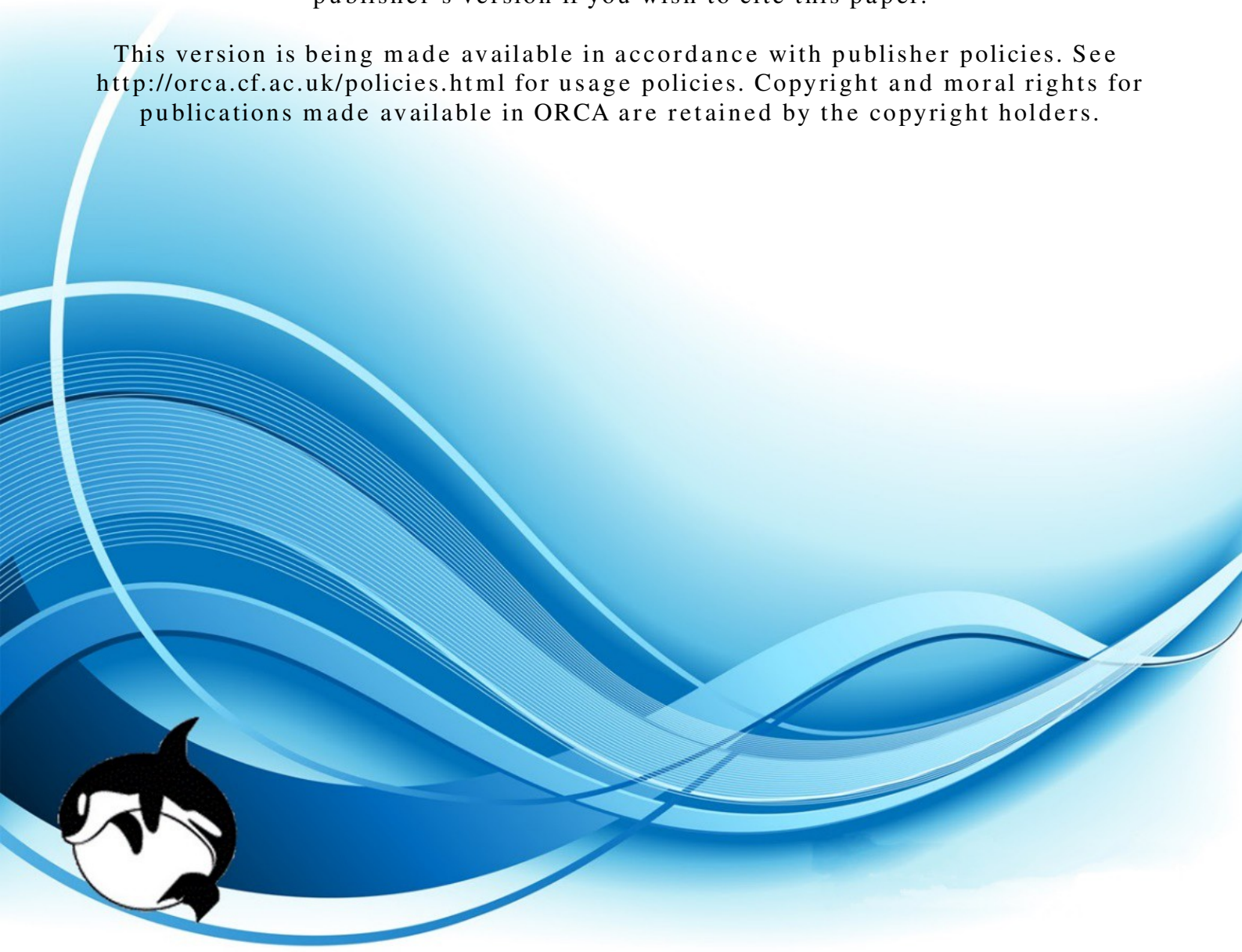
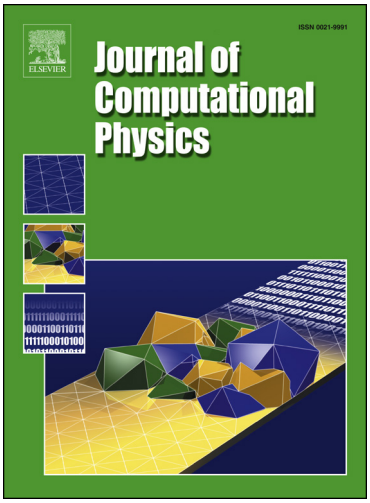


Journal Pre-proof



A quadtree-based adaptive moment-of-fluid method for interface reconstruction with filaments

Philippe Hergibo, Qiuhua Liang, Timothy N. Phillips and Zhihua Xie

PII: S0021-9991(23)00814-8
DOI: https://doi.org/10.1016/j.jcp.2023.112719
Reference: YJCPH 112719

To appear in: *Journal of Computational Physics*

Received date: 17 May 2023
Revised date: 8 November 2023
Accepted date: 16 December 2023

Please cite this article as: P. Hergibo, Q. Liang, T.N. Phillips et al., A quadtree-based adaptive moment-of-fluid method for interface reconstruction with filaments, *Journal of Computational Physics*, 112719, doi: https://doi.org/10.1016/j.jcp.2023.112719.

Highlights

- Adaptive mesh refinement is implemented with the filament moment-of-fluid method.
- Mass conservative algorithms are proposed for MOF at different adaptive level.
- Centroid error is used as the mesh refinement criterion.
- High accuracy validated with benchmarks using symmetric filament reconstruction.
- Unrestricted CFL using Lagrangian backtracking for AMR.

A quadtree-based adaptive moment-of-fluid method for interface reconstruction with filaments

Philippe Hergibo[a,b], Qiuhua Liang[c], Timothy N. Phillips[b], Zhihua Xie[a,*]

[a] *School of Engineering, Cardiff University, Queen's Buildings, Cardiff, CF24 3AA, UK*
[b] *School of Mathematics, Cardiff University, Abacws, Cardiff, CF24 4AG, UK*
[c] *School of Architecture, Building and Civil Engineering, Loughborough University, Loughborough, LE11 3TU, UK*

Abstract

Implementation of quadtree adaptive mesh refinement (AMR) to the moment-of-fluid (MOF) method is presented in the context of an interface capturing method. Filaments, thinner than a cell size, are resolved using a computationally efficient technique on an unconstrained quadtree structure. The centroid defect relative to its cell size is used as the refinement criterion, together with an enhanced refinement calculation and subsequently its volume conservation. In addition, different approaches are proposed to ensure mass conservation during the computation. This MOF-AMR framework is validated for a range of benchmark problems which are studied widely in the literature. There is no restriction on the choice of CFL number for the purely Lagrangian advection method considered here and this has advantages when combined with AMR. The current quadtree MOF-AMR method leads to much improved computational efficiency and accuracy relative to its grid size compared with a uniform grid. Higher levels of refinement can be costly, therefore the efficiency of mesh resolution is further discussed in relation to the choice of time step and number of AMR levels.

Keywords:
multiphase flow, MOF method, filament capturing, interface reconstruction, adaptive mesh refinement, quadtree
PACS: 0000, 1111
2000 MSC: 0000, 1111

*Corresponding author. Email: zxie@cardiff.ac.uk; Tel.: +44(0)29-20879375.

1. Introduction

Multiphase flow modelling has been widely used in many engineering applications. Whilst it is important, representing the interface between two or more materials to provide accurate prediction of complex topological structures is technically challenging. For simulation of natural flow processes or industrial applications such as wave breaking, droplet behaviour or bubble dynamics, highly deformable materials are of interest, which often involved curved interfaces due to surface tension. Accurate prediction of these evolving interfaces requires considerable computational resources.

To effectively improve models' computational efficiency, adaptive mesh methods have been applied in many complex and large-scale engineering modelling applications, including fluid dynamics, climate modelling and solid structure analysis. Examples include adaptive unstructured mesh [1, 2], adaptive polynomial degree [3] and adaptive mesh refinement [4, 5, 6, 7, 8, 9, 10, 11] methods. The purpose of adaptive mesh methods is to adjust dynamically the resolution of a grid in regions of interest or rapid change whilst maintaining coarse grid resolution in the regions where the solution "stagnates". Grid adaptation is triggered using a specified refinement criterion. Through "optimising" the number of grid cells used in the computation, such a grid adaptation strategy may effectively enhance computational efficiency while maintaining overall solution accuracy [9].

Methods for predicting the behaviour of fluids with complex interfaces include marker-and-cell method [12], front tracking method [13], level-set method [14, 15], volume-of-fluid (VOF) method [16, 17], and some meshless methods such as the smooth particles hydrodynamics (SPH) [18]. Some numerical techniques might be easier to implement than others, some might have better mass conservation property, and some might resolve complex interfaces in a superior way. Overall, all these techniques have been adopted widely in the literature for interface calculation.

The moment-of-fluid (MOF) method belongs to the class of methods known as interface capturing techniques and has been used to capture the interface between two materials [19]. It can be considered as an extension of the VOF method, in which the volume fraction as well as the centroid are advected to reconstruct an interface within a cell independent of neighbouring cells [19, 20, 21]. Recently, the standard MOF method [19] has been improved using symmetric reconstruction [22] and its capability has been extended to multi-materials [23, 24] which has enabled filaments and thin

structures to be reconstructed [25, 26].

Several techniques have been used over the years to capture interfaces in the context of adaptive mesh refinement (AMR) [4]. These may be broadly categorised as either patch-based AMR or quadtree(octree)-based AMR. Patch-based AMR involves dividing the computational domain into a set of refinement patches. This allows for local control over the mesh resolution, and the patches can be refined or coarsened dynamically based on the numerical solution being computed. The main advantage of patch-based AMR is its flexibility. However, it can lead to increased complexity in the maintenance of the grid being created [27]. On the other hand, a quadtree(octree)-based AMR uses a tree-based hierarchical data structure. The computational domain is recursively divided into four subcells (in 2D), or eight for octree (in 3D) when the refinement criteria are met. This approach has the advantage of being computationally efficient and easier to implement due to having a well-defined data structure [9]. This method is typically implemented on Cartesian grids, and also used to support some finite element simulations on tetrahedral meshes [28] and particle methods [29].

In the past, AMR has been adopted to simulate interfacial flows using volume-of-fluid [7, 10] and level-set [5, 11] methods. Local grid refinement has been confirmed to be effective in reducing the computational cost compared to refining the entire grid [30, 31, 32]. Furthermore, unphysical material break-up might be avoided through local refinement techniques. The refinement criterion is typically based on the volume fraction or level set function in a cell. But the estimated curvature [33] and interface gradient have also been used as refinement criteria.

Despite the potential gain in efficiency, using adaptive mesh refinement in the context of MOF has not been sufficiently explored. Undoubtedly, the associated complexity and natural computational cost of the MOF method itself is the reason why AMR has been limited in this context. In MOF situations, where zeroth and first moments are computed for interface reconstruction, the latter is used as a refinement criterion. Indeed, the centroid approximation is an estimate of the quality of the interface reconstruction. This adaptive mesh refinement method combined with the original MOF method was first developed in 2009 by Ahn & Shashkov [34]. This involved the use of an unconstrained quadtree structure with up to five levels of refinement. In addition, the refinement criterion was set to be unique for all levels and to a value smaller than machine precision. Later, Jemison et al.

[25] proposed for the first time a filament MOF approach in a patch-based AMR framework. In the advection process, their approach reached up to two levels of refinement and their criterion included a tolerance taking into account the relative subcell size. Recently, a standard MOF-AMR using a patch-based grid has been used in the flow simulation of droplets [35]. However, no attempt has been reported to combine the quadtree-based AMR and filament MOF approaches to explore their advantages in improving model performance.

The main objective of this paper is to develop a novel MOF method that can reconstruct under-resolved structures, mainly filaments, at any level of refinement in a quadtree-based AMR structure using a newly developed efficient symmetric MOF scheme [26]. Using a Lagrangian pre-image, materials are advected at a base level and then refined locally to allow for a CFL number much larger than unity. Therefore, the coarsening procedure becomes irrelevant. Fig. 1 highlights the key steps in the AMR reconstruction. In addition, a novel treatment within the MOF environment is also proposed to ensure the mass conservation property is satisfied to machine accuracy using a uniform redistribution procedure [36].

This paper is organised as follows. Section 2 describes the standard MOF method including how filaments are reconstructed within a cell on a fixed grid. Then, the adaptive mesh refinement structure and its advection procedure are presented in Section 3. Section 4 gives an insight into several volume redistribution techniques designed to conserve volume exactly in dynamic cases. In Section 5, results and analysis of several benchmark problems are presented. Finally, the efficiency and accuracy of the AMR is discussed in Section 6 and compared to a uniform grid approach. Some concluding remarks are included in Section 7.

