

Fast optimization-based conservative remap of scalar fields through aggregate mass transfer



Pavel Bochev^{a,*}, Denis Ridzal^{b,1}, Mikhail Shashkov^{c,2}

^a Numerical Analysis and Applications, Sandia National Laboratories, MS-1320, Albuquerque, NM 87185-1320, USA

^b Optimization and Uncertainty Quantification, Sandia National Laboratories, MS-1320, Albuquerque, NM 87185-1320, USA

^c XCP-4, Methods and Algorithms, Los Alamos National Laboratory, MS-F644, Los Alamos, NM 87545, USA

ARTICLE INFO

Article history:

Received 10 May 2012

Received in revised form 20 February 2013

Accepted 19 March 2013

Available online 4 April 2013

Keywords:

Constrained interpolation

Remap

Flux-corrected remap

FCT

Optimization-based remap

Quadratic programming

ABSTRACT

We develop a fast, efficient and accurate optimization-based algorithm for the high-order conservative and local-bound preserving remap (constrained interpolation) of a scalar conserved quantity between two close meshes with the same connectivity. The new formulation is as robust and accurate as the flux-variable flux-target optimization-based remap (FVFT-OB) [1,2] yet has the computational efficiency of an explicit remapper. The coupled system of linear inequality constraints, resulting from the flux form of remap, is the main efficiency bottleneck in FVFT-OB. While advection-based remappers use the flux form to directly enforce mass conservation, the optimization setting allows us to treat mass conservation as one of the constraints. To take advantage of this fact, we consider an alternative mass-variable mass-target (MVMT-OB) formulation in which the optimization variables are the net mass updates per cell and a single linear constraint enforces the conservation of mass. In so doing we change the structure of the OB problem from a global linear-inequality constrained QP to a singly linearly constrained QP with simple bounds. Using the structure of the MVMT-OB problem, and the fact that in remap the old and new grids are close, we are able to develop a simple, efficient and easily parallelizable optimization algorithm for the primal MVMT-OB QP. Numerical studies on a variety of affine and non-affine grids confirm that MVMT-OB is as accurate and robust as FVFT-OB, but has the same computational cost as the explicit, state-of-the-art FCR.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

We develop a fast, efficient and accurate optimization-based algorithm for the high-order conservative and local-bound preserving remap (constrained interpolation) of a scalar conserved quantity between two close meshes with the same connectivity. This task originates in Arbitrary Lagrangian–Eulerian (ALE) methods [3], which are the main motivation for our work. In the ALE context we are given the mean value of the primitive variable (an unknown positive scalar function, such as density) on each cell of the old (Lagrangian) mesh. The conserved variable, such as mass, is the product of this mean value and the cell volume. The objective is to find an accurate approximation of the mass on the new (rezoned) mesh. The

* Corresponding author. Tel.: +1 505 844 1990.

E-mail addresses: pbboche@sandia.gov (P. Bochev), dridzal@sandia.gov (D. Ridzal), shashkov@lanl.gov (M. Shashkov).

URLs: <http://www.sandia.gov/pbboche/> (P. Bochev), <http://www.sandia.gov/dridzal/> (D. Ridzal), <http://cnls.lanl.gov/shashkov/> (M. Shashkov).

¹ Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the US Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

² This work was carried out under the auspices of the National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396.

remapped cell mass divided by the volume of the new cell approximates the density, which must satisfy physically motivated bounds.

The paper continues the development and study of the optimization-based remap (OBR) approach initiated in [1,2]. There we rephrase remap as a global inequality-constrained quadratic program (QP) for the mass fluxes exchanged between neighboring cells. The objective is to minimize the distance between these fluxes and some given *target* fluxes subject to constraints that enforce physically motivated local bounds on the primitive variable (density). The resulting flux-variable flux-target (FVFT) OBR has valuable theoretical and computational properties, which set it apart from advection-based [4–6], mass “repair” [7,8], or flux-correction motivated [9] algorithms.

In a nutshell, these methods invoke local “worst-case” scenarios to preserve the local bounds through monotone reconstruction, mass redistribution, or convex combinations of low and high-order fluxes. This entangles accuracy considerations with the enforcement of the bounds, which tends to obscure the sources of discretization errors and complicates the analysis of the algorithms.

In contrast, the OBR strategy completely separates accuracy from the enforcement of the physically motivated local bounds. The latter define a feasible set for the QP, whereas the minimization of the objective function enforces the former. As a result, FVFT-OBR always finds a globally optimal, i.e., *the best possible*, with respect to the target fluxes, remapped mass that also satisfies these bounds. The linear constraints that express the local bounds are completely impervious to cell shape and so, OBR is applicable to arbitrary grids, including polygonal and polyhedral grids.

Thorough convergence studies in [1] confirm that the FVFT-OBR formulation is as accurate as the state-of-the-art Flux-Corrected Remap (FCR) [9] for a collection of classical remap test cases. However, a series of “torture” tests in one and two dimensions demonstrate that FVFT-OBR is significantly more robust than FCR and the representative advection-based remappers. The dual QP provides a convenient structure-exploiting setting for the effective solution of FVFT-OBR by the reflective Newton method [10]. Numerical studies in [1] indicate that the computational cost of FVFT-OBR is proportional, up to a constant, to the cost of an explicit remapper such as FCR. The proportionality constant observed in [1] varies between 1.8 and 3.2.

These figures do not count potential gains from the increased robustness of FVFT-OBR, which enables larger displacements between the old and new mesh. Nonetheless, further efficiency gains in OBR are desirable to enhance its standing as a viable competitor to explicit remappers. Accordingly, the main focus of this paper is on improving the computational efficiency of OBR. Our principal goal is to develop an OBR formulation that fully retains the robustness and accuracy of FVFT-OBR, yet has the computational efficiency of the *explicit*, state-of-the-art FCR.

The main efficiency bottleneck in FVFT-OBR is the coupled system of linear inequality constraints. The coupling of the variables stems from writing the new cell masses in flux form, which automatically conserves the total mass. While the flux form is needed for conventional remappers, it is arguably less critical in an optimization setting where conservation of mass can be treated as an explicit constraint. To take advantage of this fact, we consider an alternative *mass-variable mass-target* OBR (MVMT-OBR) formulation in which the optimization variables are the net mass updates per cell. While this formulation introduces a single global equality constraint to conserve mass, it completely decouples the inequality constraints because there is only one variable per cell. As a result, by switching to these new variables we change the structure of the OBR problem from a global linear-inequality constrained QP to a singly linearly constrained QP with simple bounds.

Of course, trading automatic mass conservation for an explicit constraint only makes sense if the resulting QP can be solved more efficiently. Using the structure of the MVMT-OBR problem, and the fact that in remap the old and the new grids are close, we are able to develop a simple, efficient and easily parallelizable algorithm for the primal MVMT-OBR QP. Preliminary studies of the computational cost suggest that this algorithm is as efficient as the explicit FCR. This makes the new formulation fully competitive in terms of speed with any conventional remapper. At the same time, thorough computational studies confirm that the MVMT-OBR formulation retains the accuracy and robustness of FVFT-OBR.

The use of the fully decoupled net mass update variables instead of the coupled flux variables opens up an additional possibility to simplify and improve MVMT-OBR by discarding the variables associated with static cells, i.e., the cells that do not move during the rezone step. Since the net mass update on a static cell equals zero, we can increase the efficiency of MVMT-OBR by not computing the variables whose values are known to be zero. This modification could bring about significant additional performance gains in applications such as propagation of waves, in which there are large numbers of static cells. We show that skipping the static cells in the MVMT-OBR formulation does not lead to a loss of key theoretical properties such as existence of optimal solutions and preservation of linearity.

The remainder of this section introduces the relevant notation. Section 2 presents the MVMT-OBR formulation. There we also prove that MVMT-OBR is well-posed and preserves linear functions. The section closes with a discussion of the swept region implementation of MVMT-OBR. Section 3 develops the optimization algorithm for the solution of the MVMT-OBR QP. Numerical studies in Section 4 focus on the accuracy, robustness and efficiency of MVMT-OBR. We summarize our conclusions and map directions for future work in Section 5.

1.1. Notation

The computational domain Ω is a bounded subset of \mathbf{R}^d , $d = 1, 2, 3$. The old (Lagrangian) grid $K_h(\Omega)$ is a conforming partition of Ω into cells κ_i , $i = 1, \dots, K$. The total numbers of vertices, edges and sides in the mesh are V , E and S , respectively.

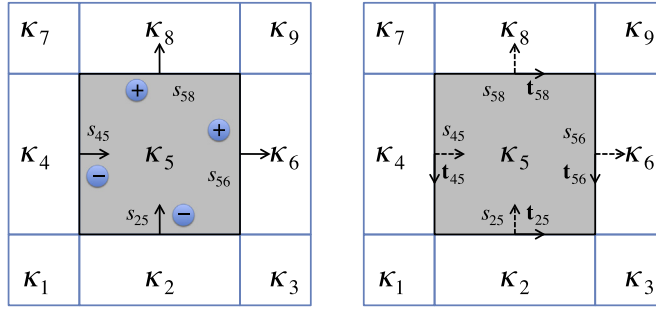


Fig. 1. Illustration of the naming and the orientation conventions for cell sides in two dimensions. The sides are s_{ij} with $i < j$. The unit normal n_{ij} on s_{ij} points away from cell κ_i . The unit tangent t_{ij} is such that the pair (t_{ij}, n_{ij}) is positively oriented. The blue circles are the signs of the non-zero entries in the row of the side-to-cell incidence matrix corresponding to κ_5 . (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The sets of all vertices, sides and cells in an entity Ξ are $V(\Xi)$, $S(\Xi)$, and $C(\Xi)$, respectively. For instance, $S(K_h)$ are the sides in the old mesh and $V(\kappa_i)$ are the vertices of cell κ_i .

When two cells κ_i and κ_j have a common side we label it by s_{ij} . Because s_{ij} and s_{ji} represent the same entity in the mesh, to avoid ambiguity we adopt the convention $i < j$. For instance, referring to the cartoon in Fig. 1, we label the side between κ_5 and κ_2 by s_{25} , the side between κ_5 and κ_6 by s_{56} and so on. This rule induces the orientation of s_{ij} by selecting the unit normal on the side that points *towards* the cell with the larger number. Returning to the example in Fig. 1, we orient s_{25} using the unit normal that points into κ_5 , while for s_{56} we use the unit normal that points towards κ_6 . The side-to-cell incidence matrix \mathbf{D} is a $K \times S$ matrix with entries $d_{i,kl}$ such that $d_{i,kl} = 0$ if s_{kl} is not a side of κ_i , $d_{i,kl} = 1$ if the unit normal to the side s_{kl} matches the direction of the outer unit normal to the boundary of κ_i , and $d_{i,kl} = -1$ otherwise. Succinctly,

$$d_{i,kl} = \begin{cases} 0 & \text{if } s_{kl} \notin S(\kappa_i), \\ 1 & \text{if } s_{kl} \in S(\kappa_i) \text{ and } i = k, \\ -1 & \text{if } s_{kl} \in S(\kappa_i) \text{ and } i = l. \end{cases} \quad (1.1)$$

The blue ovals in Fig. 1 illustrate this rule.

In two dimensions we also need to select a unit tangent on each side. We follow the rule that the unit tangent t_{ij} on s_{ij} is such that the pair (t_{ij}, n_{ij}) has positive orientation, i.e., $\det([t_{ij}, n_{ij}]) = 1$.

The new (rezoned) grid $\tilde{K}_h(\Omega)$ is another conforming partition of Ω into cells $\tilde{\kappa}_i$, $i = 1, \dots, \tilde{K}$. In this paper we restrict attention to pairs of new and old grids having the same connectivity. Therefore, $K_h(\Omega)$ and $\tilde{K}_h(\Omega)$ have the same numbers of vertices, sides and cells, i.e., $\tilde{V} = V$, $\tilde{S} = S$ and $\tilde{K} = K$. Without loss of generality we assume that the vertices \tilde{v}_i , the sides \tilde{s}_{kl} and the cells $\tilde{\kappa}_j$ in $\tilde{K}_h(\Omega)$ are numbered by the same numbers as in $K_h(\Omega)$, i.e., their Lagrangian prototypes are v_i , s_{kl} and κ_j , respectively.

The quantities and the entities on the new grid will have the tilde accent, e.g. \tilde{f} , whereas the quantities and the entities on $K_h(\Omega)$ will have no accent. For instance,

$$\mu_i = \int_{\kappa_i} dV \quad \text{and} \quad \tilde{\mathbf{b}}_j = \frac{\int_{\tilde{\kappa}_j} \mathbf{x} dV}{\tilde{\mu}_j} \quad (1.2)$$

are the *unsigned* measure of the old cell κ_i and the barycenter of the new cell $\tilde{\kappa}_j$, respectively. The *signed* measure³ of a set γ is $\mu^\star(\gamma)$.

