is clear that node-wise interpolation no longer sharply captures the interfaces in the problem and inhomogeneities have begun to appear along the boundary of the square. This is to be contrasted with Grandy and Galerkin interpolation which are able to capture the boundaries very closely. Qualitatively, the Grandy and Galerkin results are similar in this example.

Fig. 6(a) shows the integral of the scalar field over the domain for each interpolation method, and Fig. 6(b) shows the error in the three interpolation methods. The integral fluctuates in the node-wise case every time the mesh adapts and conservativeness is lost. The fluctuations in the control-volume Galerkin and Grandy cases on the other hand are negligible – of the order machine precision, implying that the methods are conservative. In all cases, the L^2 error decreases with time as the mesh resolution increases (due to the decreasing minimum element size) as expected but the error is smallest in the control-volume Galerkin case. The error is reduced compared to Grandy interpolation due to a reduction in inhomogeneities

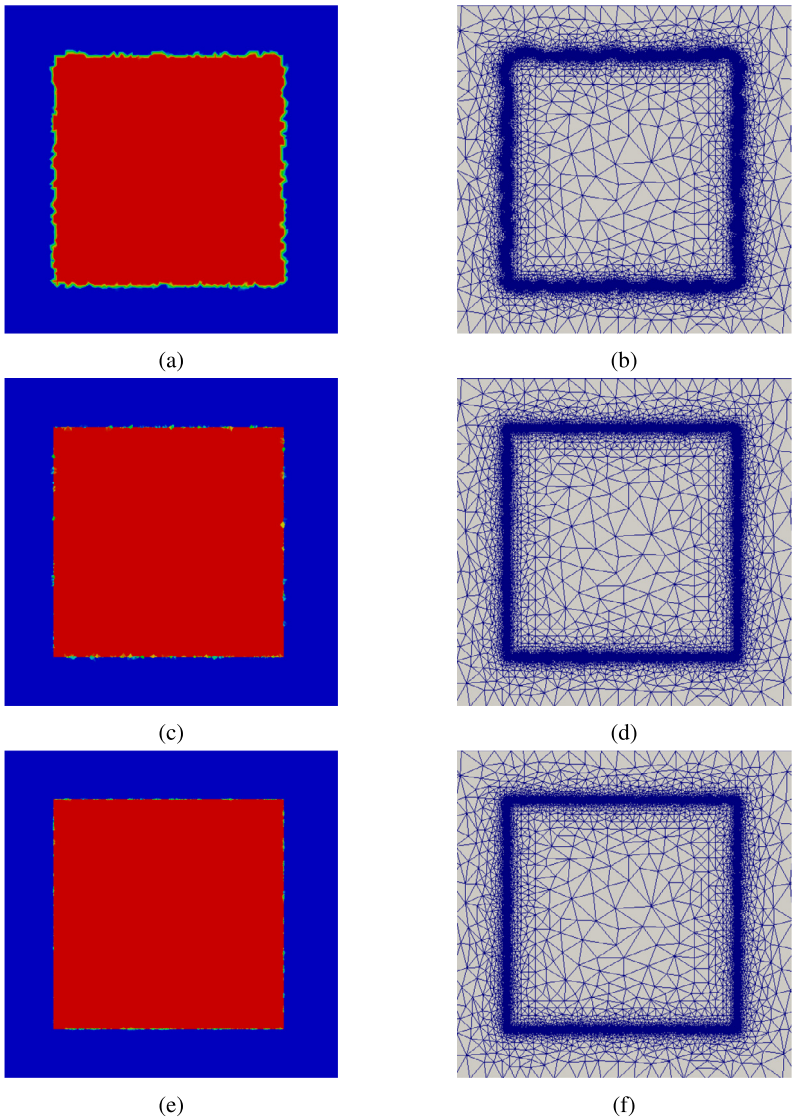


Fig. 5. Scalar field and initial mesh after 100 mesh adapts computed using node-wise interpolation (a, b) Grandy interpolation (c, d) and control-volume Galerkin interpolation (e, f).

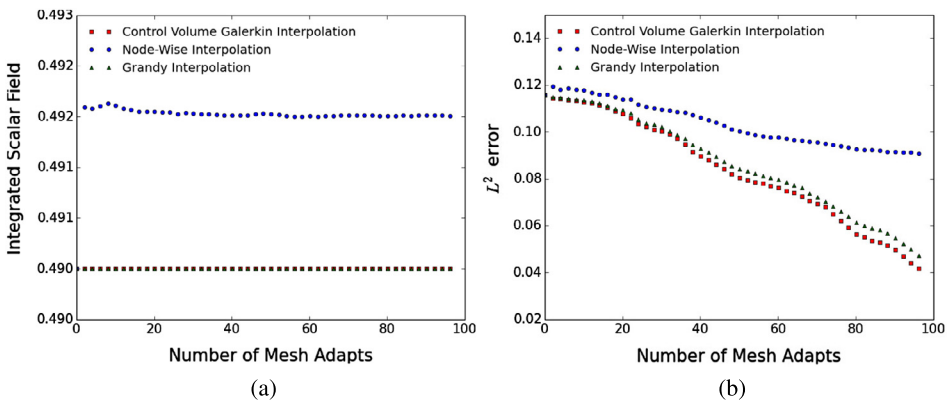


Fig. 6. (a) Integrated scalar field $\int \psi(t)dxdy$ plotted against the number of mesh adapts as computed via control-volume Galerkin projection (squares), node-wise interpolation (circles) and Grandy interpolation (triangles). (b) L^2 error in the static scalar field plotted against the number of mesh adapts for control-volume Galerkin projection (squares), node-wise interpolation (circles) and Grandy interpolation (triangles).

Table 2
Parameters used in the Gaussian scalar simulation discussed in the main text.

Size	Target mesh resolution	Min edge-length	Max edge-length
1.0×1.0	40–500	0.06–0.25	1.0

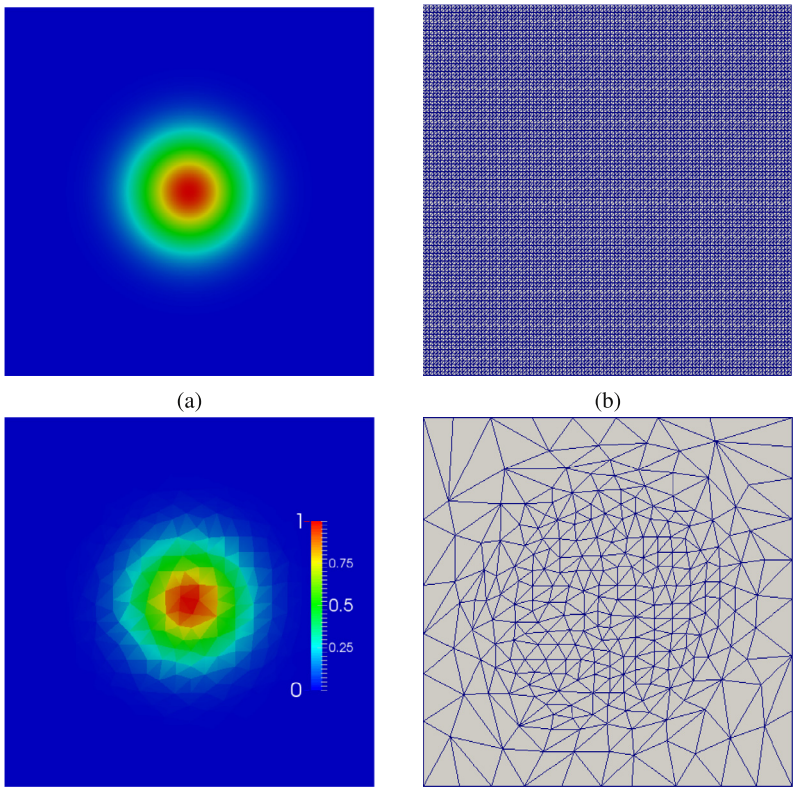


Fig. 7. (a) Initial Gaussian scalar field distribution on a fixed mesh before interpolation. (b) Initial structured mesh before interpolation. (c) Gaussian scalar field after interpolation onto a coarser target mesh. In the example shown here the interpolation was done by the control-volume Galerkin method. (d) Target mesh with prescribed minimum element edge-length (in this case 0.06).

along the interface boundary. Note as mentioned in Section 1, node-wise interpolation can be problematic for discontinuous interpolants. Whilst not so pronounced in this example, it will become so in what follows.