2. MOF method

2.1. Interface reconstruction

Reconstructing a standard interface of a desired material using the MOF method requires the computation of the zeroth and first moments, M_0 and $\mathbf{M_1}$, respectively, within a convex cell Ω. These quantities are given by the following expressions :

$$M_0 = \int_\Omega dV = \frac{1}{2}\sum_{i=1}^{n}\left(x_{i-1}y_i - x_i y_{i-1}\right) \tag{1}$$

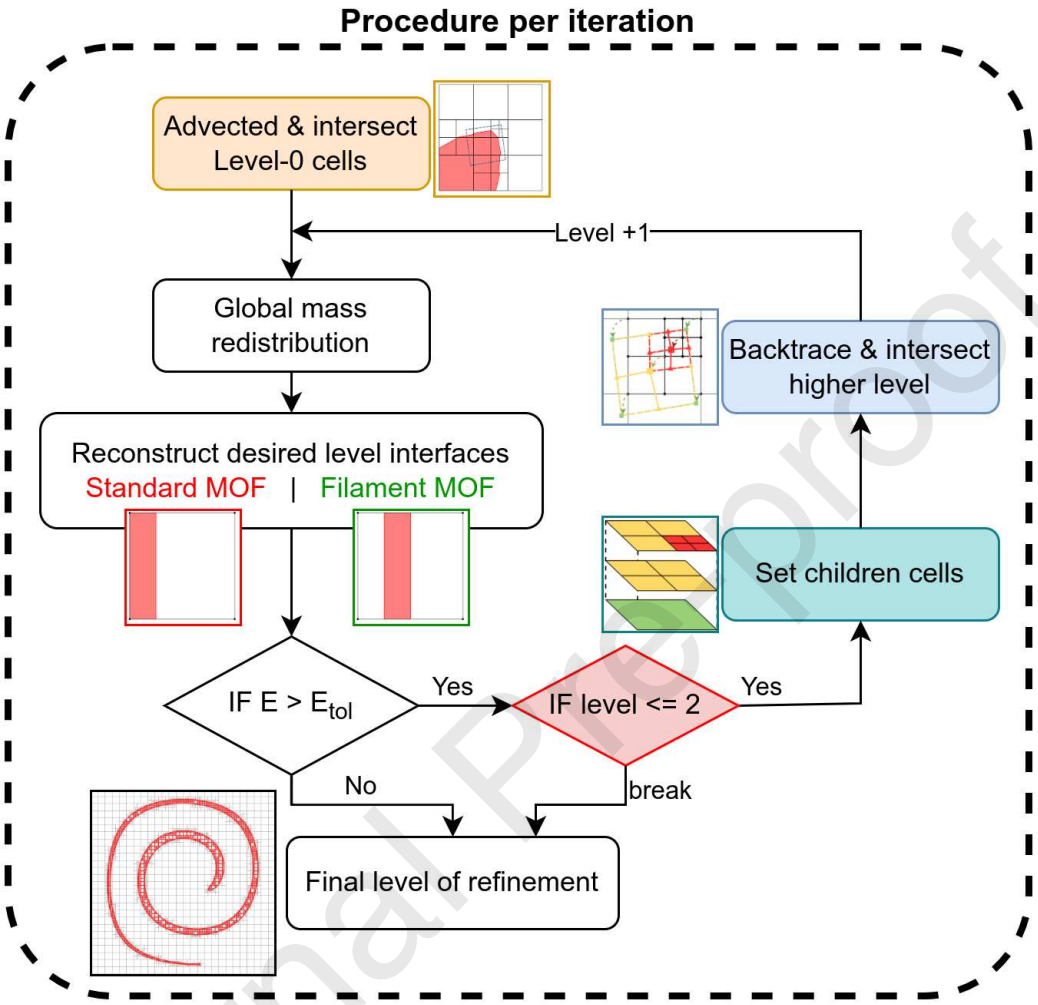


Figure 1: Flowchart highlighting the key steps to MOF-AMR reconstruction.

$$\mathbf{M_1} = \int_{\Omega} \mathbf{x} dV = \begin{bmatrix} \frac{1}{6} \sum_{i=1}^{n} (x_{i-1} + x_i)(x_{i-1} y_i - x_i y_{i-1}) \\ \frac{1}{6} \sum_{i=1}^{n} (y_{i-1} + y_i)(x_{i-1} y_i - x_i y_{i-1}) \end{bmatrix} \tag{2}$$

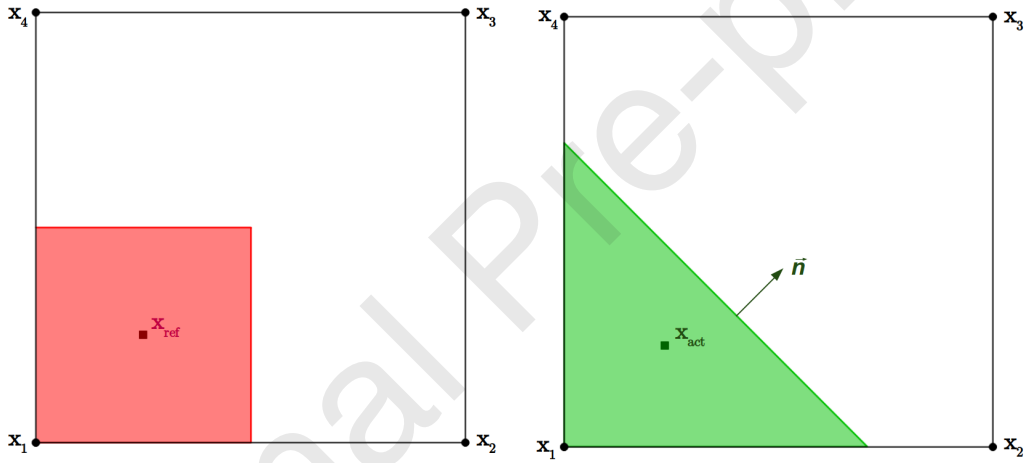
where (x_i, y_i), $i = 1, \ldots, n$, are the co-ordinates of the vertices of a polygonal cell. Note that the reference volume fraction F_{ref} corresponds to the zeroth moment relative to the convex cell area and the reference centroid

$\mathbf{x}_{ref}$ corresponds to the first moment relative to its zeroth moment.

In order to reconstruct a piece-wise linear interface in the context of the MOF method, the distance between the reference and reconstructed centroids is minimised while preserving the value of the zeroth moment. Eq. (3) describes the objective function $E_c(\mathbf{n})$ to be minimised, where $\mathbf{n}$ defines the unit outward normal to the interface. A minimisation algorithm is needed for non-rectangular cells, and a novel bisection method is used herein [26]. For rectangular cells, in particular Cartesian cells, an analytical solution is available, which eliminates the need to use a minimisation algorithm.

$$E_c(\mathbf{n}) = |\mathbf{x}_{ref} - \mathbf{x}_{act}| \tag{3}$$

where $\mathbf{x}_{act}$ refers to the reconstructed centroid.



(a) Reference interface (b) Reconstructed interface

Figure 2: Reference vs. reconstructed interface using a standard MOF approach. $\mathbf{x}_1$,...,$\mathbf{x}_4$ represent the cell vertices. $\vec{\mathbf{n}}$ denotes the outward normal of the reconstructed interface.

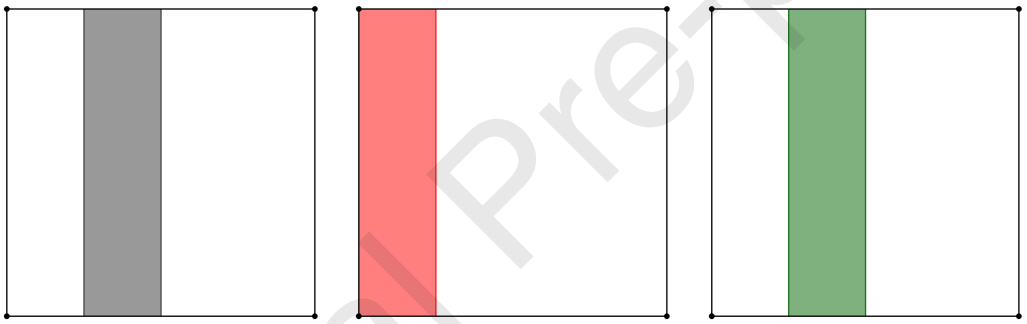
2.2. Filament reconstruction

Filaments are thin structures created during material deformation. Since they are usually smaller than a cell size, some special treatment has been developed [37, 38]. It is worth noting that a standard MOF approach cannot resolve their exact topology. When considering filaments, two interfaces appear within a cell, one on each side of the thin structure, which means that two reconstructions are needed to capture the topology accurately. In

this approach, the thin structure needs to be detected before it can be reconstructed, creating the need of an extra step in the MOF reconstruction procedure. This extra step involves the use of the conglomeration algorithm detailed in Hergibo et al. [26]. This conglomeration algorithm detects the number of groups, i.e. conglomerates, that are present in a cell by collecting all sub-polygons adjacent to each other. A multi-material reconstruction procedure is needed when filaments are involved. Therein, a symmetric reconstruction is used in the paper, which minimises both conglomerates considered. This reduces the number of combinations and is therefore computationally more efficient. Eq. (4) refers to the objective function E_c^{sym} in a symmetric reconstruction process:

$$E_c^{sym}(\mathbf{n}) = |\mathbf{x}_{ref} - \mathbf{x}_{act}| + \left|\mathbf{x}_{ref}^{rem} - \mathbf{x}_{act}^{rem}\right| \tag{4}$$

where the superscript *rem* denotes the remaining conglomerates in a cell.



(a) Reference interface displaying a filament

(b) Reconstructed interface using a standard MOF approach

(c) Reconstructed interface using a filament MOF approach

Figure 3: Schematic diagrams showing the different MOF reconstruction approaches. Filament MOF has the potential to offer exact reconstruction.

Note that the number of conglomerates is capped at three for the sake of computational efficiency. A sequential reconstruction is needed to minimise the centroid error regardless of the reconstruction of the other conglomerates considered [22]. By definition, all the possible combinations of material are tested to find the reconstruction that minimises all centroids present in a cell. The total centroid defect E can be calculated using:

$$E = \sqrt{\sum_{i}^{n} |\mathbf{x}_{ref}(\mu_i) - \mathbf{x}_{act}(\mu_i)|^2} \tag{5}$$

where μ_i characterises each material in a cell. In filament reconstructions, μ_i refers to each conglomerates until the fictitious material is assigned.

2.3. Advection on a uniform mesh

Advection of a material defines the process of its dynamic evolution through translation, rotation, and deformation. These are of particular interest when evaluating the precision of an interface tracking/capturing technique/method. This paper uses a purely Lagrangian approach as opposed to a mixed Eulerian-Lagrangian approach [21, 26]. The benefits of using a Lagrangian approach include a less restrictive choice of Courant-Friedrichs-Lewys (CFL) number and an unsplit advection approach in which a Lagrangian pre-image is used to capture the volume fraction and centroid of a material at the previous time step.

The vertices of a cell are advected backwards in time using a second order Runge-Kutta scheme (RK2) before determining its intersection with the pre-image. Advection forwards is then performed for centroids and reconstruction using the relevant volume fraction. The key steps of the advection procedure are summarised as follows:

(i) Backtrace any cells that may contain the desired material using RK2.
(ii) Intersect the backtrace cell with any material encountered to evaluate the reference volume fractions.
(iii) Advect individual centroids and compute their weighted average to obtain the reference centroid.
(iv) Reconstruct using one of the following techniques.
    (a) Standard MOF using a piecewise linear interface between two materials.
    (b) Filament MOF using conglomeration algorithm to reconstruct a filament.

Despite showing a refined grid, Fig. 5 demonstrates visually the steps for MOF advection and is valid for a uniform grid.

3. Adaptive mesh refinement

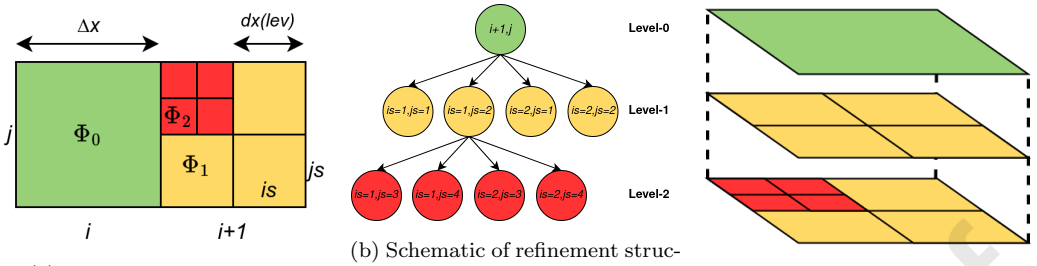
The main motivation for using AMR is to balance the trade-off between solution accuracy and computational cost. High deformation regions are of

interest for AMR in order to produce high-resolution prediction in these complex areas. The process of mesh refinement must be informed by an appropriate criterion and the reconstruction error is used to inform grid adaptation in the current MOF method [34]. Typically, the refinement process involves splitting each cell into 4 subcells in 2D, and 8 subcells in 3D; coarsening involves merging subcells into a larger (sub)cell when the specific condition is met. Generally, the refinement and coarsening processes are repeated until either a desired accuracy is achieved or a certain level of refinement is reached.