The neighborhood $N(\kappa_i)$ of κ_i includes the cell κ_i itself and all cells that share a vertex (in 1D), vertex or side (in 2D) and vertex, edge or side (in 3D) with κ_i . The *strict neighborhood* of κ_i is defined by $N'(\kappa_i) = N(\kappa_i) \setminus \kappa_i$. The *side neighborhood* $N_S(\kappa_i)$ comprises of all cells in $N(\kappa_i)$ which share a side with κ_i . The subset of cells in $N(\kappa_i)$ that share an edge but not a side with κ_i is the *edge neighborhood* $N_E(\kappa_i)$ and the subset of cells that shares a vertex but not an edge or a side with κ_i is the *vertex neighborhood* $N_V(\kappa_i)$. The strict versions $N'_S(\kappa_i)$, $N'_E(\kappa_i)$ and $N'_V(\kappa_i)$ of these neighborhoods do not include κ_i . Fig. 2 gives an example of $N(\kappa_i)$, $N'(\kappa_i)$, $N_S(\kappa_i)$ and $N_V(\kappa_i)$ in two dimensions.

We assume that $\tilde{K}_h(\Omega)$ satisfies the *locality condition*

$$\tilde{\kappa}_i \subset N(\kappa_i), \quad \text{for all } i = 1, \dots, K, \quad (1.3)$$

which makes precise the assumption that $K_h(\Omega)$ and $\tilde{K}_h(\Omega)$ are “close”.

Suppose that \mathcal{F} and $\tilde{\mathcal{F}}$ are some collections of old and new cells, respectively. The notation $i \in \mathcal{F}$, $i \in \tilde{\mathcal{F}}$ means that the index i runs over the indices of the cells in \mathcal{F} , resp. $\tilde{\mathcal{F}}$. For instance, $j \in N(\kappa_i)$ means that j loops over the indices of the cells in the neighborhood of κ_i .

³ Reduction of the volume integral to a surface integral and taking into account the surface orientation defines the signed measure of a domain.

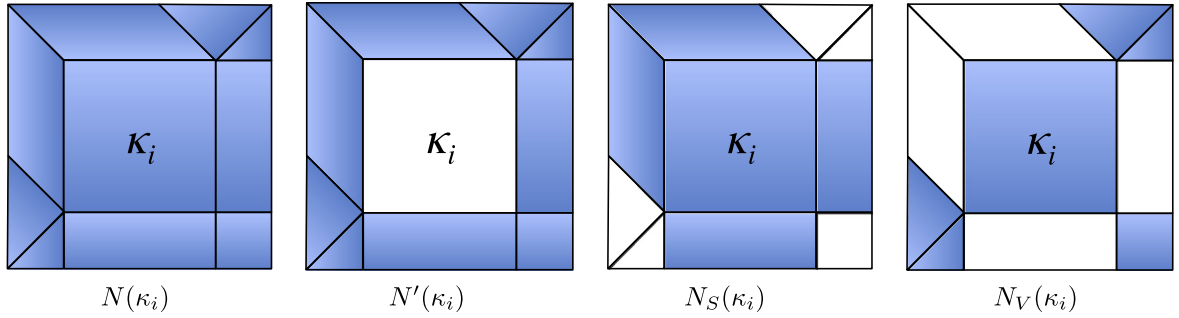


Fig. 2. Lagrangian cell κ_i and its neighborhoods.

Remark 1.1. Because a side is shared by at most two cells, dimension of $N_S(\kappa_i)$ for interior cells is exactly the number of sides in κ_i plus one. In contrast, more than two cells can share an edge in 3D and a vertex in 2D and 3D. As a result, the dimensions of $N_E(\kappa_i)$ (in 3D) and $N_V(\kappa_i)$ (in 2D and 3D) can be arbitrary. Fig. 2 shows an example where the number of cells from $N_V(\kappa_i)$ attached to each of the nodes of κ_i ranges from 0 to 2.

2. Formulation of the optimization-based remap through aggregate mass transfer

We begin with a formal statement of the mass-density remap problem. Then we review the flux-variable flux-target OBR (FVFT-OBR) and proceed to define the mass-variable mass-target OBR (MVMT-OBR). We show that MVMT-OBR is well-posed and preserves linear densities. Discussion of implementation using swept region approximation completes the section.

2.1. Statement of the mass-density remap problem

The statement of the mass-density remap problem [11,9] requires additional notation. The primitive variable (density) is a scalar function $\rho(\mathbf{x}) > 0$ on Ω . The conserved variable is the total mass

$$M = \int_{\Omega} \rho(\mathbf{x}) dV = \sum_{i=1}^K m_i = \sum_{i=1}^K \rho_i \mu_i,$$

where

$$\rho_i = \frac{\int_{\kappa_i} \rho(\mathbf{x}) dV}{\mu_i} = \frac{m_i}{\mu_i} \quad \text{and} \quad m_i = \int_{\kappa_i} \rho(\mathbf{x}) dV = \rho_i \mu_i, \quad i = 1, 2, \dots, K \quad (2.1)$$

are the mean densities and cell masses on the old mesh, respectively. For every old cell κ_i define

$$\rho_i^{\min} = \begin{cases} \min_{j \in N(\kappa_i)} \{\rho_j\} & \text{if } \kappa_i \cap \partial\Omega = \emptyset \\ \min \left\{ \min_{j \in N(\kappa_i)} \{\rho_j\}, \min_{\mathbf{x} \in N(\kappa_i) \cap \partial\Omega} \rho(\mathbf{x}) \right\} & \text{if } \kappa_i \cap \partial\Omega \neq \emptyset \end{cases}, \quad (2.2)$$

$$\rho_i^{\max} = \begin{cases} \max_{j \in N(\kappa_i)} \{\rho_j\} & \text{if } \kappa_i \cap \partial\Omega = \emptyset \\ \max \left\{ \max_{j \in N(\kappa_i)} \{\rho_j\}, \max_{\mathbf{x} \in N(\kappa_i) \cap \partial\Omega} \rho(\mathbf{x}) \right\} & \text{if } \kappa_i \cap \partial\Omega \neq \emptyset. \end{cases} \quad (2.3)$$

It is straightforward to check that the mean density trivially satisfies the local bounds

$$\rho_i^{\min} \leq \rho_i \leq \rho_i^{\max}, \quad i = 1, \dots, K. \quad (2.4)$$

The inequalities (2.4) together with (2.1) imply that cell masses satisfy similar local bounds:

$$m_i^{\min} := \rho_i^{\min} \mu_i \leq m_i \leq \rho_i^{\max} \mu_i =: m_i^{\max}, \quad i = 1, \dots, K. \quad (2.5)$$

In the mass-density remap problem we assume that we know the mean density values ρ_i on the *old* grid cells κ_i , and the boundary values of $\rho(\mathbf{x})$ on $\partial\Omega$. Given this information, we seek accurate approximations \tilde{m}_i for the masses of the *new* cells $\tilde{\kappa}_i$,

$$\tilde{m}_i \approx \tilde{m}_i^{\text{ex}} = \int_{\tilde{\kappa}_i} \rho(\mathbf{x}) dV, \quad i = 1, \dots, K, \quad (2.6)$$

such that the following conditions hold:

C1. The total mass is conserved:

$$\sum_{i=1}^K \tilde{m}_i = \sum_{i=1}^K m_i = M.$$

C2. If $\rho(\mathbf{x})$ is a global linear function on Ω , then the remapped masses are exact:

$$\tilde{m}_i = \tilde{m}_i^{\text{ex}} = \int_{\tilde{K}_i} \rho(\mathbf{x}) dV, \quad i = 1, \dots, K. \quad (2.7)$$

C3. The approximation of the mean density on the new cells

$$\tilde{\rho}_i = \frac{\tilde{m}_i}{\tilde{\mu}_i}, \quad (2.8)$$

satisfies the local bounds

$$\rho_i^{\min} \leq \tilde{\rho}_i \leq \rho_i^{\max}; \quad i = 1, \dots, K, \quad (2.9)$$

where (2.2) and (2.3) define ρ_i^{\min} and ρ_i^{\max} . Equivalently, the following local mass bounds hold:

$$\tilde{m}_i^{\min} := \rho_i^{\min} \tilde{\mu}_i \leq \tilde{m}_i \leq \rho_i^{\max} \tilde{\mu}_i =: \tilde{m}_i^{\max}. \quad \square \quad (2.10)$$

2.2. Flux-variable flux-target optimization-based remap

Recall the strict neighborhood $N'(\kappa_i) = N(\kappa_i) \setminus \kappa_i$. The *flux-form* formula [11]

$$\tilde{m}_i^{\text{ex}} = m_i + \sum_{j \in N'(\kappa_i)} F_{ij}^{\text{ex}}, \quad (2.11)$$

where

$$F_{ij}^{\text{ex}} = \int_{\tilde{K}_i \cap \kappa_j} \rho(\mathbf{x}) dV - \int_{\kappa_i \cap \tilde{K}_j} \rho(\mathbf{x}) dV, \quad j \in N'(\kappa_i) \quad (2.12)$$

are the exact *mass fluxes*, holds on any two grids that satisfy (1.3). The substitution of the exact fluxes $\{F_{ij}^{\text{ex}}\}$ in (2.11) with antisymmetric approximations $\{F_{ij}\}$, $F_{ij} = -F_{ji}$ yields the mass-conserving flux-form formula

$$\tilde{m}_i = m_i + \sum_{j \in N'(\kappa_i)} F_{ij} \quad (2.13)$$

for the approximation of the new masses. The discrete fluxes $\{F_{ij}\}$ are the variables in the FVFT-OB formulation and the constraints are (i) the antisymmetry $F_{ij} = -F_{ji}$ of the discrete fluxes and (ii) the local mass bounds (2.10). We enforce antisymmetry directly by using only fluxes F_{ij} with $i < j$ and writing the flux sum in (2.13) as

$$\sum_{j \in N'(\kappa_i)} F_{ij} = \sum_{\substack{j \in N'(\kappa_i) \\ i < j}} F_{ij} - \sum_{\substack{j \in N'(\kappa_i) \\ i > j}} F_{ji}.$$

The FVFT-OB solution minimizes the Euclidean distance between F_{ij} and some suitable target fluxes F_{ij}^T , $i < j$, subject to (2.10). Concisely, the FVFT-OB formulation is the following global inequality-constrained QP [1]:

$$\begin{cases} \min_{F_{ij}} \sum_{i=1}^K \sum_{\substack{j \in N'(\kappa_i) \\ i < j}} (F_{ij} - F_{ij}^T)^2 & \text{subject to} \\ \delta \tilde{m}_i^{\min} \leq \sum_{\substack{j \in N'(\kappa_i) \\ i < j}} F_{ij} - \sum_{\substack{j \in N'(\kappa_i) \\ i > j}} F_{ji} \leq \delta \tilde{m}_i^{\max}, & i = 1, \dots, K, \end{cases} \quad (2.14)$$

where

$$\delta \tilde{m}_i^{\min} = \tilde{m}_i^{\min} - m_i \quad \text{and} \quad \delta \tilde{m}_i^{\max} = \tilde{m}_i^{\max} - m_i. \quad (2.15)$$

A density reconstruction $\rho^h(\mathbf{x})$ on $K_h(\Omega)$ that is exact for linear functions defines the target fluxes in (2.14):

$$F_{ij}^T := \int_{\tilde{K}_i \cap \kappa_j} \rho_j^h(\mathbf{x}) dV - \int_{\kappa_i \cap \tilde{K}_j} \rho_i^h(\mathbf{x}) dV, \quad i < j. \quad (2.16)$$

Because our goal is to preserve linear densities, in this paper we restrict attention to piecewise linear density reconstructions, i.e., functions $\rho^h(\mathbf{x})$ whose restriction $\rho^h(\mathbf{x})|_{\kappa_i} = \rho^h(\mathbf{x})|_{\kappa_i}$ to cell κ_i is linear for all $i = 1, \dots, K$.

By construction, any optimal solution of QP (2.14) satisfies the local mass bounds (2.10) and conserves the total mass. Because (2.10) and (2.9) are equivalent, the FVFT-OB solution fulfills **C1** and **C3** in Section 2.1. In [1] we prove that FVFT-OB preserves linear densities by showing that if $\rho(\mathbf{x})$ is linear, then $\{F_{ij}^T\}$ are feasible and $F_{ij} = F_{ij}^T$ is optimal solution of (2.14). As a result, whenever $\rho(\mathbf{x})$ is linear,

$$\tilde{m}_i = m_i + \sum_{j \in N'(\kappa_i)} F_{ij}^T = m_i + \sum_{j \in N'(\kappa_i)} F_{ij}^{\text{ex}} = m_i^{\text{ex}}, \quad i = 1, \dots, K. \quad (2.17)$$

Therefore, FVFT-OB satisfies requirement **C2** in Section 2.1.

2.3. Mass-variable mass-target optimization-based remap

Using the flux form (2.13) with antisymmetric discrete fluxes automatically conserves the mass in QP (2.14). Yet, the flux form yields a system of globally coupled inequality constraints. This coupling complicates the solution and creates a performance bottleneck. While the flux form is imperative for conventional remappers, its importance diminishes in the optimization setting, where conservation becomes one of the many possible physically motivated constraints. We take advantage of this fact to replace (2.14) with a QP whose structure is more amenable to fast, parallelizable solution.