3.2. Higher-order convergence

As discussed in Section 2.1, the control-volume Galerkin method begins by projecting a field defined control-volume wise onto a finite element representation. In contrast to Grandy interpolation, this yields (in general) a higher-order polynomial representation as opposed to a flat function. It is expected that this higher-order representation in finite element space should lead to higher-order numerical convergence and this is indeed the case as is now demonstrated by example. We also further motivate this higher-order convergence in Appendix A.

In order to study the convergence of the interpolation methods, we consider a sequence of simulations beginning on a high resolution fixed mesh and mapping once (through mesh adaptivity) to a new mesh with different minimum edge-length (and hence resolution). The initial fixed mesh resolution is kept the same whilst the minimum element edge-length on the target mesh is varied over the range specified in Table 2. The three interpolation methods are used in turn to map the scalar fields from the old mesh onto the new mesh and the L^2 error in the interpolation results is plotted as a function of minimum element edge-length on the target mesh. The same P1DG pressure discretisation as in the previous example is adopted.

In this example, the scalar field $\psi(x, y)$ that is interpolated from the old mesh to the new mesh is taken to be a 2D Gaussian function of the form.

$$\psi(x, y) = e^{-[(x-0.5)^2+(y-0.5)^2]/0.02} \tag{22}$$

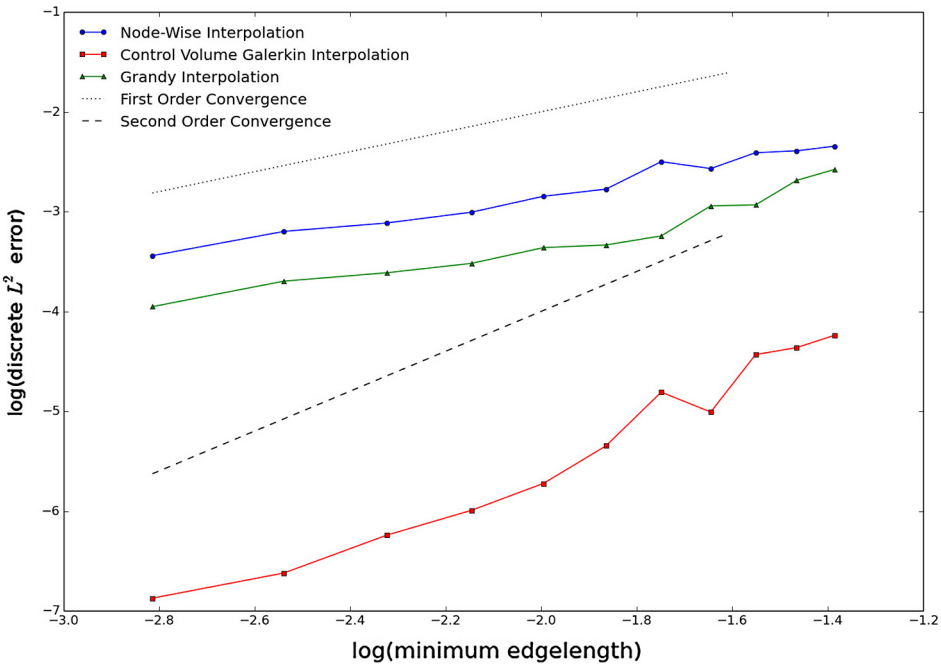


Fig. 8. Log–log plot of discrete L^2 error against minimum element edge-length for node-wise (circles), Grandy (triangles) and control-volume Galerkin interpolation (squares). The gradient of the straight line in this plot gives the order of convergence for a given interpolation method. Dashed lines indicating linear and quadratic convergence are also shown.

The initial scalar distribution together with an example target mesh are shown in Fig. 7 and the numerical parameters used in the simulation are shown in Table 2. The target resolution specifies the approximate number of elements in the target mesh and the minimum/maximum element edge-lengths bound the element sizes that enter the metric tensor in adaptivity. The initial mesh that the scalar field is initially defined on before being mapped onto the target mesh has 20000 elements (Fig. 7b). This mesh is by construction very fine so as to represent the Gaussian initial state with a very small error. A sequence of 11 simulations are performed (for each interpolation method in turn) varying the minimum element edge-length between 0.06 and 0.25.

A log–log plot of discrete L^2 error against minimum element edge-length for the three interpolation methods is shown in Fig. 8. It is clear that node-wise interpolation and Grandy interpolation exhibit slower than linear convergence, whilst the control-volume Galerkin method shows close to second order convergence and is hence a higher-order method. The order m of the trend line through the control-volume Galerkin data is $m \sim 1.85$, whilst $m \sim 0.77$ for node-wise interpolation and $m \sim 0.96$ for Grandy interpolation.

3.3. Advection of a square scalar field

The next step in complexity is to introduce a non-zero bulk velocity into the interpolation problem. We consider again initial data where the scalar is zero everywhere except on a (rotated) square. The system is initialised with a positive velocity in the x direction and the mesh is adapted to the curvature of the scalar. Since there is now advective transport of the scalar, the mesh changes with time automatically and there is no need to vary the minimum element edge-length to engineer this. Once again the results of interpolating via node-wise and Grandy interpolation are compared with those obtained by the control-volume Galerkin method presented in this paper.

The numerical parameters used in the simulation are specified in Table 3. The CFL number reported here corresponds to the largest value reached after approximately 140 adapts. The velocity is that of the initial square, and the size refers to the whole domain dimensions. Note the total domain length is purposely chosen to be much larger than the total distance travelled by the square during the simulation to minimise the influence of errors due to material advection through the outlet boundary on conservation integrals. Thus, the behaviour of the interpolation methods can be studied in a way that is decoupled as much as possible from the properties of the advection method. The resolution specifies the approximate number of elements in the initial mesh (this remains approximately constant throughout the simulation in this case for all interpolation methods except in the node-wise case, where the solution field is gradually destroyed) and the minimum/maximum edgelengths bound the element sizes that are permitted during adaptivity. Note that now the minimum element edge-length is fixed unlike in the previous example so there is no 'global' mesh refinement. The same P1DG pressure element as in the previous example is adopted together with the compressive advection scheme outlined at the start of Section 3.

Table 3
Parameters used in the square advection simulation described in the main text.

CFL Number	Velocity	Size	Resolution	Min/max edgelength
0.25	1.0	12.0×1.0	5000	0.01/0.05

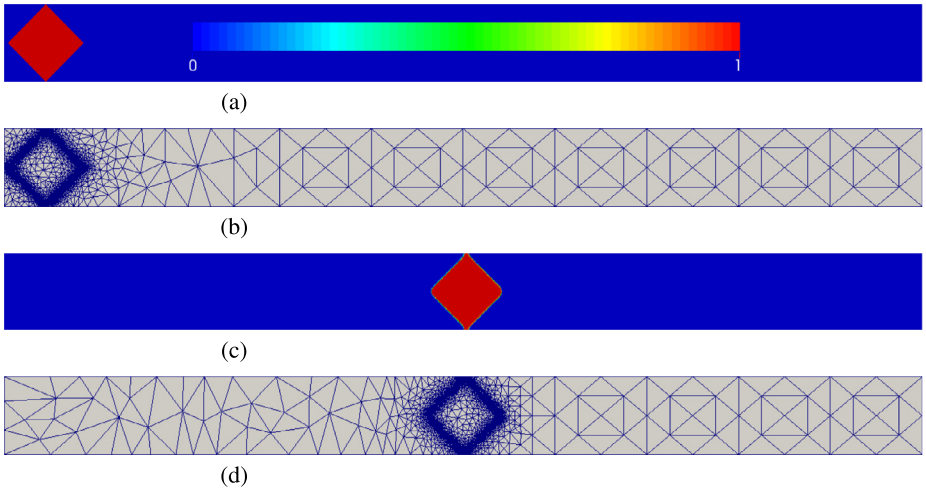


Fig. 9. (a) Initial scalar field which is non-zero in a square region of length 0.7, centred at $(0.05 + 0.7/\sqrt{2}, 0.5)$. (b) Initial mesh refined in order to sharply capture the inclined interfaces. (c) Solution field after approximately 140 mesh adapts corresponding to when the square has travelled half the length of the domain in this image computed via control-volume Galerkin interpolation. (d) Computational mesh corresponding to (c).