3.1. Data structure

This work adopts the quadtree-based AMR and the data structure is designed to store and manipulate the hierarchy of meshes with ease and efficiency, as well as allowing communication between levels [9]. The quad-tree data structure forms a tree where the root node represents the coarsest mesh (base mesh), and each additional level of refinement creates four children nodes to their parent node. This work introduces a simplified approach in which the data structure replicates a quad-tree algorithm up to two levels of refinement such that children cells can be accessed from a parent cell at every level up to two. The other advantage is that no subroutines are needed to find or access neighbours. In addition, unlike many other AMR codes, the new approach does not constrain the refinement level of neighbouring cells or subcells.

An arbitrary cell on the quad-tree mesh generated using the new approach is indexed as (i, j, is, js) where (i, j) represents the base mesh indices and $is = 1, ..., Ms$ and $js = 1, ..., Ms$ are the subcells indices, with $Ms = 2^{lev}$ and lev denoting the level of refinement starting at 0 for the base mesh. Subsequently, the size of the new subcells is defined by $dx(lev) = \Delta x/2^{lev}$ and $dy(lev) = \Delta y/2^{lev}$ with lev being $0, 1$ or 2. Naturally, at Level-0, $dx(0) = \Delta x$. In addition, the cell-centre coordinates can be directly decided through the following relationships $x_c = x(i) + (is - 1) \cdot dx(lev)$ in the x-direction and $y_c = y(j) + (js - 1) \cdot dy(lev)$ in the y-direction. The cell area $|\Omega|\,(lev)$ is then defined by $|\Omega|\,(lev) = dx(lev)dy(lev)$, and for uniform mesh, the cell area is simply defined as $|\Omega|$. Accessing children cells uses the logic from a parent cell's index parity is. Children subcells indices can be called using $(2is, 2is-1)$ when the parent index is is even, and $(is, is+1)$ when is is odd. This logic is valid up to level 2 and works in both horizontal and vertical directions. The data structure and index system are highlighted in Fig. 4.



(a) Quadtree structure grid

(b) Schematic of refinement structure using indices

(c) Schematic of levels structure

Figure 4: General idea of data structure used in an AMR framework

In order to allow for the manipulation of variables, an additional index representing the level of refinement is used in the data structure. Because the data structure may contain several variables accounting for volume fraction, centroid or polygon representation at different levels of refinement, a specific variable is used in the code for identifying which level of refinement is reached. The logical variable *last_lev_refinement(i,j,is,js,lev)* allows to enable or disable any values of unused level of refinement. A true value means that subcell i, j, is, js at level lev is the last refinement and contains a valid volume fraction to be intersected. A false value shows that the subcell i, j, is, js at level lev is not the last level of refinement and values are ignored. In general, when a higher level of refinement is triggered, the logical value of *last_lev_refinement* of the corresponding parent subcell at a lower level is set to false. The value of this logical variable is set to true for all children subcells. Mathematically, the set Φ_{lev}, including subsets Φ_0, Φ_1 and Φ_2, respectively, represents all cells at their finest refinement i.e. $\Phi_{lev} = \Phi_0 \cup \Phi_1 \cup \Phi_2$ with $\Phi_i \cap \Phi_j = \emptyset$ for $i \neq j$. This means, with reference to the color scheme of Fig. 4, Φ_0 corresponds to green cells, Φ_1 to yellow subcells and Φ_2 to red subcells. Algorithm 1 details how to loop and access any variables in our code.

3.2. Refinement criterion

In the previous VOF methods or level set methods, refinement was triggered when the volume fraction is in a certain range or when the estimated curvature gradient reaches a certain value. In the adopted MOF method, the centroid error is used as the criterion for refinement. Indeed, when a standard MOF or a filament MOF procedure is used in a cell, reconstruction error is a good indicator of how accurate the reconstruction is. Eq. (5) includes material centroid error and fictitious material in a filament case. In this paper,

Algorithm 1 AMR data structure

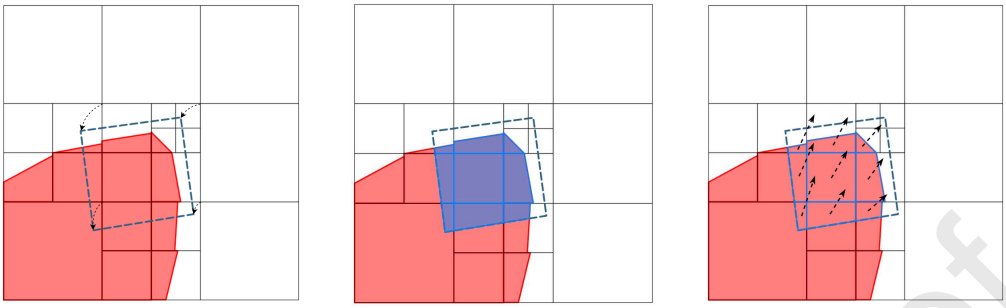
```
for lev = 0,2 do
  for i = 1, N_cell_x do
    for j = 1, N_cell_y do
      Ms = 2^lev; dx(lev) = Δx/2^lev; dy(lev) = Δy/2^lev
      for is = 1, Ms do
        for js = 1, Ms do
          % EXAMPLE : accessing the volume fraction of a subcell
          volume_fraction(i,j,is,js,lev)
          % EXAMPLE : checking the last level of refinement of a subcell
          last_lev_refinement(i,j,is,js,lev) ← true
        end for
      end for
    end for
  end for
end for
```

the refinement criterion is dependent on the cell size, here $10^{-9}dx(lev)$, finer than [25] and not set to a fixed tolerance, which is different from the previous MOF-AMR schemes [34].

3.3. Advection procedure on a refined mesh

Similar to advection on a uniform mesh, the advection on a refined mesh entails the use of a Lagrangian pre-image. This requires intersecting the material at the previous time step without omitting the different levels of refinement. All levels of refinement needs to be intersected. As per the uniform approach, the backtrace cell is advected backwards using RK2, and the area intersected in this pre-image relative to the subcell area corresponds to the volume fraction of the refined subcell. Eventually, the centroids of all polygons intersected forming the volume fraction are advected forward using the same scheme and the weighted average will define the new reference centroid. The respective reference volume fraction and centroid are used for reconstruction.

Our approach differs from other general MOF-AMR schemes since here filament MOF is enabled. The ability to capture filaments has significant advantages over a standard MOF method and has been shown to generate high accuracy on a uniform mesh [26]. For the sake of capturing filaments, the



(a) Lagrangian backtracking pre-image

(b) Intersection between back-trace and material at all levels

(c) Forward advection of individual centroids

Figure 5: Schematic showing advection of moments in an AMR framework

base mesh is used as the backtracking level, meaning that Level-0 is advected first at all time. To determine which cells need advecting, neighbouring volume fractions are used at Level-0 to evaluate its potential of being an interface, similar to a uniform advection.

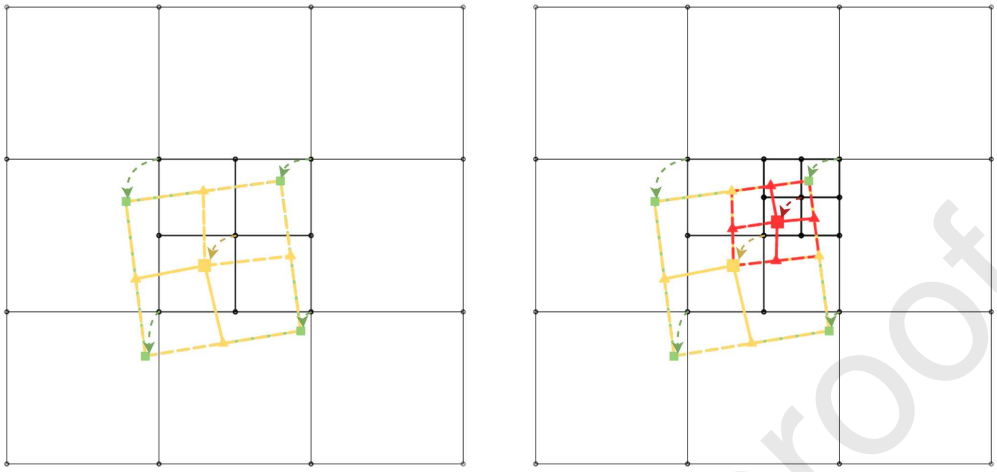
In the case of a cell being refined, the backtrace at a defined level of refinement is performed as follows. The central point common to all child subcells is advected using the usual RK2 subroutine. All other vertices are interpolated from the backtrace at Level-0. This will guarantee exact material intersection with the level of refinement below, hence exact mass conservation. This is performed in a similar fashion for Level-2.

(i) Backtrace the four vertices from a Level-0 cell using RK2.
(ii) Advect the common node using RK2.
(iii) Interpolate the four mid-points from the Level-0 backtrace cell
(iv) Create four new subcells

Fig. 6 shows the procedure in place for backtracking a subcell in this refinement framework. This ensures the intersection with a refined backtrace cell with the interface. This approach differs from the one introduced by [34] where a simplified backtracking approach is used. As stated above, our approach does not create gaps and overlaps, and therefore a simpler mass redistribution procedure can be implemented.

3.4. Time-step on a refined mesh

In numerical simulations, the typical time step is determined according to the CFL condition. In this paper, the CFL number is chosen to be unity



(a) Backtracking a subcell at Level-1 (b) Backtracking a subcell at Level-2

Figure 6: Schematic showing backtracking of moments in an AMR framework. (□) symbols refer to vertices advected using RK2. (△) symbols refer to mid-point vertices being interpolated. Color scheme shows green vertices for Level-0; Yellow vertices and dashed lines for Level-1 vertices and subcells; Level-2 vertices and subcells are in red.

unless stated otherwise. The Lagrangian approach enables an unrestricted choice of the CFL number [26]. Specifically, the time step is chosen with respect to the base mesh. When refining a mesh locally, the time step used at a refined cell is the same as the one adopted at the base mesh, and so the CFL number is 2 for Level-1 and 4 for Level-2 cells. Alternative time step strategies such as adaptive time stepping can be employed, but for simplicity these were not adopted here as no instability issues were encountered using the present approach.

4. Mass conservation during advection

4.1. Uniform global mass redistribution

Mass conservation is difficult to enforce in a grid refinement procedure. Local redistribution in a refinement step can lead to a large deformation of the interface, hence global redistribution is used in this paper. When using a global redistribution approach, over/under-filled cells are considered. These cells are formed by only one material, however their area intersected lead to a volume fraction being either less or greater than unity. These cells have their

volume fraction set to unity and the difference to unity multiplied by the cell area $|\Omega|\,(lev)$ is added to a global variable. Let us call the global redistribution variable δ. On a uniform approach, δ is redistributed to N mixed cells, i.e. cells with an interface. In fact, the $\delta/(|\Omega|\,N)$ amount is redistributed to mixed cells. In the case that not all of the mass is redistributed, an iterative procedure is enacted to ensure all mass is redistributed. Indeed, lack of mass redistribution can penalise mass conservation at other levels.

At other levels of refinement, over/under-filled cells may also occur in the intersection process, meaning that mass needs to be redistributed at all levels. However, on a refined mesh, mass has been redistributed at a lower level with cells needing refinement. Therefore, another local variable *"distributed"* is considered in the redistribution process corresponding to the mass redistributed in each cell/subcell at a lower level. It allows to keep track of redistribution to cells that may trigger refinement, with the sum of them all being δ_{low_lev}. Indeed, mass may be redistributed to cells that will be refined, hence that amount needs to be shared to the next refinement level. All cells needing refinement have their mass redistributed at a lower level added to the δ of the refinement level. Then, the new amount of mass redistributed is $\delta + \delta_{low\_lev}$. Therefore, for each cell, the new volume fraction F_i is calculated using

$$F_i \leftarrow F_i + \frac{(\delta + \delta_{low\_lev})}{|\Omega|\,(lev) N_{mix}} \tag{6}$$

where N_{mix} corresponds to the number of mixed cells in the domain at a certain level. The following subroutine gives more insight into the redistribution procedure 2. In this subroutine, the amount of mass that is not repaired *"not_repaired"* is taken into account because some "almost" full/empty cells may not be able to receive/give their contribution. In these instances, the redistribution subroutine is repeated until the amount of mass is close to machine precision. The amount redistributed is kept in the variable *"distributed"*.

Several approaches can be used when redistributing the mass globally. Two of them are presented in this section, these are termed the directly proportional and inversely proportional distribution approaches.