In a nutshell, we trade the automatic conservation of mass in (2.13) for a simpler set of box constraints plus a single linear constraint to enforce the conservation of mass explicitly. To this end, instead of (2.13) we start with the *mass-form* representation of the new cell masses

$$\tilde{m}_i^{\text{ex}} = m_i + \delta m_i^{\text{ex}}, \quad (2.18)$$

where

$$\delta m_i^{\text{ex}} = \int_{\tilde{\kappa}_i} \rho(\mathbf{x}) dV - \int_{\kappa_i} \rho(\mathbf{x}) dV \quad (2.19)$$

is the mass update between κ_i and $\tilde{\kappa}_i$. The substitution of δm_i^{ex} in (2.19) by an approximation δm_i yields the mass-form formula for the approximate masses on the new cells:

$$\tilde{m}_i = m_i + \delta m_i. \quad (2.20)$$

The approximate mass updates δm_i are the variables in the *mass-target mass-variable* optimization-based remap (MVMT-OB). The constraints on these variables are the linear equality constraint

$$\sum_{i=1}^K \delta m_i = 0, \quad (2.21)$$

which is necessary and sufficient for (2.20) to conserve the total mass and the box constraints

$$\delta \tilde{m}_i^{\min} \leq \delta m_i \leq \delta \tilde{m}_i^{\max}, \quad i = 1, \dots, K, \quad (2.22)$$

which follow from inserting (2.20) into (2.10) and using (2.15).

Remark 2.1. The equality constraint (2.21) holds for the exact mass updates:

$$\sum_{i=1}^K \delta m_i^{\text{ex}} = \sum_{i=1}^K \int_{\tilde{\kappa}_i} \rho(\mathbf{x}) dV - \sum_{i=1}^K \int_{\kappa_i} \rho(\mathbf{x}) dV = \int_{\Omega} \rho(\mathbf{x}) dV - \int_{\Omega} \rho(\mathbf{x}) dV = 0. \quad (2.23)$$

Using (2.19) in conjunction with a piecewise linear density reconstruction $\rho^h(\mathbf{x})$ on the old grid defines the mass update targets for the MVMT-OB formulation:

$$\delta m_i^T = \int_{\tilde{\kappa}_i} \rho^h(\mathbf{x}) dV - \int_{\kappa_i} \rho^h(\mathbf{x}) dV. \quad (2.24)$$

Suppose that (2.16) defines F_{ij}^T . It is easy to see that

$$\begin{aligned} \sum_{j \in N'(\kappa_i)} F_{ij}^T &= \sum_{j \in N'(\kappa_i)} \left(\int_{\tilde{\kappa}_i \cap \kappa_j} \rho_j^h(\mathbf{x}) dV - \int_{\kappa_i \cap \tilde{\kappa}_j} \rho_i^h(\mathbf{x}) dV \right) \\ &= \left(\sum_{j \in N'(\kappa_i)} \int_{\tilde{\kappa}_i \cap \kappa_j} \rho_j^h(\mathbf{x}) dV + \int_{\tilde{\kappa}_i \cap \kappa_i} \rho_i^h(\mathbf{x}) dV \right) - \left(\sum_{j \in N'(\kappa_i)} \int_{\kappa_i \cap \tilde{\kappa}_j} \rho_i^h(\mathbf{x}) dV + \int_{\tilde{\kappa}_i \cap \kappa_i} \rho_i^h(\mathbf{x}) dV \right) \\ &= \int_{\tilde{\kappa}_i} \rho^h(\mathbf{x}) dV - \int_{\kappa_i} \rho^h(\mathbf{x}) dV = \delta m_i^T. \end{aligned}$$

In other words, the target mass update on cell κ_i is the sum of the target mass fluxes corresponding to the cells in $N'(\kappa_i)$:

$$\delta m_i^T = \sum_{j \in N'(\kappa_i)} F_{ij}^T. \quad (2.25)$$

To sum it up, the MVMT-OB solution minimizes the Euclidean distance between $\{\delta m_i\}$ and the targets $\{\delta m_i^T\}$, subject to the linear constraint (2.21) and the box constraints (2.22):

$$\begin{cases} \min_{\delta m_i} \sum_{i=1}^K (\delta m_i - \delta m_i^T)^2 & \text{subject to} \\ \sum_{i=1}^K \delta m_i = 0 & \text{and } \delta \tilde{m}_i^{\min} \leq \delta m_i \leq \delta \tilde{m}_i^{\max}, \quad i = 1, \dots, K. \end{cases} \quad (2.26)$$

Problem (2.26) is a singly linearly constrained QP with simple bounds. In contrast to (2.14) the new formulation requires an additional linear equality constraint to conserve the mass, but completely decouples the inequality constraints. In Section 3 we take advantage of this structure and the locality assumption (1.3) to develop fast, efficient and parallelizable optimization algorithm for (2.26).

2.4. Properties of MVMT-OB

We first establish that (2.26) is a well-posed optimization problem. Then we prove that MVMT-OB preserves globally linear densities.

Theorem 2.1. Assume that the locality condition (1.3) holds for $K_h(\Omega)$ and $\tilde{K}_h(\Omega)$ and that $\{\rho_i\}$ are nonnegative real numbers that specify the mean density on the Lagrangian mesh. Then the QP (2.26) has a unique optimal solution.

Proof. Because (2.26) has a strictly convex objective function, it suffices to show that (2.26) has a non-empty feasible set, i.e., that there is at least one set of values $\delta m = (\delta m_1, \dots, \delta m_K)$, which satisfy the linear constraint (2.21) and the box constraints (2.22). Let

$$\delta m_i = \sum_{j \in N'(\kappa_i)} (\rho_j \mu(\tilde{\kappa}_i \cap \kappa_j) - \rho_i \mu(\kappa_i \cap \tilde{\kappa}_j)), \quad i = 1, \dots, K. \quad (2.27)$$

We prove that δm is feasible. To show (2.21) we break the equality constraint into two parts:

$$\sum_{i=1}^K \delta m_i = \sum_{i=1}^K \sum_{j \in N'(\kappa_i)} \rho_j \mu(\tilde{\kappa}_i \cap \kappa_j) - \sum_{i=1}^K \sum_{j \in N'(\kappa_i)} \rho_i \mu(\kappa_i \cap \tilde{\kappa}_j) = A - B.$$

After changing the order of summation in the first part and collecting the like terms we get

$$A = \sum_{j=1}^K \rho_j \left(\sum_{i \in N'(\kappa_j)} \mu(\tilde{\kappa}_i \cap \kappa_j) \right) = \sum_{j=1}^K \rho_j \left(\sum_{i \in N'(\kappa_j)} \mu(\tilde{\kappa}_i \cap \kappa_j) + \mu(\tilde{\kappa}_j \cap \kappa_j) - \mu(\tilde{\kappa}_j \cap \kappa_j) \right) = \sum_{j=1}^K \rho_j (\mu(\kappa_j) - \mu(\tilde{\kappa}_j \cap \kappa_j)),$$

while collecting the like terms in the second part yields

$$B = \sum_{i=1}^K \rho_i \left(\sum_{j \in N'(\kappa_i)} \mu(\kappa_i \cap \tilde{\kappa}_j) \right) = \sum_{i=1}^K \rho_i \left(\sum_{j \in N'(\kappa_i)} \mu(\kappa_i \cap \tilde{\kappa}_j) + \mu(\tilde{\kappa}_i \cap \kappa_i) - \mu(\tilde{\kappa}_i \cap \kappa_i) \right) = \sum_{i=1}^K \rho_i (\mu(\kappa_i) - \mu(\tilde{\kappa}_i \cap \kappa_i)).$$

Therefore, $A = B$, which proves that δm satisfies the linear equality constraint. To show that δm also satisfies the box constraints, note that (2.27) and $m_i = \rho_i \mu(\kappa_i)$ imply

$$m_i + \delta m_i = \rho_i \left(\mu(\kappa_i) - \sum_{j \in N'(\kappa_i)} \mu(\kappa_i \cap \tilde{\kappa}_j) \right) + \sum_{j \in N'(\kappa_i)} \rho_j \mu(\tilde{\kappa}_i \cap \kappa_j).$$

The term in the parenthesis equals $\mu(\kappa_i \cap \tilde{\kappa}_i)$ and so,

$$m_i + \delta m_i = \rho_i \mu(\kappa_i \cap \tilde{\kappa}_i) + \sum_{j \in N'(\kappa_i)} \rho_j \mu(\tilde{\kappa}_i \cap \kappa_j) = \sum_{j \in N(\kappa_i)} \rho_j \mu(\tilde{\kappa}_i \cap \kappa_j).$$

On the other hand, (2.2) and (2.3) guarantee that $\rho_i^{\min} \leq \rho_j \leq \rho_i^{\max}$ for all cells in $N(\kappa_i)$. As a result,

$$m_i + \delta m_i = \sum_{j \in N(\kappa_i)} \rho_j \mu(\tilde{\kappa}_i \cap \kappa_j) \leq \rho_i^{\max} \sum_{j \in N(\kappa_i)} \mu(\tilde{\kappa}_i \cap \kappa_j) = \rho_i^{\max} \tilde{\mu}_i;$$

$$m_i + \delta m_i = \sum_{j \in N(\kappa_i)} \rho_j \mu(\tilde{\kappa}_i \cap \kappa_j) \geq \rho_i^{\min} \sum_{j \in N(\kappa_i)} \mu(\tilde{\kappa}_i \cap \kappa_j) = \rho_i^{\min} \tilde{\mu}_i.$$

Because the box constraints (2.22) are equivalent to

$$\rho_i^{\min} \tilde{\mu}_i \leq m_i + \delta m_i \leq \rho_i^{\max} \tilde{\mu}_i, \quad i = 1, \dots, K,$$

this proves the theorem. \square

Suppose that the exact density is a globally linear function. The following theorem specializes a result from [1] and establishes sufficient conditions for the QP (2.26) to preserve linear densities.

Theorem 2.2. Let $\rho(\mathbf{x}) = c_0 + \mathbf{c}^\top \mathbf{x}$. Assume that $K_h(\Omega)$ and $\tilde{K}_h(\Omega)$ satisfy the locality condition (1.3), the density reconstruction $\rho^h(\mathbf{x})$ is exact for linear functions, and (2.24) defines the target mass updates $\{\delta m_i^T\}$. Let B_i denote the set of barycenters of the Lagrangian cells in $N(\kappa_i)$,

$$B_i = \{b_j | j \in N(\kappa_i)\}$$

and let \tilde{b}_i be the barycenter of the rezoned cell $\tilde{\kappa}_i$. The conditions

$$\tilde{b}_i \in \mathcal{H}(B_i) \quad \text{if } \kappa_i \cap \partial\Omega = \emptyset, \quad (2.28)$$

$$\tilde{b}_i \in \mathcal{H}(B_i \cup (N(\kappa_i) \cap \partial\Omega)) \quad \text{if } \kappa_i \cap \partial\Omega \neq \emptyset, \quad (2.29)$$

where $\mathcal{H}(\cdot)$ denotes the convex hull, are sufficient for the remapped masses to be exact, i.e., for $\tilde{m}_i = \tilde{m}_i^{\text{ex}}$ on every rezoned cell $\tilde{\kappa}_i$.

Proof. By assumption $\rho^h(\mathbf{x})$ is exact for linear functions and so, on every Lagrangian cell $\rho^h(\mathbf{x}) = \rho(\mathbf{x}) = c_0 + \mathbf{c}^\top \mathbf{x}$. As a result, $\delta m_i^T = \delta m_i^{\text{ex}}$ and

$$m_i + \delta m_i^T = \tilde{m}_i^{\text{ex}}.$$

Therefore, to prove $\tilde{m}_i = \tilde{m}_i^{\text{ex}}$ it suffices to show that the targets $\{\delta m_i^{\text{ex}}\}$ are feasible for QP (2.26).

Eq. (2.23) in Remark 2.1 confirms that the exact mass updates satisfy the equality constraint (2.21). On the other hand, the bounds (2.22) for $\{\delta m_i^{\text{ex}}\}$ are equivalent to the bounds on the cell masses

$$\rho_i^{\min} \tilde{\mu}_i \leq \tilde{m}_i^{\text{ex}} \leq \rho_i^{\max} \tilde{\mu}_i.$$

Assuming that (2.28) and (2.29) hold, these bounds follows from [1, Theorem 2.2]. \square

2.5. Active cell option

We call a new cell $\tilde{\kappa}_i$ *static* if $\tilde{\kappa}_i = \kappa_i$. A cell that is not static is *active*. In the context of ALE methods static cells arise when the rezoning algorithm, which aims to reduce the grid distortion accrued during the Lagrangian phase, skips cells that it deems to be of a sufficiently high “quality”. There are many practically important applications in which large numbers of cells in Ω remain static. A shock or a wave propagating through a domain is one such example. As a rule, the cells outside of a small region where the solution features reside remain static throughout the rezone phase.

Because the mass and the mean density do not change on static cells, that is, $\tilde{\rho}_i = \rho_i$ and $\tilde{m}_i = m_i$, it follows that the net mass update per static cell is zero, i.e., $\delta m_i = 0$ whenever $\tilde{\kappa}_i = \kappa_i$. On the other hand, (2.26) is a global optimization problem whose solution may not yield zero mass updates on the static cells. We can improve both the efficiency of MVMT-OBR and the physical quality of its solution by skipping the mass update computations for all static cells. Such a modification is consistent with the notion of remap as a constrained interpolation between two grids: on static cells the field representation does not change and so there is “nothing to interpolate”.