The initial state is shown in Fig. 9 and the results of the three interpolation methods after approximately 140 mesh adapts are shown in Fig. 10. Fig. 11(a) shows the integral of the scalar field over the domain for all three methods and Fig. 11(b) shows the L^2 error in the solutions. With advection included, there are now major qualitative differences in the results of the three interpolation methods at late times. In the case of node-wise interpolation, the solution field has been severely distorted. In the Grandy case, the initial state is well preserved in the interior of the square but the interfaces have become spread out. This behaviour can be attributed to the diffusivity associated with Grandy interpolation and can be problematic in certain classes of problems as will be illustrated in the final example discussed in this paper.

Of all the methods, control-volume Galerkin interpolation most closely retains the structure of the initial square, keeping both the uniform field values internally and the sharp interface boundaries. There is some overall distortion to the square-tip geometry common to all three methods but this can be attributed to numerical errors in advection and not the interpolation.

Fig. 11(a) verifies conservation in the Galerkin and Grandy cases and lack thereof in the case of node-wise interpolation. This example demonstrates that conservativeness is maintained even in scenarios with advection. Finally, Fig. 11(b) demonstrates that as expected, the error in the case of node-wise interpolation increases rapidly compared to the other two methods. The error is again smallest in the case of the control-volume Galerkin method, remaining approximately constant over the course of the simulation.

It is also instructive to compare the computational costs of the three interpolation methods in this example. The inclusion of advection, makes the problem non-trivial and the costs are representative of the general behaviour observed in other examples. The control-volume Galerkin method runs approximately 12% slower than the Grandy method, but correspondingly leads to an approximate 45% reduction in the L^2 error at the end of the simulation. Node-wise interpolation is in general very similar in cost to Grandy interpolation (only 1%–5% faster than the Grandy method) except in situations such as this example where the interpolant is discontinuous and the solution field is damaged by the interpolation. In these cases, the node-wise method is in fact much more costly (running approximately 54% slower than the Grandy method) as solving the equation system becomes progressively harder with time, ultimately leading to a very large error in the final state. These simulation times were obtained running in serial with a 2.6 GHz dual Intel Xeon processor and 128 GB of RAM. In summary, the control-volume Galerkin interpolation method is able to significantly reduce the interpolation error during mesh adaptivity compared to node-wise and Grandy interpolation whilst incurring little extra computational cost.

3.4. Advection of a circular scalar field

We now consider interpolation for a final pure advection problem, this time with a continuous interpolant. In this case, the 2D scalar field $\psi(x, y)$ is initialised to unity on a disc which is then set rotating in the anticlockwise direction for a quarter of a turn after which the sign of the velocity is changed and the disc is rotated back to its initial state. Since the flow is reversible, the final scalar field distribution should be identical to the initial state. We compare how closely node-wise

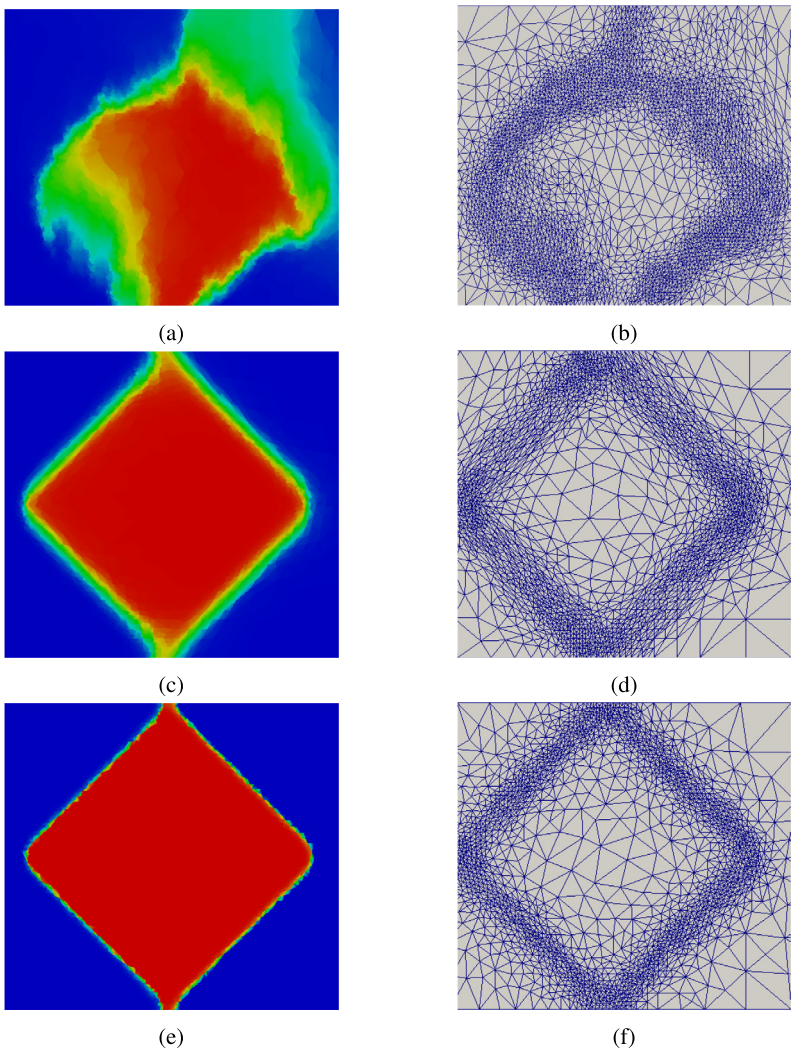


Fig. 10. Interpolated scalar field and mesh computed after approximately 140 mesh adapts by node-wise interpolation (a, b); Grandy interpolation (c, d) and control-volume Galerkin projection (e, f). The plots are zoomed in on the region of interest compared to Fig. 9.

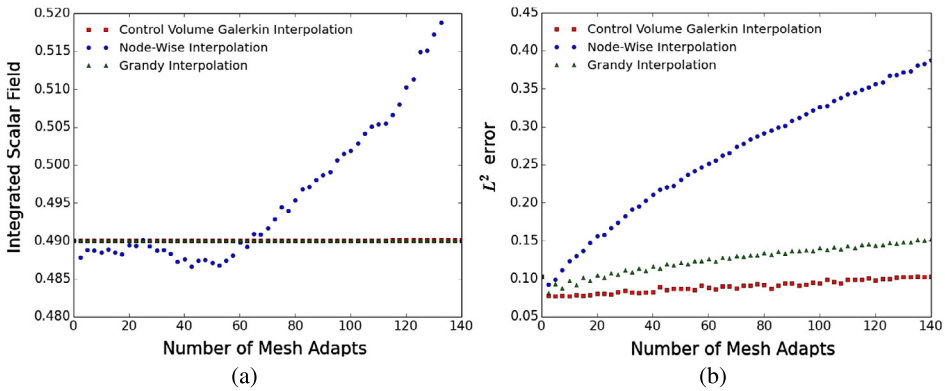
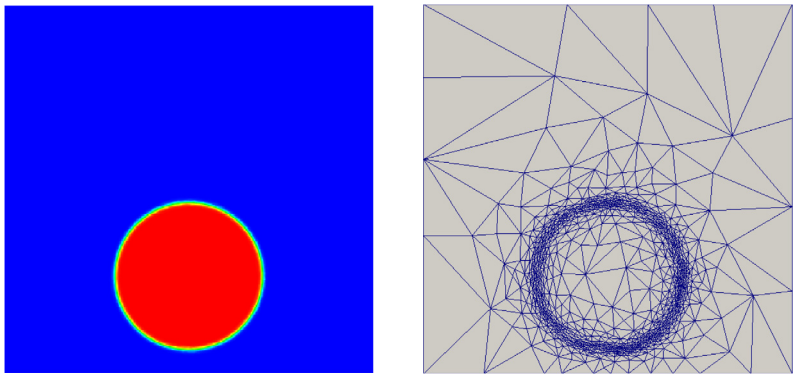


