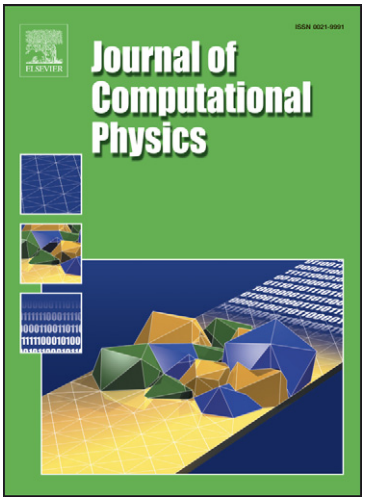
Accepted Manuscript



Adaptive Moment-of-Fluid Method

Hyung Taek Ahn, Mikhail Shashkov

PII: S0021-9991(08)00657-8
DOI: 10.1016/j.jcp.2008.12.031
Reference: YJCPH 2390

To appear in: *Journal of Computational Physics*

Received Date: 11 May 2008
Revised Date: 8 October 2008
Accepted Date: 16 December 2008

Please cite this article as: H.T. Ahn, M. Shashkov, Adaptive Moment-of-Fluid Method, *Journal of Computational Physics* (2009), doi: 10.1016/j.jcp.2008.12.031

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

Adaptive Moment-of-Fluid Method

Hyung Taek Ahn [a,*], Mikhail Shashkov [b]

[a] *School of Naval Architecture & Ocean Engineering*
University of Ulsan, Ulsan 680-749, Korea

[b] *Theoretical Division, Group T-7*
Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Abstract

A novel adaptive mesh refinement (AMR) strategy based on the moment-of-fluid (MOF) method for volume-tracking of evolving interfaces is presented. Moment-of-fluid method is a new interface reconstruction and volume advection method using volume fractions as well as material centroids. The mesh refinement criterion is based on the deviation of the actual centroid obtained by interface reconstruction from the reference centroid given by moment advection process. The centroid error indicator detects not only high curvature regions but also regions with complicated subcell structures like filaments. A new Lagrange+remap scheme is presented for advecting moments, which includes Lagrangian backtracking, polygon intersection-based remapping and forward tracking to define the material centroid. The effectiveness and efficiency of AMR-MOF method is demonstrated with classical test problems, such as Zalesak's disk and reversible vortex problem. The comparison with previously published results for these problems shows the superior accuracy of the AMR-MOF method over other methods. In addition, two new test cases with severe deformation rates are introduced, namely droplet deformation and $\mathcal{S}$-shape deformation problems, for further demonstration of the capabilities of the AMR-MOF method.

Key words: Interface reconstruction, advection, moment-of-fluid, volume-of-fluid, adaptive mesh refinement.
PACS: 02.70.Bf

* Corresponding author. Tel: +82-52-259-2164, Fax: +82-52-259-2836
Email addresses: htahn@ulsan.ac.kr (Hyung Taek Ahn), shashkov@lanl.gov (Mikhail Shashkov).

ACCEPTED MANUSCRIPT

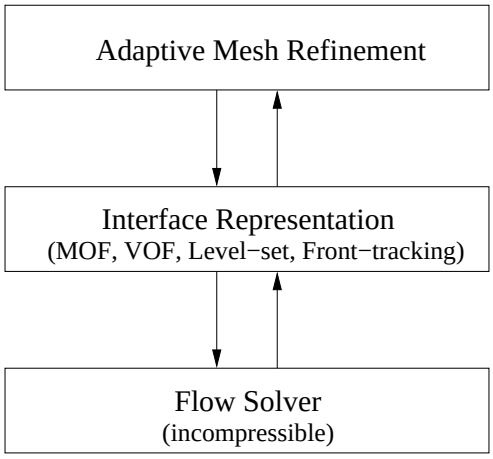


Fig. 1. Conceptual structure of the adaptive mesh refinement strategy for interfacial flow computation. In volume-tracking methods (MOF or VOF), the *Interface Representation* module is typically composed of (i) interface reconstruction and (ii) advection steps.

1 Introduction and background

One of the popular strategies for improving accuracy in computational physics is using adaptive mesh refinement (AMR). AMR technique is being widely used for various types of problems [1–12].

Although the flows with evolving interfaces are an appropriate class of problem with potential for adaptivity, the application of AMR on such problem is relatively rare compared to the flow problems without interfacial phenomena. For example, in proceedings of recent conference on AMR, [9], only two papers were related to the multi-material flows [13,14].

We also want to mention the following papers on adaptive mesh refinement for interfacial flows: for those using volume-of-fluid (VOF) type methods [15–18], those using level-set method [19,20], and front-tracking method [21].

Conceptual structure of AMR strategy for interfacial flows is presented in Fig. 1.

In this paper we are interested in the development of AMR type methods for interfacial flows which use volume tracking methods like VOF for two materials - dark and light material. In volume tracking methods, the instantaneous material interface is described by volume fractions f_c^{dark} and f_c^{light}, which indicates how much volume of each material is present in cell, i.e. c - $f_c^{dark} = V_c^{dark}/V_c$, $f_c^{light} = V_c^{light}/V_c$. Because $V_c^{dark} + V_c^{light} = V_c$ volume fractions are complimentary to each other - $f_c^{light} = 1 - f_c^{dark}$. For this reason, in VOF methods for two materials, one usually use only volume fraction of one of the materials and drops material index. Therefore, we will use notation

ACCEPTED MANUSCRIPT

$f_c = f_c^{dark}$, and where it is not ambiguous we will use term *material* meaning dark material. For cells completely filled with the dark material $f_c = 1$, and for the cell where the dark material is not present f_c=0. For *mixed cells*, partially filled by the dark material we have $1 > f_c > 0$. In most volume tracking methods, [22], interface representation phase of the AMR method (see Fig. 1) is composed of interface reconstruction and some procedure for evolving volume fraction in time (usually called advection) in accordance with velocities obtained by flow solver. The reconstructed interface is used in the advection step.

One of the most important question in AMR methods is refinement/derefinement criterion. In this paper we will only discuss refinement/derefinement criteria in mixed cells - that is, what is the appropriate level of refinement needed to represent the interface. The simplest approach is to use the same prescribed level of refinement in all mixed cells and their neighbors [23,15,24]. In [25], the authors suggest to uniformly refine all mixed cells where the volume fraction value lies in the following limits: $0.8 \geq f_c \geq 0.2$. Then volume fractions are recomputed using some remapping algorithm and the refinement procedure is repeated until some prescribed level of refinement is reached. All mentioned approaches do not take into account the complexity of the interface.

In [26], there is one example where the norm of the local gradient of the volume fraction is used as a refinement/derefinement criterion. Next level of sophistication, which is used in practice is to use some estimates for curvature of interface as criterion for refinement/derefinement [6,27,26,18,19]. There are several problems with these approaches. First of all, to obtain reliable estimate gradient of the volume fraction or estimate for curvature from volume fraction one needs sufficiently fine resolution. Theoretically, this leads to vicious circle - to obtain a good estimate one needs enough resolution and at the same time one is trying to use this estimate to decide what resolution is needed. However, in practice a refinement criterion based on curvature gives good results. The more serious problem is related to the fact that complexity of an interface not restricted just to curvature; for example, interface can have complex topology like filaments or subcell size droplets. Our opinion is that for such situations curvature estimation does not make much sense.

In series of recent reports and papers we have introduced the new moment-of-fluid (MOF) method, [28–33]. The MOF method can be thought of as a generalization of the VOF method. In the VOF method, material volume (the zeroth moment) is advected with local velocity and the interface is reconstructed based on the updated (reference) volume fraction data. In the MOF method, the material volume (zeroth moment) as well as centroid (ratio of the first moment with respect to the zeroth moment) are advected and the interface is reconstructed based on the updated moment data (reference volume and reference centroid). In the MOF method, the computed interface is

ACCEPTED MANUSCRIPT

chosen to match the reference volume exactly and to provide the best possible approximation to the reference centroid of the material.

By using the centroid information, the volume tracking with dynamic interfaces can be performed much more accurately. Furthermore, with this extension of using the moment data, the interface in a particular cell can be reconstructed independently from its neighboring cells. With the advantages of MOF method over the VOF method, our opinion is that the MOF method is a next generation of the volume-tracking interfacial flow computation method evolved from VOF method.

In this paper, we present a very accurate and efficient adaptive mesh refinement strategy for volume-tracking interfacial flow computations based on the moment-of-fluid method. In the new AMR-MOF method the distance between reference centroid and actual centroid computed from reconstructed interface is used as refinement criterion.

Below in this section, we first review the idea of piecewise linear interface calculation (PLIC) method and standard MOF interface reconstruction method. Next, we briefly describe how to obtain the data for MOF interface reconstruction. Then, we introduce the motivation and an algorithmic overview of AMR-MOF method, which is the main topic of this paper.

1.1 Piecewise linear interface calculation (PLIC)

In PLIC methods, the interface between two materials in each mixed cell is represented by a plane (line in 2D). It is convenient to specify this plane in *Hessian normal form*

$$\mathbf{n} \cdot \mathbf{r} + d = 0\,, \tag{1}$$

where $\mathbf{r} = (x, y)$ is a point on the interface, $\mathbf{n} = (n_x, n_y)$ are components of the unit normal to the interface, and d is the signed distance from the origin to the interface.

The principal reconstruction constraint is local volume conservation, i.e. the reconstructed interface must truncate the cell, c, with a volume equal to the reference volume V_c^{ref} of the material (or equivalently, the volume fraction $f_c^{ref} = V_c^{ref}/V_c$, where V_c is the volume of the entire cell c). Here we have introduced superscript *ref* to emphasize that reference quantities are input parameters at interface reconstruction stage and need of such notation will be more clear in the next section, where other reference quantities are introduced, which are not matched exactly.

ACCEPTED MANUSCRIPT

PLIC methods differ in how the interface normal $\mathbf{n}$ is computed. In the VOF method, the interface normal ($\mathbf{n}_c$) for cell-c is computed from the volume fraction data on the stencil composed of cell-c as well as its neighbors. In MOF method, the interface normal, $\mathbf{n}_c$ is computed from moment data, i.e. volume fraction and material centroids, in cell-c only.

Once the interface normal $\mathbf{n}_c$ is computed, the interface is uniquely defined by computing the distance d_c satisfying the reference volume V_c^{ref} exactly.

1.2 Moment-of-fluid interface reconstruction

The *moment-of-fluid* (MOF) interface reconstruction method was first introduced in [28,30], for interface reconstruction in 2D. The 3D extension for the arbitrary polyhedral mesh and multi-material case is described in [32].

To describe main idea of MOF method we need to introduce some definitions. For given material region, Ω, the zeroth moment (volume) and first moment are defined as follow as

$$M_0(\Omega) = \int_\Omega dV\,, \quad \mathbf{M}_1(\Omega) = \int_\Omega \mathbf{x}\, dV\,. \tag{2}$$

Centroid of the material region Ω is the ratio of first and zeroth moments

$$\mathbf{x}_\Omega = \frac{\mathbf{M}_1(\Omega)}{M_0(\Omega)}\,. \tag{3}$$

Let us assume that for each mixed cell we know, not only, the reference volume fraction f_c^{ref}, but also reference centroid $\mathbf{x}_c^{ref}$. We need to emphasize that for the interface reconstruction algorithm the reference volume fraction and the reference centroid are input data, which is supplied by some other algorithm (advection, for example). Therefore these quantities have errors and moreover there maybe no real material configuration which exactly matches both reference volume fraction and reference centroid.

In the MOF method, the computed interface is chosen to match the reference volume exactly and to provide the best possible approximation to the reference centroid of the material. That is, in MOF, the interface normal, $\mathbf{n}$, is computed by minimizing (under the constraint that the corresponding pure subcell has exactly the reference volume fraction in the cell) the following functional:

$$E_c^{MOF}(\mathbf{n}) = \| \mathbf{x}_c^{ref} - \mathbf{x}_c(\mathbf{n}) \|^2 \tag{4}$$

ACCEPTED MANUSCRIPT

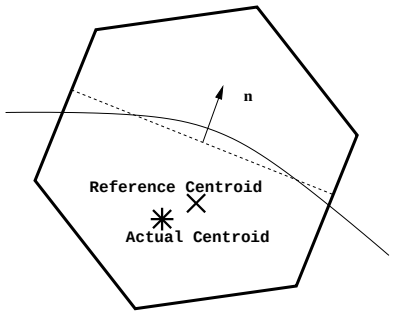


Fig. 2. Stencil for MOF in two dimensions. The stencil for MOF interface reconstruction is composed of only the cell under consideration. The MOF method can be used for arbitrary polygonal cells (polyhedral cells in 3D). The solid curved line represents the true interface, and the dashed straight line represents the piece-wise linear, volume fraction matching interface at the cell.

where $\mathbf{x}_c^{ref}$ is the reference material centroid and $\mathbf{x}_c(\mathbf{n})$ is the actual (reconstructed) material centroid with given interface normal $\mathbf{n}$.

The implementation of the MOF method requires the minimization of the nonlinear function (of one variable in 2D and of two variables in 3D) given in (4). The computation of $E_c^{MOF}(\mathbf{n})$ requires the following steps. First, for a given $\mathbf{n}$ we find the parameter d of the plane such that the volume fraction in cell c exactly matches f_c^{ref}. Second, we compute the centroid of the resulting subcell containing the reference material. This is a simple calculation, described, for example, in [33,34]. Finally, one computes the distance between the actual and reference centroids. The MOF method is linearity-preserving, that is, it reconstructs linear interfaces exactly.

The MOF method uses information about the volume fraction, f_c^{ref} as well as centroid, $\mathbf{x}_c^{ref}$ of the material, but *only* from the cell c under consideration. No information from neighboring cells is used, as illustrated in Fig. 2.

1.3 Obtaining reference volume fraction and reference centroid information

To use MOF method for interface reconstruction one needs to have reference volume fractions f_c^{ref} and reference centroid $\mathbf{x}_c^{ref}$ for each mixed cell c. There are two distinct situations: *static* reconstruction and *dynamic* reconstruction.

Static reconstruction, described in Section 3 is used to represent "exact" material configuration on given mesh using PLIC. Exact material configuration can be provided in different ways but in any case it allows us to compute reference volume fractions and reference centroids for any mesh with the same accuracy with which material configuration is described. Static interface reconstruction is used for initialization of the problem.

ACCEPTED MANUSCRIPT

In case of dynamic reconstruction (Section 4), which is used during the time evolution, the reference volume fractions and reference centroids are obtained by "advection" of these quantities using velocity field provided by flow solver. There are a lot of different methods for advection of volume fraction (see for example, [22], for review). In context of MOF method we also need to advect centroids. One of the possible methods to advect volume fractions and centroids is described in [28,30]. This method is close in flavor to semi-Lagrangian techniques [35,36] and has a lot of similarities with methods described in [37–40] and can be characterized as cell-based Lagrange plus remap. In case of MOF it is used both for advecting volume fractions and centroids.

In case AMR-MOF we have found that to improve accuracy we need to modify method from [28,30]. A new method is described in Sections 4.2, 4.3 and 4.4.

1.4 AMR-MOF: Design principles

In many physical simulations, the region of interest is often localized (e.g. boundary layer, wake behind of a body, shock front, or multi-material/phase interfaces) and the computational resources can be selectively utilized for improving the accuracy in such regions. Refining the mesh in such regions, that is adaptive mesh refinement, is very natural way of improving accuracy for a given set of computational resources.

For interfacial flows, there is a clear definition of the localized region of interest: the region around the interface. In most interfacial flow computations based on volume-tracking, the major issue is how to accurately resolve the material configuration which is again defined by the interface. Ideally, adaptation algorithm is supposed to detect the interface complexity intelligently. The two immediate properties characterizing the interface complexity would be *curvature* and *topology* of the interface.

Fig. 3 illustrates representative interface features. We note that all features illustrated in the Fig. 3 are at the subcell scale (their length scale is less than those of the unrefined mesh) and also independent from the features of their neighboring cells (neighboring cells may not have similar features). It is interesting to note that after we have created illustrative Fig. 3 we have discovered a very similar figure in [18]. We also note that the subcell scale filament can be represented by two non-contiguous linear segments [41].

The cornerstone of any AMR method is the refinement criterion. In the context of modeling of interfacial flows, the refinement criterion is supposed to detect severe deformation of the interface in a wide spectrum of length scales. In this paper, the refinement criterion is based on the error indicator, defined as the deviation of the actual centroid of the reconstructed material configuration

ACCEPTED MANUSCRIPT

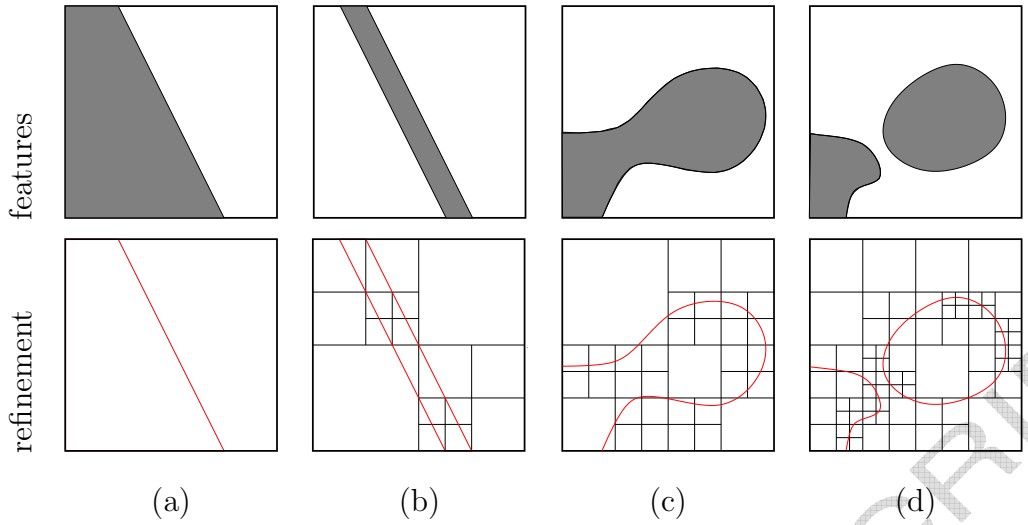


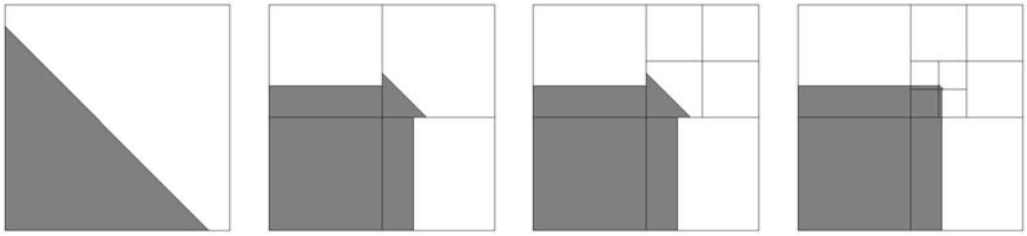
Fig. 3. Subcell scale interface features with different curvature and topology. The thick solid line indicates the square cell boundary, and gray region indicates material configuration. Top row – material configuration, bottom row – possible AMR-MOF refinement pattern. Four representative interface features within a square cell are illustrated: (a) one piece of the material inside the cell — interface is the segment of the straight line (curvature is zero); (b) two disjoint pieces of the white material — subcell thickness filament of dark material, curvature has meaning only for each segment of the straight line and equal to zero, but one curvature per cell does not make sense; (c) one piece of dark material with complicated shape, only average curvature makes sense; (d) disjoint pieces of dark material (subcell size droplet), each of pieces has high average curvature.

from the reference centroid. As we will show in Section 2 the centroid error is the effective measure of the discrepancy between the reconstructed and the reference material configurations defined by the reference volume fraction and the reference centroid. If the centroid error is higher than a certain tolerance, then the cell is refined. It is important to note that refinement criterion is based on the same data, namely centroid information, which is used in MOF interface reconstruction.