Let K_a denote the number of active cells in $K_h(\Omega)$. Without loss of generality assume that the active cells are the first K_a cells in the mesh. The mass-variable mass-target formulation with an active cell option, MVMT(a)-OBR, restricts the optimization in (2.26) to the net mass updates on the active cells only. Specifically, we set the net mass updates to zero on all static cells, that is,

$$\delta m_i = 0, \quad i = K_a + 1, \dots, K$$

and solve a QP for the net updates on the active cells only:

$$\begin{cases} \min_{\delta m_i} \sum_{i=1}^{K_a} (\delta m_i - \delta m_i^T)^2 & \text{subject to} \\ \sum_{i=1}^{K_a} \delta m_i = 0 & \text{and } \delta \tilde{m}_i^{\min} \leq \delta m_i \leq \delta \tilde{m}_i^{\max}, \quad i = 1, \dots, K_a. \end{cases} \quad (2.30)$$

The active cell option does not change the key properties of MVMT-OBR, i.e., the conclusions of Theorems 2.1 and 2.2 remain in full force for MVMT(a)-OBR.

Let us first confirm that Theorem 2.1 continues to hold for (2.30). Removing the terms that correspond to the static cells from the objective function in (2.26) does not change its strict convexity. Consequently, as before, it suffices to show that

MVMT(a)-OBR has a non-empty feasible set. In particular, we will show that (2.27) defines δm that is also feasible for the MVMT(a)-OBR. Suppose that κ_i is a static cell. Since $\tilde{\kappa}_i = \kappa_i$ it follows that

$$\mu(\tilde{\kappa}_i \cap \kappa_j) = \mu(\kappa_i \cap \kappa_j) = 0 \quad \text{and} \quad \mu(\kappa_i \cap \tilde{\kappa}_j) = \mu(\tilde{\kappa}_i \cap \tilde{\kappa}_j) = 0.$$

As a result, formula (2.27) yields

$$\delta m_i = \sum_{j \in N'(\kappa_i)} \left(\rho_j \mu(\tilde{\kappa}_i \cap \kappa_j) - \rho_i \mu(\kappa_i \cap \tilde{\kappa}_j) \right) = 0$$

for all δm_i corresponding to static cells and so, $\delta m = (\delta m_1, \dots, \delta m_{K_a}, 0, \dots, 0)$. Theorem 2.1 implies that $\delta \tilde{m}_i^{\min} \leq \delta m_i \leq \delta \tilde{m}_i^{\max}$ for $i = 1, \dots, K_a$, i.e., δm is feasible for MVMT(a)-OBR.

To show that MVMT(a)-OBR preserves linear densities, note that for static cells

$$\sum_{j \in N'(\kappa_i)} F_{ij}^T = \sum_{j \in N'(\kappa_i)} \left(\int_{\tilde{\kappa}_i \cap \kappa_j} \rho_j^h(x) dV - \int_{\kappa_i \cap \tilde{\kappa}_j} \rho_i^h(x) dV \right) = \sum_{j \in N'(\kappa_i)} \left(\int_{\kappa_i \cap \kappa_j} \rho_j^h(x) dV - \int_{\tilde{\kappa}_i \cap \tilde{\kappa}_j} \rho_i^h(x) dV \right) = 0$$

and so, (2.25) implies that $\delta m^T = 0$ whenever $\tilde{\kappa}_i = \kappa_i$. Therefore, the proof of linearity preservation in Theorem 2.2 trivially specializes to MVMT(a)-OBR.

Because MVMT-OBR and MVMT(a)-OBR have identical theoretical properties and differ only in the number of variables, in what follows we write both methods as the QP

$$\begin{cases} \min_{\delta m_i} \sum_{i=1}^{K'} (\delta m_i - \delta m_i^T)^2 & \text{subject to} \\ \sum_{i=1}^{K'} \delta m_i = 0 & \text{and} \quad \delta \tilde{m}_i^{\min} \leq \delta m_i \leq \delta \tilde{m}_i^{\max}, \quad i = 1, \dots, K', \end{cases} \quad (2.31)$$

where $K' = K$ for MVMT-OBR, $K' = K_a$ – the number of active cells, for MVMT(a)-OBR. The last $K - K'$ variables are set to zero:

$$\delta m_i = 0, \quad i = K' + 1, \dots, K.$$

Remark 2.2. Because the active cell option is based on the fact that field representation does not change on static cells, its scope extends beyond the optimization-based remap. In fact, the strategy to exclude static cells from the remap can be used in conjunction with any remap algorithm, including advection-based remap and flux-corrected remap (FCR).

2.6. Swept region approximation

Recall that $\rho^h(x)$ is a piecewise linear function defined with respect to the old mesh. As a result,

$$\delta m_i^T = \int_{\tilde{\kappa}_i} \rho^h(x) dV - \int_{\kappa_i} \rho^h(x) dV = \sum_{j \in N(\kappa_i)} \int_{\tilde{\kappa}_i \cap \kappa_j} \rho_j^h(x) dV - \int_{\kappa_i} \rho_i^h(x) dV, \quad (2.32)$$

where $\rho_j^h(x)$ is the restriction of $\rho^h(x)$ to the Lagrangian cell κ_j . An implementation of formula (2.32) requires the computation of the intersections between the new and old cells. While software tools for this task exist, the swept region approach [11, Section 4] offers a simpler and more efficient alternative, especially in three dimensions. Swept regions are defined by the movement of the sides of the old cells into a new configuration. As a result, swept regions are completely determined by the coordinates of the old and new cell vertices.

For simplicity, we explain the swept region approach in two dimensions. The sketch in Fig. 3 uses the cell numbering from Fig. 1 and shows the swept regions resulting from connecting the side vertices of $\tilde{\kappa}_5$ and its Lagrangian prototype κ_5 . Each swept region σ_{ij} corresponds to a side s_{ij} in the old mesh. The unit tangent \mathbf{t}_{ij} to that side induces orientation on σ_{ij} . Because swept regions are associated with the sides of the cells, their use with the flux form (2.13) of the new masses is natural. Specifically, to develop the swept region approximation of FVFT-OBR we restrict the summation in (2.13) to the side neighborhood of κ_i :

$$\tilde{m}_i = m_i + \sum_{j \in N_s(\kappa_i)} F_{ij}. \quad (2.33)$$

The optimization variables F_{ij} are mass fluxes associated with the swept regions. To define the targets for these variables we consider integration of a piecewise linear density reconstruction ρ^h on σ_{ij} . As before, ρ_i^h is the linear restriction of ρ^h to the old cell κ_i .

Note that a swept region σ_{ij} can intersect more than one cell. Consequently, exact integration of ρ^h on σ_{ij} would also require intersections between the swept region and the old cells. To avoid the computation of these intersections we

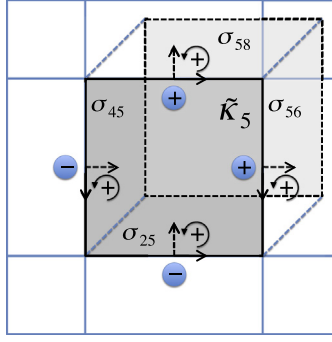


Fig. 3. Swept region nomenclature in two dimensions. Cell and side numbers follow the example in Fig. 1. The unit face tangent t_{ij} indicates the orientation of the swept region σ_{ij} . The blue ovals indicate the signs of the entries in the side-to-cell incidence matrix (1.1). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

approximate $\int_{\sigma_{ij}} \rho^h dV$ using either ρ_i^h or ρ_j^h to compute the target flux for F_{ij} . The choice of the two possible restrictions of ρ^h depends on (i) the signed area $\mu^*(\sigma_{ij})$ of the swept region, and (ii) the orientation of s_{ij} encoded in the side-to-cell incidence matrix \mathbf{D} . Specifically, instead of (2.16) we define the target fluxes according to the following formula:

$$F_{ij}^{TS} = \begin{cases} \int_{\sigma_{ij}} \rho_i^h(x) dV & \text{if } \mu^*(\sigma_{ij}) d_{i,ij} < 0, \\ \int_{\sigma_{ij}} \rho_j^h(x) dV & \text{if } \mu^*(\sigma_{ij}) d_{i,ij} > 0. \end{cases} \quad (2.34)$$

Fig. 4 illustrates the choice of the reconstruction component as a function of the signed area and the side orientation. The swept region approximation (2.34) is exact for linear functions [12]. As a result, all theoretical conclusions about FVFT-OBK continue to hold when the approximate target fluxes (2.34) are used in lieu of the “true” targets (2.16); see [1].

We now proceed to motivate the swept region approximation for the MVMT-OBK formulation. According to (2.25), when using exact cell intersections the target mass update δm_i^T on κ_i equals the sum of the target mass fluxes for that cell. This relationship prompts the approximation

$$\delta m_i^{TS} = \sum_{j \in N_s(\kappa_i)} F_{ij}^{TS}, \quad (2.35)$$

where (2.34) defines the target fluxes F_{ij}^{TS} .

It is easy to see that the swept region approximation does not change the theoretical properties of the MVMT-OBK formulation. Theorem 2.1 does not depend on the choice of the targets and so it remains in full force. In contrast, Theorem 2.2 does depend on the target selection. However, because the swept region approximation is exact for linear densities, it follows that formula (2.35) is also exact for such densities, i.e., there holds $\delta m_i^{TS} = \delta m_i^{\text{ex}}$, whenever ρ is linear. This is sufficient to carry out the proof of the theorem in the case when (2.35) defines the targets.

2.7. Flux-variable mass-target formulation

In this section we briefly mention an alternative QP formulation of remap, which combines the features of FVFT-OBK and MVMT-OBK. We explain why this formulation is less attractive for the development of an optimization-based remap algorithm.

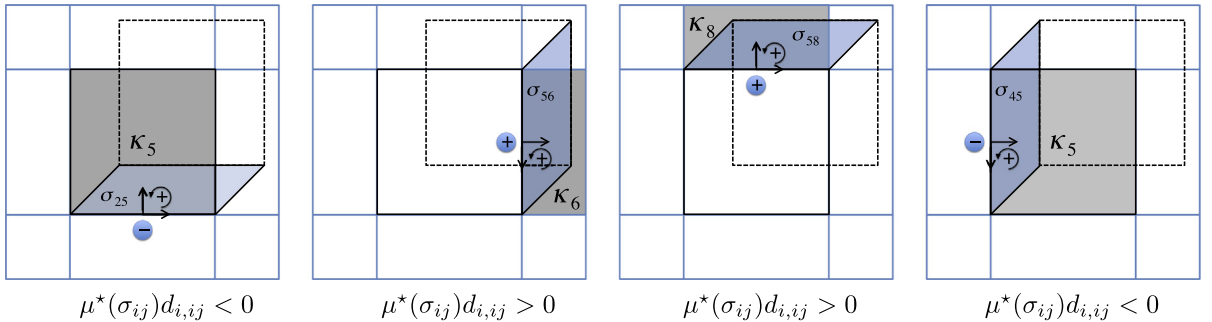


Fig. 4. The grey color indicates the Lagrangian cell, which provides the reconstruction component of ρ^h for the computation of the target flux in (2.34). Cell numbering follows the example from Fig. 1. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The FVFT-OBOR objective minimizes the Euclidean distance between the flux variables and the target fluxes. We can define another optimization objective by combining the flux variables that contribute to the same cell. The resulting QP minimizes the Euclidean distance between the sums of the mass fluxes exchanged between new and old cells and the sums of the corresponding target fluxes, subject to the same bounds on the flux variables as in (2.14):

$$\left\{ \begin{array}{l} \min_{F_{ij}} \left(\left(\sum_{\substack{j \in N'(\kappa_i) \\ i < j}} F_{ij} - \sum_{\substack{j \in N'(\kappa_i) \\ i > j}} F_{ji} \right) - \left(\sum_{\substack{j \in N'(\kappa_i) \\ i < j}} F_{ij}^T - \sum_{\substack{j \in N'(\kappa_i) \\ i > j}} F_{ji}^T \right) \right)^2 \quad \text{subject to} \\ \delta \tilde{m}_i^{\min} \leq \sum_{\substack{j \in N'(\kappa_i) \\ i < j}} F_{ij} - \sum_{\substack{j \in N'(\kappa_i) \\ i > j}} F_{ji} \leq \delta \tilde{m}_i^{\max} \quad i = 1, \dots, K. \end{array} \right. \quad (2.36)$$

From (2.25) we know that the sum of the target fluxes in the objective function of (2.36) gives the mass target δm_i^T of the MVMT-OBOR formulation. Furthermore, the sum of the flux variables is effectively a mass update

$$\delta m_i := \sum_{\substack{j \in N'(\kappa_i) \\ i < j}} F_{ij} - \sum_{\substack{j \in N'(\kappa_i) \\ i > j}} F_{ji} \quad (2.37)$$

for cell κ_i . As a result, we can write (2.36) in the following flux-variable mass-target (FVMT) form:

$$\left\{ \begin{array}{l} \min_{F_{ij}} \sum_{i=1}^K (\delta m_i - \delta m_i^T)^2 \quad \text{subject to} \\ \delta \tilde{m}_i^{\min} \leq \delta m_i \leq \delta \tilde{m}_i^{\max} \quad i = 1, \dots, K, \end{array} \right. \quad (2.38)$$

with the tacit understanding that $\{\delta m_i\}$ are functions of the flux variables F_{ij} .

The FVMT-OBOR formulation combines the flux variables of FVFT-OBOR with the objective function of MVMT-OBOR. Compared to MVMT-OBOR this formulation suffers from two key drawbacks. First, since (2.38) uses flux variables, it inherits the coupled system of linear inequality constraints from FVFT-OBOR, which is the main efficiency bottleneck of the FVFT-OBOR algorithm. Second, one can show that the objective function of (2.36) has a large null space. Therefore, the FVMT-OBOR formulation is not strictly convex and the optimal solution of (2.36) is not unique. This complicates the efficient numerical solution of FVMT-OBOR. For these reasons we do not pursue FVMT-OBOR in this paper.