Fig. 11. (a) Integrated scalar field $\int \psi(x, y)dxdy$ against the number of mesh adapts as computed via control-volume Galerkin projection (squares) node-wise interpolation (circles) and Grandy interpolation (triangles) for the square advection problem described in the main text. (b) L^2 error in the interpolation of the scalar field $\psi(x, y)$ plotted as a function of the number of mesh adapts for control-volume Galerkin projection (squares), node-wise interpolation (circles) and Grandy interpolation (triangles).

Table 4
Parameters used in the circular advection simulation described in the main text.

CFL number	Velocity	Size	Initial/inter resolution	Min/max edgelength
0.3	$u = \begin{pmatrix} \sin(x)\cos(y) \\ -\cos(x)\sin(y) \end{pmatrix}$	$\pi \times \pi$	2000/4000	$\frac{\pi}{200} / \frac{\pi}{2}$



(a) (b)

Fig. 12. Initial circular scalar field distribution (a) and initial mesh (b). The disk has radius $r = \frac{\pi}{5}$ and is centered at $\left(\frac{\pi}{2}, \frac{\pi+1}{5}\right)$.

and control-volume Galerkin interpolation are able to capture this reversibility by analysing the scalar fields in the final state.

The initial state of the system is shown in Fig. 12 and the numerical settings used in the simulation are shown in Table 4. The Courant number reported here corresponds to the largest value reached. The initial resolution refers to the approximate number of elements in the initial/final mesh whilst the inter(mediate) resolution specifies the number of elements after a quarter turn before the velocity is reversed. The minimum/maximum edgelength control the element sizes in adaptivity. A P2CG pressure element was used in the simulation.

We show the results after a quarter turn in Fig. 13 and after returning to the initial state in Fig. 14. After this time 400 mesh adapts have taken place. In this example, there is again a qualitative difference between the results of the two interpolation methods. In particular, it is apparent from Fig. 14 that the control-volume Galerkin results more closely capture the original scalar field distribution. In particular, the node-wise interpolation results have developed oscillatory modes along the boundary of the disc that are not present in the initial state.

3.5. Viscous instabilities in two phase immiscible displacement in a porous medium

As a final and more complex application of the new interpolation method presented in this paper, we consider simulation of an immiscible two-phase displacement in a porous medium. This process is of great importance occurring in the water-flooding of oil reservoirs [20]. In such scenarios where a more viscous fluid is displaced by a less viscous fluid, it can be shown that the interface between the two phases is unstable to infinitesimal transverse perturbations. This instability may be viewed as analogous to the Saffman–Taylor instability observed in Hele-Shaw cell experiments but now in the context of two-phase porous media flow [21]. These perturbations can grow, becoming macroscopic viscous fingers. Simulation of viscous fingering is known to be challenging requiring a combination of higher-order numerical methods and very fine resolution to minimise the influence of numerical diffusion [22–24]. Mesh adaptivity is advantageous in such situations, allowing very fine mesh resolution to be maintained in the vicinity of the fingers as they grow with coarser resolution elsewhere [25].

When modelling unstable fluid phenomena with mesh adaptivity it is important that the interpolation method adopted is conservative since any fluctuations in field values introduced by interpolation can lead to artificial, unphysical perturbations that may grow and affect the solution field. In this example, mesh adaptivity is used to simulate viscous fingering at early-times, using a discontinuous representation of water saturation, and contrasting the results of the new interpolation method presented in the paper with those of Grandy interpolation. A discontinuous representation is used to maximise the effective number of control-volumes to capture the instability. Note we do not use node-wise interpolation since, as shown by the previous examples, it is unreliable in applications to discontinuous fields. This example also serves as a demonstration of when the Grandy method, which has hitherto performed well, can yield poor results due to its highly diffusive nature.

The equations governing two phase flow in a porous medium are the generalised Darcy law for the volumetric fluid flux q_α:

$$q_\alpha = -\frac{\mathbf{K} k_{r,\alpha}}{\mu_\alpha} \nabla p_\alpha \qquad (\alpha = o, w), \tag{23}$$

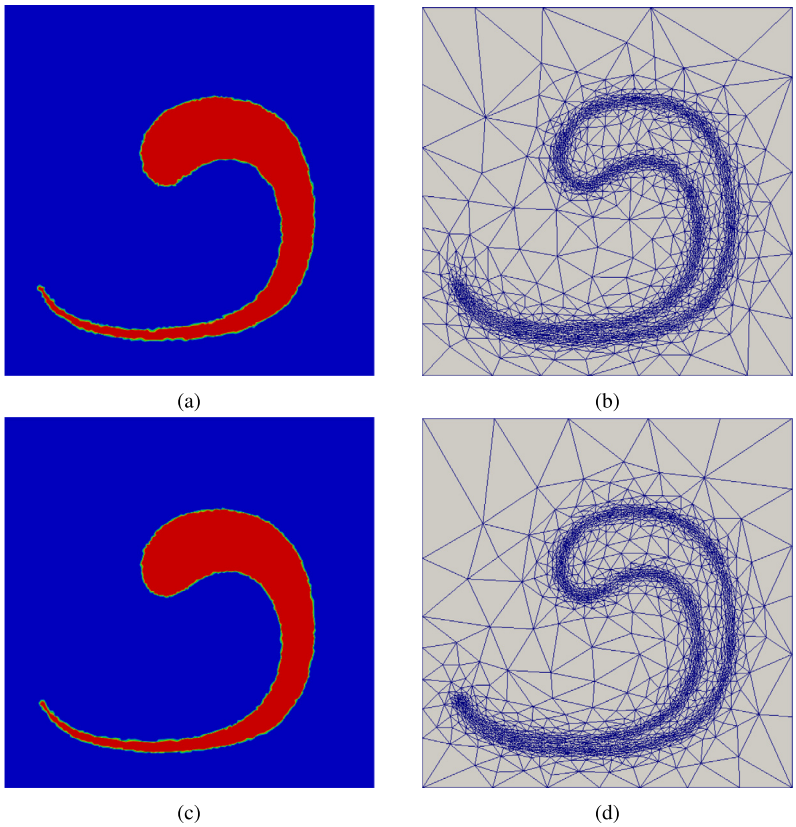


Fig. 13. Scalar field and mesh as computed after a quarter of a turn ($\theta = \pi/4$) corresponding to 200 adapts by node-wise interpolation (a, b) and control-volume Galerkin interpolation (c, d) at the point where the instantaneous velocity of the system is reversed.

together with the continuity equation:

$$\phi \frac{\partial S_\alpha}{\partial t} + \nabla \cdot q_\alpha = 0 \qquad (\alpha = o, w), \tag{24}$$

and finally the constraint that the volume fractions occupied by each of the two fluids sum to unity [26]:

$$S_o + S_w = 1. \tag{25}$$

In the above, $\mathbf{K}$ is the permeability tensor, α indexes the two fluids, $k_{r,\alpha}$ is the relative permeability of each fluid, p_α the pressure, S_α the phase volume fraction (saturation) and ϕ the porosity of the porous medium. Note that capillary pressure is neglected in the discussion here.