Next question is how to refine? This issue is closely related to what data structures are used to describe the refined mesh. According to [20], two most popular types of refinement are patch based [1,42,43], and tree based refinement [16,18,6,26,25]. For general discussion and more references related to spatially adaptive techniques, we refer the interested reader to [20].

In this paper we use quadtree refinement, where a cell flagged for refinement is subdivided into four subcells. From this point of view, this is isotropic refinement as opposed to anisotropic refinement [10]. Many codes which use quadtree data structures have constraint that the level of refinement in neighboring cells can differ only by one level. This constraint is usually related to the available flow solver and simplicity of data communication between different

ACCEPTED MANUSCRIPT



level-0 level-1 level-2 level-3

Fig. 4. AMR-MOF interface reconstruction on adaptively refined meshes. From the left (level-0) to the right (level-3) refinement.

levels.

In this paper we are not dealing with flow solver but we want to mention that modern discretization techniques allows us to use quadtree meshes without constraints related to level of refinement in neighboring cells, [44,20], and therefore, we use such *unconstrained* quadtree meshes in this paper.

To give an idea how quadtree refinement and corresponding data structures in application to interface reconstruction may look like we consider simple illustrative example of static interface reconstruction (initialization), Fig. 4, and Fig. 5. It is the reconstruction of a square material region occupying $[0., 0.64]^2$ within a cell covering $[0, 1]^2$ square domain (equivalent to a level 0 mesh, which consist only of one cell). The quadtree data structure developed in the process of the corner reconstruction example, as shown in Fig. 4, is illustrated in Fig. 5.

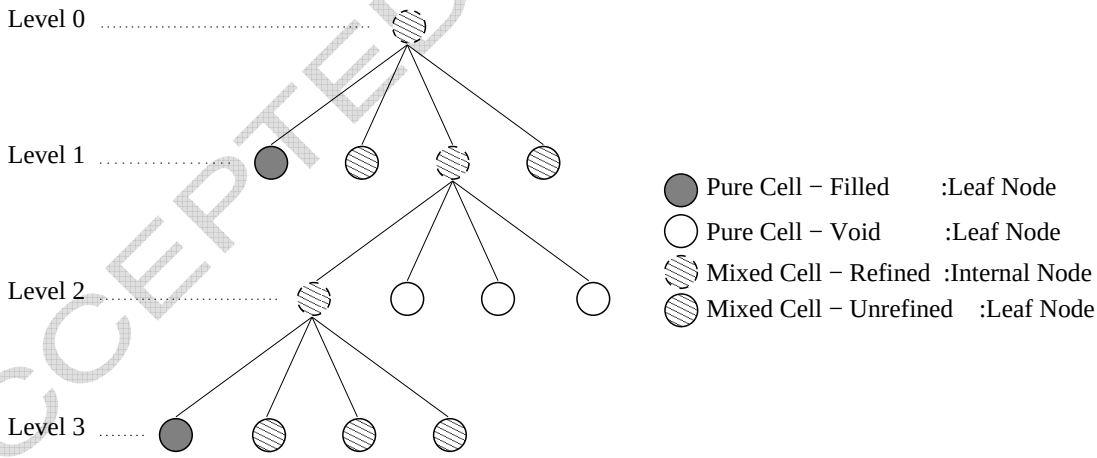


Fig. 5. Quadtree structure of AMR-MOF reconstruction shown in Fig. 4. Correspondence between Fig. 4 established by introducing local enumeration of children of parent cell counter-clockwise starting from left-bottom child to left-top child. Then for each level the left circle corresponds to left-bottom child and the right circle correspond to left-top child.

At each AMR iteration, mixed cells are refined into four child cells. Once a

ACCEPTED MANUSCRIPT

cell is refined, then the reference moment data is recomputed on the child cells for the next interface reconstruction stage. If a mixed cell has centroid error less than a given tolerance (e.g. child cell with linear interface), the mixed cell is not flagged into refinement.

This adaptive refinement strategy results in adequate piece-wise linear representation of interface on adaptive mesh.

1.5 Organization of the paper

The rest of the paper is organized as follow. In Section 2 we numerically justify the use of the error in centroid position as a criterion for mesh refinement. Static AMR-MOF interface reconstruction and numerical example of interface reconstruction of a multi-element airfoil geometry are described in Section 3. Dynamic AMR-MOF is described in Section 4. The moment advection is first explained for case of uniform mesh case, Section 4.3, and then extended to AMR meshes, Section 4.4. Different approaches for time-stepping are described in Section 4.5. To demonstrate the effectiveness of the AMR-MOF method, various test problems are presented in Section 4.6. In Section 4 we describe a numerical tests related to dynamic interface reconstruction. The effect of quality of the initial interface representation is investigated in Section 4.6.1. In Sections 4.6.2, 4.6.3, and 4.6.4 two classical test problems are presented, namely Zalesak's notched disk rotation and single reversible vortex problem. Comparative studies with other published results are presented for both standard MOF and AMR-MOF. In addition to those classical problems, two new test problems with severe deformation rates are presented in Sections 4.6.5, 4.6.6. Finally in Section 5, we present summary of the results obtained in the paper and consider future work.

2 Centroid error as refinement criterion

In this section, we demonstrate that the centroid error, the error indicator for AMR-MOF method, can detects different feature of the interface. This includes not only the local curvature of the interface but also the topology of material region within the cell.

We first demonstrate the local curvature sensing capability of AMR-MOF method. If the true interface is straight line, the MOF method reconstructs the interface exactly, i.e. centroid error is zero. If the interface is curved, the linear interface computed by MOF method will deviate from the true curve. In this case, MOF computes non-zero centroid error. It can also be expected

ACCEPTED MANUSCRIPT

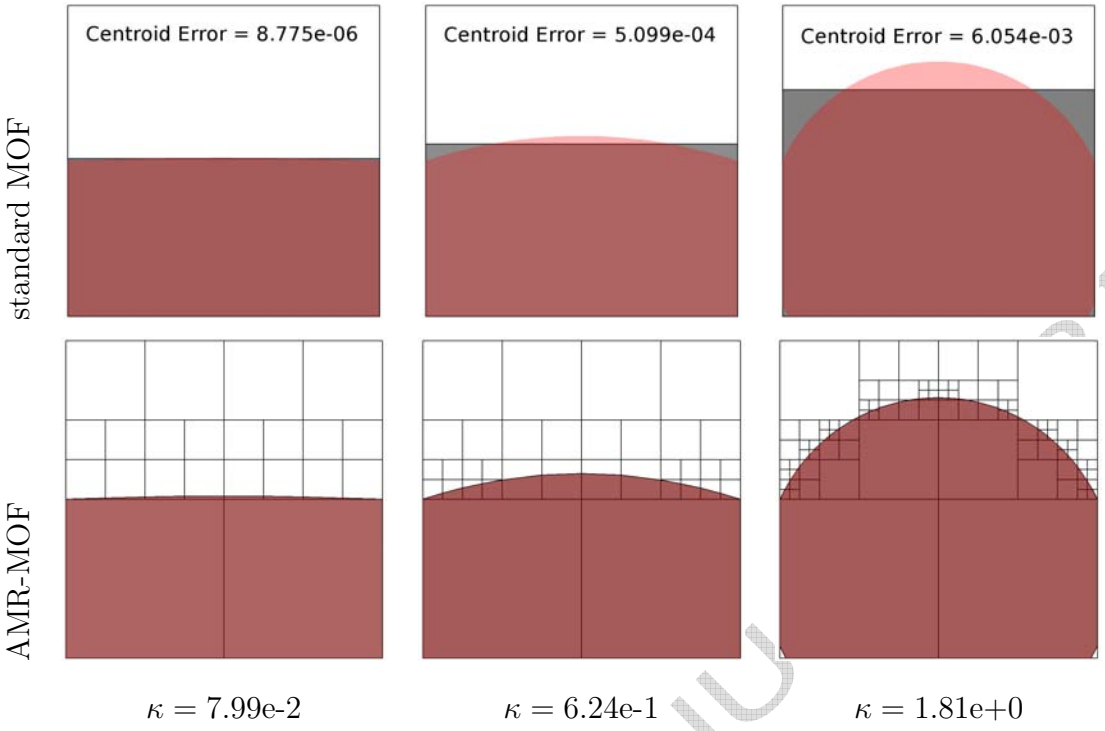


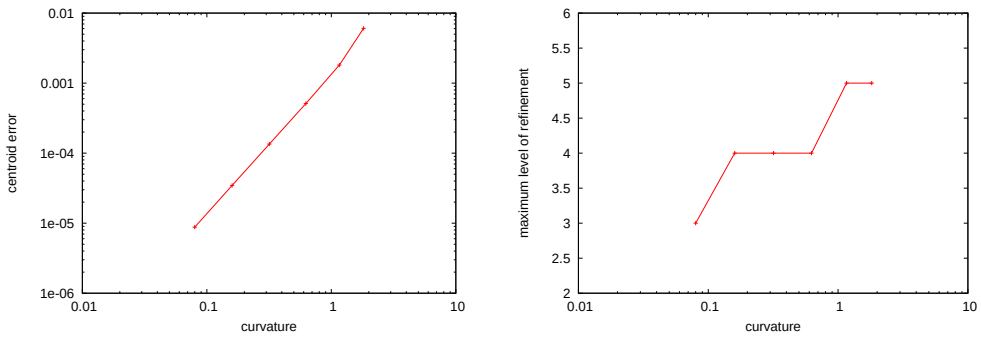
Fig. 6. Curvature effect on the interface reconstruction on a square cell of $[0,1]^2$. Top row shows the standard MOF interface reconstruction. Bottom row shows the AMR-MOF interface reconstruction. To emphasize the quality of the reconstruction, the true material region (red) is overlapped on top of reconstructed material region (gray). For standard MOF reconstruction, higher curvature results in higher deviation of reconstructed material region from the true material region. This is directly indicated by the *centroid error* (Euclidean distance between the actual and reference centroids, $\sqrt{E_c^{MOF}}$) and also as displayed in Fig. 7-(a). For AMR-MOF reconstruction, the higher curvature results in higher level of refinement to decrease of the centroid error below the prescribed tolerance. This trend is also confirmed in Fig. 7-(b).

that *the higher curvature of the interface, the higher centroid error* due to the linear approximation of the curved interface.

This implies that the AMR-MOF method based on the centroid error detects the curvature of the interface. This claim is supported by the examples illustrated in Fig. 6. As the interface curvature ($\kappa = \frac{1}{r}$, where κ is curvature and r is the radius of circular interface) increases, the linear interface produced by standard MOF method results in large discrepancy between the true and reconstructed regions, but with AMR-MOF the discrepancy between the reconstructed material region and highly curved original material region is removed.

Fig. 7 confirms this observation. The centroid error produced by standard MOF method (see, Fig. 7-(a)) shows that the error is increasing quadratically

ACCEPTED MANUSCRIPT



(a) (b)

Fig. 7. Curvature effects on the centroid error and level of refinement required. Top graph shows the centroid error computed by standard MOF reconstruction. It confirms that the centroid error (here, it is measured by Euclidean distance between the actual and reference centroids, $\sqrt{E_c^{MOF}}$) is *quadratic* with respect to the curvature, i.e. the centroid error quadruples as the curvature doubles. Bottom graph shows that more levels of refinement are required to decrease the high centroid error induced by the high curvature interface. Refinement is performed until $E_c^{MOF} < 1.e\text{-}12$. These two figures directly correspond to the results presented in Fig. 6.

with respect to the interface curvature. This result is in accordance to the analysis in [29]. We note that the analysis in [29] assumes the local radius ($r = 1/\kappa$) is larger than local cell size. The slight super-quadratic behavior of the centroid error at the highest curvature is due to the local radius falling below the cell size, where the analysis is not valid. The maximum level of refinement required for the AMR-MOF to achieve a centroid error less than the given tolerance is displayed in Fig. 7-(b). It is clear that higher the curvature of the interface, the more refinement is required to decrease the centroid error.

Another important aspect of interface complexity is its topology within the cell. For example, there can be multiple disjoint pieces of the material within a cell. Such subcell-scale material configuration cannot be correctly reconstructed with methods without refinement. Example of filament reconstruction with subcell thickness is illustrated in Fig. 8. As shown in top row of the Figure, the standard MOF reconstruction cannot resolve the filament configuration as it falls inside of the cell. However, as shown in the bottom row, the AMR-MOF reconstruction, based on the centroid error indicator, correctly resolves the subcell configuration of the filament.

The previous examples confirm that our error indicator, the centroid error, is not only senses the local curvature but also reflects the overall accuracy of reconstructed interface for complex material configurations. This centroid error indicator eventually guides the AMR-MOF method to produce the accurate interface reconstruction. Numerical examples presented in following Sections confirm this conclusion.

ACCEPTED MANUSCRIPT

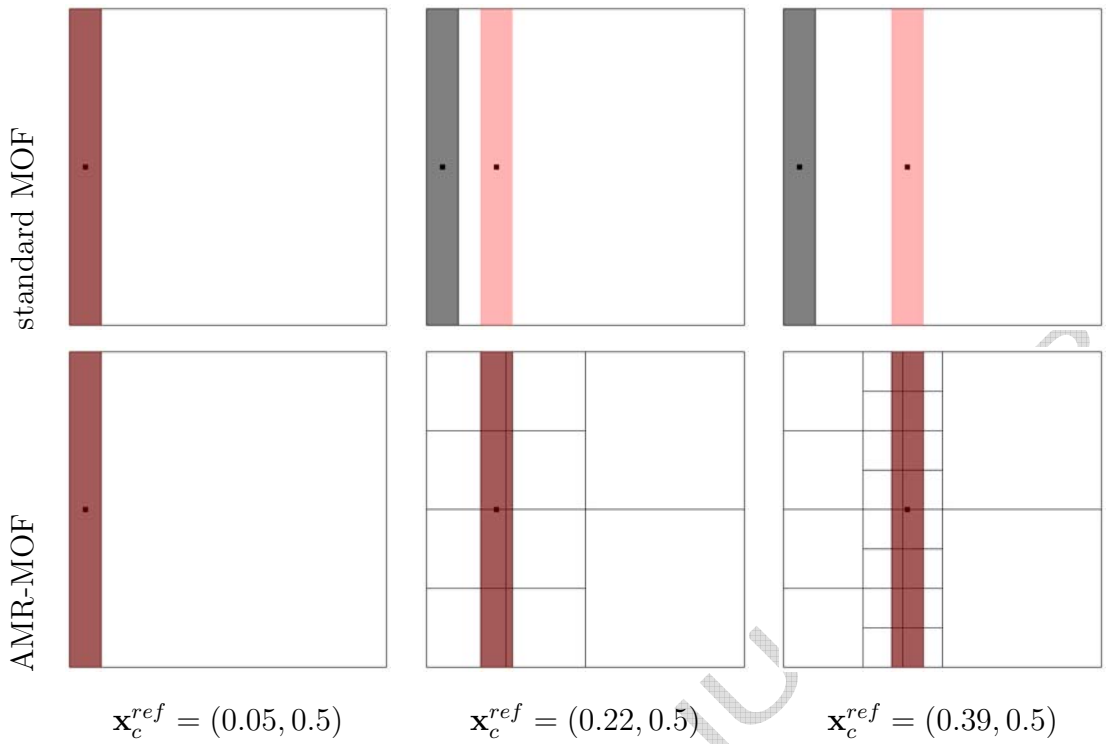


Fig. 8. Reconstruction of subcell thickness ($w = 0.1$) filament within a square cell covering $[0, 1]^2$ domain. From the left, the reference filament configuration (indicated by transparent red) is translated to the right with increment of $\Delta x = 0.17$. The gray region indicates reconstructed filament region. Actual and reference centroids are marked with black square dots. Top – standard MOF interface reconstruction, bottom – AMR-MOF interface reconstruction. As filament moves inside of the cell, standard MOF fails to represent the true material configuration, while AMR-MOF resolves the true material configuration.

3 Static interface reconstruction - initialization

3.1 Logic of static interface reconstruction

The statement of the problem for AMR-MOF static interface reconstruction is as follows: for given original material configuration, represent the reconstructed material region by PLIC on adaptively refined mesh. The main algorithm is composed of the following three steps:

(i) identify the cells to be refined (refinement criterion)
(ii) compute reference moment data (de-referencing)
(iii) reconstruct interface on the AMR mesh using MOF.