4.2. Directly proportional global mass redistribution

Amongst redistribution procedures, the directly proportional redistribution seems intuitive. The redistribution occurs in a similar fashion as the

Algorithm 2 Redistribution in a refined mesh

Initialise δ, δ_{low_lev}
$repair \leftarrow \delta + \delta_{low\_lev}$
$not\_repaired \leftarrow repair$
%Note : $\delta_{low\_lev} = 0$ at Level-0
while $(not\_repaired > 10^{-13})$ **do**
  if $(mixed\_cell \leftarrow$ **true .AND.** $repair < 0)$ **then**
    %REPAIR IS NEGATIVE - REMOVE VOLUME FRACTION
    if $(volfrac(i,j,is,js,lev) + repair/(|\Omega|\,(lev)N\_mix) < 0)$ **then**
      %CELL VOLUME FRACTION CANNOT BE NEGATIVE - RESET TO 0
      $not\_repaired = not\_repaired + volfrac(i,j,is,js,lev) * (|\Omega|\,(lev)N\_mix)$
    else
      $volfrac(i,j,is,js,lev) = volfrac(i,j,is,js,lev) + repair/((|\Omega|\,(lev)N\_mix)$
      $distributed(i,j,is,js,lev) = distributed(i,j,is,js,lev) + repair/N\_mix$
      $not\_repaired = not\_repaired - repair/N\_mix$
    end if
  else if $(mixed\_cell \leftarrow$ **true .AND.** $repair > 0)$ **then**
    %REPAIR IS POSITIVE - ADD VOLUME FRACTION
    if $(volfrac(i,j,is,js,lev) + repair/(|\Omega|\,(lev)N\_mix) > 1)$ **then**
      %CELL VOLUME FRACTION CANNOT MORE THAN UNITY - RESET TO 1
      $not\_repaired = not\_repaired + (1 - volfrac(i,j,is,js,lev)) * (|\Omega|\,(lev)N\_mix)$
    else
      $volfrac(i,j,is,js,lev) = volfrac(i,j,is,js,lev) + repair/((|\Omega|\,(lev)N\_mix)$
      $distributed(i,j,is,js,lev) = distributed(i,j,is,js,lev) + repair/N\_mix$
      $not\_repaired = not\_repaired - repair/N\_mix$
    end if
  end if
end while
% WHEN REDISTRIBUTION IS TRIGGERED AT NEW REFINEMENT LEVEL
$\delta_{low\_lev} = \sum distributed(i,j,is,js,lev)$ **if**$(last\_lev\_refinement(i,j,is,js,lev) ==$ **false**)

uniform case. However, the mass is redistributed proportional to the volume fraction in a cell. Hence, the total volume fraction of all mixed cells is calculated. The repair is then performed using the redistribution process in which the new volume fraction is given by

$$F_i \leftarrow F_i + \frac{(\delta + \delta_{low\_lev})}{|\Omega|\,(lev)} \frac{F_i}{\sum_{i=1}^{N_{mix}} F_i} \tag{7}$$

This approach may alter the shape of the interface to a lesser extent.

4.3. Inversely proportional global mass redistribution

Opposite to the previous concept, mass is redistributed inversely proportional to its volume fraction in this approach. Conceptually, a proportional approach may lead to several iterations of redistribution because a large mass is redistributed to an "almost" full/empty cell, and therefore the mass that is not repaired may be large. By using the inversely proportional approach, more mass is redistributed to those almost empty cells which intuitively would reduce the number of redistribution iterations, but may alter the shape of the interface more. The repair would be redistributed as follows

$$F_i \leftarrow F_i + \frac{(\delta + \delta_{low\_lev})}{|\Omega|\,(lev)} \frac{(1 - F_i)}{\sum_{i=1}^{N_{mix}} (1 - F_i)} \tag{8}$$

5. Results

5.1. Error evaluation

Computing errors play an important part of interface capturing methods as it is the primary indicator of the relevance of a method. Comparing errors enables one to evaluate the merits of different methods. The L_1 error norm E_{L_1}, which is based on a volume fraction approach, is one of these numerical indicators. On a refined grid, the L_1 error is evaluated on the base mesh, which means that refined subcells are grouped together to form a single volume fraction on the base mesh. Using the data structure described in Section. 3, the error can be evaluated using

$$E_{L_1} = \sum_{i,j} |(F_{final}(i,j) - (F_{initial}(i,j))|\,|\Omega|\,(0) \tag{9}$$

where $|\Omega|(0)$ represents the cell area at Level-0, $F_{final}(i,j)$ and $F_{initial}(i,j)$ are calculated in similar fashion :

$$F_{final}(i,j) = \sum_{lev} \sum_{is,js \subset \{\Phi_{lev}\}} \frac{F(i,j,is,js,lev)\,|\Omega|\,(lev)}{|\Omega|\,(0)} \tag{10}$$

where F represents the volume fraction in a subcell and Φ_{lev} corresponds to the set including subsets Φ_0, Φ_1 and Φ_2, respectively. If a cell contains subcells at level 1 and level 2, then Eq. (10) aims at summing up their volume fraction with respect to their subcell grid size. If a cell has not been refined, then level 0 remains and $F_{final}(i,j)$ is the volume fraction at level 0. The relative error norm E_r is given by

$$E_r = \frac{E_{L_1}}{\sum_{i,j} |(F_{initial}(i,j))|\,|\Omega|\,(0)} \tag{11}$$

The symmetric error is another indicator of the error in reconstruction. This error indicator provides an estimation of the discrepancy in the area between the initial and final states. The symmetric error E_{sym} is similarly given by

$$E_{sym} = \sum \left| \omega^{ref} \cup \omega^{act} - \omega^{ref} \cap \omega^{act} \right| \tag{12}$$

where ω^{ref} denotes the initial state reference interface, which is potentially curved, and ω^{act} denotes the final state reconstructed polygon.

Eventually, the mass difference is also used as an indicator. Mass conservation is critical during dynamic cases. In this paper, mass corresponds to the area encompassed within the original interface i.e.

$$\Delta m = \sum |F_{final}|\,|\Omega|\,(lev) - \sum |F_{initial}|\,|\Omega|\,(lev) \tag{13}$$

While the order of convergence is always calculated on a uniform mesh, an attempt at finding the pseudo-order of convergence of the mesh is presented here. Indeed, the order of convergence calculated with a uniform mesh is related to the mesh size but also to the ratio of number of cells between refinement levels. In that respect, we aim to give a ratio of maximum grid cells. The pseudo-order of convergence OC is given by

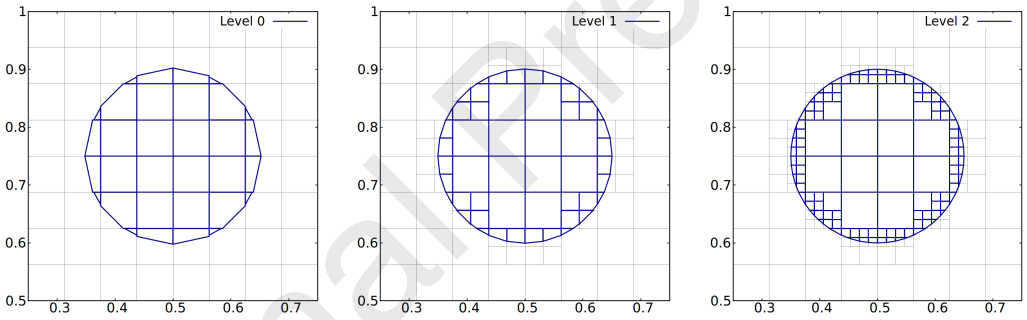
$$OC = \log\left(\frac{E_1^{sym}}{E_2^{sym}}\right) / \log\left(\sqrt{\frac{n_2^{max}}{n_1^{max}}}\right) \tag{14}$$

where E^{sym} corresponds to the symmetric difference error of a particular grid and n^{max} its maximum number of grid cells. Note that Eq. (14) is valid for

a uniform mesh and so the ratio of maximum number of cells in a constant environment gives the same order of convergence equation as in a uniform mesh.

5.2. Static reconstruction

Static reconstruction consists of reconstructing the interface of a material using the same AMR logic as that described in Section. 3. The only variation is that no advection is necessary. The refinement procedure still applies and the refinement criterion remains. The intersection of a circle of radius $r = 0.15$ centred at $[0.5, 0.75]$ in a unit domain on a finer grid is determined using the exact interface rather than the material configuration at the previous time step. Exact mass conservation is achieved at all levels. Fig. 7 highlights the difference in precision during reconstruction when refining the interface using zero, one or two levels of refinement. The symmetric difference error gives a good insight into the increased precision and accuracy obtained when using a higher level of refinement.



(a) Zero refinement level; $E_{sym} = 1.04 \times 10^{-4}$

(b) One refinement level; $E_{sym} = 4.24 \times 10^{-5}$

(c) Two refinement levels; $E_{sym} = 1.29 \times 10^{-5}$

Figure 7: Static reconstruction for a 16×16 base grid with zero, one and two levels of refinement and the associated symmetric difference error

5.3. Benchmark: Zalesak slotted disc

This benchmark test case involves a slotted disc which is rotated anti-clockwise in a rigid body rotation around the centre of the domain [39]. The circle of radius $r = 0.15$ has a rectangular slot of width $w = 0.05$ in its centre part with a maximum height of $h = 0.85$. The velocity field for this test case is given by

$$\mathbf{u}(x, y) = \begin{bmatrix} 0.5 - y \\ x - 0.5 \end{bmatrix} \tag{15}$$

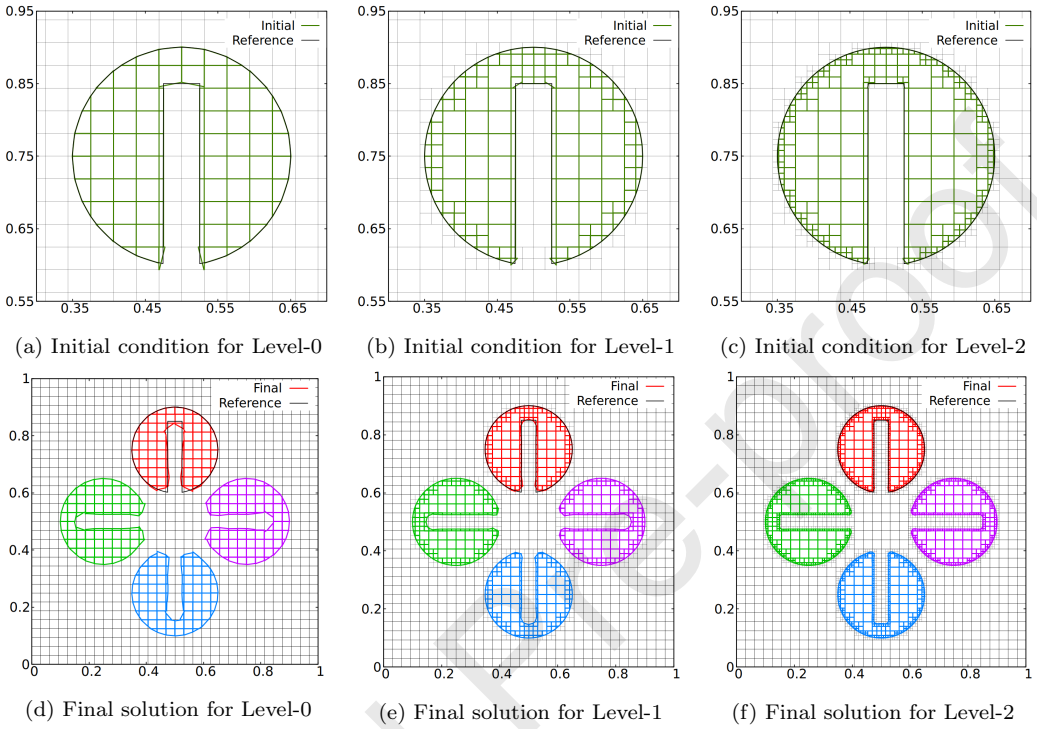
Even though no filaments are formed during the advection process, the filament capability of our code is still enabled. The rotational nature of this test case also highlights the fact that no deformation occurs in the material, hence the mass redistribution algorithms are enabled but not used as the backtrace is always of the same size as the cell area. This highlights the powerful choice of backtrace when refining a mesh as described in Section. 3. Three different grids are presented, explicitly 32×32 as base mesh and a Level-1 and Level-2 of refinement. The number of iterations is $n_{it} = 300$ and $\Delta t = 2\pi/n_{it}$.

Table 1: Dependence of the L_1 error, E_{L_1}, and relative error, E_r, on refinement level for the Zalesak slotted disc problem using a 32×32 base mesh

Refinement level	E_{L_1}	E_r
0	2.55×10^{-3}	4.38×10^{-2}
1	5.31×10^{-4}	9.13×10^{-3}
2	1.98×10^{-4}	3.41×10^{-3}

The error indicator used in this test case is the interpolated L_1 error. Table 1 presents the error for different levels of refinement. Fig 8 emphasises the difference between initial and final reconstructions, as well as the intermediate reconstructions captured during the full rigid body rotation. The shape of the interface is maintained well, except around the sharp edges of the rectangular slot. The MOF method, as it stands, is not able to reconstruct these sharp edges even when refining the grid locally. Note, however, that the straight interface around the longer edges of the rectangle is not refined during the initial condition. Indeed, because MOF reconstructs these cells exactly, the refinement criterion is not triggered. The main difference compared with the method of Ahn et al. [34] is that the tolerance used in that paper is independent of the cell dimension. This means that, with a tolerance set to be smaller than machine precision, even cells that are reconstructed exactly will be refined. Fig. 9 taken from Ahn et al.[34] shows that

the neighbourhood of slots are refined while it is not in our initial reconstruction.