In the case of 1D symmetric solutions of the form $S(x, y) = S(x)$, Eq. (23) and Eq. (24) reduce to the Buckley–Leverett equation [20]:

$$\phi \frac{\partial S_w}{\partial t} + \frac{\partial}{\partial x}\left(\frac{k_{r,w}(S_w)}{k_{r,w}(S_w) + \frac{1}{M} k_{r,o}(S_w)} q_w \right) = 0 \tag{26}$$

where M is the viscosity ratio of the two fluids and q_w is the fluid inflow velocity. Eq. (26) admits shock solutions composed of an advancing compression wave together with a rarefaction. In order to simulate viscous fingering we are interested in the behaviour of pressure and saturation perturbations about solutions to the Buckley–Leverett equation (26). Solution of the linearised equations can be shown to reduce to an eigenvalue problem that relates the growth rate of perturbations to their frequency. Further theoretical discussions of viscous fingering can be found in for example [22,23,26–29].

We consider simulation of an immiscible displacement using mesh adaptivity to accurately resolve the shock front and the growth of viscous fingers. The simulation was carried out using a higher-order CVFE simulator developed at Imperial College detailed in [7,10]. The simulator solves a modified form of the equation system above (Eq. (23)–Eq. (25)) where a new velocity ($u_\alpha = q_\alpha / S_\alpha$), distinct from the Darcy velocity, is solved for. Velocity is eliminated from the equation system and the resulting pressure-saturation equation is then solved. The simulations were run with a P2DG-P1DG element (second order discontinuous finite element representation for velocity, first order discontinuous finite element representation for pressure). The numerical solution is interpolated between meshes after each adapt.

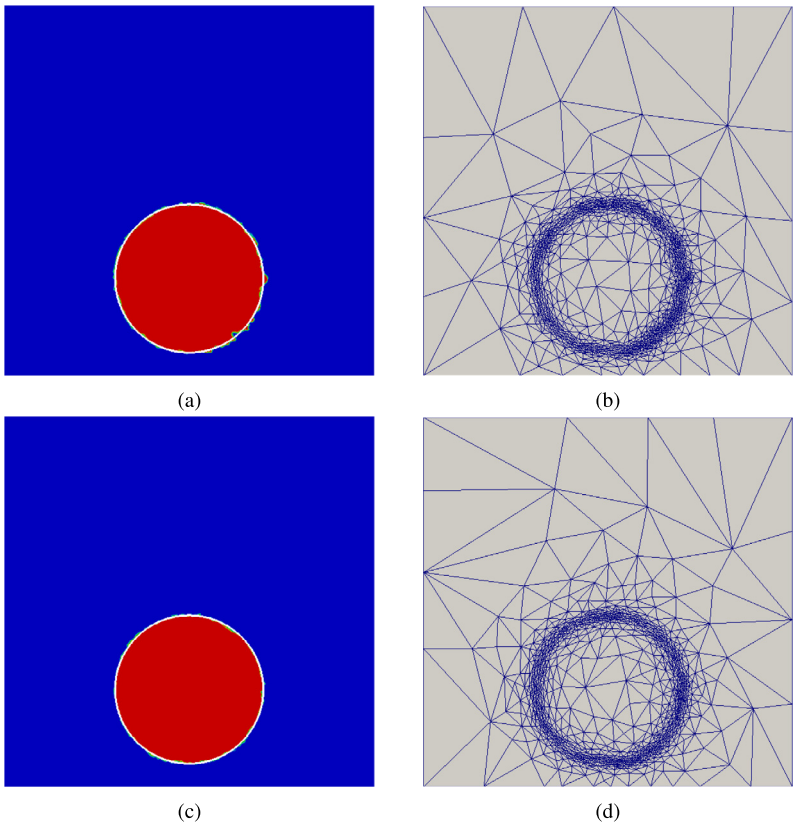


Fig. 14. Scalar field and mesh on returning to the initial orientation $\theta = 0$ after 400 adapts as computed by node-wise interpolation (a, b) and control-volume Galerkin interpolation (c, d). The initial state is overlaid on these images in white to compare how closely it is retained by the interpolation methods.

Table 5
Multiphase flow parameters used in the immiscible displacement simulation described in the main text. The last two columns specify the irreducible saturations S_{wr}, S_{or} of the two phases used in the relative permeability functions.

Viscosity ratio	Permeability	Porosity	Irred. sat. phase 1	Irred. sat. phase 2
$\mu = 303$	$K = 3.7 \times 10^{-13}$	$\phi = 0.205$	$S_{wr} = 0.3$	$S_{or} = 0.4$

Table 6
Simulation parameters used in the immiscible displacement simulation described in the main text.

CFL number	Velocity (m/s)	Size	Adaptive resolution	Min/max edgelength
0.125	2.5×10^{-5}	0.05×0.05	7000/10500	0.000375/0.005

As can be seen from Eq. (26), closure of the system of equations requires specification of the functional form of the relative permeabilities. A Corey-type correlation [30] given by [24] is used:

$$k_{r,w} = 0.06S^2, \qquad k_{r,o} = 0.74(1 - S), \qquad S = \frac{S_w - S_{wr}}{1 - S_{wr} - S_{or}} \tag{27}$$

where S_{wr} and S_{or} are the immobile fractions of the two fluids (the fractions of the two fluids that cannot be displaced). The remaining multiphase and numerical parameters used in the simulation are specified in Table 5 and Table 6 respectively. The displacing phase (water) is injected from the left with the specified velocity. The resolution specifies the approximate number of elements in the adaptive mesh. The maximum value 10500 is the number of elements after the first adapt, whilst the value 7000 corresponds to the approximate number of elements at the end of the simulation. The number of elements in the initial mesh is approximately 20000 whilst the whole domain fixed mesh we compare the adaptive results against has approximately 40000 elements.

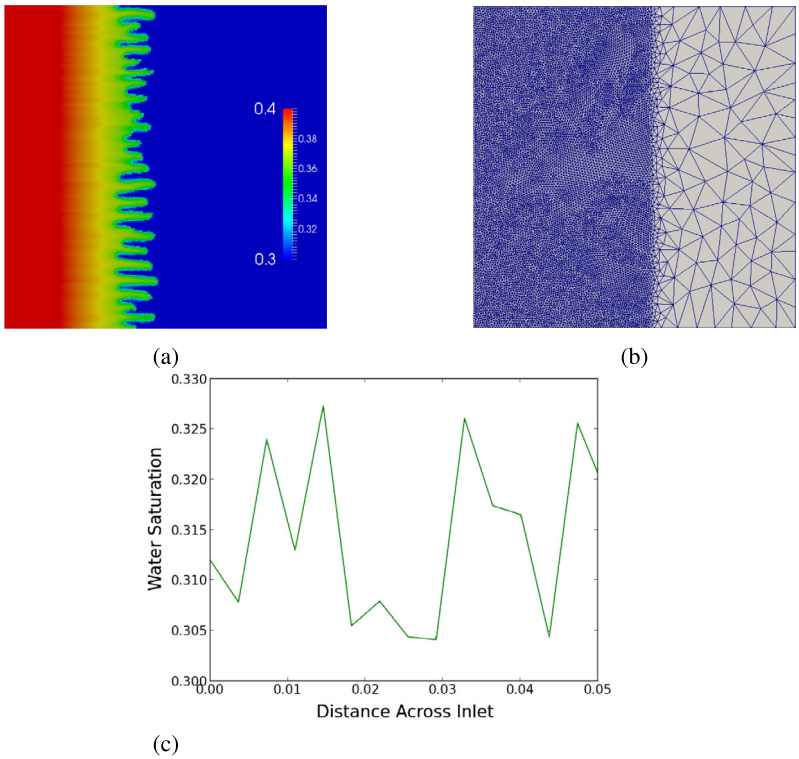


Fig. 15. (a) Water saturation profile immediately before adapting the mesh for the immiscible two-phase displacement simulation described in the main text. Water is injected from the left boundary and displaces the resident second phase (oil). (b) Initial mesh, by construction fine in the region where the viscous fingers are growing and coarser elsewhere. (c) Saturation perturbation used on the initial time-step to trigger viscous fingering.