The refinement criterion is based on the centroid error, the departure of the actual (reconstructed) centroid from the reference centroid. If this error is big-

ACCEPTED MANUSCRIPT

ger than a certain tolerance, then the cell is refined. The second part can be referred to as de-referencing. For the static cases (e.g. initial stage of interfacial flow simulation), the reference material configuration is usually given by an analytical form or body fitted meshes describing the original geometry. In examples presented in this paper the reference moment data (volume and centroid) representing the true material configuration is computed by exact intersection of the cell and the original geometry. Finally, the interface is reconstructed on the AMR mesh using MOF with the provided reference data. It completes one AMR-MOF iteration. One can continue to the next level of refinement depending on desired error in the centroid and maximum allowed level of refinement. The flow-chart for the static AMR-MOF interface reconstruction of a given geometry is presented in Fig. 9. We note that the static

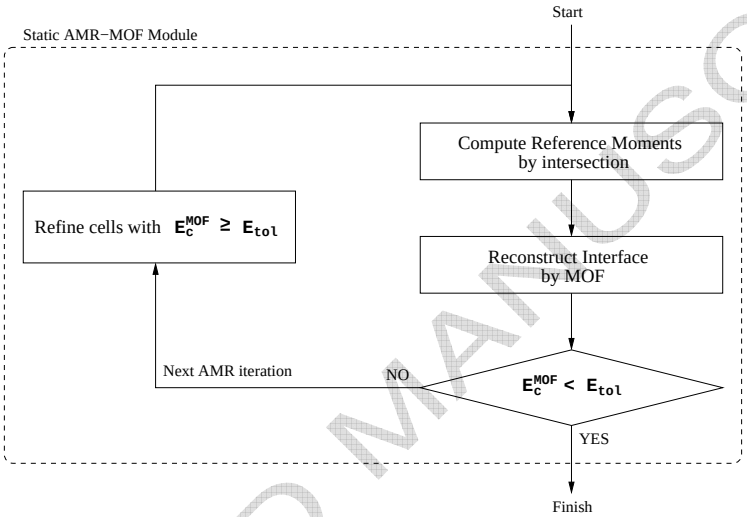


Fig. 9. Flow-chart for static AMR-MOF interface reconstruction for initial representation of material configuration on AMR mesh.

AMR-MOF interface reconstruction, described in Fig. 9 is only for the *initial* representation of given material configuration on AMR mesh.

3.2 Example of static interface reconstruction

In this Section we present static interface reconstruction for a multi-element airfoil configuration, as shown in Fig. 10. The AMR-MOF reconstruction starts with a single cell $[0, 1]^2$ - level-0 mesh.

The reference moment data is computed by exact intersection of mixed cells and original geometry, i.e. the body fitted unstructured mesh representing the airfoil, as shown in Fig. 13. The mesh is generated by using Gmsh [45] based on the boundary data as shown in Fig. 10. Adaptive refinement is performed up to level-8 from the level-0 mesh. First six levels of AMR-MOF interface

ACCEPTED MANUSCRIPT

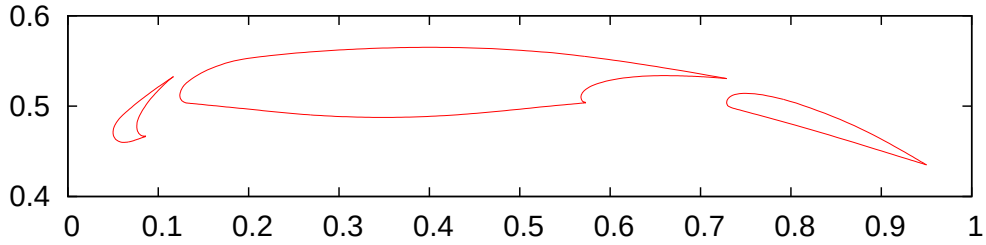


Fig. 10. Multi-element airfoil configuration.

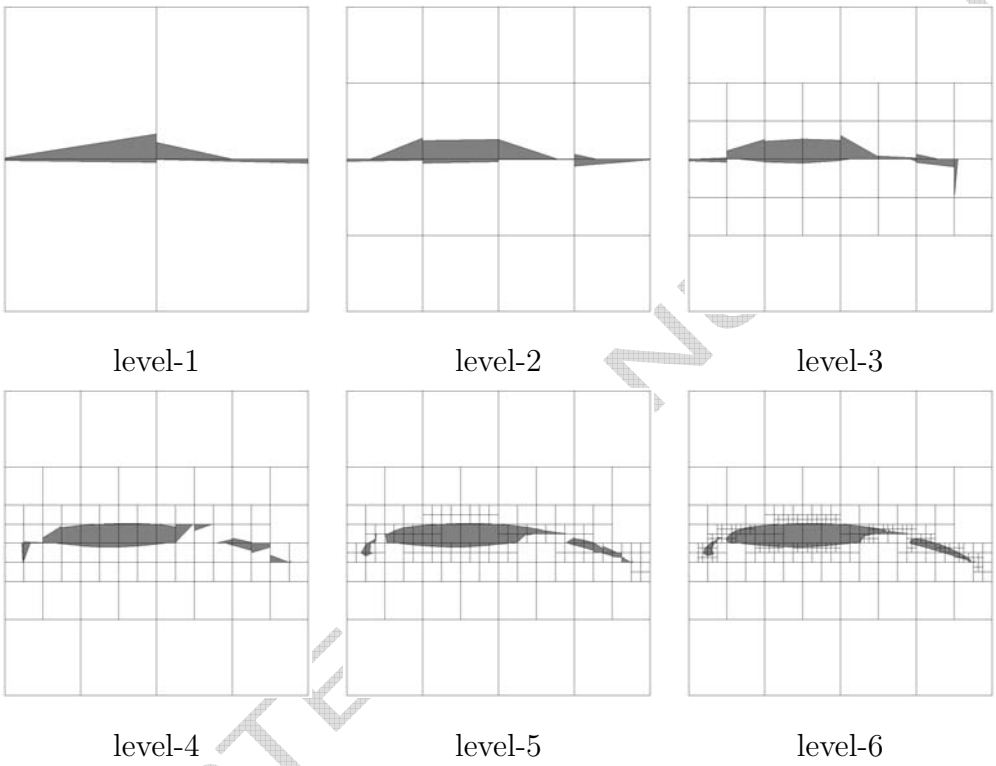


Fig. 11. AMR-MOF interface reconstruction of multi-element airfoil configuration starting with one cell, i.e. the level-0 mesh is 1×1 covering the domain of $[0, 1]^2$. Different levels of AMR-MOF reconstruction process are displayed. $E_{tol} = 1.e\text{-}15$ is used as the refinement criterion.

reconstruction is displayed in Fig. 11.

The initial geometry as shown in Fig. 13-(a) is triangulated so that the reconstruction error can also be easily computed in the sense of area of the symmetric difference between the true (original) and reconstructed configurations.

The symmetric difference of regions T and R is defined as follows:

$$T \triangle R = (T \cup R) - (T \cap R) \tag{5}$$

ACCEPTED MANUSCRIPT

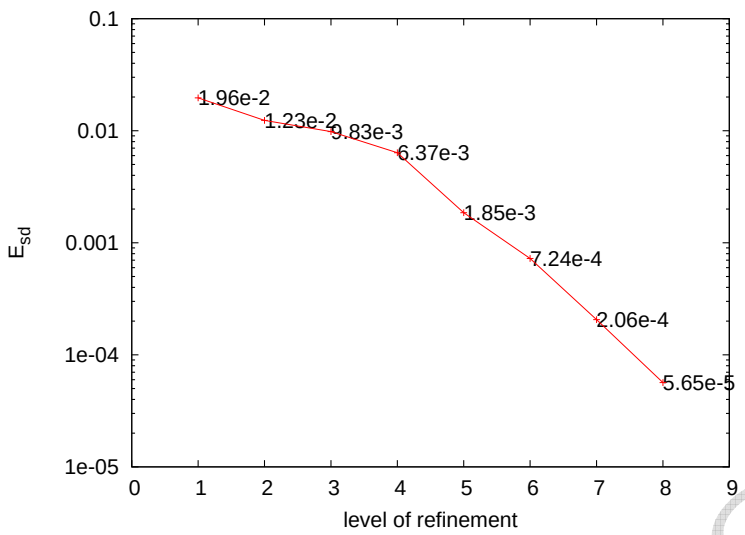


Fig. 12. Reduction of error, E_{sd}, computed by the area of symmetric difference by AMR-MOF interface reconstruction. $E_{tol} = 1.e\text{-}15$ is used as the refinement criterion.

where T represents the set of true material regions and R represents the set of actual (reconstructed) material regions on a given mesh.

The actual computation of the error, the area related to symmetric difference, is carried out cell-wise manner as follows:

$$E_{sd} = \sum_{c \in \mathcal{M}} |T_c \triangle R_c| = \sum_{c \in \mathcal{M}} |(T_c \cup R_c) - (T_c \cap R_c)| \tag{6}$$

where $\mathcal{M}$ is the set of cells; $T_c = T \cap c$ is the intersection of the material region with cell c, and R_c is reconstructed material within the cell-c. $|T_c \triangle R_c|$ represents the area of the region defined by $T_c \triangle R_c$.

The error E_{sd} is displayed in Fig. 12 as function the refinement level.

The close up view on the final reconstruction is compared with the original configuration in Fig. 13. Most of refinement structure is performed around high curvature region, especially leading and trailing edges of each airfoil. This result supports our claim that the error indicator, based on the departure of reference and actual centroids, detects high curvature region effectively.

ACCEPTED MANUSCRIPT

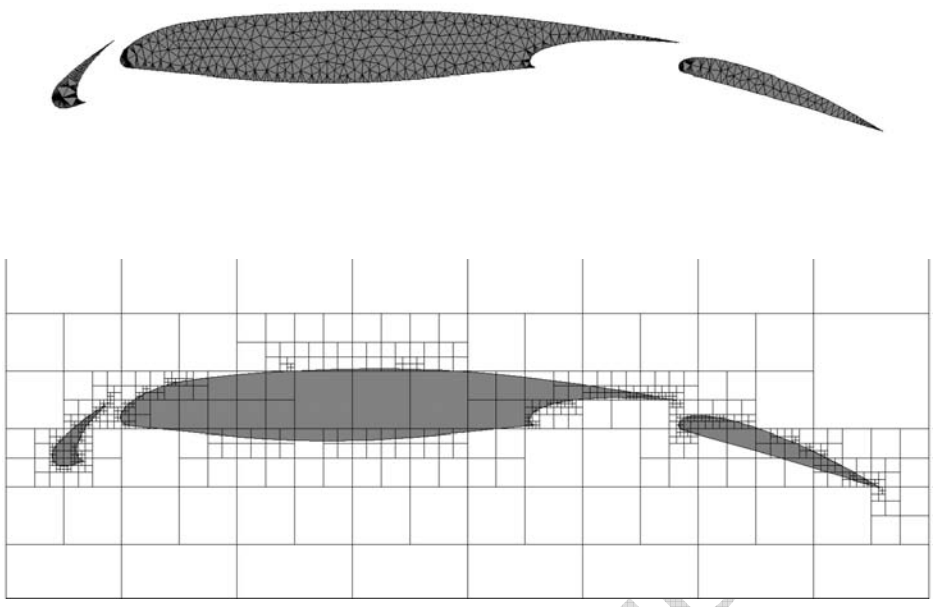


Fig. 13. AMR-MOF interface reconstruction of stationary object. Three-element airfoil geometry within 1×1 mesh (level-0) covering the square domain $[0, 1]^2$ is considered. Refinement is carried up to level-8, i.e. the maximum effective mesh resolution is 256×256. Top – original, bottom – AMR-MOF reconstruction. $E_{tol} = 1.e\text{-}15$ is used in the refinement criterion.

4 Dynamic interface reconstruction

4.1 Logic of dynamic interface reconstruction

The algorithm of the AMR-MOF for dynamically evolving interface is illustrated in Fig. 14. The essential difference of AMR-MOF algorithm for the dynamic interfaces compared to the static interface reconstruction is in the way the reference moment data is computed. In static AMR-MOF, the reference moment data is computed by exact intersection of the mesh with original geometric description, usually represented by bodyfitted unstructured mesh. In dynamic AMR-MOF, the reference moment data is computed by de-referencing the material configuration on the AMR mesh at the previous time step. The material configuration at the previous time step is represented by pure subcells obtained by AMR-MOF interface reconstruction of the previous time step. In dynamic AMR-MOF, the reference moment data for each refined cell is computed by moment advection between the material configurations at the previous time step and the current time step.

The MOF method can be applied to volume-tracked evolving interface prob-

ACCEPTED MANUSCRIPT

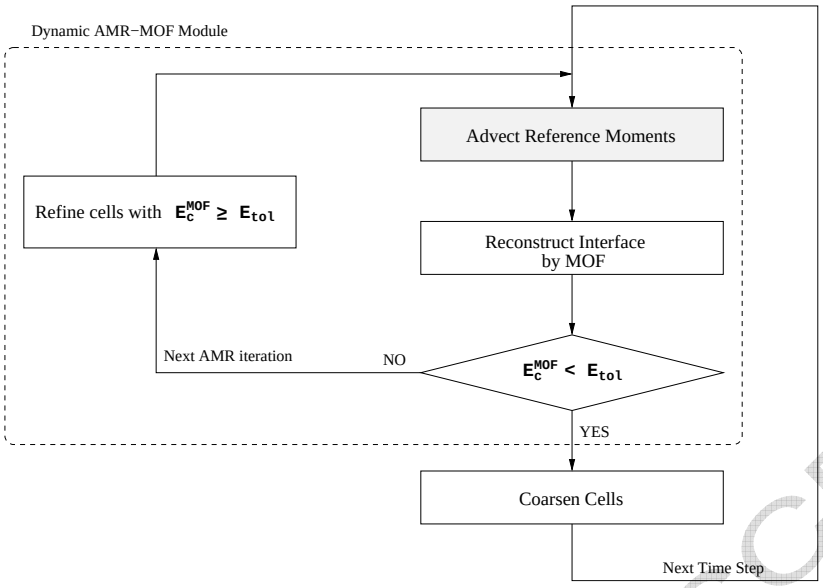


Fig. 14. Flow-chart for dynamic AMR-MOF interface reconstruction and moment advection. The difference of the dynamic AMR-MOF module from the static AMR-MOF module, as shown in Fig. 9, is reference moment computation step. For dynamic case, the reference moment is computed by advection step, as indicated with gray box.

lems once the moment advection scheme is augmented.

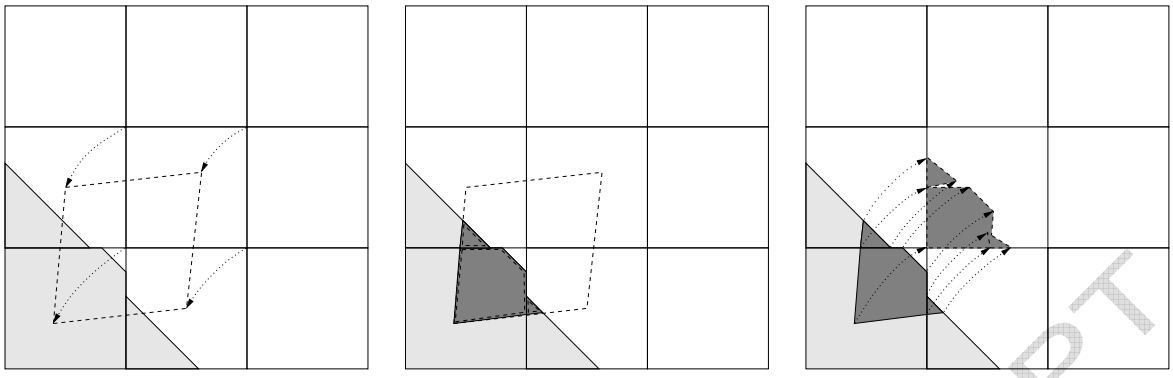
4.2 Principles of advecting moments

We explain principles of advection on the example of mesh without refinement. At the initial time we know the exact material configuration and we can use static interface reconstruction, described in the previous Section, to approximately represent the material configuration by a set of pure and mixed (multimaterial) cells of the mesh. Each mixed cell is subdivided in two pure subpolygons representing corresponding materials.

Now we can assume that at time $t = t^n$ we have a similar representation of material configuration and our goal is to represent material configuration at time $t = t^{n+1}$, which has changed due to prescribed velocity field. Let us denote a cell of the stationary Eulerian mesh by $\{c\}$. The known pure subpolygon representing a dark material in cell c at $t = t^n$ is denoted by Ω_c^n (if cell c completely filled with dark material then $\Omega_c^n = c$, and if cell c is empty then $\Omega_c^n = \emptyset$). These subpolygons for cells surrounding central cell c are presented in gray in Fig. 15 (a). Our goal is to construct Ω_c^{n+1}.

We will need to introduce several notions and notations. According to the book [46] a *material volume* "...is an arbitrary collection of fluid of fixed

ACCEPTED MANUSCRIPT



(a) Lagrangian backtracking (b) Polygon intersection-Remapping (c) Forward tracking

Fig. 15. Moment advection by Lagrange+remap strategy. The moment advection process for the central cell-c on 3×3 local stencil is illustrated.

identity and enclosed by surface also formed by fluid particles. All points of the material volume, including the points of its boundary, move with the local continuum velocity. A material volume moves with the flow and deforms in shape as the flow progresses, with stipulation that no mass ever fluxes in or out of the volume, viz., both the volume and its boundary are always composed of the same fluid particles."

To avoid expressions like "volume of the material volume", we will use term *material element* (ME) instead of *material volume*.

In the context of our paper ME corresponding to cell c can consist of two materials and each of these materials can be considered as a material element by itself.

Our goal is to represent the material configuration at $t = t^{n+1}$ on the Eulerian mesh. We know the geometry of the material element at $t = t^{n+1}$ (which is just cell c - central cell in Fig. 15), but we do not know what materials it will consist of. To find this, we need to know what material element at time $t = t^n$ (*departure* element) will *arrive* at cell c at $t = t^{n+1}$. We will denote geometry of departure element by $\overleftarrow{c}$. The cell $\overleftarrow{c}$ can be approximately defined by tracking back in time[1] the vertices of cell c (in the following Sections this process will be referred as *Lagrangian backtracking*), and connecting these vertices by segments of straight lines. The boundary of the departure element $\overleftarrow{c}$ is shown in Fig. 15 (a) in dashed lines. Clearly this procedure is an approximate procedure because there is an error in time integration of positions of vertices as well as an error related to connecting vertices by straight lines. Implications related to these errors are considered in the later Sections of the paper. Here for simplicity we also assume that $\overleftarrow{c} \subset \bigcup_{c' \in C(c)} c'$, where $C(c)$ is the union of the immediate neighbors of cell c and cell c itself ($C(c)$ consists of nine cells

[1] using 4th order Runge-Kutta scheme

ACCEPTED MANUSCRIPT

presented in Fig. 15.