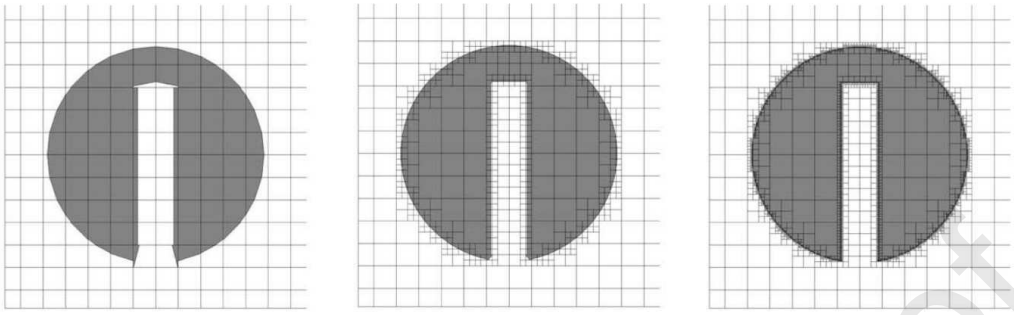


(a) Initial condition for Level-0 (b) Initial condition for Level-1 (c) Initial condition for Level-2

(d) Final solution for Level-0 (e) Final solution for Level-1 (f) Final solution for Level-2

Figure 8: Zalesak slotted disc test case for a 32×32 base grid and one and two levels of refinement. Top row of figures shows the initial reconstruction. Bottom row of figures shows the evolution of the shape of the interface.

The Zalesak slotted disc is also a good benchmark to evaluate the efficiency of the method through a time distribution. The time investigation is an average percentage of time per iteration. Five main blocks exist in this code, the first involves identifying level 0 cells that will need to be advected. This second and third part involves backtracking cells at any levels and also the intersection procedure. The final blocks involves global mass redistribution and interface reconstruction. Fig.10 highlights the percentage of time taken in each block of the code both for a 32×32 and 64×64 grid. The second plot shows data for the same finest level of refinement. Note the advection identification subroutine is insignificant, so is the redistribution procedure. Most of the time is taken in the intersection procedure as expected due to looping through all cells and subcells. As more levels are considered, the



(a) AMR up to Level-0 (b) AMR up to Level-2 (c) AMR up to Level-4

Figure 9: Figure taken from Ahn et al. [34] highlighting their initial reconstruction and choice of refinement for the Zalesak slotted disc.

percentage of time increases in the intersection procedure. However, the time spent in reconstruction does not increase significantly because of limited number of cells reaching higher levels. In addition, the likelihood of them not being filament reconstruction cells mitigates the computational cost.

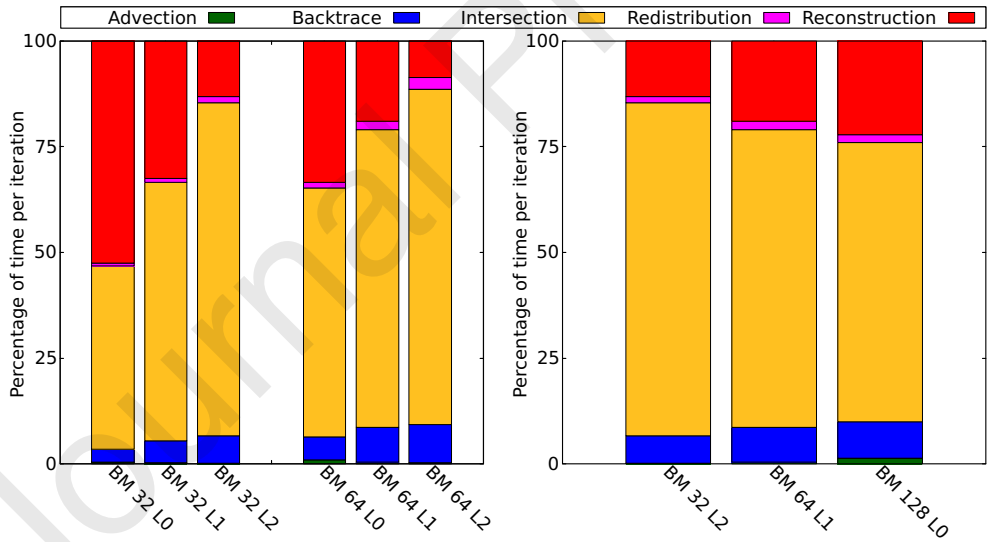


Figure 10: Percentage of time spent on key MOF processes per iteration for a 32×32 and 64×64 grid for level 0, level 1 and level 2, respectively. Comparison of time distribution for same finest level of refinement.

5.4. Benchmark: Reversible vortex T=8

The reversible vortex is an advection benchmark that has been widely studied in the literature [40]. This deformation case sees a circle of radius $r = 0.15$ within a unit domain and centered at $[0.5, 0.75]$ shearing its body along a divergence-free velocity field given by

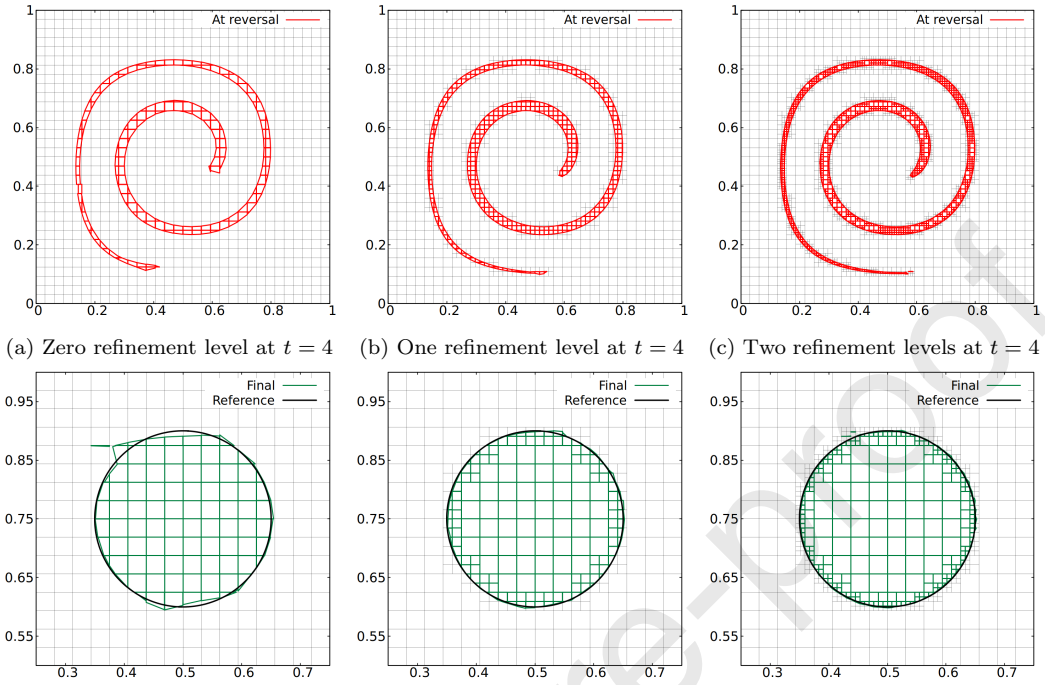
$$\mathbf{u}(x, y, t) = \begin{bmatrix} -\sin^2(\pi x)\sin(2\pi y) \\ \sin^2(\pi y)\sin(2\pi x) \end{bmatrix} \cos(\pi t/T) \tag{16}$$

in which T represents the full period. In most cases, $T = 8$ and in our case, the Courant-Friedrichs-Lewy (CFL) number is 1. In that respect, the number of iterations $n_{it} = 256$ and $\Delta t = \Delta x$. The structure of the deformed interface exhibits filaments which indicates that the filament procedure is activated within our AMR scheme. Fig. 11 highlights the results for a base mesh of 32×32 with 0, 1 and 2 levels of refinement. The maximum deformation at $t = T/2$ is shown as well as the final state at $t = T$. Indeed, during the final state, the symmetric difference error can be used when comparing with the initial reconstruction. It is important to note that during the refinement process, the local CFL number reaches 2 and 4, respectively, for refinement at Level-1 and Level-2. Mass difference and runtime are also explicitly displayed in Table 2.

Table 2: Symmetric difference error, order of convergence, mass difference and runtime for the reversible vortex test case at final reconstruction using a filamentary approach. The pseudo-order of convergence is given in parenthesis.

Refinement level	0	1	2
E_{sym}	3.05×10^{-3}	1.14×10^{-3}	8.93×10^{-4}
Order of convergence	-	1.41(4.25)	0.35(0.77)
Mass difference	3.3×10^{-15}	-6.7×10^{-15}	2.5×10^{-13}
Runtime (s)	15.7	35.2	92.2

The evolution of the number of cells is displayed in Fig. 12. As expected, Level-0 offers a constant number of cells throughout the iterations, while the number of cells for Level-1 and Level-2 increase gradually until the vortex is reversed. Note the small drop in the number of cells in the final iteration before reversal. At this instant in time the magnitude of the velocity field vanishes which limits the error in reconstruction.



(a) Zero refinement level at $t = 4$ (b) One refinement level at $t = 4$ (c) Two refinement levels at $t = 4$

(d) Zero refinement level at $t = 8$ (e) One refinement level at $t = 8$ (f) Two refinement levels at $t = 8$

Figure 11: Reversible vortex test case using $T = 8$ for the base grid 32×32 with zero, one and two levels of refinement. Top row of figures shows the maximum deformation. Bottom row of figures shows the final interface.

5.4.1. Influence of the mass redistribution procedure

In this section the influence of the mass redistribution procedure is examined. In most cases, mass is redistributed uniformly. However, as discussed in Section 4, directly proportional and inversely proportional redistributions are implemented and explored in this paper. Fig. 13 shows the seemingly marginal differences between these approaches in terms of reconstruction. Runtime are also comparable with a uniform distribution. However, in terms of mass conservation, machine precision is not achieved. The main difference lies in the way the redistribution of mass is achieved. While a directly proportional approach seems to be a natural way to follow, the number of iterations necessary to redistribute mass is increased compared to a uniform approach. Similarly, the inversely proportional approach iterates more times without increasing the runtime significantly.

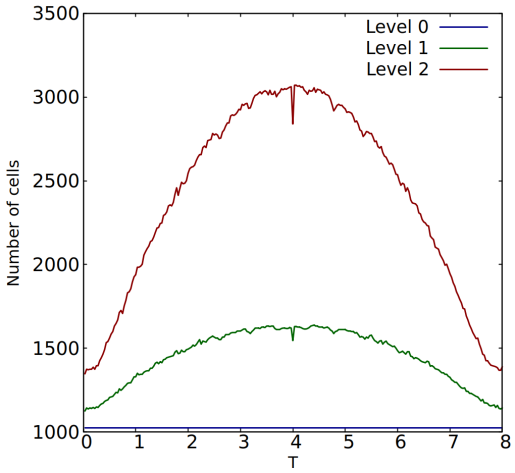
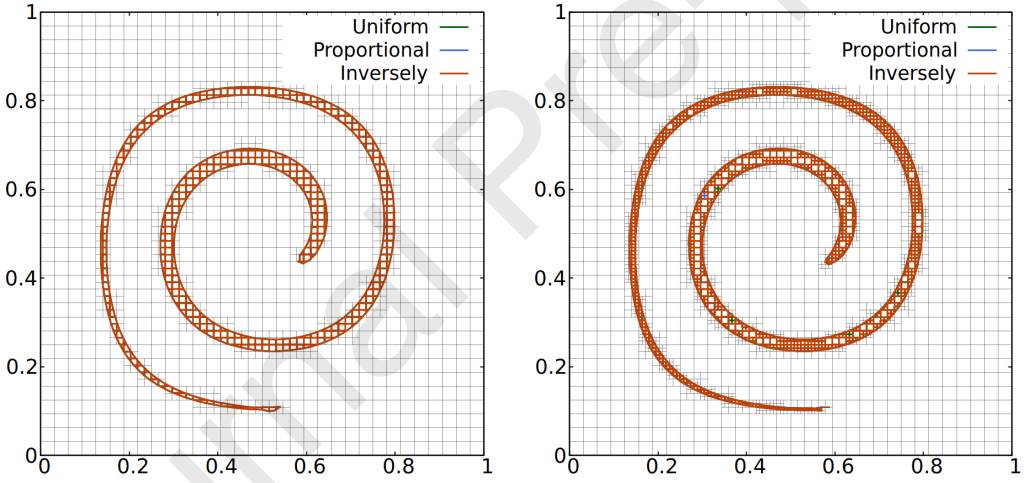


Figure 12: Evolution of the number of cells when using different levels of refinement during the reversible vortex test case.