Viscous fingering can be triggered numerically by introducing a perturbation to the saturation, pressure or permeability fields. It is triggered here by a saturation perturbation across the inlet boundary at the first time-step (shown in Fig. 15(c)). The form of the perturbation used is the same as that considered in [24]. This perturbation is allowed to grow on a fixed mesh until the fingers reach just under half-way across the domain at which point the mesh adapts to the water saturation field. The mesh is initially kept fixed so as to allow the fingering pattern time to form. Adapting too early before the fingering pattern has fully formed could damage the solution field. The saturation profile and mesh immediately before the first adapt are shown in Fig. 15 whilst the results after the fingers have nearly reached the outlet (after 680 mesh adapts) are shown in Fig. 16 for both methods of interpolation together with results on a fixed mesh (with edge-length equal to the minimum allowed in the adaptive simulations) at the same point in time.

It is clear from these results that the control-volume Galerkin method is able to preserve the structure of the viscous fingers after many adapts whilst in contrast the diffusivity of the Grandy method smears the fingering pattern, capturing only the large-scale structure of the front and diffusing away any small scale details. This behaviour can be attributed to the lower order nature of Grandy interpolation – smaller variations in field values are eroded during the interpolation process which leads to a runaway adaptive coarsening of the mesh.

Further evidence to support the conclusion that the control-volume Galerkin method is able to capture viscous fingering with comparable accuracy to results obtained on a fixed mesh can be seen in saturation cross sections through the fingering pattern for the three simulations. From Fig. 17(a) it is clear that mesh adaptivity with control-volume Galerkin interpolation produces very similar average saturations to those found in the fixed mesh simulations. In contrast, Grandy interpolation shows deviations from the fixed mesh in the region where the viscous fingers are forming. A water saturation cross section as a function of vertical spatial coordinate along a transverse slice ($x = 0.04$) that intersects the fingering pattern is shown in Fig. 17(b). The control-volume Galerkin results more closely match the pattern of maxima and minima in saturation seen on the fixed mesh when compared to the Grandy method which tends to average out nearby extrema. Similarly, the saturation cross section along a longitudinal slice ($y = 0.022$) is shown in Fig. 17(c). The control-volume Galerkin results again more closely match the fixed mesh results. The growth rate in particular appears to be under-predicted in the Grandy case with the fingers being too short. Finally, a plot of the water flux across the outlet boundary as a function of dimensionless simulation time is shown in Fig. 17(d). The water flux is initially zero until the viscous fingers reach the outlet boundary and cause water 'breakthrough'. The control-volume Galerkin interpolation results show a very similar breakthrough time to that seen on the fixed mesh whilst the Grandy results show a later breakthrough time due to the diffused fingering pattern.

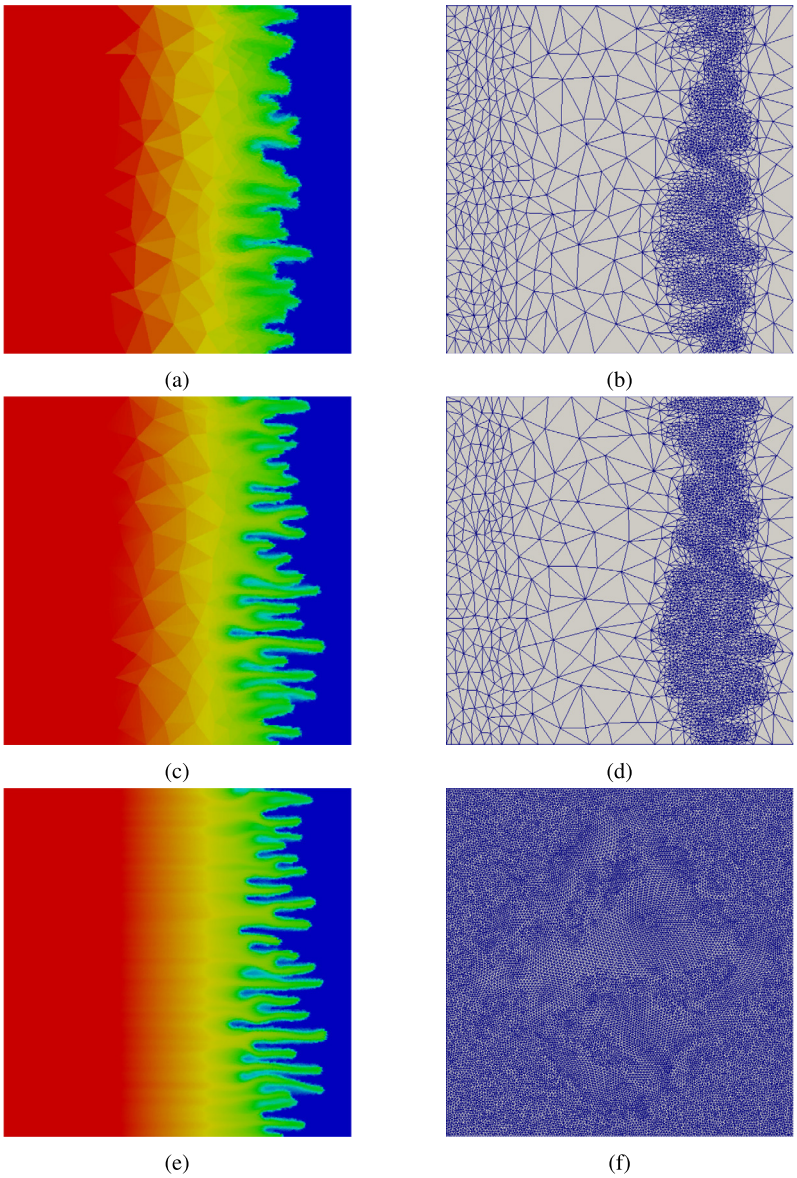


Fig. 16. Water saturation and computational mesh after 680 adapts as computed via Grandy interpolation (a, b) and control-volume Galerkin interpolation (c, d). At this time, the initial viscous finger distribution has grown substantially. The results of the same simulation performed on a fixed mesh are shown in (e, f) where the fixed mesh edge-length is equal to the minimum value in the adaptive simulations.

4. Conclusions

We have presented a general higher order, conservative and bounded method for interpolating fields between control-volume meshes. This is achieved by first mapping the control-volume field into a finite element representation on the donor mesh by Galerkin projection, then interpolating onto the target mesh by constructing a finite element supermesh from the intersection of the donor and target meshes and finally projecting back to a control-volume representation on the target mesh. Boundedness is achieved by using a simple diffusion algorithm on the resulting interpolated solution on the target mesh.

A series of test cases have demonstrated that the method is superior to both node-wise and Grandy interpolation for control-volume fields with a discontinuous dual finite element representation. Node-wise interpolation was shown to result in significant distortion in the solution field when using a discontinuous (higher order) interpolant. Grandy interpolation did not result in such distortion but was found to be much more diffusive than the Galerkin projection method. For the test case investigating the simulation of immiscible viscous fingering in porous media this meant that the Grandy interpolation smeared out the structure of the viscous fingers after a large number of mesh adapts. Overall the Galerkin projection method