Now using the definition of the ME we can determine pieces of the material it consists of, by intersection of $\overleftarrow{c}$ with pure subpolygons representing material configuration at $t = t^n$:

$$\Omega^n_{\overleftarrow{c},c'} = \overleftarrow{c} \cap \Omega^n_{c'} \,. \tag{7}$$

In this paper we assume that $\overleftarrow{c}$ is convex. This assumption is not critical, but if we allow $\overleftarrow{c}$ to be nonconvex then we need to use more complicated algorithms for polygon-polygon intersection, and the result of this intersection can be several disjoint pieces.

Subpolygons $\Omega^n_{\overleftarrow{c},c'}$ are shown in Fig. 15 (b) in dark grey. In the following Sections this process will be referred as *remapping*.

According to the definition of a ME, the mass of the ME does not change with time. Here we are considering incompressible fluid and it means that volume (zeroth moment) of the ME does not change. Therefore, we can define a reference zeroth moment of the dark material for cell c at time $t = t^{n+1}$ as follows

$$M^{c,n+1}_{0,ref} = \sum_{c' \in C(c)} M_0 \left(\Omega^n_{\overleftarrow{c},c'} \right) \,. \tag{8}$$

To use MOF for interface reconstruction at $t = t^{n+1}$ we also need to define reference first moment. The first moment is not constant in time. Because of that we do the following. We trace forward in time (*forward tracking* [2] from t^n to t^{n+1} - Fig. 15 (c)) pieces of the materials $\Omega^n_{\overleftarrow{c},c'}$, defined by (7). We denote polygons resulting from this operation as follows $\overrightarrow{\Omega}^n_{\overleftarrow{c},c'}$. Schematically they are shown in Fig. 15 (c) in dark grey and are located in central cell c. Now we can define reference first moment as

$$M^{c,n+1}_{1,ref} = \sum_{c' \in C(c)} M_1 \left(\overrightarrow{\Omega}^n_{\overleftarrow{c},c'} \right) \,. \tag{9}$$

We want to note that polygons $\overrightarrow{\Omega}^n_{\overleftarrow{c},c'}$ can slightly overlap each other or there can be gaps between them. The reason for this is again error related to defining trajectories as well as result of connecting vertices by segments of straight lines. The errors in definition of first moment are small and are not corrected

[2] again using 4th order Runge-Kutta scheme

ACCEPTED MANUSCRIPT

because first moment is used in definition of reference centroid only and not suppose to be preserved exactly anyway.

In following Sections we often will refer to described moment advection method as *Lagrange+remap* strategy.

In the next Section we describe details of moment advection on an unrefined (uniform) mesh, and in Section 4.4 we extend it to the case of AMR mesh.

4.3 Implementation of moment advection on uniform mesh

First we find cells which will be pure (completely filled by dark material - $f_c = 1$ at $t = t^{n+1}$). The simplest situation, when we can guarantee that cell will be pure (we will call such cells (an *a priori* pure cells) is when cell c is pure cell at $t = t^n$ and all its nearest neighbors are also pure cells at $t = t^n$. For such a cell c we perform Lagrangian backtracking step to find $\overleftarrow{c}$. According to definition of the material element we are supposed to have that

$$\tilde{V}_c^{n+1} = \mathrm{Vol}(\overleftarrow{c}).$$

Here $\tilde{}$ indicates that this volume may be not the final volume which will be assigned to cell c at $t = t^{n+1}$. As we have mentioned before, the Lagrangian backtracking procedure is not exact and therefore $\tilde{V}_c^{n+1}$ maybe not equal to volume of c as it supposed to be because cell c is declared a pure cell and we are dealing with an incompressible velocity field.

If $\tilde{V}_c^{n+1} > \mathrm{Vol}(c)$ then this cell is declared *overfilled*; if $\tilde{V}_c^{n+1} < \mathrm{Vol}(c)$ then this cell is declared *underfilled*. Volume $\tilde{V}_c^{n+1}$ and discrepancy $\tilde{V}_c^{n+1} - \mathrm{Vol}(c)$ are stored and will be used in repair stage of the algorithm which will be described later.

Next we consider *potentially-mixed* cells, i.e. the cells may contain material interface at $t = t^{n+1}$. The cell c is a potentially-mixed cell if at least one of the following conditions is satisfied at $t = t^n$:

(1) cell-c is mixed
(2) some of the immediate neighbors of c are mixed,
(3) cell-c is a pure cell - $f_c = 1$, but at least one of its neighbors is an empty cell - $f_c = 0$.

The cells satisfying one of the above conditions are flagged as a potentially-mixed cells (PMCs). PMCs reside within a narrow bands around the interface. As we will see, not all PMCs will be *really* mixed cells; instead, some of them will be pure cells. Once a set of PMCs is identified, we perform Lagrangian

ACCEPTED MANUSCRIPT

backtracking for those cells. Next we perform remapping, that is finding of subpolygons $\Omega^n_{\overleftarrow{c},c'}$ by intersections of $\overleftarrow{c}$ with $\Omega^n_{c'}$, $c' \in C(c)$. The volume $\tilde{V}^{n+1}_c$ for PMC is defined as

$$\tilde{V}^{n+1}_c = \sum_{c' \in C(c)} \mathrm{Vol}\left(\Omega^n_{\overleftarrow{c},c'}\right) .$$

Now if for all cells c' we have $\overleftarrow{c} \cap c' \subset \Omega^n_{c'}$ (if cell c' is empty cell then we set $\Omega^n_{c'} = \emptyset$) then cell c is flagged as pure cell and it can be declared overfilled or underfilled as it was described before for *a priori* pure cells.

PMC is also flagged as a pure cell when the following condition is satisfied

$$\tilde{V}^{n+1}_c > \mathrm{Vol}(c) .$$

In this case this pure cell is declared to be overfilled.

In all other cases PMCs are declared *mixed cells*. Mixed cells *cannot* be overfilled or underfilled. The *preliminary* zeroth moment (volume) and *final* first moment for dark material in these cells are defined as described in previous Section. The zeroth moment is preliminary because it may be corrected at the repair stage to accommodate the discrepancy in the volume for pure underfilled and overfilled cells.

To obtain the final reference volume we use a new variant of the repair process [47–49]. The repair is a conservative redistribution of a conservative quantity with the goal of preserving local physical bounds of this quantity. In the context of this paper the conservative quantity is the volume of the dark (and light) material. The amount of total volume, V_{total} of the dark material at time $t = t^n$ is given by sum of the volumes of the Ω^n_c, and we want to preserve this total volume at $t = t^{n+1}$:

$$V_{total} = \sum_c \mathrm{Vol}(\Omega^n_c) = \sum_c \mathrm{Vol}(\Omega^{n+1}_c) .$$

Our assumption is that mesh $\{\overleftarrow{c}\}$ covers the computational domain without gaps and overlaps, and therefore

$$\sum_c \tilde{V}^{n+1}_c = \sum_c \sum_{c' \in C(c)} \mathrm{Vol}\left(\Omega^n_{\overleftarrow{c},c'}\right) = V_{total} . \tag{10}$$

Therefore preliminary volumes $\tilde{V}^{n+1}_c$ sum to correct total volume. However, $\tilde{V}^{n+1}_c$ for pure cells maybe not correct because pure cells maybe overfilled

ACCEPTED MANUSCRIPT

or underfilled and have to be repaired. Using terminology of repair process we need to establish upper, $U_b(V_c^{n+1})$, and lower $L_b(V_c^{n+1})$, bound for V_c^{n+1}. Clearly for *pure* cells we have

$$U_b(V_c^{n+1}) = L_b(V_c^{n+1}) = \mathrm{Vol}(c)\,.$$

For *mixed* cells we only require that

$$L_b(V_c^{n+1}) = 0\,, \quad U_b(V_c^{n+1}) = \mathrm{Vol}(c)\,.$$

The goal of repair is conservatively redistribute volume such that resulting volume for each cell lies in its bounds.

Redistribution is performed among neighbors of the repaired cell.

We will refer to this repair process as *local* repair as opposed to *global* repair which will be needed for AMR meshes. Details of the repair step can be found in [50].

After the repair we get final reference volumes (zeroth moments) which will be used in MOF interface reconstruction. The reference centroid is obtained from reference zeroth moment and first moment obtained using formula (9).

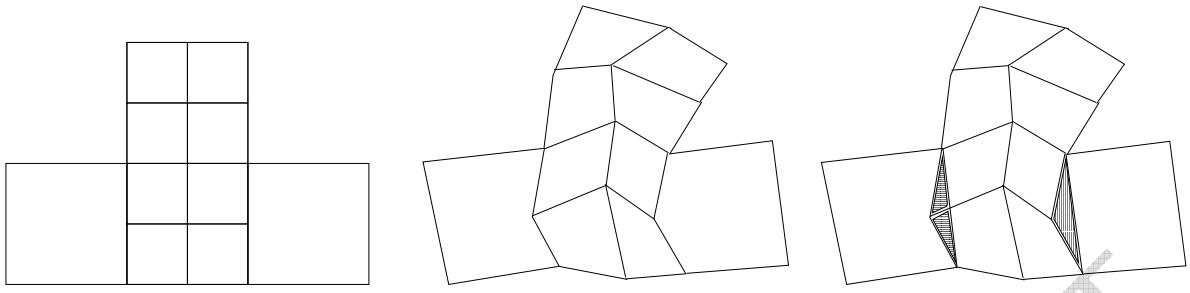
4.4 Moment advection on AMR mesh

In case of an AMR mesh we need more definitions. We will denote AMR mesh at $t = t^n$ by $\{c^n\}$, where c^n refers to generic cell from AMR mesh. For the purpose of this section, it is not important to distinguish which level of refinement it represents. The next mesh is an AMR mesh at $t = t^{n+1}$ for which we have to define moment data. In case of advection on uniform mesh $\{c^{n+1}\} = \{c^n\} = \{c\}$. Finally we have mesh $\{\overleftarrow{c}^{n+1}\}$ which is obtained by Lagrangian backtracking of mesh $\{c^{n+1}\}$.

There are several ways to backtrack AMR mesh. Let us consider a fragment of an AMR mesh presented in Fig. 16 (a). The central cell and top cell are refined once (level 1 refinement). The left and the right cell are not refined and have one "hanging" node at their boundary which comes from the refined central cell. From a formal point of view, the left and the right cells can be considered as pentagons with the hanging node being a "degenerate" vertex. There are no hanging nodes on the boundary of the central and top cells because they have the same level of refinement.

The correct way of backtracking of an AMR mesh is to move all vertices (in-

ACCEPTED MANUSCRIPT



(a) Fragment of the $\{c^{n+1}\}$ AMR mesh (b) Correct backtracking (c) Simplified backtracking

Fig. 16. Lagrangian backtracking of the AMR mesh. Overlap area for simplified backtracking is shaded by horizontal lines, and gap is shaded by vertical lines.

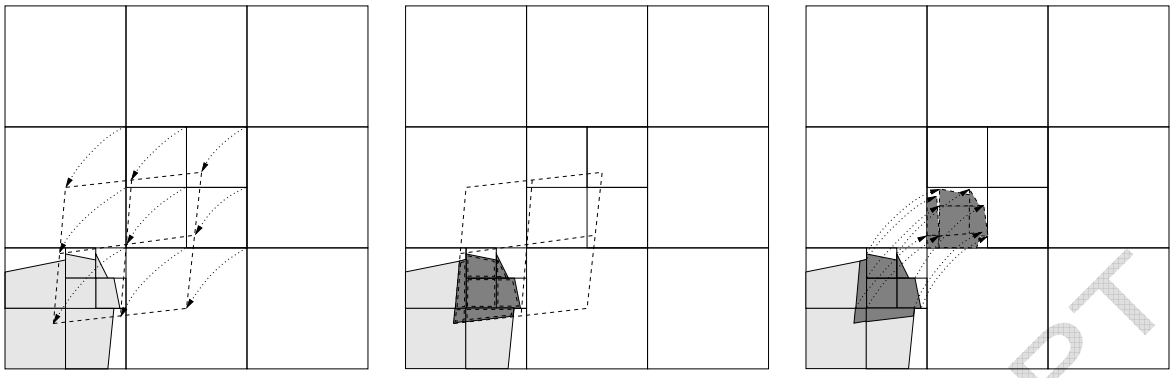
cluding degenerate vertices) of all cells. The result of correct backtracking is shown in Fig.16 (b). In this case backtracked mesh $\{\overleftarrow{c}^{n+1}\}$ covers computational domain without gaps and overlaps similar to case of advection on uniform mesh. For correct backtracking advection of moments (including repair) is essentially the same as for uniform mesh. The complications come from the fact that cells of mesh $\{\overleftarrow{c}^{n+1}\}$ maybe nonconvex, that is, one needs to use more complicated algorithm for intersection of polygons; also because both meshes $\{\overleftarrow{c}^{n+1}\}$ and $\{c^n\}$ are AMR meshes,the logic of what cells has to be intersected is more complicated.

There is another simplified way of backtracking AMR mesh when each cell is backtracked independently, that is, each cell is considered to be square, for example, hanging nodes are ignored when backtracking left and right cell in Fig.16 (a). Result of such simplified backtracking is shown in Fig.16 (c). It may lead to overlaps and gaps in mesh $\{\overleftarrow{c}^{n+1}\}$, which is now just collection of convex quadrilaterals.

In this paper we have chosen to use simplified backtracking and numerically demonstrate that it still gives very good results.

In Fig. 17 we have shown main stages of moment advection remap on AMR meshes. These stages are conceptually the same as for advection on a uniform mesh. In Fig. 17 the central cell represents the AMR mesh $\{c^{n+1}\}$ and neighboring cells represent AMR mesh $\{c^n\}$ with superimposed subpolygons representing material configuration at $t = t^n$. The main difference of the moment advection on AMR the mesh in comparison to the advection on uniform mesh is that, at the previous time step (on backtracked configuration) each mixed cell on level-0 can have multiple layers of quadtree structure of child cells, and each mixed child cell can have it own pure subcell configuration. Hence the polygon/polygon intersection have to be carried out for all child cells contained in the quadtree hierarchical structure.

ACCEPTED MANUSCRIPT



(a) Lagrangian backtracking (b) Polygon intersection-Remapping (c) Forward tracking

Fig. 17. Moment advection on AMR mesh. Moment advection for child cells originated from the central parent cell is illustrated.

Logic of defining what cells of AMR mesh $\{c^{n+1}\}$ are pure is exactly the same as for advection on uniform mesh.

The main difference is how to perform repair. The main problem here comes from the fact that mesh $\{\overleftarrow{c}^{n+1}\}$, has gaps and overlaps and therefore

$$\sum_c \tilde{V}_c^{n+1} = \sum_c \sum_{c'} \mathrm{Vol}\left(\Omega^n_{\overleftarrow{c},c'}\right) \neq V_{total}\,. \tag{11}$$

This fact makes the use of *local* repair very difficult, and we use another version of repair which is called *global* repair, [47]. In this case one compute the discrepancy in the total volume and distributes it among all cells proportional to the amount they can accept. Again details are presented in [50].

4.5 Time-stepping

The dynamic test cases presented in this paper use a one-step method time integration, i.e. the material configuration only at the present time step t^n is used for computing the material configuration at the next time step t^{n+1}. In the one-step method, the AMR-MOF routine keeps two different meshes. One represents the material configuration at time t_n, and the other for reconstructing the material configuration at time t^{n+1}. The AMR mesh representing the material configuration at time t^n is used as the reference for the moment advection for reconstructing material configuration on new AMR mesh at time t^{n+1}.

Construction of AMR mesh at $t = t^{n+1}$ starts with uniform mesh (that is, level-0 refinement). Clearly, more sophisticated strategy for de-refinement can

ACCEPTED MANUSCRIPT

be developed to save the computational time in the de-refinement step, e.g. selective de-refinement for re-cycling the AMR structure at the previous time.

For coupled simulation with flow solvers, this one-step approach is enough for one-step time integrators, such as a second-order accurate trapezoidal scheme. If a flow solver employs multi-step time integrators, such as the second-order accurate backward difference formula (BDF2), the material configurations at the previous time steps $t^{n-1}, t^{n-2}, \cdots$ can be easily incorporated with minor memory overhead.

In an AMR-MOF computation, the mesh is being refined locally. Hence, the time steps Δt based on local cell size differs over the computational domain. For adaptively refined meshes, two different strategies for time integration can be used. First, successively refined time steps Δt^{AMR}, like the uniformly refined meshes, can be used depending on the maximum level of refinement allowed. The second option is using the fixed time step Δt^0 as in the level-0 mesh regardless of the maximum level of refinement allowed. Of course, any combination of the above two strategy can be used. If the time scale to resolve the flow feature also has to be refined as the mesh refines, then the first strategy Δt^{AMR} can be used. If not and the moment advection scheme is robust and accurate to handle the time step Δt^0 corresponding to level-0 mesh, then the second strategy would be preferred. In our results, we prefer Δt^0 to Δt^{AMR} but both of the time steps are tested with AMR-MOF method and the results are compared with uniform refinement cases in the next section.

4.6 Test problems - Numerical results

We start this section with investigation of how the quality of initial interface reconstruction affects accuracy for dynamic problems, Section 4.6.1. Next in Section 4.6.2 we consider a classic example of advection, namely, Zalesak's notched disk. On this example we show how accuracy depends on the allowed level of refinement. To compare results obtained by our new method with published results, we consider the reversible vortex problem [22] with short period $T = 2$ - Section 4.6.3. The reversible vortex problem with long period $T = 8$ is considered in Section 4.6.4. Additionally, results for new examples, namely droplet deformation case and $\mathcal{S}$-shape flow case are presented in Sections 4.6.5 and 4.6.6.