(a) One level of refinement

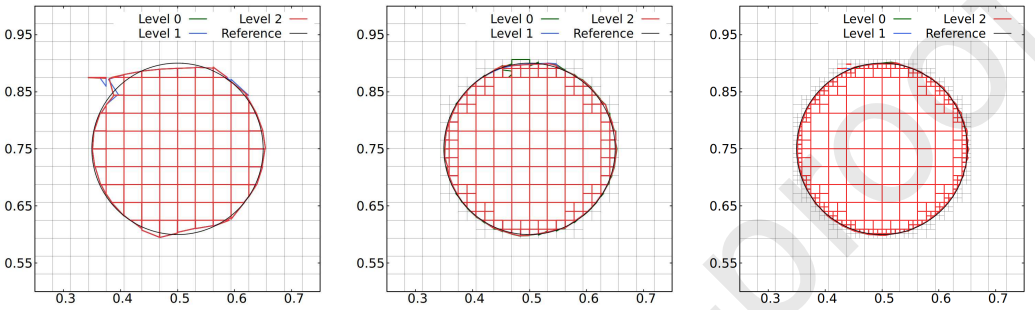
(b) Two levels of refinement

Figure 13: Visual comparison between a uniform, direclty proportional and inversely proportional mass redistribution at maximum deformation.

5.4.2. Influence of the initial refinement

The initial reconstruction is the lower limit of error possible when reconstructing the interface. Indeed, it may differ between the initial reconstruction and the dynamic case. In general, when using a refinement structure, the

initial refinement is the same as the advection process. This is the case for all other cases in this present study. Hence, this section assesses the influence of the initial refinement on the final reconstruction. In the following case, the circle is reconstructed using different levels of refinement at the initial stage, then advected using either level 0, level 1 or level 2. Fig. 14 shows the final reconstruction for different levels of refinement at the initial stage.



(a) Use of Level-0 for advection (b) Use of Level-1 for advection (c) Use of Level-2 for advection

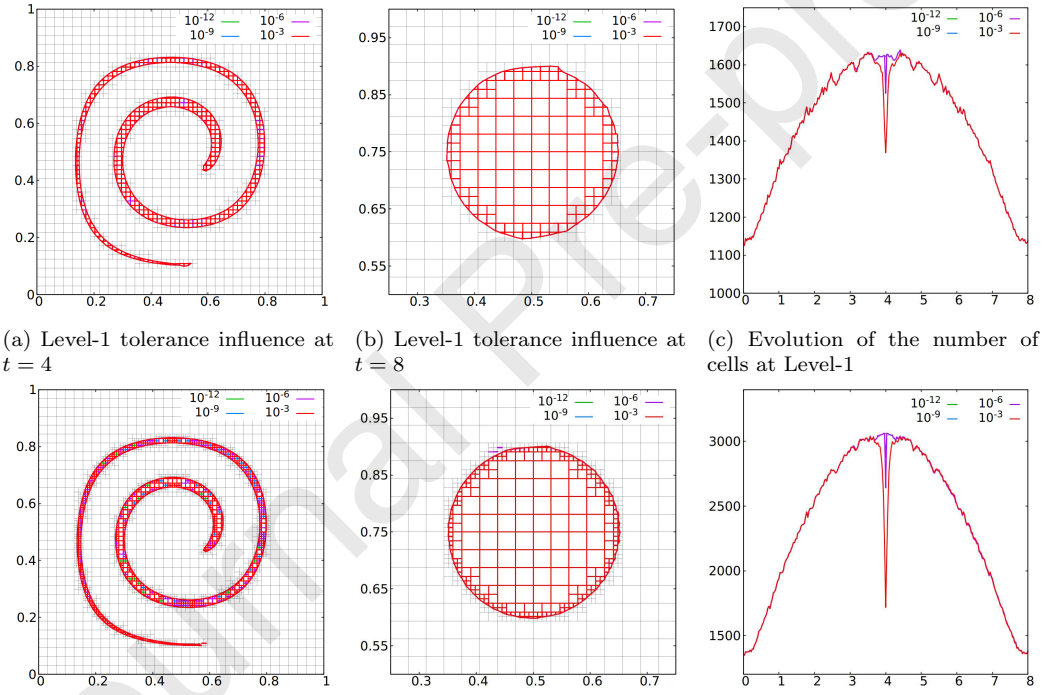
Figure 14: Final reconstruction for the reversible vortex test case using $T = 8$ for the base grid 32×32, one and two levels of refinement. Levels indicate the level of refinement at the initial stage.

A slightly adapted data structure is used to accommodate the correct segmentation. In our code, (i, j, is, js, lev) is the data structure used for adaptive mesh refinement. However, space allocation is performed at the start using the desired maximum refinement level div_max, i.e. is has an allocation of 2^{div_max} and so has js. Note that this allocation would not work if the maximum level was 0 at the initial stage but then 2 during the advection process as the allocation would not be performed. This allows us to use any initial condition in terms of refinement levels as one can see in Fig. 7. Note that the final state is not highly dependent on the initial level of refinement.

5.4.3. Influence of the mesh refinement criteria

As described above the refinement criteria in a MOF framework is the discrepency between the reference and reconstructed centroid. This section discusses the influence of having a finer or coarser criterion. Note that the criterion is non-dimensionalised by the cell size so that it is more meaningful use than using machine precision. A lower tolerance has a great influence on the reconstruction precision, but it also has implications on the number of

cells in the domain and indeed the runtime. In setting a suitable tolerance, one has to consider the trade-off between accuracy and runtime. Runtime is comparable for all test cases and a significant difference is not found. Fig. 15 shows the intermediate and final reconstruction as well as the evolution of the number of cells in the domain. The maximum number of cells is also comparable, however the evolution shows an interesting feature where the last iteration before reversal exhibits a large drop in the number of cells. Indeed, the last iteration corresponds to the $cos(\pi t/T)$ term vanishing, meaning the reconstruction is an almost-static reconstruction. The Level-0 advection will be able to reconstruct more filaments. Combined with a low tolerance, the number of cells in the domain decreases significantly.



(a) Level-1 tolerance influence at $t = 4$

(b) Level-1 tolerance influence at $t = 8$

(c) Evolution of the number of cells at Level-1

(d) Level-2 tolerance influence at $t = 4$

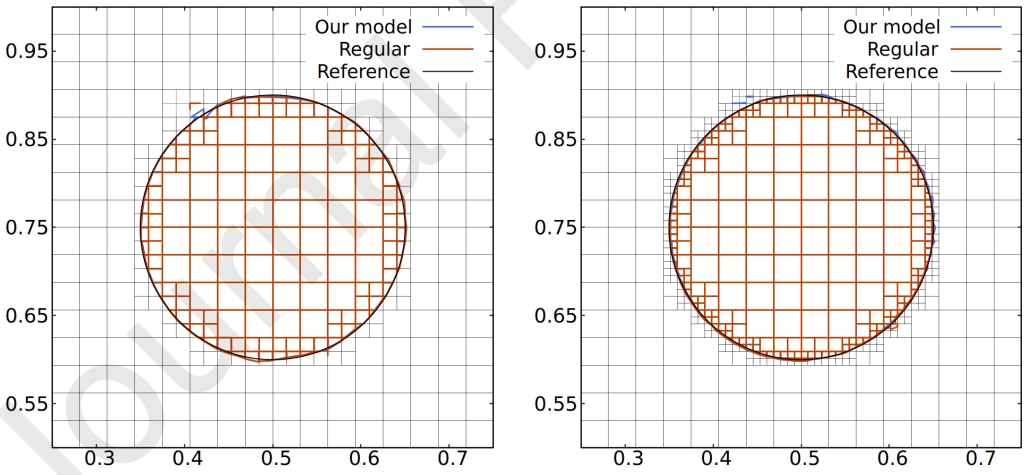
(e) Level-2 tolerance influence at $t = 8$

(f) Evolution of the number of cells at Level-2

Figure 15: Influence of the mesh refinement criterion tolerance on intermediate and final reconstruction and evolution of the number of cells in the domain.

5.4.4. *Influence of the backtrace on interface reconstruction*

The choice of backtrace within a refinement framework can influence results greatly. Indeed, the natural choice is to perform backtracking on the subcell itself, ascribed here as *Regular*. However, there are some advantages and disadvantages which are explained below. On the one hand, the intersection procedure of our approach must intersect the entirety of the desired material at all times. In this regard, our backtracking approach is to use Level-0 as reference and make sure that all refined levels intersect the same area as previous levels. This ensures exact mass conservation. However, the refined backtrace subcells are slightly deformed, which means the reference volume fraction and centroid are somewhat distorted. On the other hand, the regular backtrace creates gaps and overlaps that are very small [34]. This does not guarantee a full intersection of the material, leading to poor conservation of mass. Despite this loss of mass, the interface reconstruction is not distorted which may indicate a smoother interface reconstruction. The correct backtracking consisting of advected hanging nodes may also create non-convex cells, which means more complex algorithms are needed. This approach has been discarded. Fig. 16 emphasises the difference between a regular backtrace and our proposal for one and two levels of refinement.



(a) One level of refinement

(b) Two levels of refinement

Figure 16: Comparison between a regular backtrace and our choice of backtracking a refined subcell at the final state.

Table 3 shows that the error is smaller when using a regular backtrace.

Yet, having a smaller symmetric difference error may not guarantee good mass conservation. In addition, the regular backtrace choice seems to be computationally faster. Indeed, the backtracking procedure is only relevant for the subcell itself, whereas in our model, higher levels need to account for the previous levels of refinement for its backtracking procedure.

Table 3: Comparison between our model and a regular backtrace regarding interface reconstruction.

	Our model	**Regular backtrace**
	Level-1	
E_{sym}	1.14×10^{-3}	7.84×10^{-4}
Mass difference	-6.7×10^{-15}	9.8×10^{-5}
Runtime (s)	35.3	31.0
	Level-2	
E_{sym}	8.93×10^{-4}	6.26×10^{-4}
Mass difference	2.5×10^{-13}	-2.6×10^{-4}
Runtime (s)	92.2	75.2

5.5. Benchmark: Droplet flow

Originally proposed by Ahn and Shashkov [34] and further developed by Jemison et al. [25], the droplet flow test case deforms an initial circle of radius $r = 0.125$ centred in a unit domain using a nonlinear divergence-free velocity field given by

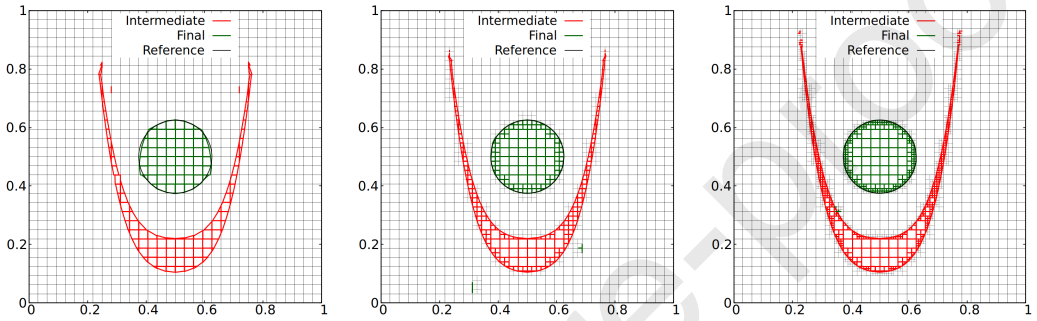
$$\mathbf{u}(x, y, t) = \begin{bmatrix} 0.125(8x - 4) \\ 0.125\left[-(8y - 4) - 4 - (1 - (8x - 4)^2 - (8x - 4)^4)\right] \end{bmatrix} f(t) \quad (17)$$

where

$$f(t) = \begin{cases} 1 & 0 \leq t < T_{max} - t_\epsilon/2 \\ \cos\left(\frac{\pi(t - T_{max} + t_\epsilon/2}{t_\epsilon}\right) & T_{max} - t_\epsilon/2 \leq t \leq T_{max} + t_\epsilon/2 \\ -1 & T_{max} + t_\epsilon/2 < t \leq 2T_{max} \end{cases} \quad (18)$$

represents the amplitude of the velocity field which varies in time so that at time $t = T_{max}$ the initial droplet is recovered to its original position.

Filaments are formed during the advection process. A leading tip is generated, making this case challenging. The base mesh is 32×32, the number of iterations is $n_{it} = 160$ and $\Delta t = 0.01$. Two levels of refinement are tested. Fig. 17 highlights the shape of the intermediate $t = T_{max}$ and final interface using different levels of refinement. All figures show adequate results compared to the original circle. In addition, filaments are well reconstructed except when the tip needs to be reconstructed using refinements. This tends to lead to spurious break ups in the material.