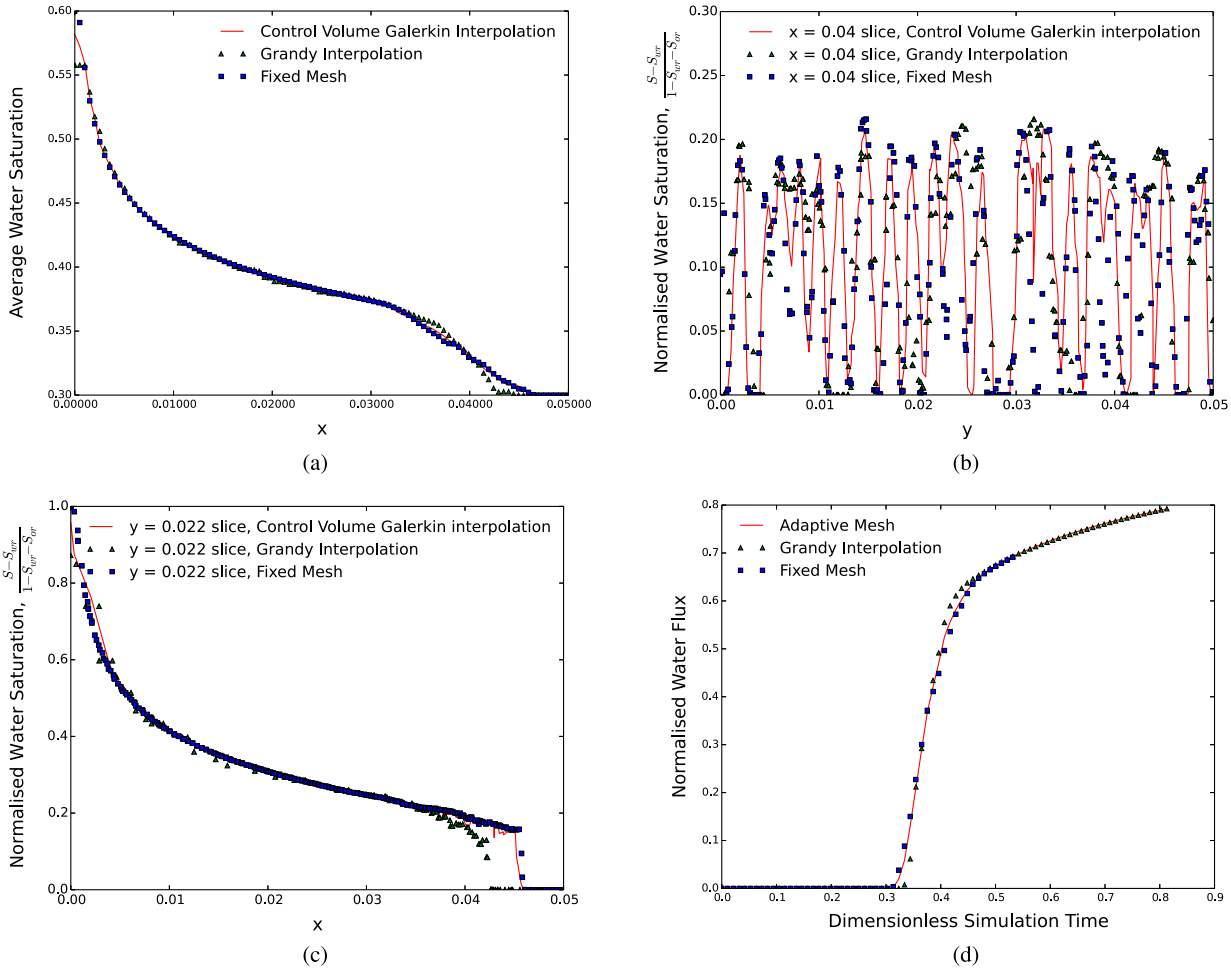


Fig. 17. (a) Average water saturation as a function of the horizontal spatial coordinate. (b) Water saturation cross section as a function of vertical spatial coordinate along a transverse slice ($x = 0.04$) that intersects the fingering pattern. (c) Water saturation cross section as a function of horizontal spatial coordinate along a longitudinal slice ($y = 0.022$). (d) Water flux across the outlet (right) boundary as a function of (dimensionless) simulation time.

has a lower error than both the node-wise and Grandy interpolation methods, when compared with a fine fixed mesh, and causes less numerical diffusion.

The error in the Galerkin projection method grows much more slowly than in both the node-wise and Grandy interpolation so would be the method of choice for simulations requiring frequent mesh adapts. It is slightly more CPU intensive that the Grandy method (12% slower in our test case) but the L^2 error is reduced at the end of the simulation (45% in this test case). The node-wise interpolation was even slower than the Galerkin projection method for discontinuous interpolants because it distorts the solution field making the equation system harder to solve as the simulation progresses. For lower order interpolants the performance of the node-wise interpolation was similar to that of the Grandy interpolation.

The proposed control-volume Galerkin projection method is currently the only conservative, higher order interpolation technique available for control-volume fields. The method was applied to a CVFE model in this paper but it should be possible to generalise this to other interpolations of control-volume fields.

Acknowledgements

Dr. A. Adam would like to thank and acknowledge TOTAL for funding this work. Funding for Dr. P. Salinas from ExxonMobil is also acknowledged. Dr. D. Pavlidis acknowledges support from the research grants: EPSRC 'Computational Modelling for Advanced Nuclear Power Plants' (EP/I003010/1) and EU/FP7 THINS (FP7-249337). Prof. C.C. Pain, Dr. Z. Xie and Dr. J.R. Percival acknowledge support from EPSRC 'Multi-Scale Exploration of Multiphase Physics in Flows: MEMPHIS' (EP/K003976/1). The research leading to these results has also received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 603663 for the research project PEARL,'Preparing for Extreme and Rare events in coastaL regions'. Part funding for Prof. M.D. Jackson and Prof. A.H. Muggeridge under the TOTAL Chairs programme at Imperial College is also acknowledged.

Appendix A. Higher-order convergence of the CV interpolation method

In this appendix, we provide arguments to support the claim that the control-volume interpolation introduced in this paper exhibits higher-order convergence. As discussed in Section 2.1, the method is composed of three steps: A map from a control-volume to finite element representation on the old mesh, a map between finite element representations on the old and new meshes and finally a mapping from a finite element representation back into control-volumes on the new mesh. The overall convergence of the method can be studied by bounding the interpolation errors in each of these three projections.

A.1. Error analysis for projections (1) and (3)

We begin by considering the error in the projection between control-volume and finite element meshes. In the case of a finite element representation, the errors in Eq. (4) and Eq. (7) are identically zero. This follows from the definition of finite element interpolation from or onto a control-volume mesh and can be seen clearly in Fig. 2 by regarding the error as the difference between the finite element interpolation and the initial control-volume functions.

A.2. Error analysis for projection (2)

We now consider the error in Eq. (6), the FEM Galerkin projection from old to new mesh. Let an element E_k be a tetrahedron over a 3D domain Ω, $\mathbf{I}_{new}$ and $\mathbf{I}_{old}$ are the interpolation operators over the new and old meshes respectively. $f = \psi_{FEM}^{new} - \psi_{FEM}^{old} \in H^1(\Omega)$ (where $\psi_{FEM}^{new} = \mathbf{I}_{new}\psi$ and $\psi_{FEM}^{old} = \mathbf{I}_{old}\psi$). According to the Poincaré inequality, there then exists a constant C, such that

$$\|\psi_{FEM}^{new} - \psi_{FEM}^{old}\|_{L_2(\Omega)} \leq C \sum_{k=1}^{\mathcal{K}} \sum_{i=1}^{d} h_i(E_k) \left\| \frac{\partial \psi}{\partial x_i} \right\|_{L_2(E_k)} \tag{A.1}$$

where $E_k \in (E_1, \ldots, E_{\mathcal{K}})$ ($\mathcal{K}$ is the total number of elements over the domain Ω), $d = 3$ is the dimensional size of the domain, $h_i(E_k)$ are suitably defined element sizes on the element E_k [31].

The error bound in Eq. (A.1) can be further estimated [32] as:

$$\|\psi_{FEM}^{new} - \psi_{FEM}^{old}\|_{L_2(\Omega)}^2 \leq C \sum_{k=1}^{\mathcal{K}} h^{2\kappa}(E_k) \sigma_{min}^{-2}(\mathbf{A}_{E_k}) V_{E_k} \tag{A.2}$$

where $\mathbf{A}_{E_k}$ is the edge shape matrix, σ_{min} is the minimal singular value of $\mathbf{A}_{E_k}$, κ is the order of approximation of the finite element degrees of freedom and V_{E_k} is the volume of the element E_k. Since we have argued that the convergence behaviour of the interpolation method is governed by the second step, we can conclude from the above that the method is higher-order. The error bound Eq. (A.2) is consistent with the higher-order numerical example in the main text. In that case, a first order discretisation ($\kappa = 1$) for pressure was used, and thus quadratic $h^2(E_k)$ convergence is expected, as seen in Fig. 8.