4.6.1 Effect of quality of the initial interface representation

The accurate representation of material configuration at the initial stage is extremely important for long term evolving interfaces because the entire dynamic simulation depends on the quality of initial configuration. The significance of

ACCEPTED MANUSCRIPT

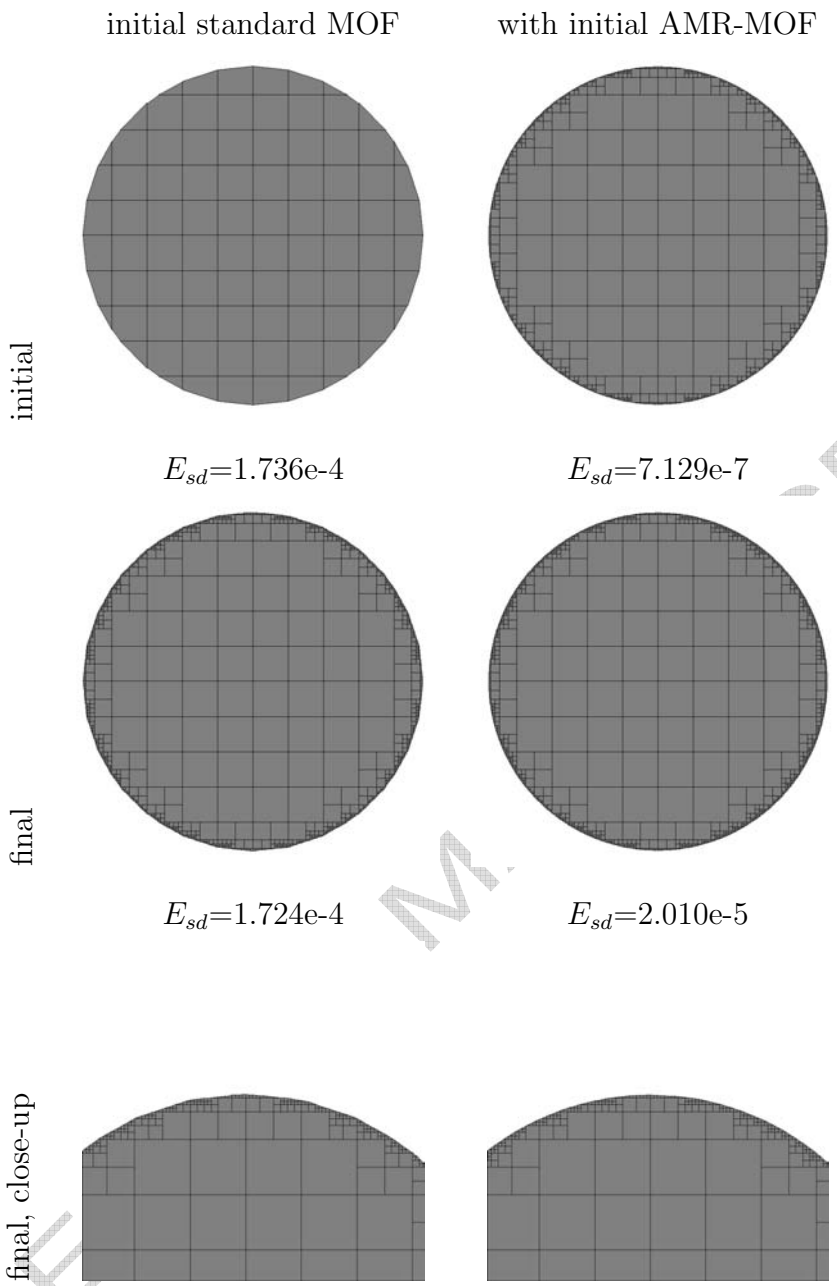


Fig. 18. Initial interface reconstruction effect. Reversible vortex case with long period ($T = 8$) as described in Section 4.6.4. Left–no AMR-MOF interface reconstruction at the initial stage, right – AMR-MOF interface reconstruction at the initial stage. Both cases are computed with same AMR-MOF from the first advection step to the final stage. $E_{tol} = 1.e\text{-}20$ is used as the refinement criterion.

initial AMR-MOF interface reconstruction is demonstrated in Fig. 18. The initial material configuration is a circle, a relatively *simple* geometry which can be easily represented by any interface reconstruction method. The given circle is deformed under the nonlinear divergence-free reversible vortical velocity

ACCEPTED MANUSCRIPT

field (see, for example, [22]):

$$\mathbf{v}\left(x,y,t\right)=\begin{bmatrix}\sin^2(\pi x)\sin(2\pi y)\\ -\sin^2(\pi y)\sin(2\pi x)\end{bmatrix}\cos\left(\frac{\pi t}{T}\right) \tag{12}$$

where T is the period of reversing vortical flow. $T = 8$ is used for this case.

The effect of initial interface quality is estimated by two different initial material reconstructions: one with initial standard MOF and the other with initial AMR-MOF interface reconstruction. For both cases, the same AMR-MOF advection and interface reconstruction is performed from the first time step to the last. At the final reversed time moment, the two results are compared and the error E_{sd} is computed with respect to the exact. It is clear that the initial AMR-MOF interface reconstruction results in much more accurate final material configuration for long term evolution of volume-tracking computation.

We note that the final material configuration, for the case started with initial standard MOF, is as accurate as the initial material configuration and even slightly more accurate. We believe that this is mostly because the AMR-MOF moment advection scheme is very accurate and the error generated during the advection is several orders of magnitude smaller than the error of initial standard interface reconstruction and also partly because of the possible error cancellation during successive time steps (total number of time steps, $n_t = 256$ for the period of $T = 8$). The effect of the initial interface reconstruction quality is further clarified with Fig. 19. The error E_{sd} at the final stage is measured with different levels of AMR-MOF computation while the initial interface is reconstructed only by standard MOF on level-0 mesh. The Fig. 19 clearly shows that the accuracy at final stage is bounded by the accuracy of the initial interface reconstruction. This confirms that regardless of the level of refinement, the accuracy of the final material configuration is bounded by the accuracy of the initial material configuration.

We like to emphasize that the initial configuration is relatively simple. If the initial material configuration is described with more complex geometries (e.g. sharp corners or filaments with subcell size thickness), the quality of initial interface reconstruction must be even more crucial for the accuracy of entire dynamic simulation. For the rest of test cases in this paper, we employ AMR-MOF starting with initial representation of the material configuration.

4.6.2 Zalesak's notched circle

As the first case of dynamic test, we consider the rigid body rotation of Zalesak's notched circle. The initial configuration of the Zalesak's case is described

ACCEPTED MANUSCRIPT

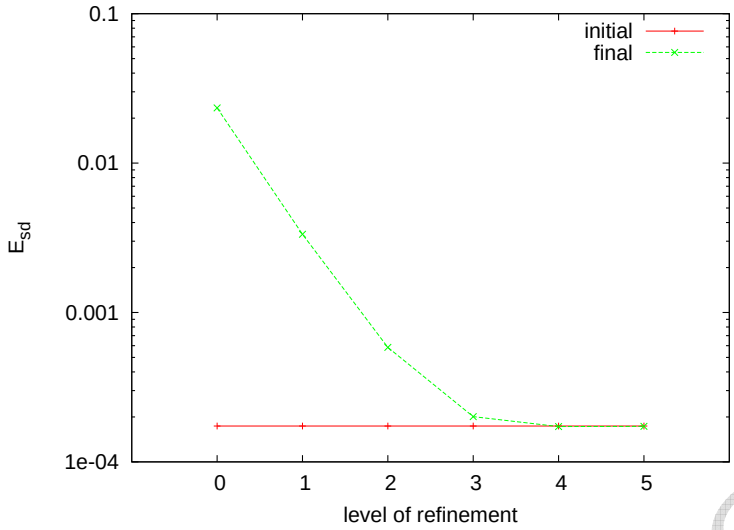


Fig. 19. Reduction of error E_{sd} by AMR-MOF computation while the initial interface is reconstructed by the standard MOF on level-0 mesh. The initial interface reconstruction error is fixed because no mesh refinement is allowed at the initial stage, and the error at the final stage is decreasing for first a few levels of refinement but eventually bounded by the accuracy of the initial interface reconstruction.

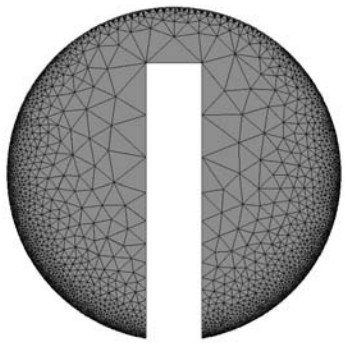


Fig. 20. Initial configuration of Zalesak's notched disk. Using the triangulation of the initial configuration, the reference moment data and E_{sd} by area of the symmetric difference can be easily computed.

in [51], and its body fitted unstructured triangulation is displayed in Fig. 20. The circular perimeter is defined by the circle centered at $(0.5, 0.75)$ with radius $r = 0.15$. The vertical notch is produced with the width of $w = 0.05$ and the maximum vertical thickness of the upper connection is also 0.05, i.e. the maximum height of the notch is $h = 0.85$. The initial geometry is triangulated so that the advection and reconstruction error can also be easily computed in the sense of symmetric difference between the true (original) and reconstructed configurations.

ACCEPTED MANUSCRIPT

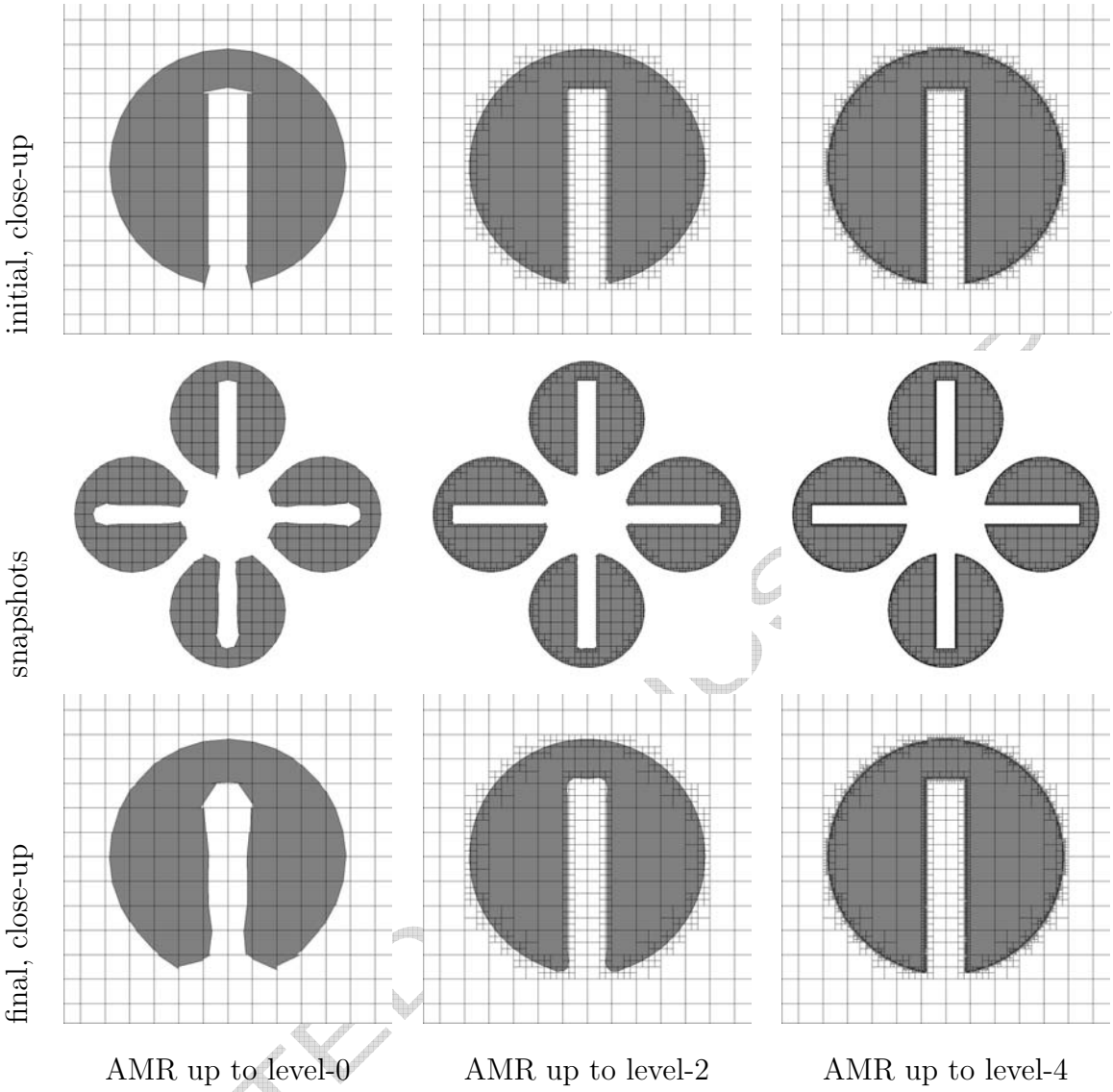


Fig. 21. Rotation test of Zalesak's notched disk. Top row shows close-up view of the initial reconstruction before the rotation. Middle row shows the snap shots at $\frac{0}{4}$, $\frac{1}{4}$, $\frac{2}{4}$, and $\frac{3}{4}$ of a full rotation along the counter-clock-wise direction starting from the top. Bottom row shows close-up view of the final configuration after one full rotation. Different levels of refinement is allowed. From the left, refinement is performed up to level-0 (32×32), level-2, and level-4. Value of $E_{tol} = 1.e$-20 is used.

The actual AMR-MOF interface reconstruction and advection results are displayed in Fig. 21. The top row shows the initial AMR-MOF interface reconstruction with various level of refinement, and the bottom row shows the result of AMR-MOF advection and interface reconstruction method after one full rotation. Total number of time step $n_t = 128$ is taken for all cases. 32×32 mesh is used as the level-0. The centroid error tolerance of $e_{tol} = 1.$e-20 is used for all cases, i.e. mixed cells with $e > e_{tol}$ are refined up to the maximum allowed refinement level. The error measured by the area of symmetric difference is

ACCEPTED MANUSCRIPT

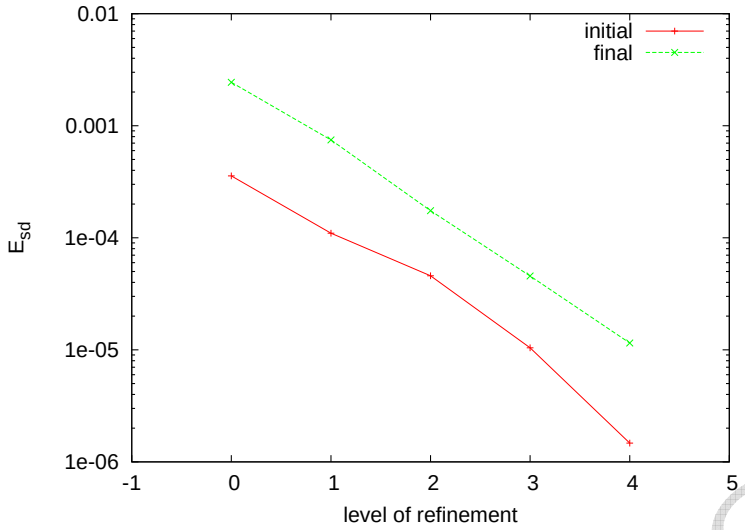


Fig. 22. Error of Zalesak's notched disk. The error in computed by the area of symmetric difference between the original material configuration as shown in Fig. 20 and the material configuration by AMR-MOF interface reconstruction and moment advection as shown in Fig. 21. The order of accuracy is same for both before (initial) and after (final) the rotation.

refinement level	0	1	2	3	4
	E_{sd}	E_{sd}	E_{sd}	E_{sd}	E_{sd}
initial	3.571e-04	1.099e-04	4.578e-05	1.042e-05	1.472e-06
final	2.438e-03	7.475e-04	1.744e-04	4.561e-05	1.150e-05

Table 1
Error by symmetric difference between the original geometry (shown in Fig. 21) and AMR-MOF computation (reconstruction and advection as shown in Fig. 22). Level-0 mesh is 32×32 covering $[0, 1]^2$ computational domain.

listed in Table. 1.

Each level of refinement reduces error approximately by a factor of four, i.e. second order accuracy. The quadratic reduction of error E_{sd} with respect to the level of refinement is also displayed in Fig. 22. The slight over/under quadratic convergence of the initial reconstruction error is attributed to the sharp corner effect of the notch. The difference of the errors between the initial interface reconstruction and the final material configuration after one full rotation clearly indicates the error gained by the advection and reconstruction. This result indicates that the AMR-MOF advection and interface reconstruction preserves the second order accuracy of the initial AMR-MOF interface reconstruction.

ACCEPTED MANUSCRIPT

max. refinement level	initial ($time = 0.0$)	final ($time = 2.0$)
	E_{sd}	E_{sd} / $\Delta\mathcal{V}$
0	1.736e-04	5.908e-04 / 1.282e-13
1	4.061e-05	1.161e-04 / 8.154e-14
2	1.279e-05	2.329e-05 / 1.811e-14

Table 2
Reversible vortex with short period ($T = 2$): error computed by the area of symmetric difference between AMR-MOF computation and reference solution obtained by front tacking and mesh generation. Total volume gain/loss is also indicated by $\Delta\mathcal{V} = \mathcal{V}^{final} - \mathcal{V}^{initial}$.

4.6.3 Reversible vortex: Short period ($T = 2$)

The reversible vortex case was described by Rider and Kothe [22]. The initial configuration of the material is defined by a circle with radius $r_0 = 0.15$ centered at $(0.50, 0.75)$ within the square domain of $[0, 1]^2$. The circular region deforms under the nonlinear unsteady velocity field defined by the following stream function,

$$\Psi(x, y, t) = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y) \cos\left(\frac{\pi t}{T}\right), \tag{13}$$

which results in the nonlinear divergence-free vortical velocity field as described in Eq. (12).

The AMR-MOF advection and interface reconstruction results are displayed in Fig. 23. The top row shows the initial interface reconstruction, the middle row shows the material configuration at maximum stretch ($time = 1.0$), and the bottom row shows the final reversed material configuration ($time = 2.0$). Total number of time steps $n_t = 64$ (i.e. $\Delta t = \frac{1}{32}$) is used for all cases. A 32^2 mesh is used as the level-0, and adaptive mesh refinement is allowed up to level-2, i.e. maximum effective mesh resolution is 128×128. The error E_{sd} computed by the area of symmetric difference is summarized in Table 2. The total volume error $\Delta\mathcal{V}$ with respect to the initial stage is also listed at the final stage.