3. Optimization algorithm

In this section we discuss the solution of the optimization problem (2.31), which we restate for ease of reference:

$$\left\{ \begin{array}{l} \min_{\delta m_i} \sum_{i=1}^{K'} (\delta m_i - \delta m_i^T)^2 \quad \text{subject to} \\ \sum_{i=1}^{K'} \delta m_i = 0 \quad \text{and} \quad \delta \tilde{m}_i^{\min} \leq \delta m_i \leq \delta \tilde{m}_i^{\max}, \quad i = 1, \dots, K' \end{array} \right.$$

We remind that $K' = K$ – the number of all cells, in the case of MVMT-OBOR, $K' = K_a$ – the number of active cells, in the case of MVMT(a)-OBOR and the last $K - K'$ variables are set to zero:

$$\delta m_i = 0; \quad i = K' + 1, \dots, K.$$

Optimization problems of this type are known in the optimization literature as the *singly linearly constrained quadratic programs with simple bounds*. A recent publication with new techniques for their solution is [13], where the authors propose an algorithm that runs in expected linear time and is directly applicable to (2.31). While the algorithm proposed here is motivated by Dai and Fletcher [13], it is considerably simpler due to several features of (2.31) that are unique to the context of remap.

A key observation in developing an algorithm for (2.31) is that the related optimization problem without the mass conservation constraint, $\sum_{i=1}^{K'} \delta m_i = 0$, is fully separable. This problem can be solved by independently solving K' one-dimensional quadratic programs with simple bounds, i.e., its solution cost is $\mathcal{O}(K')$. Thus, our goal is to satisfy the remaining constraint $\sum_{i=1}^{K'} \delta m_i = 0$ in a few iterations of cost $\mathcal{O}(K')$ each.

We solve (2.31) by a direct application of the Karush–Kuhn–Tucker (KKT) theory, see, e.g. [14, Ch.12]. We define the Lagrangian functional $\mathcal{L} : \mathbb{R}^{K'} \times \mathbb{R} \times \mathbb{R}^{K'} \times \mathbb{R}^{K'} \rightarrow \mathbb{R}$,

$$\mathcal{L}(\delta m, \lambda, \mu_1, \mu_2) = \frac{1}{2} \sum_{i=1}^{K'} (\delta m_i - \delta m_i^T)^2 - \lambda \sum_{i=1}^{K'} \delta m_i - \sum_{i=1}^{K'} \mu_{1,i} (\delta m_i - \delta \tilde{m}_i^{\min}) - \sum_{i=1}^{K'} \mu_{2,i} (\delta \tilde{m}_i^{\max} - \delta m_i),$$

where $\delta m \in \mathbb{R}^{K'}$, with $(\delta m)_i = \delta m_i$, is the vector of primal optimization variables, and $\lambda \in \mathbb{R}$, $\mu_1 \in \mathbb{R}^{K'}$, with $(\mu_1)_i = \mu_{1,i}$, and $\mu_2 \in \mathbb{R}^{K'}$, with $(\mu_2)_i = \mu_{2,i}$, are the Lagrange multipliers. The gradient of the Lagrangian with respect to the primal variables is given by

$$\frac{\partial}{\partial \delta m_i} \mathcal{L}(\delta m, \lambda, \mu_1, \mu_2) = \delta m_i - \delta m_i^T - \lambda - \mu_{1,i} + \mu_{2,i}, \quad \text{for } i = 1, \dots, K'.$$

As shown in Section 2, $\delta \tilde{m}_i^{\min}$ and $\delta \tilde{m}_i^{\max}$ are such that the constraints in (2.31) are consistent, i.e., the optimization problem has a solution. Furthermore, strict convexity yields a unique global minimizer. The necessary and sufficient optimality conditions for (2.31) are:

$$\delta m_i = \delta m_i^T + \lambda + \mu_{1,i} - \mu_{2,i} \quad i = 1, \dots, K' \quad (3.1a)$$

$$\delta \tilde{m}_i^{\min} \leq \delta m_i \leq \delta \tilde{m}_i^{\max} \quad i = 1, \dots, K' \quad (3.1b)$$

$$\mu_{1,i} \geq 0, \quad \mu_{2,i} \geq 0 \quad i = 1, \dots, K' \quad (3.1c)$$

$$\mu_{1,i}(\delta m_i - \delta \tilde{m}_i^{\min}) = 0, \quad \mu_{2,i}(-\delta m_i + \delta \tilde{m}_i^{\max}) = 0 \quad i = 1, \dots, K' \quad (3.1d)$$

$$\sum_{i=1}^{K'} \delta m_i = 0. \quad (3.1e)$$

We solve the conditions (3.1) directly. First, we focus on the conditions (3.1a)–(3.1d), which are separable in the index i . For any fixed value of λ a solution to (3.1a)–(3.1d) is given by

$$\begin{cases} \delta m_i = \delta m_i^T + \lambda; & \mu_{1,i} = \mu_{2,i} = 0 & \text{if } \delta \tilde{m}_i^{\min} \leq \delta m_i^T + \lambda \leq \delta \tilde{m}_i^{\max} \\ \delta m_i = \delta \tilde{m}_i^{\min}; & \mu_{2,i} = 0, \quad \mu_{1,i} = \delta m_i - \delta m_i^T - \lambda & \text{if } \delta m_i^T + \lambda < \delta \tilde{m}_i^{\min} \\ \delta m_i = \delta \tilde{m}_i^{\max}; & \mu_{1,i} = 0, \quad \mu_{2,i} = \delta m_i^T - \delta m_i + \lambda & \text{if } \delta m_i^T + \lambda > \delta \tilde{m}_i^{\max}, \end{cases} \quad (3.2)$$

for all $i = 1, \dots, K'$. Ignoring the Lagrange multipliers μ_1 and μ_2 and treating δm_i as a function of λ yields

$$\delta m_i(\lambda) = \text{median}(\delta \tilde{m}_i^{\min}, \delta m_i^T + \lambda, \delta \tilde{m}_i^{\max}), \quad i = 1, \dots, K', \quad (3.3)$$

where we note that this is an $\mathcal{O}(K')$ computation.

Second, we adjust λ in an outer iteration in order to satisfy the constraint $\sum_{i=1}^{K'} \delta m_i(\lambda) = 0$. When we find the λ^* such that $\sum_{i=1}^{K'} \delta m_i(\lambda^*) = 0$ holds, we will have solved the optimality conditions (3.1). From (3.3) it is clear that $\sum_{i=1}^{K'} \delta m_i(\lambda)$ is a piecewise linear, monotonically increasing function of a single scalar variable λ . Therefore, a secant method can be efficiently employed as the outer iteration.

A basic secant approach, such as the one described in [15, pp. 28–31], typically requires modifications to ensure fast global convergence. The algorithm outlined in [13] tailors such modifications to the context of secant methods applied to general singly linearly constrained QPs with simple bounds. However, in the context of MVMT-OBOR it turns out that globalization is unnecessary. This is primarily a consequence of the strict convexity of (2.31) and the quality of the initial guess $\lambda_0 = 0$. Substituting $\lambda_0 = 0$ into (3.3) yields $\delta m_i(\lambda_0) = \text{median}(\delta \tilde{m}_i^{\min}, \delta m_i^T, \delta \tilde{m}_i^{\max})$, for $i = 1, \dots, K'$, in other words the solution of the optimization problem (2.31) without the constraint $\sum_{i=1}^{K'} \delta m_i = 0$. In Section 4 we demonstrate on several examples that $\delta m_i(\lambda_0)$, $i = 1, \dots, K'$, violate the mass conservation constraint very slightly. This can be traced back to the locality assumption (1.3). In summary, at each remap step we have $|\sum_{i=1}^{K'} \delta m_i(\lambda_0)| \approx 0$, so only a few (1–5) iterations of the basic secant method are needed to achieve full feasibility. We also note that since $\sum_{i=1}^{K'} \delta m_i(\lambda)$ is a piecewise linear function, the secant method always gives a solution that is accurate to machine precision for our choice of stopping tolerances. The complete algorithm is stated below.

Algorithm 1. Secant method for solving the MVMT-OBOR problem, (2.31)

1. Initialization: Set $\lambda_0 \leftarrow 0$, $\eta \leftarrow 10^{-12}$ and $\Delta \lambda^{FD} \leftarrow 10^{-8}$.
 2. Finite difference step:
 - (a) Compute $\delta m_i(\lambda_0) \leftarrow \text{median}(\delta \tilde{m}_i^{\min}, \delta m_i^T + \lambda_0, \delta \tilde{m}_i^{\max})$ for $i = 1, \dots, K'$. Compute residual $r_p \leftarrow \sum_{i=1}^{K'} \delta m_i(\lambda_0)$. **If** $|r_p| < \eta$, then return $\delta m_i(\lambda_0)$ for $i = 1, \dots, K'$ and **stop**.
 - (b) Compute $\delta m_i(\lambda_0 + \Delta \lambda^{FD}) \leftarrow \text{median}(\delta \tilde{m}_i^{\min}, \delta m_i^T + \lambda_0 + \Delta \lambda^{FD}, \delta \tilde{m}_i^{\max})$ for $i = 1, \dots, K'$. Compute residual $r_c \leftarrow \sum_{i=1}^{K'} \delta m_i(\lambda_0 + \Delta \lambda^{FD})$.
 - (c) Set $\alpha \leftarrow \Delta \lambda^{FD} / (r_c - r_p)$.
 - (d) Set $\lambda_p \leftarrow \lambda_0$. Set $\lambda_c \leftarrow \lambda_p - \alpha r_p$.
 3. **While** $|r_c| > \eta$ (Secant Iteration)
 - (a) Compute $\delta m_i(\lambda_c) \leftarrow \text{median}(\delta \tilde{m}_i^{\min}, \delta m_i^T + \lambda_c, \delta \tilde{m}_i^{\max})$ for $i = 1, \dots, K'$. Compute residual $r_c \leftarrow \sum_{i=1}^{K'} \delta m_i(\lambda_c)$.
 - (b) Set $\alpha \leftarrow (\lambda_p - \lambda_c) / (r_p - r_c)$. Set $r_p \leftarrow r_c$.
 - (c) Set $\lambda_p \leftarrow \lambda_c$. Set $\lambda_c \leftarrow \lambda_c - \alpha r_c$.
- End While**
4. Return $\delta m_i(\lambda_c)$ and **stop**.
-

4. Computational studies

4.1. Preservation of shape, monotonicity and linearity

In this section we present numerical examples that demonstrate that FVFT-OB and MVMT-OB give virtually identical results when subjected to the “torture” tests developed in [1]. The overall purpose of these tests is to numerically confirm the theoretical properties of the methods in extreme scenarios. For completeness we include the results of the same studies performed with a flux-corrected remap (FCR) algorithm. For all algorithms we use swept regions for the target flux reconstruction.

The first test examines the ability of the algorithms to preserve the shape of a given density function. The test involves a one-dimensional mesh consisting of three cells. In the original or “old” configuration the cells are of equal lengths. In the remapped or “new” configuration the middle cell is compressed by a factor of approximately 6.6, see [1] for the detailed setup. The density function has the shape of an asymmetrical peak. An accurate method should preserve the shape of the peak on the new mesh. It is evident from Fig. 5 that FVFT-OB and MVMT-OB preserve the peak, giving identical results, while FCR loses accuracy due to the local approximation of the global FVFT-OB formulation.

The second test focuses on the preservation of monotonicity for a given density function. We use a simple two-dimensional extension of the previous one-dimensional experiment. The same setup is used in [1]. As in one dimension, the original mesh is uniform, while in the new mesh the middle cell is compressed equally in both spatial directions, see Fig. 6. The density function is linear. We report loss of monotonicity if the bounds in (2.14) and (2.26) are violated after a single remap step. We note that the compressive mesh motion illustrated in Fig. 6 is permitted according to the locality assumption (1.3) and the linearity preservation conditions (2.28) and (2.29) established for FVFT-OB and MVMT-OB. Table 1 confirms that FVFT-OB and MVMT-OB preserve monotonicity, which is a direct consequence of the construction of the schemes as constrained optimization algorithms. At the same time, as explained in [1], the condition on mesh motion for the monotonicity of the swept-region donor-cell method [11, p. 279] is violated by this example. This defect is inherited by FCR, resulting in the loss of monotonicity for high compressions of the middle cell.

The third test examines whether the methods preserve linear density functions. We use the same torture mesh as in the second experiment. The L_1 error is evaluated after two remap steps (original mesh \rightarrow compressed mesh \rightarrow original mesh). Since the linearity preservation conditions (2.28) and (2.29) are satisfied for the compressive mesh motion, we expect that FVFT-OB and MVMT-OB preserve linear densities regardless of how much the middle cell is compressed. This is confirmed in Table 2, where FVFT-OB and MVMT-OB give very similar results. In contrast, FCR fails to preserve linear densities at the compression ratios of $4 \times 4 : 1$ and above.

4.2. Asymptotic accuracy

In this section we compare the asymptotic accuracy of FVFT-OB and MVMT-OB in the context of a continuous rezone strategy. The appropriate notions of remap error and convergence rates can be defined through cyclic remap tests [11]. The methodology is described in great detail in [1,12,11].