(a) Zero refinement level (b) One refinement level (c) Two refinement levels

Figure 17: Droplet flow test case for a 32×32 base grid with zero, one and two levels of refinement. Figures shows the maximum deformation and the final interface.

Table 4: Symmetric difference error, mass difference and runtime for the droplet flow test case at final reconstruction using a 32×32 base mesh compared to reference solutions.

Refinement level	0	1	2
E_{sym} in [25]	2.48×10^{-3}	6.37×10^{-4}	2.96×10^{-4}
Order of convergence	-	1.96	1.10
Runtime (s)	191.3	529.3	940.4
E_{sym}	1.53×10^{-3}	2.55×10^{-4}	1.90×10^{-4}
Mass difference	-2.82×10^{-4}	-4.78×10^{-7}	-4.92×10^{-16}
Runtime (s)	2.9	7.8	21.2

Table 4 provides the information on the symmetric difference error, mass difference and runtime. Note that the mass difference is not as accurate as expected. Indeed, for the coarser refinement, some material tends to leave the domain near the bottom edge. The Level-1 figure shows that some material

at the final state was advected very near the edge of domain, suggesting that for levels 0 and 1, some has left the domain. This highlights a major drawback of our approach. Indeed, when using a Level-0 advection scheme combined with filaments, material that breaks away from the main material tends to stay detached, or is reconstructed poorly even when using some levels of refinement.

5.6. Benchmark: S-shape

First tested by Ahn and Shashkov [34] and Jemison et al. [25], the S-shape benchmark case is a challenging material deformation test where an initial circle of radius $r = 0.25$ centred in a unit domain is deformed in a nonlinear divergence-free velocity field given by

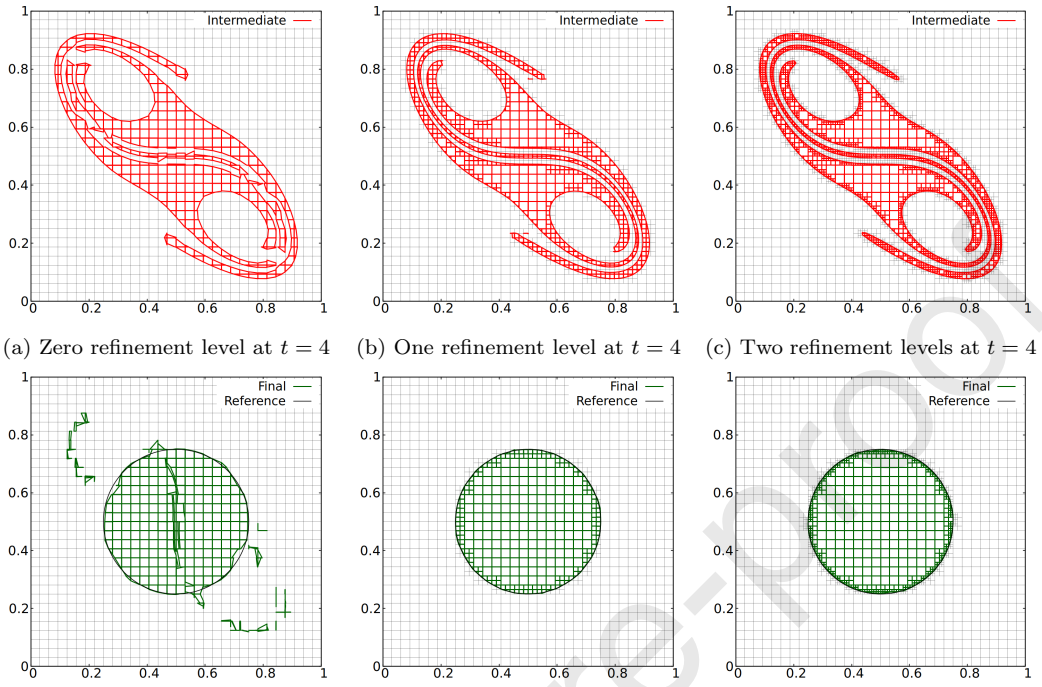
$$\mathbf{u}(x, y, t) = \begin{bmatrix} 0.25[(4x-2) + (4y-2)^3] \\ -0.25[(4y-2) + (4x-2)^3] \end{bmatrix} f(t) \tag{19}$$

where $f(t)$ is given in Eq. (18). In this case, $T_{max} = 4$ and $t_\epsilon = 2$. The total number of iterations for a base mesh 32×32 is $n_{it} = 320$ and $\Delta t = 0.025$.

The deformation creates a highly deformed material creating thin filamentary structures in the centre of the domain. For this benchmark, our filament capable MOF procedure is used. Fig. 18 shows the maximum deformation of the material and its final state. The Level-0 grid shows poor reconstruction because the thin strand of material in the centre of the domain is difficult to reconstruct even with a filament approach using three conglomerates. When more than three conglomerates exist, a standard MOF reconstruction is used which tends to merge materials together [26]. Using one or two levels of refinement exhibits a better reconstructed interface. However, mass conservation is not well maintained for this challenging case due to the reversion of large portion of thin filamentary structures. The significant loss of mass affects the symmetric difference error at Level-2, which is larger than the reconstruction at Level-1.

6. Discussion on the efficiency of MOF-AMR filament capability

Any AMR framework is known to use a reasonable trade-off between accuracy and runtime, refining regions of interest while decreasing the total number of cells used in computation compared to a uniform grid. In general AMR practices, runtime increases with refinement levels while the error



(a) Zero refinement level at $t = 4$ (b) One refinement level at $t = 4$ (c) Two refinement levels at $t = 4$

(d) Zero refinement level at $t = 8$ (e) One refinement level at $t = 8$ (f) Two refinement levels at $t = 8$

Figure 18: S-shape test case for a 32×32 base grid with zero, one and two levels of refinement. Top row of figures shows the maximum deformation. Bottom row of figures shows the final interface.

Table 5: Symmetric difference error, mass difference and runtime for the S-shape test case at final reconstruction compared to Jemison et al. [25].

Refinement level	0	1	2
E_{sym} in [25]	2.11×10^{-2}	1.34×10^{-3}	4.74×10^{-4}
Runtime (s)	157.2	773.1	1871.5
E_{sym}	1.57×10^{-2}	1.11×10^{-3}	1.41×10^{-3}
Mass difference	-3.47×10^{-10}	-2.44×10^{-4}	3.30×10^{-3}
Runtime (s)	35.9	61.7	180.7

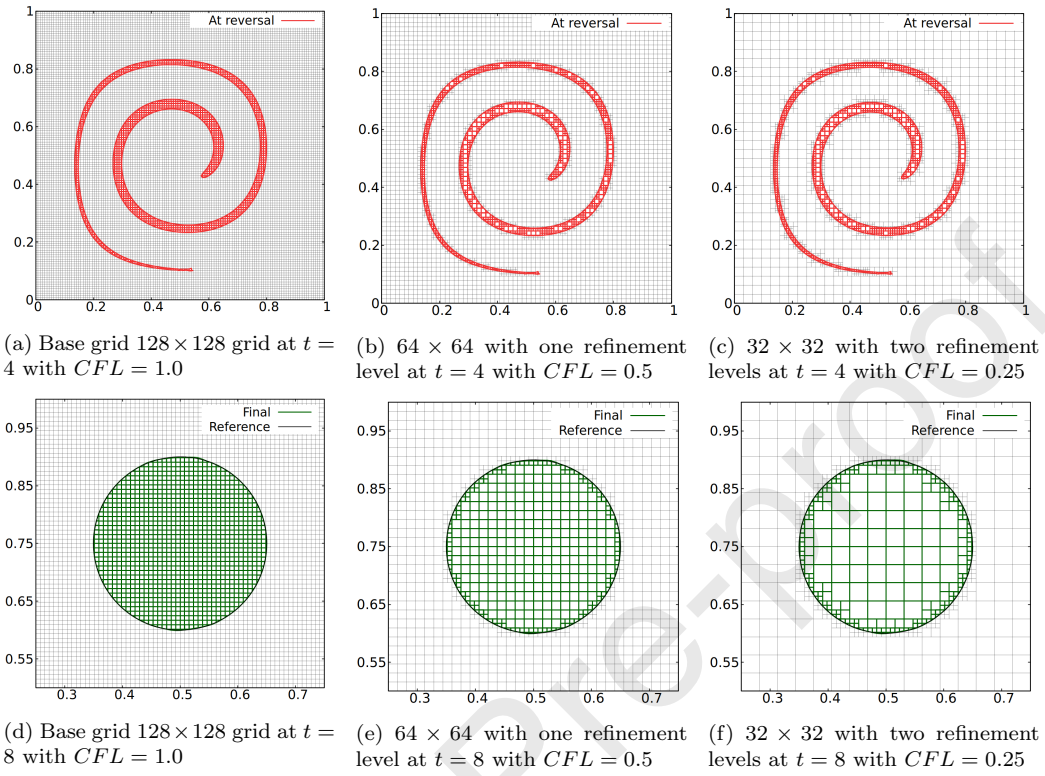
decreases (or the region of interest becomes more accurately defined). However, in the MOF context, regions of high deformation can be reconstructed with ease using filaments while maintaining a reasonable computational cost. In this regard, one can try to compare the efficiency of different levels of a

MOF-AMR filament capable procedure. Indeed, a filament reconstruction with a higher base resolution but with a lower level of refinement may be equivalent to a lower base resolution reconstruction but with a higher level of refinement. This section tries to give an insight into compromising runtime and error for the well-known reversible vortex benchmark. At first we use a constant unity CFL number on the base mesh, meaning the local CFL number for refined grids is 2 and 4, respectively, for Level-1 and Level-2. Secondly, we consider an effective CFL number for the finest resolution meaning that the number of iterations is constant for all three configurations. The base mesh CFL number for one level of refinement is 0.5 and for two levels of refinement 0.25.

Table 6: Efficiency table testing three different grids with the same maximum level of refinement. BM 128 relates to Base Mesh and its resolution. CFL numbers are expressed for the base mesh.

	BM 128 Level-0	**BM 64 Level-1**	**BM 32 Level-2**
CFL	1.0	1.0	1.0
E_{sym}	1.56×10^{-4}	2.36×10^{-4}	8.93×10^{-4}
Max number of cells	16384	5530	3061
Number of iterations	1024	512	256
Runtime (s)	115.3	81.2	92.9
CFL	1.0	0.5	0.25
E_{sym}	1.56×10^{-4}	1.65×10^{-4}	2.42×10^{-4}
Runtime (s)	115.3	165.5	296.9

One can see from Table 6 that with a constant CFL number, runtime is better for one level of refinement, which is also better than two levels of refinement. This is due to the fact that Level-1 has to be reconstructed first. In addition, the number of cells used is very small compared to a uniform mesh even with the highest refinement levels. When using the same effective CFL number, i.e. equivalent at the finest resolution, runtime increases significantly with the increased number of iterations. Similarly, the symmetric difference error increases. Fig. 19 shows the improved final reconstruction. Comparison of performance on a fine uniform grid and a grid using one level of refinement, both using filament capable methods, shows that there is a significant improvement in runtime and number of cells for the latter while the error is very similar in both cases. This solution may be a more desirable



(a) Base grid 128×128 grid at $t = 4$ with $CFL = 1.0$

(b) 64×64 with one refinement level at $t = 4$ with $CFL = 0.5$

(c) 32×32 with two refinement levels at $t = 4$ with $CFL = 0.25$

(d) Base grid 128×128 grid at $t = 8$ with $CFL = 1.0$

(e) 64×64 with one refinement level at $t = 8$ with $CFL = 0.5$

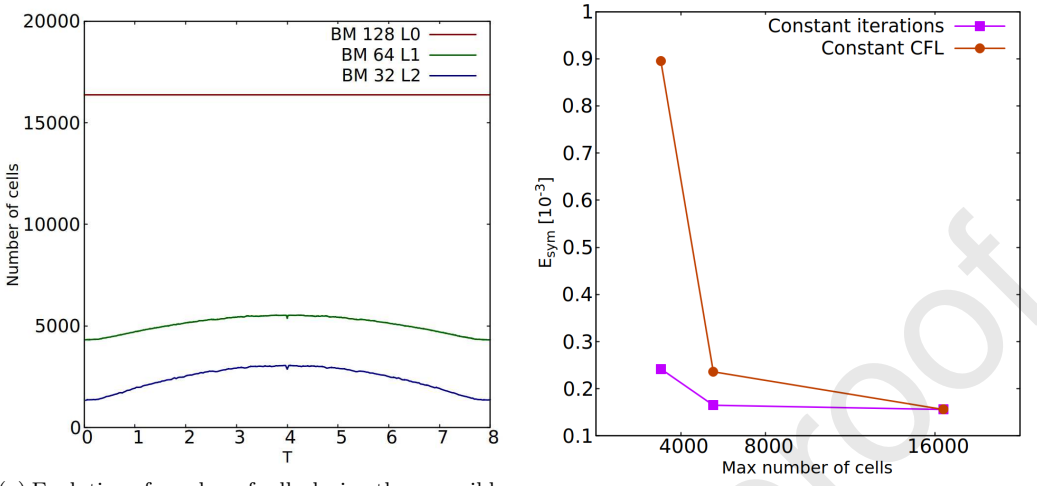
(f) 32×32 with two refinement levels at $t = 8$ with $CFL = 0.25$

Figure 19: Visual results of the efficiency test of the MOF-AMR filament capable procedure using different CFL numbers therefore a constant number of iterations.

option. Fig. 20 shows a significantly smaller number of cells used. In addition, much better reconstruction is achieved with a smaller CFL number. The influence of high CFL numbers (> 2) on interface accuracy has not been demonstrated.