A different measure of the error E_{vf}, the error based on cell-wise volume fraction difference, is computed in the following way for comparison as summarized in Table 3,

$$E_{vf} = \sum_{c \in \mathcal{M}} \mathcal{V}_c |f_c^{ref} - f_c^{act}| \tag{14}$$

ACCEPTED MANUSCRIPT

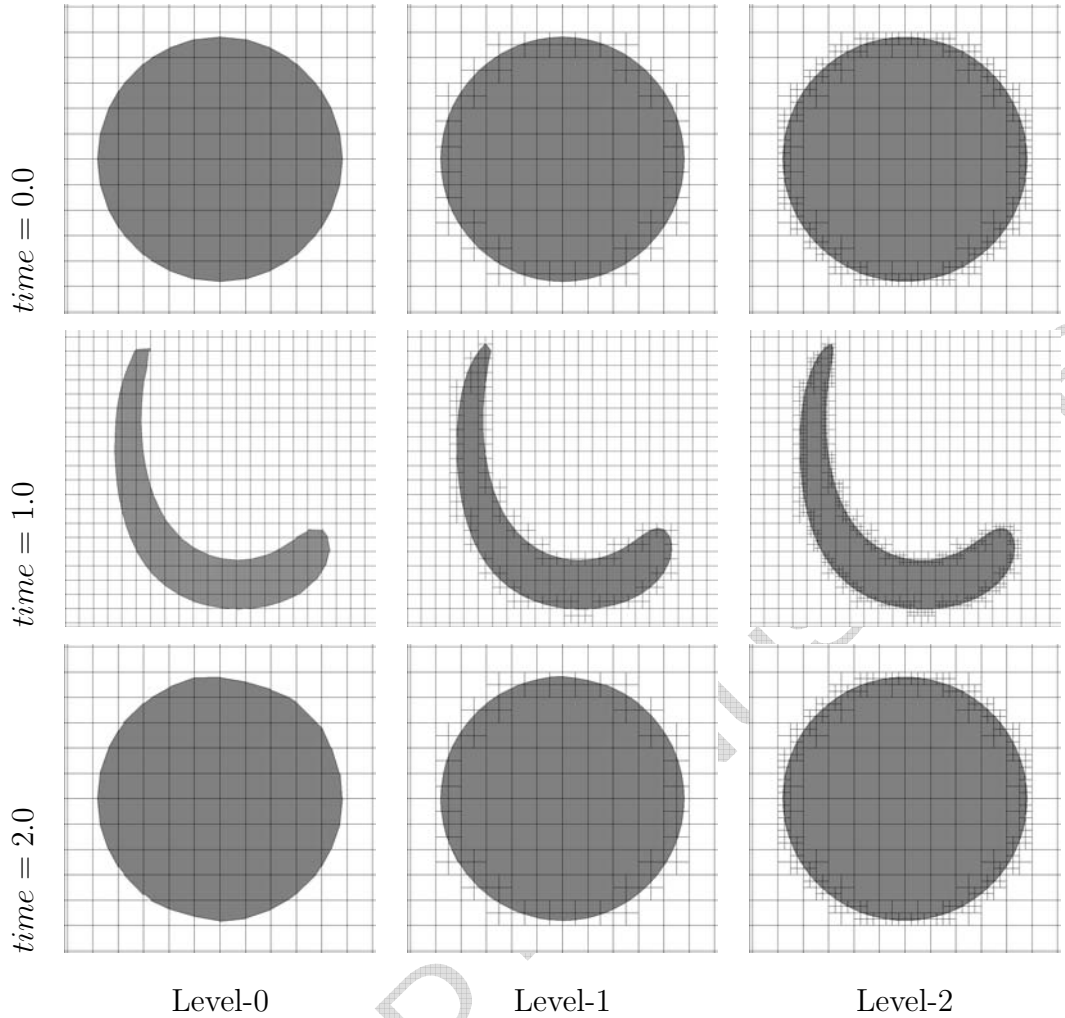


Fig. 23. Single vortex flow (period, $T = 2$). The top row shows the material configuration at the initial stage ($time = 0.0$), middle row shows the material configuration at maximum stretch ($time = 1.0$), and the bottom row shows at the reversed configuration ($time = 2.0$). 32×32 mesh is used as the level-0, and result with different maximum level of refinement is displayed. $E_{tol} = 1.e\text{-}20$ is used as the refinement criterion

where $\mathcal{M}$ is the set of all cells and $\mathcal{V}_c$ is the volume of cell-c (area in 2D), and f_c^{ref} is the reference (initial) volume fraction and f_c^{act} is the actual (final) volume fraction at the reversed stage. We note that the above error definition is different from our previous error E_{sd} based on the symmetric difference as expressed in Eq. (6) in two reasons. First, E_{vf} computed by Eq. (14) measures a relative error with respect to the initial material configuration not to the exact material configuration. This error is a measure of error that occurred during the advection process between the initial and the final reversed stage. Second, the error E_{vf} is blind to the subcell interface configuration in the cell. The intra-cell interface configuration cannot be captured by this error. For the above reasons, we prefer the error E_{sd} by symmetric difference to E_{vf} by

ACCEPTED MANUSCRIPT

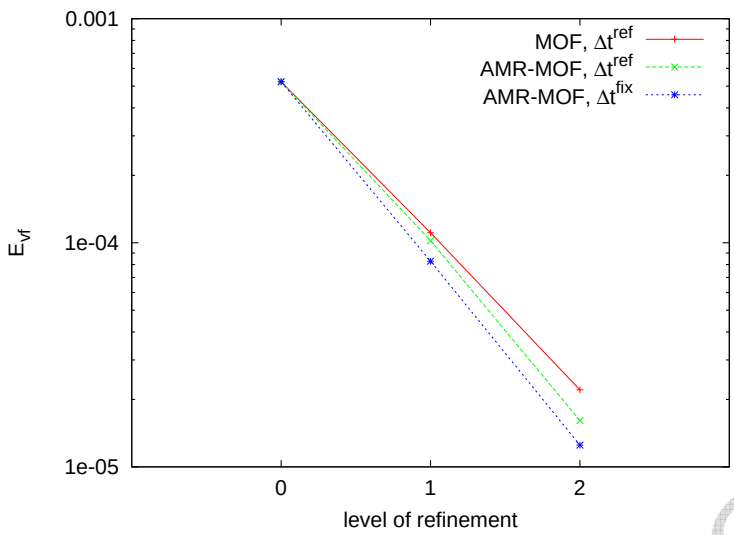


Fig. 24. Mesh convergence study: MOF vs. AMR-MOF. For adaptive meshes, mesh resolution is taken by the maximum level of refinement. The error E_{vf} is computed with Eq. (14). Δt^{ref} refers to the time step that is also refined as the mesh refines. Δt^{fix} refers to the fixed time of level-0 mesh that is used regardless of mesh refinement.

volume fraction. For comparison purpose, however, we also provide the error E_{vf}.

For comparison with previous works [22,52,53], the reversible vortex case is presented first with a relatively short period of time, $T = 2$. The mesh convergence study is performed for AMR-MOF as well as uniform mesh MOF. For 32^2 level-0 mesh, time step of $\Delta t = \frac{1}{32}$ is used. For uniformly refined meshes, time step is also successively refined by the factor of two, i.e. uniform 64^2 mesh case uses time step $\Delta t = \frac{1}{64}$.

The convergence of the error E_{vf}, obtained from the present MOF and AMR-MOF computation, are displayed in Fig. 24. For AMR-MOF, the error E_{vf} is computed on level-0 mesh by summation of the signed error, because of the non-uniform distribution of mesh resolution, and we note that this may result in minor error cancellation effect for AMR-MOF cases. Our present results (both the error and CPU time) are summarized in Table 3 and also compared with other results published in literature. This table shows that even the results from the standard MOF using uniform meshes show much less error compared to other published results. For AMR-MOF, the error shows even less with successively refined time steps. We believe this may attribute to the error cancellation effect on level-0 mesh where the error E_{vf} is computed for AMR-MOF cases. For AMR-MOF with fixed time steps, the error is even less. We speculate that this is due to less frequent interface reconstruction (hence, less error from inexact interface representation) compared to the refined time step cases, which needs more interface reconstruction.

ACCEPTED MANUSCRIPT

Resolution	(a)	(b)	(c)	(d)	(e)	(f)
	E_{vf}	E_{vf}	E_{vf}	E_{vf} (CPU time)	E_{vf} (CPU time)	E_{vf} (CPU time)
32^2 (level-0)	2.36e-3	2.37e-3	1.09e-3	5.22e-4 (7.22)	5.22e-4 (7.22)	5.22e-4 (7.22)
64^2 (level-1)	5.85e-4	5.65e-4	2.80e-4	1.10e-4 (34.62)	1.02e-4 (22.52)	8.27e-5 (13.96)
128^2 (level-2)	1.31e-4	1.32e-4	5.72e-5	2.20e-5 (183.72)	1.61e-5 (107.36)	1.25e-5 (27.10)

Table 3
Mesh convergence study and comparison with other published results. First three columns (a–c) are from others, and last three columns (d–f) are from the present results (MOF and AMR-MOF). The error E_{vf} computed by Eq. (14) is presented at the first row for each mesh resolution, and the CPU time in [sec] is presented within the parenthesis at the second row of each corresponding mesh resolution. For AMR-MOF, mesh resolution is taken by the finest level of mesh. The column (a) is taken from Rider and Kothe [22], the column (b) is from Harvie and Fletcher [52], the column (c) is from Scardovelli and Zaleski [53]. The column (d) is from standard MOF using uniform meshes, and column (e–f) is from AMR-MOF. The column (e) is AMR-MOF with the refined time step, i.e. time step is also refined for each mesh refinement. Finally, column (f) is AMR-MOF with fixed time step, i.e. the time step computed on level-0 mesh is used for all levels of AMR-MOF computation.

level of refinement	# of cells	% with respect to uniform mesh
0	1024	100 %
1	1237	30.2 %
2	1672	10.2 %

Table 4
Total number of cells required for AMR-MOF and its comparison to the uniform mesh case. For AMR-MOF, maximum number of cells are taken.

We note that the larger time step can also be used for uniform mesh cases with extra computational overhead. This will require extended stencil for the advection scheme. The advection with larger time step will also incur additional computational cost for extended search for polygon intersection algorithms.

The efficiency of the AMR-MOF is demonstrated with the comparison of computational cost, CPU time as summarized in Table 3 and also total number of cells involved as listed in Table 4.

Table 4 shows the total number of cells produced in the AMR-MOF computation. Since the number of cells are changing as the evolution of the material

ACCEPTED MANUSCRIPT

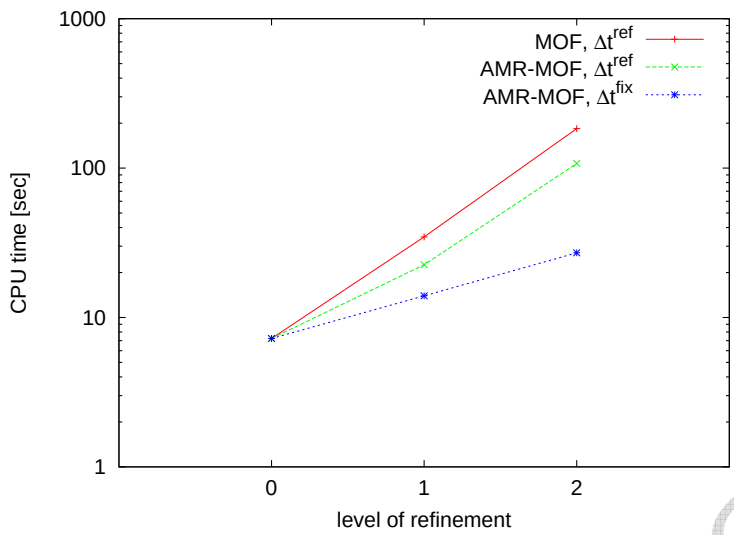


Fig. 25. Computational cost: AMR-MOF vs. standard MOF. Δt^{ref} refers to the time step is also refined as the mesh refines. Δt^{fix} refers to the fixed time step regardless of mesh refinement.

interface (the number of cells increases as vortex stretches, and *vice versa*), the maximum number of cells produced around the maximum stretch moment are counted for AMR-MOF cases. The total number of cells increases quadratically for uniform mesh cases, but only sub-linearly for the AMR-MOF case. This is because the mesh refinement is carried only on the mixed cells with high centroid errors. For example, in the level-2 mesh case, the maximum number of cells for the AMR-MOF is only about 10% of the uniform mesh case. This is a clear indication that AMR-MOF uses much less memory spaces.

Next, the actual CPU time is compared with uniform mesh MOF and AMR-MOF cases. The machine used for this benchmark test is an Opteron 2 GHz with 24 GB memory running 64 bit Fedora Core 3 as its operating system. For MOF with uniform mesh computations, the CPU time quadruples for every refinement (at each uniform refinement, roughly the number of mixed cell doubles and also the total number of time stepping doubles with halved Δt). For AMR-MOF with refined time steps, the CPU time also quadruples however much less CPU time is required compared to the uniform mesh refinement case. This is because AMR-MOF use much less number of cells compared to the uniform mesh case, as listed in Table 4. Finally the AMR-MOF with fixed time steps (not refined time step) shows only linear increases in CPU time for every increment of refinement level, and the amount of CPU time is less than 15% of it for 128^2 uniform mesh MOF. This is clear that AMR-MOF significantly improves both *accuracy* and also decreases the *computational cost* for volume-tracking evolving interface computation.

ACCEPTED MANUSCRIPT

Resolution	Rider/Kothe	Harvie/Fletcher	AMR-MOF
	E_{vf}	E_{vf}	E_{vf}
32^2 (level-0)	4.78e-2	3.72e-2	2.33e-2
64^2 (level-1)	6.96e-3	6.79e-3	3.15e-3
128^2 (level-2)	1.44e-3	1.18e-3	5.04e-4

Table 5
Comparison with results published in literature [22,52] for long period ($T = 8$) vortex case. For the present AMR-MOF, mesh resolution is taken by the finest level of the mesh. The error E_{vf} is computed with Eq. (14). The first column is taken from Rider and Kothe [22], the second column is from Harvie and Fletcher [52]. For AMR-MOF, time step computed on level-0 mesh, $\Delta t = \frac{1}{32}$, is used for all cases.

4.6.4 Reversible vortex, long period ($T = 8$)

The reversible vortex problem is presented with longer period, $T = 8$. Time steps of $\Delta t = \frac{1}{32}$ (total number of time steps, $n_t = 256$) are used for all AMR-MOF computations. First, for comparison purpose with uniform mesh results, published in literature [22,52], the maximum level of refinement is allowed up to level-2 (to be consistent with the mesh resolution of others [22,52]) starting with 32^2 level-0 mesh. The errors are computed using E_{vf} as defined in Eq. (14). The results are summarized in Table 5. Due to the non-uniform cell distribution of adaptively refined mesh, the E_{vf} is computed on level-0 mesh for AMR-MOF cases. This could result in minor error cancellation of adaptively refined mesh. However, we note that the AMR-MOF with maximum level-0 refinement (essentially a fixed mesh MOF on 32^2 mesh) already shows the most accurate result. Like for the short period ($T = 2$) vortex case, AMR-MOF gives the most accurate result in long period ($T = 8$) vortex example.

Next, a more aggressive refinement is allowed for this long period vortex case. The result of AMR-MOF computation, with maximum refinement up to level-4, is displayed in Fig. 26 at various time steps. Each snapshot taken symmetrically with respect to the maximum stretch for showing the *reversing* vortex.

The close up views are also shown in Fig. 27 at three critical time steps, namely initial ($time = 0$), at maximum stretch ($time = 4$), and final time steps ($time = 8$). The close up views at the initial and final configuration is focused around upper region of the circle, the material region corresponds to the tail of the vortex at the maximum stretch.

The difference between the initial and the final configuration cannot be detected visually even in close up views as displayed in Fig. 27. Hence, we computed the error by using a series of reference solution. The reference solution, as displayed in Fig. 28, is prepared using the front-tracking technique and mesh generation *in lieu of* the exact solution. First, the interface is tracked

ACCEPTED MANUSCRIPT

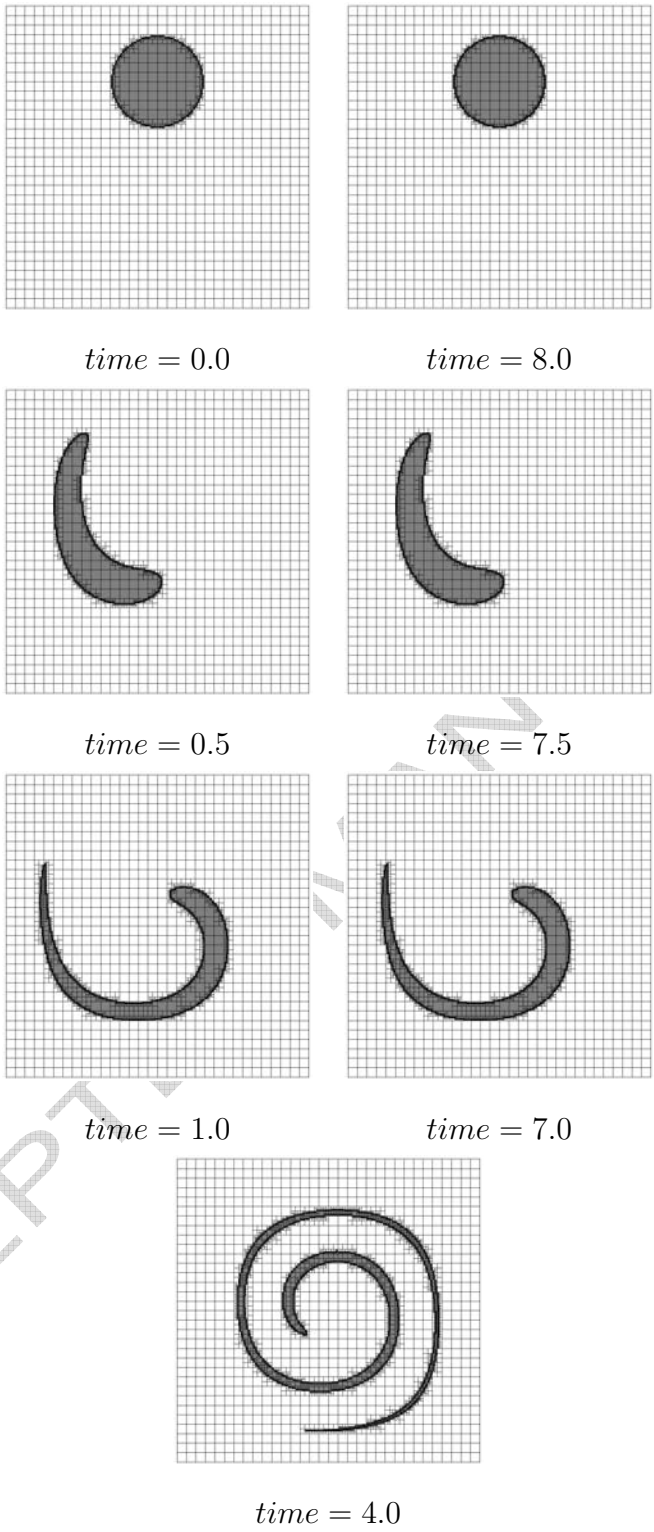


Fig. 26. AMR-MOF computation of reversible vortex problem, $T = 8$. Level-0 mesh is 32^2 covering the domain of $[0, 1]^2$. Adaptive refinement is performed up to level-4 (maximum effective mesh resolution is 512^2).