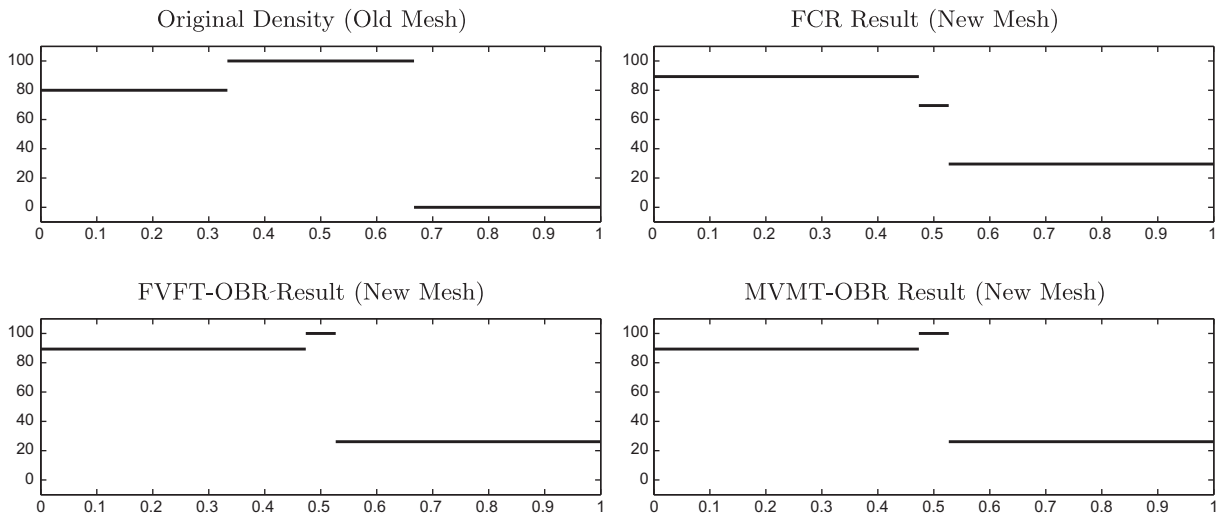


Fig. 5. FVFT-OB and MVMT-OB preserve the shape of the peak, giving identical results, while FCR loses accuracy and transforms the peak into a step function.

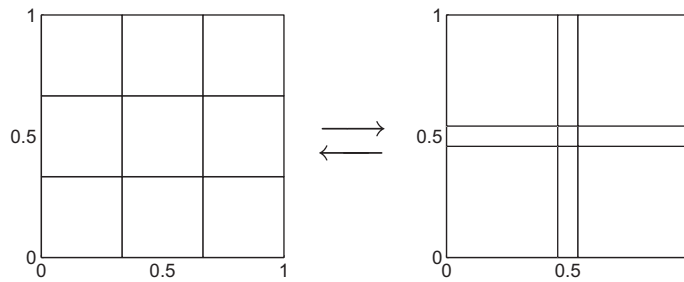


Fig. 6. A 3×3 uniform mesh (left pane) and a “compressed” mesh (right pane) with a 4×4 -fold compression of the middle cell.

Table 1

Monotonicity of FVFT-OB, MVMT-OB and FCR, implemented using swept regions, with respect to the remap of a linear density function in two dimensions, for different compression ratios $\ell \times \ell : 1$ of the middle cell. FVFT-OB and MVMT-OB are monotone throughout, while FCR is not.

	$\ell = 5$	$\ell = 6$	$\ell = 7$	$\ell = 14$	$\ell = 15$	$\ell = 16$	$\ell = 100$
FVFT-OB	Yes	Yes	Yes	Yes	Yes	Yes	Yes
MVMT-OB	Yes	Yes	Yes	Yes	Yes	Yes	Yes
FCR	Yes	Yes	Yes	Yes	No	No	No

Table 2

L_1 errors in the FVFT-OB, MVMT-OB and FCR remap of a linear density function in two dimensions, for different compression ratios $\ell \times \ell : 1$ of the middle cell. Errors close to machine precision are highlighted. FVFT-OB and MVMT-OB preserve linear densities for arbitrarily compressed middle cells, while FCR does not.

		$\ell = 3$	$\ell = 4$	$\ell = 5$	$\ell = 15$	$\ell = 16$	$\ell = 100$
FVFT-OB	(L_1 err)	1.02e-16	3.02e-16	2.22e-16	6.17e-16	2.22e-15	1.26e-13
MVMT-OB	(L_1 err)	6.78e-17	3.05e-16	2.25e-16	1.54e-15	1.70e-15	1.46e-13
FCR	(L_1 err)	9.25e-17	3.01e-03	7.68e-03	3.28e-02	3.49e-02	1.51e-01

We consider three cyclic mesh motions from [1]: the smooth *tensor-product* motion, resulting in a sequence of slowly evolving rectangular meshes (Fig. 7(a)); the smooth *wave* motion, based on a sinusoidal transformation of the mesh coordinates (Fig. 7(b)); and the *repeated-repair* motion, giving a sequence of rapidly changing rectangular meshes, derived from the torture motion illustrated in Fig. 6. We additionally introduce the *random* motion (Fig. 7(c)). A uniform partition of the unit square provides the first and the last grids in a random motion sequence. All other grids are obtained by random perturbations of the nodes (ξ_i, η_j) of the initial uniform grid, defined by

$$x_i = \xi_i + 0.25hr_x; \quad y_j = \eta_j + 0.25hr_y,$$

where h is the initial cell width and r_x, r_y are uniformly distributed random numbers in $[-1, 1]$.

For the convergence studies involving the tensor-product, wave and random motions, we use the sine, peak and shock densities referenced in [1,11]. We test convergence under the repeated-repair motion using the sine density. We recall from [1] that FCR is only first-order accurate in this test.

For the repeated-repair motion, the absolute accuracy of MVMT-OB equals that of FVFT-OB, see Table 3. Additionally, both methods exhibit second-order convergence.

For the tensor-product, wave and random motions, while the convergence rates of FVFT-OB and MVMT-OB are similar for all test densities, FVFT-OB has a slight advantage in terms of absolute accuracy, see Tables 4–6. As shown in Section 4.6 this accuracy can be easily matched by MVMT-OB using only slightly finer computational meshes.

4.3. Robustness

The purpose of the test in this section is to subject the FVFT-OB and MVMT-OB algorithms to mesh motions that may violate the locality assumption (1.3) and the linearity preservation conditions (2.28) and (2.29). In particular, we examine the behavior of the methods on 64×64 tensor-product cyclic meshes as the number of remaps, R , is decreased much beyond the previously chosen value of $R = 320$. Equivalently, we may say that the so-called pseudo-time step, $1/R$, is increased significantly beyond $1/R = 1/320$. For completeness, we include the FCR results from [1].

It is evident from Table 7 that FVFT-OB and MVMT-OB perform similarly, in other words they are both robust when subjected to mesh deformations that are beyond the theoretical guarantees of the methods. In contrast, the FCR algorithm violates linearity preservation early, and eventually becomes unstable.

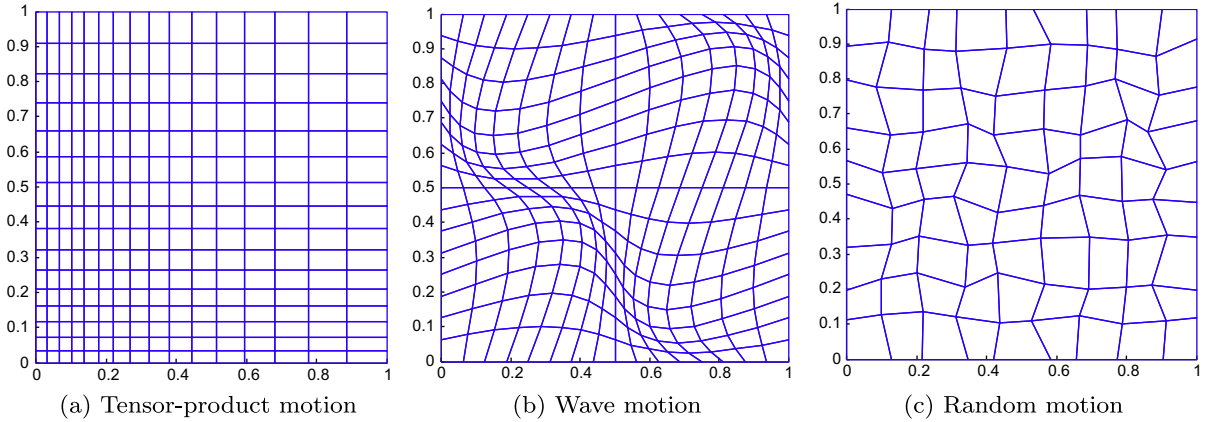


Fig. 7. Cyclic mesh motions used in our numerical studies; we also use the repeated-repair motion derived from Fig. 6.

Table 3

FVFT-OB and MVMT-OB L_1 errors and convergence rate estimates for the sine density using 4 repeated-repair cyclic meshes. The absolute errors and convergence rates are very similar.

# Cells	# Remaps	FVFT-OB		MVMT-OB	
		L_1 err	L_1 rate	L_1 err	L_1 rate
64×64	320	1.09e-03	–	1.15e-03	–
128×128	640	2.69e-04	2.03	2.77e-04	2.06
256×256	1280	6.71e-05	2.01	6.82e-05	2.04
512×512	2560	1.68e-05	2.01	1.69e-05	2.03

4.4. Locality

An interesting question is whether the optimization-based remap algorithms respect the notion of “locality” or “causality”, which is a feature of purely local, albeit less accurate and less robust methods like FCR. To examine the behavior of OB methods, we construct an example in which only a single mesh vertex is displaced, and measure the mass updates computed by the methods on every cell. We note that if FCR is applied to such single-vertex motion, the mass updates will be zero everywhere except in the immediate neighborhood of the vertex (a four-cell neighborhood in this case).

The single-vertex motion is defined by moving the center vertex of the mesh, with coordinates (x, y) , to the position $(x + h \cos(\alpha), y + h \sin(\alpha))$, where $\alpha = 23\pi/180$ and $h \in \{1/16, 1/32, 1/64, 1/128\}$, for the mesh sequence $\{16 \times 16, 32 \times 32, 64 \times 64, 128 \times 128\}$, respectively. The remapped density function is a symmetric Gaussian of amplitude 1 with standard deviation 0.1, centered at $(0.5, 0.5)$. We plot the magnitudes of mass updates, δm_i , after a single remap step. The mass updates are scaled by K , the total number of cells.

While MVMT-OB generates nonzero mass updates everywhere, we observe from Fig. 8, top pane, that there is a sharp drop-off in the magnitudes of mesh updates between the four-cell neighborhood surrounding the displaced vertex and the remainder of the domain. The size of the drop-off increases as the mesh is refined. For the 128×128 mesh, see Fig. 9, bottom pane, the mass updates are approximately seven orders of magnitude smaller for all cells outside of the four-cell neighborhood of the displaced vertex. Thus we may argue that MVMT-OB is an “asymptotically” local method.

In contrast, FVFT-OB generates mass updates that are considerably more localized, see Fig. 8, middle pane. Nonetheless, the behavior of FVFT-OB is not identical to that of FCR, as eight cells, instead of four, contribute with non-zero mass updates. Thus we may label FVFT-OB “weakly” local.

Finally, we examine the behavior of the mass-variable mass-target algorithm with the active cell option, MVMT(a)-OB. We recall that MVMT(a)-OB eliminates the mass-update variables that correspond to static cells. Fig. 8 shows that this algorithmic modification yields an optimization-based remap scheme with purely local behavior, when applied to the single-vertex mesh motion. We recommend the use of MVMT(a)-OB whenever significant portions of the computational domain are static.

4.5. Conservation properties

This section examines the numerical behavior of MVMT-OB with respect to mass conservation. Our test is designed to verify a claim made in Section 3, namely that the mass updates $\delta m_i(\lambda_0)$, $i = 1, \dots, K$, where $\lambda_0 = 0$ is the initial Lagrange multiplier guess, violate the mass conservation constraint only slightly. We demonstrate that this is true in a cumulative sense

Table 4

FVFT-OBDR and MVMT-OBDR L_1 errors and convergence rate estimates for the sine, peak and shock densities using 4 tensor-product cyclic meshes. The convergence rates are very similar. The absolute errors are slightly lower for FVFT-OBDR.

# Cells	# Remaps	Sine		Peak		Shock	
		L_1 err	L_1 rate	L_1 err	L_1 rate	L_1 err	L_1 rate
<i>FVFT-OB</i>							
64×64	320	4.91e-04	–	2.55e-03	–	2.88e-02	–
128×128	640	6.16e-05	3.00	8.90e-04	1.52	1.75e-02	0.72
256×256	1280	7.82e-06	2.99	3.10e-04	1.52	1.06e-02	0.72
512×512	2560	9.89e-07	2.98	1.09e-04	1.52	6.35e-03	0.73
<i>MVMT-OB</i>							
64×64	320	6.38e-04	–	2.88e-03	–	3.03e-02	–
128×128	640	8.09e-05	2.98	1.08e-03	1.42	1.83e-02	0.73
256×256	1280	1.04e-05	2.97	4.04e-04	1.42	1.10e-02	0.73
512×512	2560	1.32e-06	2.97	1.51e-04	1.42	6.65e-03	0.73

Table 5

FVFT-OBDR and MVMT-OBDR L_1 errors and convergence rate estimates for the sine, peak and shock densities using 4 wave cyclic meshes. The convergence rates are very similar. The absolute errors are slightly lower for FVFT-OBDR.