7. Conclusions

In this paper, a new quadtree-based adaptive MOF method has been presented where filament structures are resolved using a symmetric multi-material approach on a refined grid. A simplified quadtree structure has been implemented with logical connection between parent and children cells up to two levels of refinement. A Lagrangian backtracking approach for refined grids is proposed that enables exact material intersection during the advection process, hence ensuring mass conservation. The refinement crite-



(a) Evolution of number of cells during the reversible vortex test case

(b) Symmetric difference error as a function of max number of cells

Figure 20: Number of cells and symmetric difference error for the efficiency test.

rion is based on the centroid defect relative to the cell or subcell size, ensuring linear interfaces are reconstructed exactly without the need for refinement. As a result, the proposed framework achieved good results in terms of accuracy and runtime while using computational resources in a more efficient manner. Comparison between different levels of refinement for the same minimum cell size provides insight into the most efficient use of this framework and the MOF method in general.

This MOF-AMR method is tested on several benchmark problems with high material deformation. All of these benchmark problems are compared with a couple of similar MOF approaches using refinement. First, the Zalesak slotted disc shows less refined cells at the initial stage and achieved good qualitative results. Other benchmarks such as the reversible vortex show highly precise reconstruction at maximum deformation under different levels of refinement. The droplet flow and the S-shape test case yielding highly deformed structures are presented with filament reconstruction. Qualitatively, results are comparable to other MOF methods. The limitation of our method lies in the number of refinement levels available in an unconstrained adaptive grid structure. Our refinement approach differs from other MOF-AMR reference methods [25, 34] but shows acceptable results. Machine precision mass conservation algorithms are achieved for benchmark problems such as the re-

versible vortex, whilst further improvements are required for other problems such as the droplet flow or the S-shape case. Furthermore, runtime has been significantly decreased compared to previous methods. In this study, no high-performance libraries are used and calculations are carried out on a single core. High-performance frameworks would offer strong scalability and efficient algorithms for handling large parallel octree operations [37, 41, 42, 43]. Yet, complexity and potential resource requirements may be challenging. In comparison, our data structure offers ease of use and accessibility, suitable for smaller-scale efforts. Many advantages follow from this decision such as the absence of load balancing, numbering, and neighbouring search. However, the authors are aware of potential issues related to limited scalability and versatility, memory access, parent node data optimisation and general computing performance.

The present MOF-AMR method tends to decrease the cross-stream diffusion of advected material and can reconstruct sharp edges or tips of filaments with greater accuracy using up to two levels of refinement. Further improvement may be made to address these numerical issues by using the recent new moment-of-fluid method [44] or the parabolic interface reconstruction [45]. In addition, the number of conglomerates has less influence on the centroid defect as these scenarios tend to trigger refinement. In future work we would like to advect and reconstruct several materials within the same domain which will most likely involve reconstructing more than three materials. In this AMR framework, optimising the levels of refinement could be of interest to reduce the computational cost by using prediction algorithms. Coupling our MOF-AMR framework with a fluid flow solver is our next aim targeting complex multiphase flow problems, which can potentially reduce the computational cost without sacrificing accuracy.

Acknowledgments

Constructive comments from anonymous reviewers have helped to improve the manuscript and these are gratefully acknowledged. The first author would like to thank the United Kingdom Engineering and Physical Sciences Research Council (EPSRC) for providing the funding to support his doctoral study (EP/T517951/1 with project reference 2558593). Z.X. was financially supported by EPSRC grant (EP/V040235/1), the Royal Society Newton Advanced Fellowship (NAF/R1/201156) and International Exchanges Award (IEC/NSFC/211143, IES/R2/202095).

References

[1] X. Zheng, J. Lowengrub, A. Anderson, V. Cristini, Adaptive unstructured volume remeshing–ii: Application to two-and three-dimensional level-set simulations of multiphase flow, Journal of Computational Physics 208 (2) (2005) 626–650.

[2] Z. Xie, D. Pavlidis, P. Salinas, J. R. Percival, C. C. Pain, O. K. Matar, A balanced-force control volume finite element method for interfacial flows with surface tension using adaptive anisotropic unstructured meshes, Computers & Fluids 138 (2016) 38–50.

[3] C. Cantwell, D. Moxey, A. Comerford, A. Bolis, G. Rocco, G. Mengaldo, D. De Grazia, S. Yakovlev, J.-E. Lombard, D. Ekelschot, B. Jordi, H. Xu, Y. Mohamied, C. Eskilsson, B. Nelson, P. Vos, C. Biotto, R. Kirby, S. Sherwin, Nektar++: An open-source spectral/hp element framework, Computer Physics Communications 192 (2015) 205–219.

[4] M. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, Journal of Computational Physics 53 (3) (1984) 484–512.

[5] 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 (1) (1999) 81–124.

[6] H. D. Ceniceros, A. M. Roma, A. Silveira-Neto, M. M. Villar, et al., A robust, fully adaptive hybrid level-set/front-tracking method for two-phase flows with an accurate surface tension computation, Communications in Computational Physics 8 (1) (2010) 51–94.

[7] S. Popinet, An accurate adaptive solver for surface-tension-driven interfacial flows, Journal of Computational Physics 228 (16) (2009) 5838–5866.

[8] D. Hartmann, M. Meinke, W. Schröder, A strictly conservative cartesian cut-cell method for compressible viscous flows on adaptive grids, Computer Methods in Applied Mechanics and Engineering 200 (9-12) (2011) 1038–1052.

[9] Q. Liang, A simplified adaptive Cartesian grid system for solving the 2D shallow water equations, International Journal for Numerical Methods in Fluids 69 (2) (2012) 442–458.

[10] X. Chen, V. Yang, Thickness-based adaptive mesh refinement methods for multi-phase flow simulations with thin regions, Journal of Computational Physics 269 (2014) 22–39.

[11] M. Mirzadeh, A. Guittet, C. Burstedde, F. Gibou, Parallel level-set methods on adaptive tree-based grids, Journal of Computational Physics 322 (2016) 345–364.

[12] F. H. Harlow, J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Physics of Fluids 8 (12) (1965) 2182–2189.

[13] S. O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, Journal of Computational Physics 100 (1) (1992) 25–37.

[14] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, Journal of Computational Physics 79 (1) (1988) 12–49.

[15] J. A. Sethian, P. Smereka, Level set methods for fluid interfaces, Annual Review of Fluid Mechanics 35 (1) (2003) 341–372.

[16] C. W. Hirt, B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, Journal of Computational Physics 39 (1) (1981) 201–225.

[17] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annual Review of Fluid Mechanics 31 (1) (1999) 567–603.

[18] J. J. Monaghan, Smoothed particle hydrodynamics, Annual Review of Astronomy and Astrophysics 30 (1) (1992) 543–574.

[19] V. Dyadechko, M. Shashkov, Moment-of-fluid interface reconstruction, Los Alamos Report LA-UR-05-7571 (2005) 49.

[20] A. Lemoine, S. Glockner, J. Breil, Moment-of-fluid analytic reconstruction on 2D Cartesian grids, Journal of Computational Physics 328 (2017) 131–139.

[21] A. A. Mukundan, T. Ménard, J. C. B. de Motta, A. Berlemont, A hybrid moment of fluid–level set framework for simulating primary atomization, Journal of Computational Physics 451 (2022) 110864.

[22] R. N. Hill, M. Shashkov, The symmetric moment-of-fluid interface reconstruction algorithm, Journal of Computational Physics 249 (2013) 180–184.

[23] H. T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, Journal of Computational Physics 226 (2) (2007) 2096–2132.

[24] V. Dyadechko, M. Shashkov, Reconstruction of multi-material interfaces from moment data, Journal of Computational Physics 227 (11) (2008) 5361–5384.

[25] M. Jemison, M. Sussman, M. Shashkov, Filament capturing with the multimaterial moment-of-fluid method, Journal of Computational Physics 285 (2015) 149–172.

[26] P. Hergibo, T. N. Phillips, Z. Xie, A moment-of-fluid method for resolving filamentary structures using a symmetric multi-material approach, Journal of Computational Physics 491 (2023) 112401.

[27] Y. Zeng, H. Liu, Q. Gao, A. Almgren, A. P. S. Bhalla, L. Shen, A consistent adaptive level set framework for incompressible two-phase flows with high density ratios and high Reynolds numbers, Journal of Computational Physics 478 (2023) 111971.

[28] O. Antepara, N. Balcázar, A. Oliva, Tetrahedral adaptive mesh refinement for two-phase flows using conservative level-set method, International Journal for Numerical Methods in Fluids 93 (2) (2021) 481–503.

[29] L. Fu, S. Litvinov, X. Y. Hu, N. A. Adams, A novel partitioning method for block-structured adaptive meshes, Journal of Computational Physics 341 (2017) 447–473.

[30] 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 (2) (2001) 302–335.

[31] D. Greaves, A quadtree adaptive method for simulating fluid flows with moving interfaces, Journal of Computational Physics 194 (1) (2004) 35–56.

[32] M. Malik, E. S.-C. Fan, M. Bussmann, Adaptive VOF with curvature-based refinement, International Journal for Numerical Methods in Fluids 55 (7) (2007) 693–712.

[33] J. Wang, A. Borthwick, R. E. Taylor, Finite-volume-type VOF method on dynamically adaptive quadtree grids, International Journal for Numerical Methods in Fluids 45 (5) (2004) 485–508.

[34] H. T. Ahn, M. Shashkov, Adaptive moment-of-fluid method, Journal of Computational Physics 228 (8) (2009) 2792–2821.

[35] A. Islam, M. Sussman, H. Hu, Y. Lian, Simulation of drop impact on substrate with micro-wells, Physics of Fluids 34 (6) (2022) 062108.

[36] M. Shashkov, B. Wendroff, The repair paradigm and application to conservation laws, Journal of Computational Physics 198 (1) (2004) 265–277.

[37] K. Saurabh, M. Ishii, M. A. Khanwale, H. Sundar, B. Ganapathysubramanian, Scalable adaptive algorithms for next-generation multiphase flow simulations, in: 2023 IEEE International Parallel and Distributed Processing Symposium (IPDPS), IEEE, 2023, pp. 590–601.

[38] C. Harrison, J. Weiler, R. Bleile, K. Gaither, H. Childs, A distributed-memory algorithm for connected components labeling of simulation data, in: Topological and Statistical Methods for Complex Data: Tackling Large-Scale, High-Dimensional, and Multivariate Data Spaces, Springer, 2015, pp. 3–19.

[39] S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, Journal of computational physics 31 (3) (1979) 335–362.

[40] W. J. Rider, D. B. Kothe, Reconstructing volume tracking, Journal of Computational Physics 141 (2) (1998) 112–152.

[41] C. Burstedde, L. C. Wilcox, O. Ghattas, p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees, SIAM Journal on Scientific Computing 33 (3) (2011) 1103–1133.

[42] M. Ishii, M. Fernando, K. Saurabh, B. Khara, B. Ganapathysubramanian, H. Sundar, Solving pdes in space-time: 4d tree-based adaptivity, mesh-free and matrix-free approaches, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, 2019, pp. 1–61.

[43] H. Sundar, R. S. Sampath, G. Biros, Bottom-up construction and 2: 1 balance refinement of linear octrees in parallel, SIAM Journal on Scientific Computing 30 (5) (2008) 2675–2708.

[44] M. Shashkov, E. Kikinzon, Moments-based interface reconstruction, remap and advection, Journal of Computational Physics 479 (2023) 111998.

[45] R. A. Remmerswaal, A. E. Veldman, Parabolic interface reconstruction for 2d volume of fluid methods, Journal of Computational Physics 469 (2022) 111473.

Declaration of interests

√ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

P.H.: Methodology, Software, Data curation, Validation, Investigation, Visualization, Writing- Original draft, Writing- Reviewing and Editing.
Q.L.: Methodology, Software, Investigation, Writing- Reviewing and Editing.
T.N.P.: Conceptualization, Investigation, Writing- Reviewing and Editing, Supervision.
Z.X.: Conceptualization, Methodology, Validation, Investigation, Writing- Reviewing and Editing, Supervision, Funding acquisition.