References

[1] J. Peraire, J. Peiró, K. Morgan, Adaptive remeshing for three-dimensional compressible flow computations, J. Comput. Phys. 103 (2) (1992) 269–285.
[2] C.C. Pain, A.P. Umpleby, C.R.E. de Oliveira, A.J.H. Goddard, Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations, Comput. Methods Appl. Mech. Eng. 190 (29–30) (2001) 3771–3796.
[3] P.L. George, H. Borouchaki, Delaunay Triangulation and Meshing: Application to Finite Elements, Hermes, Paris, 1998.
[4] D. Pavlidis, Z. Xie, J.R. Percival, J.L.M.A. Gomes, C.C. Pain, O.K. Matar, Two- and three-phase horizontal slug flow simulations using an interface-capturing compositional approach, in: A Collection of Papers in Honor of Professor G. Hewitt on the Occasion of his 80th Birthday, Int. J. Multiph. Flow 67 (Supplement) (2014) 85–91.
[5] J.R. Percival, D. Pavlidis, Z. Xie, J.L.M.A. Gomes, M. Sakai, Y. Shigeto, H. Takahashi, O.K. Matar, C.C. Pain, Control volume finite element modelling of segregation of sand and granular flows in fluidized beds, in: A Collection of Papers in Honor of Professor G. Hewitt on the Occasion of his 80th Birthday, Int. J. Multiph. Flow 67 (Supplement) (2014) 191–199.
[6] D. Pavlidis, J.L.M.A. Gomes, Z. Xie, J.R. Percival, C.C. Pain, O.K. Matar, Compressive advection and multi-component methods for interface-capturing, Int. J. Numer. Methods Fluids 80 (4) (2016) 256–282, fld. 4078.
[7] M.D. Jackson, J.R. Percival, P. Mostaghimi, B.S. Tollit, D. Pavlidis, C.C. Pain, J.L.M.A. Gomes, A.H. El-Sheikh, P. Salinas, A.H. Muggeridge, M.J. Blunt, Reservoir modeling for flow simulation by use of surfaces, adaptive unstructured meshes, and an overlapping-control-volume finite-element method, SPE Reserv. Eval. Eng. 18 (2015) 115–132.
[8] P. Mostaghimi, J.R. Percival, D. Pavlidis, R.J. Ferrier, J.L.M.A. Gomes, G.J. Gorman, M.D. Jackson, S.J. Neethling, C.C. Pain, Anisotropic mesh adaptivity and control volume finite element methods for numerical simulation of multiphase flow in porous media, Math. Geosci. 47 (4) (2015) 417–440.
[9] P. Salinas, J.R. Percival, D. Pavlidis, Z. Xie, J.L.M.A. Gomes, C.C. Pain, M.D. Jackson, A discontinuous overlapping control volume finite element method for multi-phase porous media flow using dynamic unstructured mesh optimization, in: Society of Petroleum Engineers, SPE-173279-MS, 2015, pp. 1511–1528.
[10] K. Su, J.P. Latham, D. Pavlidis, J. Xiang, F. Fang, P. Mostaghimi, J.R. Percival, C.C. Pain, M.D. Jackson, Multiphase flow simulation through porous media with explicitly resolved fractures, Geofluids 15 (2015) 592–607.
[11] P.E. Farrell, J.R. Maddison, Conservative interpolation between volume meshes by local Galerkin projection, Comput. Methods Appl. Mech. Eng. 200 (1–4) (2011) 89–100.

[12] P.E. Farrell, M.D. Piggott, C.C. Pain, G.J. Gorman, C.R. Wilson, Conservative interpolation between unstructured meshes via supermesh construction, Comput. Methods Appl. Mech. Eng. 198 (33–36) (2009) 2632–2642.
[13] R. Löhner, Robust, vectorized search algorithms for interpolation on unstructured grids, J. Comput. Phys. 118 (2) (1995) 380–387.
[14] D.R. Davies, J.H. Davies, O. Hassan, K. Morgan, P. Nithiarasu, Investigations into the applicability of adaptive finite element methods to two-dimensional infinite Prandtl number thermal and thermochemical convection, Geochem. Geophys. Geosyst. 8 (5) (2007).
[15] P. Clément, Approximation by finite element functions using local regularization, Rev. Fr. Autom. Inform. Rech. Opér., Anal. Numér. 9 (2) (1975) 77–84.
[16] C. Carstensen, Clément interpolation and its role in adaptive finite element error control, in: Partial Differential Equations and Functional Analysis, Springer, 2006, pp. 27–43.
[17] X. Jiao, M.T. Heath, Common-refinement-based data transfer between non-matching meshes in multiphysics simulations, Int. J. Numer. Methods Eng. 61 (14) (2004) 2402–2427.
[18] J. Grandy, Conservative remapping and region overlays by intersecting arbitrary polyhedra, J. Comput. Phys. 148 (2) (1999) 433–466.
[19] P.E. Farrell, Galerkin projection of discrete fields via supermesh construction, PhD thesis, Imperial College London, 2009.
[20] S.E. Buckley, M.C. Leverett, Mechanism of fluid displacement in sands, Trans. AIME 146 (01) (1942) 107–116.
[21] P.G. Saffman, G. Taylor, The penetration of a fluid into a porous medium or hele-shaw cell containing a more viscous liquid, Proc. R. Soc. Lond. A: Math. Phys. Eng. Sci. 245 (1958) 312–329, The Royal Society.
[22] G.M. Homsy, Viscous fingering in porous media, Annu. Rev. Fluid Mech. 19 (1) (1987) 271–311.
[23] A. Riaz, H.A. Tchelepi, Numerical simulation of immiscible two-phase flow in porous media, Phys. Fluids 18 (1) (2006) 014104.
[24] S. Jaure, A. Moncorge, R. de Loubens, Reservoir simulation prototyping platform for high performance computing, in: SPE Large Scale Computing and Big Data Challenges in Reservoir Simulation Conference and Exhibition, Society of Petroleum Engineers, 2014.
[25] P. Mostaghimi, F. Kamali, M.D. Jackson, A.H. Muggeridge, C.C. Pain, A dynamic mesh approach for simulation of immiscible viscous fingering, in: SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, 2015.
[26] Y.C. Yortsos, F.J. Hickernell, Linear stability of immiscible displacement in porous media, SIAM J. Appl. Math. 49 (3) (1989) 730–748.
[27] Y.C. Yortsos, A.B. Huang, Linear-stability analysis of immiscible displacement: Part 1-simple basic flow profiles, SPE Reserv. Eng. 1 (4) (1986) 378–390.
[28] R.L. Chuoke, P. Van Meurs, C. van der Poel, The instability of slow, immiscible, viscous liquid–liquid displacements in permeable media, in: Society of Petroleum Engineers, SPE-1141-G, 1959.
[29] A. Riaz, H.A. Tchelepi, Linear stability analysis of immiscible two-phase flow in porous media with capillary dispersion and density variation, Phys. Fluids 16 (12) (2004) 4727–4737.
[30] R.H. Brooks, A.T. Corey, Hydraulic properties of porous media, in: Hydrology Papers, No. 3, Colorado State University, Collins, Colo, 1964.
[31] Th. Apel, M. Berzins, P.K. Jimack, G. Kunert, A. Plaks, I. Tsukerman, M. Walkley, Mesh Shape and Anisotropic Elements: Theory and Practice, 2000.
[32] Igor Tsukerman, Approximation of conservative fields and the element 'edge shape matrix', IEEE Trans. Magn. 34 (5) (1998) 3248–3251.