# Cells	# Remaps	Sine		Peak		Shock	
		L_1 err	L_1 rate	L_1 err	L_1 rate	L_1 err	L_1 rate
FVFT-OBRR							
64×64	320	1.92e−04	–	6.71e−04	–	9.89e−03	–
128×128	640	2.47e−05	2.96	2.54e−04	1.40	6.33e−03	0.64
256×256	1280	3.17e−06	2.96	9.45e−05	1.41	3.93e−03	0.67
512×512	2560	4.00e−07	2.97	3.50e−05	1.42	2.44e−03	0.68
MVMT-OBRR							
64×64	320	2.41e−04	–	9.43e−04	–	1.16e−02	–
128×128	640	3.10e−05	2.96	3.67e−04	1.36	7.18e−03	0.69
256×256	1280	3.99e−06	2.96	1.39e−04	1.38	4.44e−03	0.69
512×512	2560	5.11e−07	2.96	5.30e−05	1.39	2.81e−03	0.68

Table 6

FVFT-OBDR and MVMT-OBDR L_1 errors and convergence rate estimates for the sine, peak and shock densities using 4 random cyclic meshes. The convergence rates are very similar. The absolute errors are slightly lower for FVFT-OBDR.

# Cells	# Remaps	Sine		Peak		Shock	
		L_1 err	L_1 rate	L_1 err	L_1 rate	L_1 err	L_1 rate
<i>FVFT-OB</i>							
64×64	320	2.89e-04	–	1.38e-03	–	1.88e-02	–
128×128	640	4.69e-05	2.62	4.89e-04	1.50	1.16e-02	0.70
256×256	1280	9.13e-06	2.49	1.72e-04	1.50	7.09e-03	0.70
512×512	2560	2.04e-06	2.38	6.08e-05	1.50	4.32e-03	0.70
<i>MVMT-OB</i>							
64×64	320	3.49e-04	–	1.65e-03	–	2.01e-02	–
128×128	640	5.43e-05	2.69	6.25e-04	1.40	1.23e-02	0.71
256×256	1280	1.01e-05	2.55	2.32e-04	1.41	7.47e-03	0.71
512×512	2560	2.18e-06	2.44	8.55e-05	1.42	4.51e-03	0.72

Table 7

L_1 errors in the FVFT-OBDR, MVMT-OBDR and FCR remap of a linear density function on the 64×64 tensor-product cyclic mesh, for different values of the pseudo-time step $1/R$. Errors smaller than $1e-8$ are highlighted. The OBR methods fail to preserve linear densities at $R = 154$, while FCR fails at $R = 212$, resulting in a pseudo-time step advantage for FVFT-OBDR and MVMT-OBDR of $212/154 \approx 1.4$. Beyond this point, the OBR methods exhibit a graceful loss of accuracy; FCR becomes numerically unstable. Bold face indicates preservation of linear functions to computing precision.

	$R = 213$	$R = 212$	$R = 211$	$R = 155$	$R = 154$	$R = 153$	$R = 100$	$R = 50$
FVFT-OBDR	1.32e-13	1.42e-13	1.60e-13	4.60e-09	4.06e-06	1.53e-05	1.97e-03	6.48e-03
MVMT-OBDR	1.32e-13	1.42e-13	1.60e-13	4.59e-09	2.07e-06	7.53e-06	1.27e-03	6.71e-03
FCR	1.32e-13	5.32e-08	1.10e-06	2.26e-03	2.35e-03	2.44e-03	5.73e+04	8.50e+11

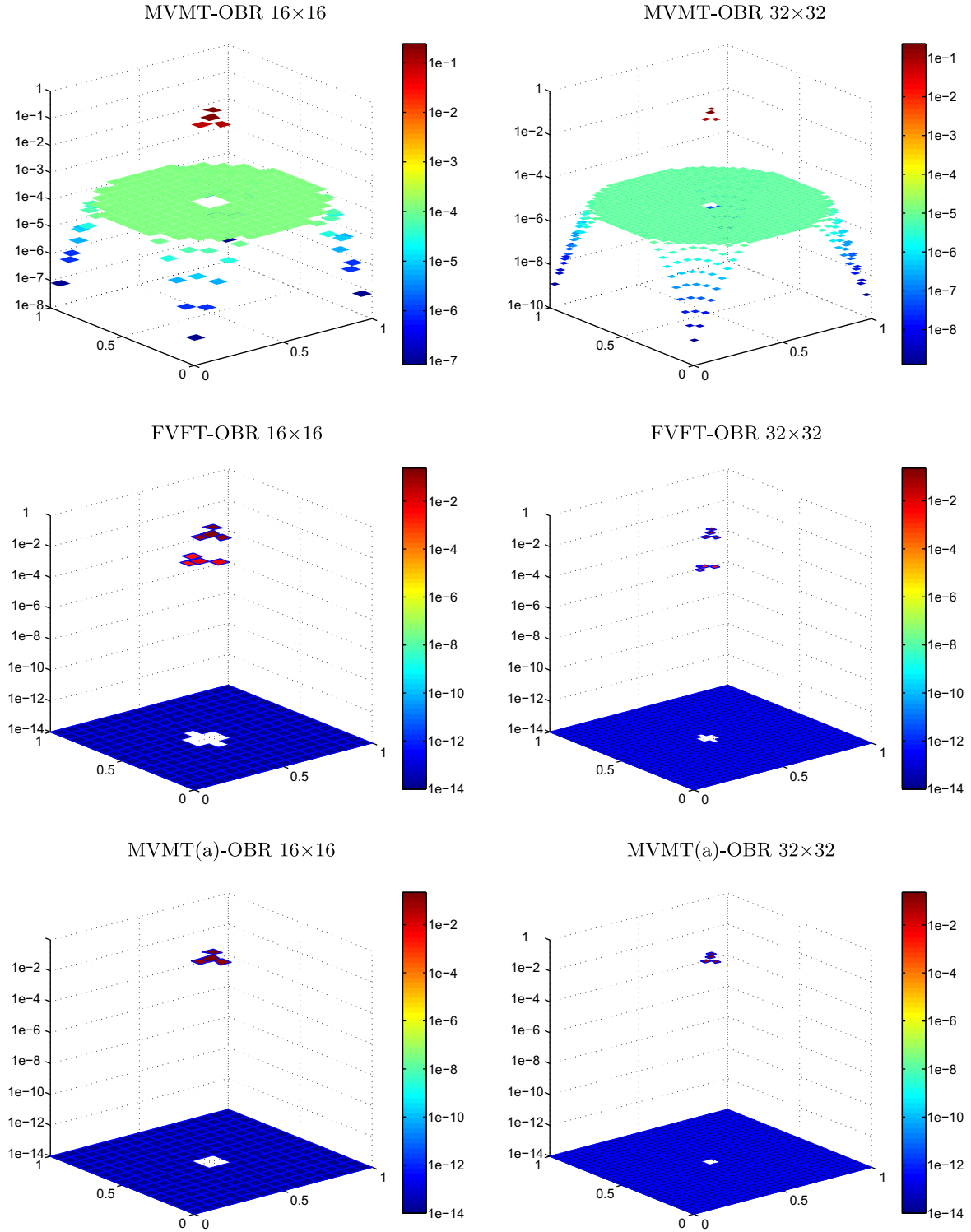


Fig. 8. The magnitudes of mass updates scaled by the number of cells in the mesh, $K|\delta m_i|$, for one step of OBR algorithms subject to the single-vertex motion. The top pane shows the mass updates generated by MVMT-OBR for the 16×16 and 32×32 meshes, resulting in an "asymptotically" local method. The middle pane shows the mass updates generated by FVFT-OBR – a "weakly" local method; the bottom pane shows the mass updates generated by MVMT(a)-OBR – a local method.

(and so, clearly, at each remap step). We also show that only a handful of iterations of our secant algorithm are needed to fully restore mass conservation. The test is set up so that Algorithm 1 is terminated immediately after Step 2. (a), in other

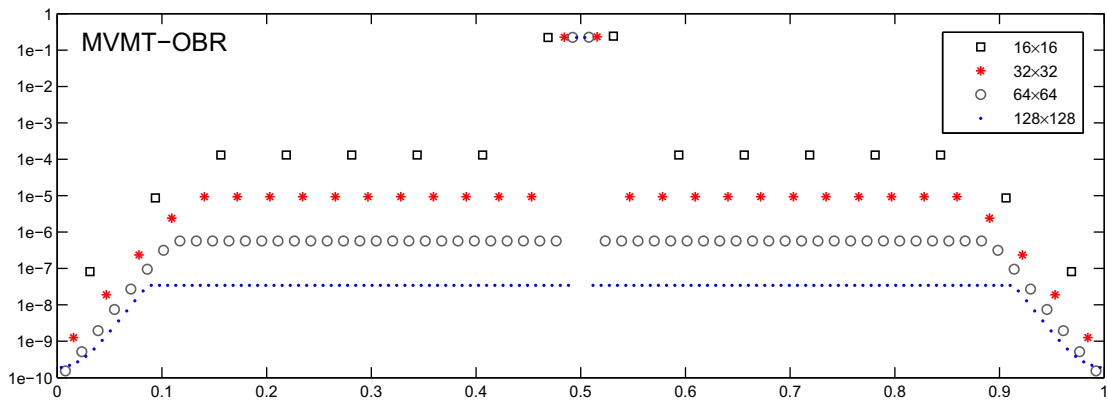


Fig. 9. The magnitudes of mass updates scaled by the number of cells in the mesh, $K|\delta m_i|$, for one step of MVMT-OBOR subject to the single-vertex motion, shown for the mesh sequence $\{16 \times 16, 32 \times 32, 64 \times 64, 128 \times 128\}$, across the $(0,0) - (1,1)$ diagonal of the domain.

Table 8
A comparison of the total mass (i) on the original mesh, (ii) as computed by the “faulty” (nonconservative) MVMT-OBOR algorithm after 320 remaps and (iii) as computed by the proper MVMT-OBOR algorithm, Algorithm 1, after 320 remaps. The average number of secant iterations for the proper MVMT-OBOR algorithm, rounded to the nearest integer, is also reported.

	Sine	% Diff	Peak	% Diff	Shock	% diff
(i) Original mass $\sum_{i=1}^K m_i$	1.0000000000	–	1.0654579749	–	1.5500488281	–
(ii) Final mass $\sum_{i=1}^K m_i(\lambda_0)$	1.0000082955	0.0008	1.0663039593	0.08	1.5499147318	0.009
(iii) Final mass $\sum_{i=1}^K m_i(\lambda^*)$	1.0000000000	0	1.0654579749	0	1.5500488281	0
Avg. secant iterations		1		3		5

Table 9
Comparison of the computational costs of FCR, MVMT-OBOR and FVFT-OBOR, as measured by Matlab™ wall-clock times in seconds, on a single Intel Xeon X5680 3.33 GHz processor, for sine, peak and shock densities and the tensor-product cyclic grid. Ratios of run times of MVMT-OBOR and FVFT-OBOR with respect to FCR are included. The cost of MVMT-OBOR is very close to the cost of FCR, while FVFT-OBOR is on average two times slower.

# Cells	# Remaps	FCR	MVMT-OBR		FVFT-OBR	
		Time (sec)	Time (sec)	Ratio	Time (sec)	Ratio
<i>Sine</i>						
64 × 64	320	4.1	4.1	1.0	6.8	1.7
128 × 128	640	25.1	24.6	1.0	48.7	1.9
256 × 256	1280	177.2	173.1	1.0	384.6	2.2
512 × 512	2560	2049.1	1918.0	0.9	3677.5	1.8
<i>Peak</i>						
64 × 64	320	4.9	4.9	1.0	7.7	1.6
128 × 128	640	28.0	28.6	1.0	53.9	1.9
256 × 256	1280	194.5	192.8	1.0	400.8	2.1
512 × 512	2560	2060.1	2096.5	1.0	4410.5	2.1
<i>Shock</i>						
64 × 64	320	4.6	4.9	1.1	9.0	2.0
128 × 128	640	27.4	29.1	1.0	85.1	3.1
256 × 256	1280	192.5	195.6	1.0	414.4	2.2
512 × 512	2560	2064.9	2146.7	1.0	3117.1	1.5

words, to minimize the given objective function while ensuring bound preservation yet completely ignoring conservation of mass. In this case, the final mass is given by $\sum_{i=1}^K m_i(\lambda_0)$. In a second run, we additionally enforce mass conservation and compare the final masses given by $\sum_{i=1}^K m_i(\lambda^*)$ to those generated by the “faulty” algorithm. We use the 64×64 tensor-product cyclic mesh with 320 remaps. Table 8 fully confirms our claims, i.e., the violation of mass conservation is relatively small and is fully corrected by a few iterations of the MVMT-OBOR algorithm.

Table 10

Comparison of the computational costs of FCR, MVMT-OB and FVFT-OB, as measured by Matlab™ wall-clock times in seconds, on a single Intel Xeon X5680 3.33 GHz processor, for sine, peak and shock densities and the wave cyclic grid. Ratios of run times of MVMT-OB and FVFT-OB with respect to FCR are included. The cost of MVMT-OB is very close to the cost of FCR, while FVFT-OB is on average two times slower.