ACCEPTED MANUSCRIPT

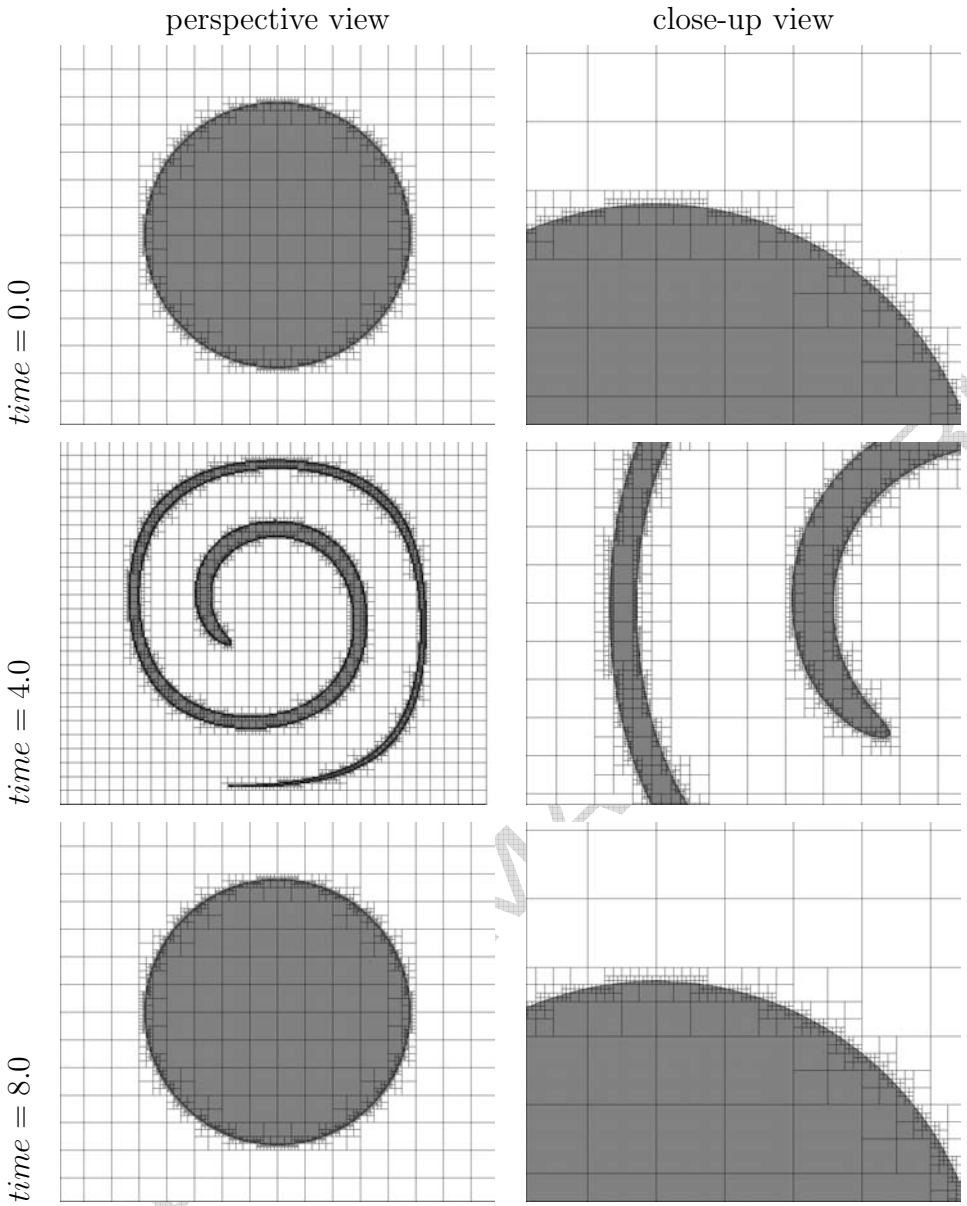
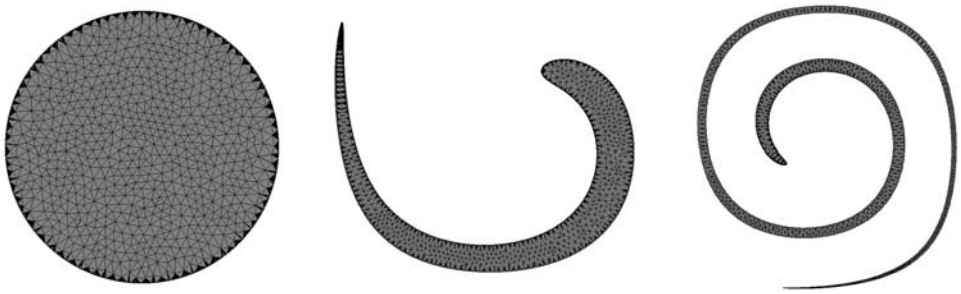


Fig. 27. Single vortex flow, $T = 8$. Level-0 mesh is 32^2 and maximum 4 level of AMR is allowed (maximum effective mesh resolution is 512^2). $E_{tol} = 1.e\text{-}20$ is used as the refinement criterion

by 2,000 equi-distributed points on the circular interface of the initial configuration. These points are advected with 4th-order accurate Runge-Kutta (RK4) method with time step of $\Delta t = \frac{1}{3200}$, which is $\frac{1}{100}$-th of the time step used for AMR-MOF advection. At the intermediate time steps corresponding to those of AMR-MOF computation, unstructured triangular meshes are generated and used as the reference solution for those particular time steps. Total 17 meshes are generated for being used as the reference solution at the time index of $i_t = 0, 8, 16, \cdots, 128$ for covering the first half of the period (note the total number of time stepping is $n_t = 256$). The same meshes are used in the

ACCEPTED MANUSCRIPT



(a)$time = 0.0$ (b)$time = 1.0$ (c)$time = 4.0$

Fig. 28. Reference material configuration for the error (E_{sd}), the area of the symmetric difference, computation. The reference solution is prepared by front tracking and mesh generation at each time step. Three representative time steps are chosen for display. Initially 2,000 points are equally distributed on the circular interface of the initial material configuration. Each points are tacked with RK4 scheme with time step of $\Delta = 1/3200$, i.e. 100 time steps are taken for each time step of AMR-MOF.

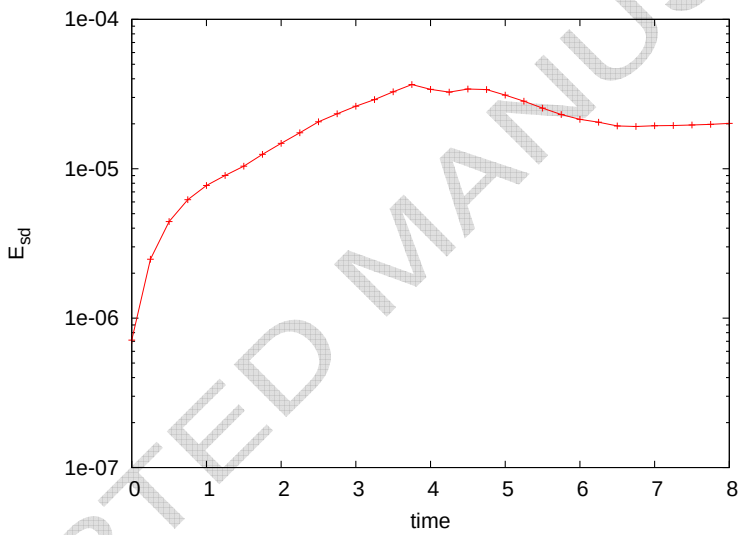


Fig. 29. Error E_{sd} computed by the area of symmetric difference between reference solution (as shown in Fig. 28) and AMR-MOF computation with refinement up to level-4 (as shown in Fig. 26).

reversing order for the second half of the period.

By using the set of reference material configuration, the error E_{sd} is computed at the intermediate time steps of the AMR-MOF computation. The evolution of error is displayed in Fig. 29 for a full period computation. The errors from AMR-MOF computation with different refinement levels are also listed in Table 6 at three representative time steps: initial ($time = 0.0$), maximum stretch ($time = 4.0$), and final ($time = 8.0$) time steps. It is interesting to note that the error increases until the maximum stretch (in the first half of the period) and then decreases in the reversing phases.

ACCEPTED MANUSCRIPT

max. refinement level	$time = 0.0$	$time = 4.0$	$time = 8.0$
	E_{sd}	E_{sd}	E_{sd} / $\Delta\mathcal{V}$
0	1.736e-04	3.402e-02	2.342e-02 / 1.423e-13
1	4.061e-05	5.826e-03	3.313e-03 / -1.761e-14
2	1.279e-05	9.722e-04	5.781e-04 / 6.591e-15
3	2.906e-06	1.311e-04	1.224e-04 / -6.591e-15
4	7.129e-07	3.401e-05	2.010e-05 / -5.551e-17

Table 6
Errors computed by the area of symmetric difference between reference solution and AMR-MOF computation. From the left, the errors at initial ($time = 0.0$), maximum stretch ($time = 4.0$), and finally reversed ($time = 8.0$) stages are listed. Total volume gain/loss is also indicated by $\Delta\mathcal{V} = \mathcal{V}^{final} - \mathcal{V}^{initial}$. 32^2 mesh is used as level-0 and AMR is performed up to level-4.

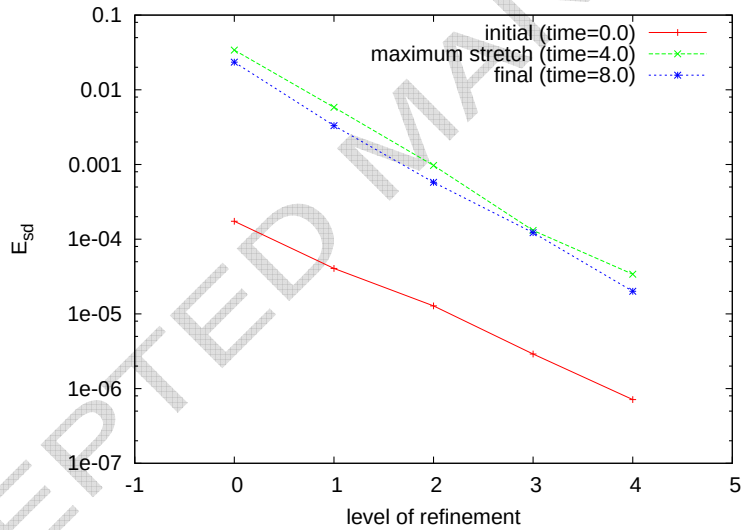
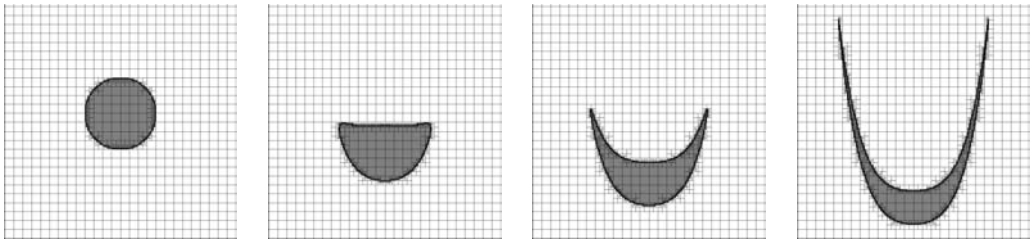


Fig. 30. Reduction of error E_{sd}, as summarized in Table 6, with respect to the maximum level of refinement allowed. The 32^2 mesh is used as level-0 and refinement is allowed up to level-4.

The reduction of error E_{sd} with respect to the level of refinement is displayed in Fig. 30. This shows that errors at three representative time steps are converging with second order accuracy. This also suggests that our reference material configuration, as displayed in Fig. 28 is adequate to be used as the reference.

ACCEPTED MANUSCRIPT



time=0.00 time=0.25 time=0.50 time=0.75

Fig. 31. Droplet flow. Level-0 mesh is 32^2 covering the domain of $[0,1]^2$. Maximum 5 levels of refinement is allowed (maximum effective mesh resolution is 1024^2). $E_{tol} = 1.e\text{-}20$ is used as the refinement criterion

4.6.5 Droplet flow

In addition to the more classical test cases, we introduce two new test cases. The first case is a divergence free nonlinear velocity field

$$\mathbf{v} = \begin{bmatrix} \frac{1}{8}(8x-4) \\ \frac{1}{8}\left\{-(8y-4)-4-(1-(8x-4)^2-(8x-4)^4)\right\} \end{bmatrix} \tag{15}$$

deforming a circular region to a droplet shape with two sharp edges.

The initial configuration is a circular region of radius $r_0 = 0.125$ center at the center of $[0,1]^2$ domain. The final time is $T = 0.75$, and the total number of time steps are $n_t = 75$, i.e. $\Delta t = 0.01$. Since the initially circular material region develops very sharp edges, it is a very appropriate test case for demonstrating the sharp corner as well as filament resolving capability of AMR-MOF method.

The actual computation is displayed in Fig. 31 for different time moments. For better visualization of final material configuration, the close up view of the primary material region is compared with the reference material configuration as shown in Fig. 32.

The reference material configuration is prepared by front tracking of interface points and mesh generation. Since the deformation rate is severe in this example, 10,000 points are equidistributed along the initial circular interface. Each points are tracked by RK4 scheme with a time step of $\Delta t = \frac{0.75}{7500}$ which is $\frac{1}{100}$-th of the time step used for AMR-MOF advection. At the final time step, the tracked points are selectively removed so that the minimum distance between the neighboring points are $\Delta s_{min} = \frac{\pi r_0}{1000}$. This results in a total $6,036$ points on the boundary of the final material configuration and also makes the size of the final mesh much smaller.

ACCEPTED MANUSCRIPT

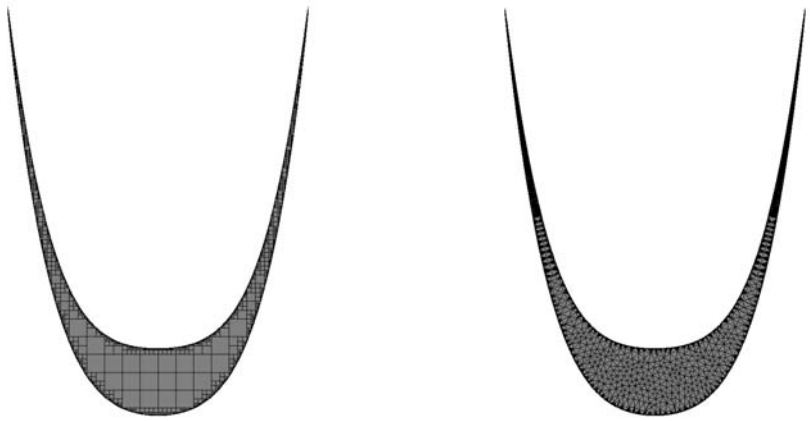


Fig. 32. Final configuration ($time = 0.75$) of droplet flow case. Left – AMR-MOF advection and interface reconstruction. Level-0 mesh is 32^2, and maximum effective mesh resolution is 1024^2 (level-5). Right – Mesh generated in the domain obtained by front tracking of boundary points.

max. refinement level	initial ($time = 0.00$)	final ($time = 0.75$)
	E_{sd}	E_{sd} / $\Delta\mathcal{V}$
0	1.506e-04	1.348e-02 / -1.513e-13
1	4.718e-05	3.352e-03 / -6.559e-14
2	1.010e-05	7.202e-04 / -3.767e-14
3	3.008e-06	1.664e-04 / -2.574e-15
4	7.267e-07	2.660e-05 / 6.931e-15
5	2.211e-07	5.549e-06 / -9.756e-15

Table 7
Droplet case: error computed by the area of symmetric difference between AMR-MOF computation and reference solution obtained by front tacking and mesh generation. Total volume gain/loss is also indicated by $\Delta\mathcal{V} = \mathcal{V}^{final} - \mathcal{V}^{initial}$.

By using the reference material configuration, the error E_{sd} is computed at the start and end of the AMR-MOF computation with different refinement levels and summarized in Table 7. The reduction of the error with successive refinement is also displayed in Fig. 33.

ACCEPTED MANUSCRIPT

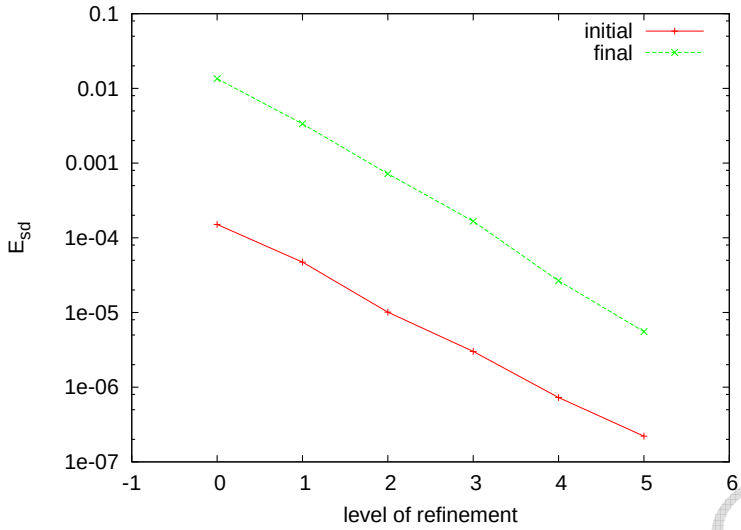


Fig. 33. Reduction of error E_{sd}, as summarized in Table 7, with respect to the maximum level of refinement allowed. The 32^2 mesh is used as level-0 and refinement is allowed up to level-5.

4.6.6 $\mathcal{S}$-shape flow

The last example problem is character $\mathcal{S}$-shape flow. It is also a divergence-free nonlinear velocity field

$$\mathbf{v} = \begin{bmatrix} \frac{1}{4}\left\{(4x-2)+(4y-2)^3\right\} \\ \frac{-1}{4}\left\{(4y-2)+(4x-2)^3\right\} \end{bmatrix} \tag{16}$$

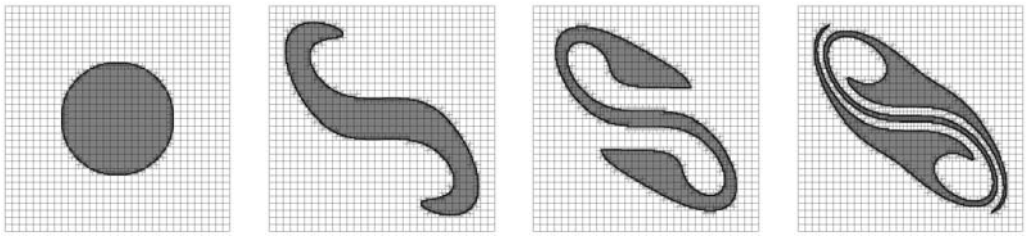
deforming a circular region to a character $\mathcal{S}$ shape.

The initial configuration is a circular region of radius $r_0 = 0.25$ center at the center of $[0,1]^2$ domain. The velocity field is defined as follows: The final time is $T = 3.0$, and total number of time steps are $n_t = 120$, i.e. $\Delta t = 0.025$. Since the initially circular material region develops thin filament region as well as sharp edges, it is also a very appropriate test case for demonstrating the capabilities of AMR-MOF method.

The actual AMR-MOF computation is displayed in Fig. 34 for different time moments. The refinement is performed up to level-4. For better visualization of final material configuration, the close up view of the primary material region is compared with the reference material configuration as shown in Fig. 35.

The reference material configuration is prepared in the same way as the previous droplet case. Since the deformation rate is also very severe example (especially in the central filament region), 10,000 points are equidistributed along the initial circular interface. Each point is tracked by a RK4 scheme

ACCEPTED MANUSCRIPT



time=0.0 time=1.0 time=2.0 time=3.0

Fig. 34. Superman flow. Level-0 mesh is 32^2 covering the domain of $[0,1]^2$. Maximum 4 levels of refinement is allowed (maximum effective mesh resolution is 512^2). $E_{tol} = 1.e\text{-}20$ is used as the refinement criterion

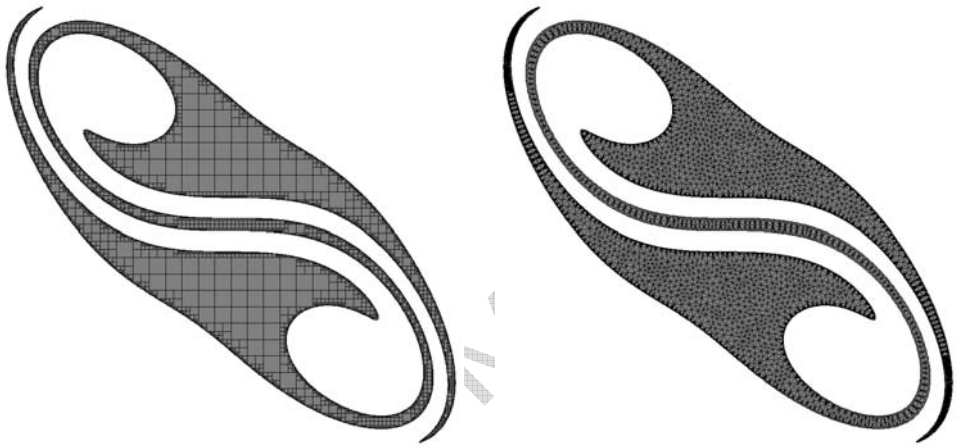


Fig. 35. Final configuration ($time = 3.0$) of $\mathcal{S}$-shape flow case. Left – AMR-MOF advection and interface reconstruction. Level-0 mesh is 32^2, and maximum effective mesh resolution is 512^2 (level-4). Right – Mesh generated in the domain obtained by front tracking of boundary points.

with time step $\Delta t = \frac{3}{12000}$ which is $\frac{1}{100}$ of the time step used for AMR-MOF advection. At the final time step, the points are selectively removed so that the minimum distance between the boundary points are $\Delta s_{min} = \frac{\pi r_0}{1000}$. This results in total $5,470$ point on the boundary of final material configuration and also makes the size of final mesh much manageable.

By using the reference material configuration, the error E_{sd} is computed at the initial and final configuration of the AMR-MOF computation with different levels of refinement, and it summarized in Table 8. The reduction of the error with successive refinement is displayed in Fig. 36, and confirms the second order accuracy both at the initial and final configurations.

ACCEPTED MANUSCRIPT

max. refinement level	initial ($time = 0.0$)	final ($time = 3.0$)
	E_{sd}	E_{sd} / $\Delta\mathcal{V}$
0	1.887e-04	2.878e-02 / 1.848e-13
1	4.043e-05	2.503e-03 / 5.842e-14
2	1.203e-05	5.403e-04 / 6.783e-14
3	2.907e-06	9.642e-05 / -3.608e-15
4	8.583e-07	2.683e-05 / 1.074e-14

Table 8
$\mathcal{S}$-shape flow case: Error E_{sd} computed by the area of symmetric difference between reference solution and AMR-MOF computation is listed. Total volume gain/loss is also indicated by $\Delta\mathcal{V} = \mathcal{V}^{final} - \mathcal{V}^{initial}$.

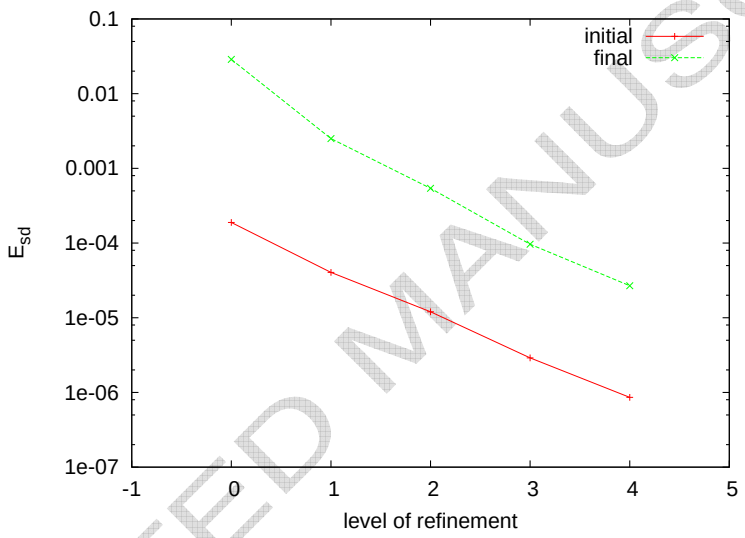


Fig. 36. Reduction of error E_{sd}, as summarized in Table 8, with respect to the maximum level of refinement allowed. The 32^2 mesh is used as level-0 and refinement is allowed up to level-4.

5 Conclusion

A new adaptive mesh refinement strategy based on the moment-of-fluid method was presented. The new method uses information about the material volume fraction and the material centroid position for interface reconstruction. Advection of these quantities is based on a cell-based semi-Lagrangian algorithm. We have presented a new advection algorithm for centroids. The AMR-MOF method uses new refinement criterion based on error in centroid position. Numerical examples demonstrate that the error in the centroid position can correctly detect not only regions with high curvature of the interface but also regions with subcell structures like filaments.

ACCEPTED MANUSCRIPT

We have demonstrated that new AMR-MOF method significantly improves the accuracy of volume-tracking evolving interface computations in comparison with other published results and the standard MOF method, which already gives more accurate result than any published results. The advantage of the AMR-MOF method is also strengthened by its superior efficiency (less computational cost) compared to the standard MOF method using uniform meshes.

In [54] we have coupled standard MOF without AMR with incompressible Navier-Stokes solver for two materials. In the future, we are planning to couple AMR-MOF with incompressible Navier-Stokes AMR solver for two materials. The extension towards multi-material (i.e. the number of material $n_{mat} \geq 3$) cases and also coupled simulation of incompressible multi-material flow is expected in our forthcoming research.

References

[1] M. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, Journal of Computational Physics 82 (1989) 64–84.

[2] R. B. Pember, J. B. Bell, P. Colella, W. Y. Curtchfield, M. L. Welcome, An adaptive cartesian grid method for unsteady compressible flow in irregular regions, Journal of Computational Physics 120 (1995) 278–304.

[3] A. Khawaja, T. Minyard, Y. Kallinderis, Adaptive hybrid grid methods, Computer Methods in Applied Mechanics and Engineering 189 (2000) 1231–1245.

[4] A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, M. L. Welcome, A conservative adaptive projection method for the variable density incompressible Navier-Stokes equations,, Journal of Computational Physics 142 (1998) 1–46.

[5] P. Colella, D. T. Graves, B. J. Keen, D. Modiano, A cartesian grid embedded boundary method for hyperbolic conservation laws, Journal of Computational Physics 211 (2006) 347–366.

[6] S. Popinet, Gerris: a tree-based adaptive solver for the incompressible euler equations in complex geometries, Journal of Computational Physics 190 (2003) 572–600.

[7] M. Aftosmis, M. Berger, J. Melton, S. Murman, Cart3D, http://people.nas.nasa.gov/∼aftosmis/cart3d/.

[8] G. F. Carey, A perspective on adaptive modeling and meshing (AM & M), Computer Methods in Applied Mechanics and Engineering 195 (2006) 214–235.

[9] T. Plewa, T. Linde, V. G. Weirs (Eds.), Adaptive Mesh Refinement - Theory and Applications, Vol. 41 of Lecture Notes in Computational Science and

ACCEPTED MANUSCRIPT

Engineering, Springer, 2003, proceedings of the Chicago Workshop on Adaptive Mesh Refinement Methods, Sept. 3-5, 2003.

[10] J. Morrell, P. Sweby, A. Barlow, A cell by cell anisotropic mesh ale method, Int. J. Numer. Meth. Fluids, DOI:10:1002/fld.1599.

[11] R. Anderson, N. Elliott, R. Pember, An arbitrary Lagrangian-Eulerian method with adaptive mesh refinement for the solution of the Euler equations, Journal of Computational Physics 199 (2004) 598–617.

[12] A. Kongies, R. Anderson, P. Wang, B. Gunney, R. Becker, D. Eder, B. MacGowan, M. Schneider, Modeling NIF experimental designs with adaptive mesh refinement and Lagrangian hydrodynamics, Intertial Fusion Sciences and Applications, 2005, J. Phys. IV France 133.

[13] A. Dawes, Prallel multi-dimensional and multi-material Eulerian staggered mesh schemes using localised patched based adaptive mesh refinement (AMR) for strong shock wave phenomena, in: T. Plewa, T. Linde, V. G. Weirs (Eds.), Adaptive Mesh Refinement - Theory and Applications, Vol. 41 of Lecture Notes in Computational Science and Engineering, Springer, 2003, pp. 295–302, proceedings of the Chicago Workshop on Adaptive Mesh Refinement Methods, Sept. 3-5, 2003.

[14] H. Jourdren, HERA: A Hydrodynamic AMR platform for multi-physics simulations, in: T. Plewa, T. Linde, V. G. Weirs (Eds.), Adaptive Mesh Refinement - Theory and Applications, Vol. 41 of Lecture Notes in Computational Science and Engineering, Springer, 2003, pp. 284–294, proceedings of the Chicago Workshop on Adaptive Mesh Refinement Methods, Sept. 3-5, 2003.

[15] I. Ginzburg, G. Wittum, Two-phase flows on interface refined grids modeled with vof, staggered finite volumes, and spline interpolants, Journal of Computational Physics 166 (2001) 302–335.

[16] D. Greaves, A quadtree adaptive method for simulating fluid flows with moving interfaces, Journal of Computational Physics 194 (2004) 35–56.

[17] J. Wang, A. Borthwick, R. Taylor, Finite-volume-type vof method on dynamically adaptive quadtree grids, International Journal for Numerical Methods in Fluids 45 (2004) 485–508.

[18] M. Malik, E. S.-C. Fan, M. Bussmann, Adaptive vof with curvature-based refinement, International Journal for Numerical Methods in Fluids 55 (2007) 693–712.

[19] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, M. L. Welcome, An adaptive level set approach for incompressible two-phase flows, Journal of Computational Physics 148 (1999) 81–124.

[20] F. Losasso, R. Fedkiw, S. Osher, Spatially adaptive techniques for level set methods and incompressible flow, Computers and Fluids 35 (2006) 995–1010.

ACCEPTED MANUSCRIPT

[21] M. Dai, D. Schmidt, Adaptive tetrahedral meshing in free-surface flow, Journal of Computational Physics 208 (2005) 228–252.

[22] W. J. Rider, D. B. Kothe, Reconstructing volume tracking, Journal of Computational Physics 141 (1998) 112–152.

[23] Z. Wang, Z. Wang, Multi-phase flow computation with semi-Lagrangian level set method on adaptive Cartesina grids, Tech. Rep. AIAA-205-1390, AIAA (2005).

[24] S. Vincent, J.-P. Caltagirone, A one-cell local multigrid method for solving unsteady incompressible multiphase flows, Journal of Computational Physics 163 (2000) 172–215.

[25] A. Theodorakakos, G. Bergeles, Simulation of sharp gas-liquid interface using VOF method and adaptive grid local refinement around the interface, Int. J. Numer. Meth. Fluids 45 (2004) 421–439.

[26] S. Popinet, Gerris Flow Solver, http://gfs.sourceforge.net/wiki/index.php/Main_Page/.

[27] S. Popinet, An accurate adaptive solver for surface-tension-driven interfacial flows, Tech. Rep. Submitted to Journal of Computational Physics, Prerint of National Institute of Water and Atmospheric Research, New Zeland (2008).

[28] V. Dyadechko, M. Shashkov, Moment-of-fluid interface reconstruction, Tech. Rep. LA-UR-05-7571, Los Alamos National Laboratory, `http://cnls.lanl.gov/~shashkov/` (2005).

[29] V. Dyadechko, M. Shashkov, Moment-of-fluid interface reconstruction – revised, Tech. Rep. LA-UR-07-1537, Los Alamos National Laboratory, `http://cnls.lanl.gov/~shashkov/` (2007).

[30] V. Dyadechko, M. Shashkov, Reconstruction of multi-material interfaces from moment data, Journal of Computational Physics 227 (2008) 5361–5384.

[31] H. T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, Tech. Rep. LA-UR-07-0656, Los Alamos National Laboratory, `http://cnls.lanl.gov/~shashkov/` (2007).

[32] H. T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, Journal of Computational Physics 226 (2007) 2096–2132.

[33] H. T. Ahn, M. Shashkov, Geometric algorithms for 3d interface reconstruction, in: M. L. Brewer, D. Marcum (Eds.), Proceedings of the 16th International Meshing Roundtable, Springer, 2007, pp. 405–422.

[34] B. Mirtich, Fast and accurate computation of polyhedral mass properties, Journal of Graphics Tools 1 (1996) 31–50.

[35] A. Staniforth, J. Cote, Semi-Lagrangian integration schemes for atmospheric models- a review, Monthly Weather Review 119 (1991) 2206–2223.

ACCEPTED MANUSCRIPT

[36] P. Smolarkiewicz, , J. Pudykewicz, A class of semi-Lagrangian approximations for fluids, Journal of the Atmospheric Sciences 49 (1992) 2082–2096.

[37] J. Dukowicz, J. Baumgardner, Incremental remapping as transport/advection algorithm, Journal of Computational Physics 160 (2000) 318–335.

[38] Q. Zhang, P.-F. Liu, A new interface tracking method: the polygonal area mapping method, Journal of Computational Physics, doi:10.1016/j.jcp.2007.12.014.

[39] N. Ashgriz, T. Barbat, G. Wang, A computational Lagrangian-Eulerian advection remap for free surface flow, Int. J. Numer. Meth. Fluids 44 (2003) 1–32.

[40] R. Nair, B. Machenhauer, The mass-conservative cell-integrated semi-Lagrangian advection scheme on the sphere, Monthly Weather Review 130 (2002) 649–667.

[41] J. López, J. Hernández, P. Gómez, F. Faura, An improved plic-vof method for tracking thin fluid structures in incompressible two-phase flows, Journal of Computational Physics 208 (2005) 51–74.

[42] M. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, Journal of Computational Physics 53 (1984) 484–512.

[43] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, M. L. Welcome, An adaptive level set approach for incompressible two-phase flows, Journal of Computational Physics 148 (1999) 81–124.

[44] K. Lipnikov, J. Morel, M. Shashkov, Mimetic finite difference methods for diffusion equations on non-orthogonal non-conformal meshes, Journal of Computational Physics 199 (2004) 589–597.

[45] C. Geuzaine, J.-F. Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, http://www.geuz.org/gmsh/.

[46] Z. Warsi, Fluid dynamics. Theoretical and Computational Approaches, CRC Press, 1993.

[47] M. Shashkov, B. Wendroff, The repair paradigm and application to conservation laws, Journal of Computational Physics 198 (2004) 265–277.

[48] M. Kucharick, M. Shashkov, B. Wendroff, An efficient linearity-and-bound-preserving remapping methods, Journal of Computational Physics 188 (2003) 462–471.

[49] R. Loubere, M. Staley, B. Wendroff, The repair paradigm: New algorithms and applications to compressible flow, Journal of Computational Physics 211 (2006) 385–404.

[50] H. Ahn, M. Shashkov, Adaptive moment-of-fluid method, Tech. Rep. LA-UR-08-2153, Los Alamos National Laboratory, `http://cnls.lanl.gov/~shashkov/` (2008).

ACCEPTED MANUSCRIPT

[51] S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, Journal of Computational Physics 31 (1979) 335–362.

[52] D. J. E. Harvie, D. F. Fletcher, A new volume of fluid advection algorithm: The stream scheme, Journal of Computational Physics 162 (2000) 1–32.

[53] R. Scardovelli, S. Zaleski, Interface reconstruction with least-square fit and split eulerian-lagrangian advection, International Journal for Numerical Methods in Fluids 41 (2003) 251–274.

[54] H. Ahn, M. Shashkov, M. Christon, The moment-of-fluid method in action, Communications in Numerical Methods in Engineering, to appear Tech. Rep. of Los Alamos National Laboratory - LA-UR-07-6854, also available at `http://cnls.lanl.gov/~shashkov/`.

ACCEPTED MANUSCRIPT