# Cells	# Remaps	FCR	MVMT-OB		FVFT-OB	
		Time (sec)	Time (sec)	Ratio	Time (sec)	Ratio
<i>Sine</i>						
64 × 64	320	4.4	4.1	0.9	6.9	1.6
128 × 128	640	26.6	24.6	0.9	45.4	1.7
256 × 256	1280	186.3	183.5	1.0	375.2	2.0
512 × 512	2560	1978.4	1858.1	0.9	3977.8	2.0
<i>Peak</i>						
64 × 64	320	4.8	5.0	1.0	6.6	1.4
128 × 128	640	29.0	28.7	1.0	38.5	1.3
256 × 256	1280	189.2	197.3	1.0	551.5	2.9
512 × 512	2560	2002.9	2066.4	1.0	5727.5	2.9
<i>Shock</i>						
64 × 64	320	4.7	5.0	1.1	15.0	3.2
128 × 128	640	29.3	30.0	1.0	86.0	2.9
256 × 256	1280	199.9	218.0	1.1	458.0	2.3
512 × 512	2560	2005.5	2327.4	1.2	3397.0	1.7

Table 11

Comparison of the computational costs of FCR, MVMT-OB and FVFT-OB, as measured by Matlab™ wall-clock times in seconds, on a single Intel Xeon X5680 3.33 GHz processor, for sine, peak and shock densities and the random cyclic grid. Ratios of run times of MVMT-OB and FVFT-OB with respect to FCR are included. The cost of MVMT-OB is very close to the cost of FCR, while FVFT-OB is on average 1.5 times slower.

# Cells	# Remaps	FCR	MVMT-OB		FVFT-OB	
		Time (sec)	Time (sec)	Ratio	Time (sec)	Ratio
<i>Sine</i>						
64 × 64	320	4.2	4.2	1.0	7.4	1.8
128 × 128	640	25.1	25.1	1.0	49.3	2.0
256 × 256	1280	186.1	188.9	1.0	438.3	2.4
512 × 512	2560	2337.2	2120.1	0.9	4599.4	2.0
<i>Peak</i>						
64 × 64	320	5.4	4.9	0.9	6.5	1.2
128 × 128	640	29.4	29.0	1.0	36.9	1.3
256 × 256	1280	196.3	210.2	1.1	263.3	1.3
512 × 512	2560	2479.9	2254.9	0.9	2877.1	1.2
<i>Shock</i>						
64 × 64	320	5.2	5.2	1.0	6.5	1.2
128 × 128	640	30.3	30.4	1.0	34.0	1.1
256 × 256	1280	211.6	243.6	1.2	244.4	1.2
512 × 512	2560	2403.5	2800.3	1.2	2470.5	1.0

4.6. Computational efficiency

Before proceeding to the comparisons of MVMT-OB, FVFT-OB and FCR, it is important to point out that the MVMT-OB algorithm, Algorithm 1, possesses several advantages over the FVFT-OB scheme. The FVFT-OB algorithm relies on a dual QP formulation and a reflective Newton method for its solution, see [1]; the MVMT-OB algorithm, based on the secant method, is much simpler to implement. The average number of iterations needed by the FVFT-OB scheme (2–15) is about twice the number of iterations observed in using MVMT-OB (1–5). Each iteration of FVFT-OB entails the solution of a sparse but very large, $\mathcal{O}(K \times K)$, linear system; each iteration of MVMT-OB solves K fully decoupled single-variable quadratic programs. Finally, the parallelization of FVFT-OB necessitates the parallelization of the linear system solve; MVMT-OB is trivially parallelized and requires minimal communication.

Tables 9–11 compare the computational cost of FCR, MVMT-OB and FVFT-OB for three sequences of cyclic meshes (tensor-product, wave and random) and three target densities (sine, peak and shock).⁴ It is evident that the cost of MVMT-OB is very close to the cost of FCR. Overall, they are about twice as fast as FVFT-OB, with the exception of the remap of the peak and shock densities on random cyclic grids. To investigate this apparent anomaly, we also present results for a tougher, more

⁴ All experiments are performed in Matlab™, Version 7.11.0.584 (R2010b) 64-bit; as in [1], we note that the computational cost of our Matlab™ implementation will be close to that of a Fortran or C implementation, due to a consistent vectorization of algebraic operations.

Table 12

Comparison of the computational costs of FCR, MVMT-OB and FVFT-OB, as measured by Matlab™ wall-clock times in seconds, on a single Intel Xeon X5680 3.33 GHz processor, for LeVeque's combination [16] of the slotted cylinder, cone and smooth hump densities, and the random cyclic grid. Ratios of run times of MVMT-OB and FVFT-OB with respect to FCR are included. The cost of MVMT-OB is very close to the cost of FCR, while FVFT-OB is on average 3.9 times slower.

# Cells	# Remaps	FCR	MVMT-OB		FVFT-OB	
		Time (sec)	Time (sec)	Ratio	Time (sec)	Ratio
<i>LeVeque's combination of slotted cylinder, cone and smooth hump</i>						
64 × 64	320	8.0	7.9	1.0	24.6	3.1
128 × 128	640	40.1	41.7	1.0	149.7	3.7
256 × 256	1280	252.5	255.9	1.0	978.0	3.9
512 × 512	2560	2084.4	2292.7	1.1	9725.4	4.7

Table 13

Comparison of the computational costs of FVFT-OB and MVMT-OB, as measured by Matlab™ wall-clock times in seconds, on a single Intel Xeon X5680 3.33 GHz processor, for sine, peak and shock densities and the tensor-product cyclic grid. In this example the mesh resolution and the number of remaps for MVMT-OB, given in parentheses, are adjusted to match the L_1 error of FVFT-OB. The cost of MVMT-OB is still significantly lower than the cost of FVFT-OB.

# Cells	# Remaps	FVFT-OB		MVMT-OB	
		L_1 error	time (sec)	L_1 error	time (sec)
<i>Sine</i>					
64×64 (71×71)	320 (350)	$4.91\text{e-}4$	6.8	$4.84\text{e-}4$	5.5
128×128 (142×142)	640 (705)	$6.16\text{e-}5$	48.7	$6.04\text{e-}5$	33.8
256×256 (283×283)	1280 (1410)	$7.82\text{e-}6$	384.6	$7.76\text{e-}6$	254.4
512×512 (566×566)	2560 (2825)	$9.89\text{e-}7$	3677.5	$9.93\text{e-}7$	2541.1
<i>Peak</i>					
64×64 (71×71)	320 (350)	$2.55\text{e-}3$	7.7	$2.56\text{e-}3$	6.3
128×128 (148×148)	640 (735)	$8.90\text{e-}4$	53.9	$8.88\text{e-}4$	42.8
256×256 (310×310)	1280 (1545)	$3.10\text{e-}4$	400.8	$3.11\text{e-}4$	339.0
512×512 (646×646)	2560 (3225)	$1.09\text{e-}4$	4410.5	$1.09\text{e-}4$	4252.0
<i>Shock</i>					
64×64 (70×70)	320 (345)	$2.88\text{e-}2$	9.0	$2.87\text{e-}2$	6.0
128×128 (138×138)	640 (685)	$1.75\text{e-}2$	85.1	$1.74\text{e-}2$	36.6
256×256 (271×271)	1280 (1350)	$1.06\text{e-}2$	414.4	$1.06\text{e-}2$	256.4
512×512 (547×547)	2560 (2730)	$6.35\text{e-}2$	3117.1	$6.36\text{e-}3$	2612.2

realistic example that uses the random grid motion. For this purpose, we use a test, suggested by LeVeque, that combines various density profiles – a smooth hump, a cone and a slotted cylinder – in one, see [16]. From Table 12 it is clear that for such problems FVFT-OB is significantly slower than MVMT-OB and FCR. In this case, the average performance penalty is 3.9. As before, the computational costs of MVMT-OB and FCR are virtually identical. We expect these trends to hold in real-world simulations.

Table 13 addresses an observation made in Section 4.2, namely that FVFT-OB may exhibit better absolute accuracy in comparison with MVMT-OB. Therefore, in order to make the cost comparisons fair, we adjust the mesh resolution and the number of remaps for MVMT-OB, given in parentheses, to match the L_1 error of FVFT-OB. The computational cost of MVMT-OB is still significantly lower than the cost of FVFT-OB.

5. Conclusions

We have developed a new optimization-based conservative, local bounds and linearity-preserving remapping method – MVMT-OB. Contrary to the existing flux-based remapping methods, which consider mass fluxes exchanged between the new and old mesh cells, MVMT-OB is based on aggregate mass transfer, that is, the primary unknowns in the optimization formulation are the net mass updates between the new and old cells. The optimization setting lets us treat mass conservation as one of the constraints. The choice of primary variables leads to a singly linearly constrained quadratic optimization problem with simple bounds. The structure of the optimization problem and the fact that in remap the old and new meshes are close allow us to develop an efficient and easily parallelizable optimization algorithm.

We have demonstrated numerically that MVMT-OB is “asymptotically” local, that is, the domain of influence of a mesh change is limited, as the mesh size goes to zero, to its immediate neighborhood. We have also presented a version of MVMT-OB with the “active cell option” – MVMT(a)-OB, which preserves all key features of MVMT-OB yet treats limited changes in the mesh much like explicit, purely local remappers. We recommend the use of MVMT(a)-OB whenever a large number of cells are static.

Numerical studies demonstrate that the new algorithm is as accurate and robust as the previously developed flux-based optimization algorithm – FVFT-OBR, but has the same computational cost as the state-of-the-art explicit flux-corrected remapping method – FCR. As future work we plan to develop an algorithm for the recovery of mass fluxes from the aggregate mass updates. This may be needed, for example, in the context of remapping multiple tracers in climate applications. Another possible extension is a high-order (for example, quadratic-preserving) remapping algorithm, which will require the development of new local bounds that take into account the presence of local extrema.

Acknowledgments

All authors acknowledge funding by the DOE Office of Science Advanced Scientific Computing Research (ASCR) Program. D. Ridzal and M. Shashkov also acknowledge funding by the Advanced Simulation & Computing (ASC) Program.

Our colleagues Kara Peterson, Guglielmo Scovazzi and Joseph Young provided valuable comments that helped improve this work. We also thank Dmitri Kuzmin for useful insights.

References

- [1] P. Bochev, D. Ridzal, G. Scovazzi, M. Shashkov, Formulation, analysis and numerical study of an optimization-based conservative interpolation (remap) of scalar fields for arbitrary Lagrangian–Eulerian methods, *Journal of Computational Physics* 230 (2011) 5199–5225.
- [2] P. Bochev, D. Ridzal, G. Scovazzi, M. Shashkov, Constrained optimization based data transfer – a new perspective on flux correction, in: D. Kuzmin, R. Löhner, S. Turek (Eds.), *Flux-Corrected Transport*, second ed., Principles, Algorithms and Applications, Springer Verlag, Berlin, Heidelberg, 2012, pp. 345–398.
- [3] C. Hirt, A. Amsden, J. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, *Journal of Computational Physics* 14 (1974) 227–253.
- [4] J.S. Peery, D.E. Carroll, Multi-material ale methods in unstructured grids, *Computer Methods in Applied Mechanics and Engineering* 187 (2000) 591–619.
- [5] P. Colella, Multidimensional upwind methods for hyperbolic conservation laws, *Journal of Computational Physics* 87 (1990) 171–200.
- [6] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, *Computer Methods in Applied Mechanics and Engineering* 99 (1992) 235–394.
- [7] R. Loubere, M.J. Shashkov, A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian–Eulerian methods, *Journal of Computational Physics* 209 (2005) 105–138.
- [8] R. Loubere, M. Staley, B. Wendroff, The repair paradigm: new algorithms and applications to compressible flow, *Journal of Computational Physics* 211 (2006) 385–404.
- [9] R. Liska, M. Shashkov, P. Váchal, B. Wendroff, Optimization-based synchronized flux-corrected conservative interpolation (remapping) of mass and momentum for arbitrary Lagrangian–Eulerian methods, *Journal of Computational Physics* 229 (2010) 1467–1497.
- [10] T.F. Coleman, L.A. Hulbert, A globally and superlinearly convergent algorithm for convex quadratic programs with simple bounds, *SIAM Journal on Optimization* 3 (1993) 298–321.
- [11] L.G. Margolin, M. Shashkov, Second-order sign-preserving conservative interpolation (remapping) on general grids, *Journal of Computational Physics* 184 (2003) 266–298.
- [12] L. Margolin, M. Shashkov, Second-order sign-preserving remapping on general grids, Technical Report LA-UR-02-525, Los Alamos National Laboratory, Los Alamos NM 87545, 2003.
- [13] Y.-H. Dai, R. Fletcher, New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds, *Mathematical Programming* 106 (2006) 403–421.
- [14] J. Nocedal, S.J. Wright, *Numerical Optimization*, Springer Verlag, Berlin, Heidelberg, New York, 1999, first ed..
- [15] J.E. Dennis Jr., R.B. Schnabel, *Numerical Methods for Nonlinear Equations and Unconstrained Optimization*, SIAM, Philadelphia, 1996.
- [16] R.J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow, *SIAM Journal on Numerical Analysis* 33 (1996) 627–665.