Delft University of Technology

Faculty EEMCS

Master Thesis Applied Mathematics

Specialisation Computational Science and Engineering

An innately mass conserving interface capturing method for the modelling of interface advection on unstructured grids

D.J.B.V. Pols

Supervisor

Dr.ir. D.R. van der Heul

Defence Date

24 January 2019

Summary

In solving two-phase flows, the location of the interface between the phases is necessary to handle interface jump conditions when solving the Navier-Stokes equation. Current interface capturing and advection methods, however, suffer from various issues. The level set method uses the signed distance to the interface and the interface being the zero level set of this function allows the evolution of the level set field to be described with a simple advection equation. This means that no additional steps are required, but solving the advection equation generally does not conserve mass. On the other hand, Volume of Fluid methods utilise the local fluid fractions to represent the phase interface. For these methods, a mass conserving advection algorithm exists [1], but the absence of an explicit interface requires expensive reconstruction methods to be used instead. Additionally, this Volume of Fluid advection method is subject to a restrictive CFL condition on the time-step and, being dimensionally split, requires a structured grid to be used. Other Volume of Fluid or Moment of Fluid advection methods that do not have these conditions are not mass conserving. Dual interface methods that combine information from the level set and volume fractions are able to achieve higher accuracy, but these method still use Volume of Fluid advection to remain mass conserving and are thus subject to the same conditions. These methods use the level set field to create a cheaper and better justified reconstruction method, which negates one drawback of VoF methods.

In this thesis, a method is formulated to allow interface advection on unstructured grids without such a strict CFL condition compared to the MCLS method. To do this, the finite volume level set advection method of the MCLS method is replaced with a nodal-modal discontinuous Galerkin method [2]. The DG method is analogous to the Galerkin finite element methods, but the basis functions are now only valid on one element, and the solution can be multiply defined on the cell boundaries. This allows for level set advection on polygonal cells, and has the added benefit that since cells are semi-independent, level set correction now only has to be applied locally. This, like the finite volume level set advection method, is not necessarily mass conserving though, so the volume of fluid will need to solve this issue. The difference with MCLS is that no volume of fluid advection method is used, since these methods are the parts that introduce the aforementioned problems that are being avoided. Instead, a minimisation method is done on the VOF field in order to obtain a correction which can be applied to the level set and keep it mass conserving. For this minimisation method, a flux condition is used to impose additional constraints on the solution. This condition effectively attempts to match the interface intersections on a cell edge for both adjacent cells, for all cell edges that have a flux going through.

The resulting method is tested for four test cases, with solid body translation and rotation, the corner flow test, and the vortex deformation test. In the translation test, there is only vertical velocity, so the flux condition does not take vertical edges into account. To stabilise this test case, level set reinitialisation is applied, which is a technique used to maintain the signed distance property of the level set function to use in the Navier-Stokes evolution method. This keeps the solution in the translation test somewhat representative of the exact interface, but makes the obtained results no longer representative of the actual method. For the other test cases, the method appears to converge when the grid is refined, except for the solid body rotation test using Euler Forward, as this result becomes unstable. However, no apparent order of convergence is present, and the effect of a more accurate time integration method is very inconsistent. To solve this, the method is altered to allow multiple level set advection steps for every optimisation routine. This makes the method better follow the order of convergence of the discontinuous Galerkin method, although this order is slightly lower for the vortex deformation test. The obtained results show that in it's current form, this optimisation based method is likely not usable for general applications, but that with additional work, this kind of method may prove to be more applicable than MCLS.

Contents

1 Introduction

In various application of science and industry, the modelling of two-phase flows is required for the simulation of various systems. Examples of cases where accurate numerical solutions are needed are the flows of gas-fluid mixtures in pipelines [3], microscale ink jetting [4], and bubble and shockwave dynamics [5]. At it's base, the dynamics of the two-phase flow field are governed by the Navier-Stokes equations. However, solving these equations requires knowledge of the separation of the two phases since the various variables, such as pressure and density, do not have to be the same on both sides of the interface that separates the two phases. As such, interface advection methods have been created that model the movement of the phase interface in a velocity field, which can then be fed back into the Navier-Stokes equation to evolve the velocity field by one time step.

These interface advection methods can roughly be divided into two categories, namely *interface tracking* methods and *interface capturing* methods. Tracking methods attempt to explicitly represent the interface, which can be done in two dimensions by chains of line segments, and shift this representation according to the local flow velocity. This principle for interface advection can work in many cases, but trouble occurs if the method has to deal with changing topologies. When an interface splits into two, or when two interfaces merge, the chains of segments need to be reworked and ordered correctly. The logic behind this can quickly become too complicated and can result in large errors if not done correctly. Additionally, the checks that are required to detect such intersections of line segments are not cheap. As a result, this type of method is only applicable if the nature and location of such a topology change is anticipated, and can also not be trivially extended to three dimensions. Interface capturing methods, on the other hand, define the phases according to the regions they occupy, and move these regions in time. This way, the interface is implicitly represented by such a method, and topology changes should automatically be accounted for. This makes interface capturing methods more generally applicable.

There are already many interface capturing methods that are able to accurately find the location of the interface [1, 3, 4, 6, 7, 8, 9, 10], these methods suffer from certain drawbacks that ultimately inhibit general application. In incompressible flow fields, the total mass of the fluid should be conserved, but for these methods this does not have to be the case, requiring arbitrary mass correction steps to ensure that no mass loss or gain is present (e.g. [6, 7]). The problem with the mass conserving methods is that they ultimately have to rely on the concept of donating regions, which imposes a strict CFL condition on the time step and requires dimensional splitting to remain mass conserving. Methods that are not based on donating regions require arbitrary correction steps to ensure mass conservation. The CFL condition imposed by the donating regions has as an effect that for a formulation of the equation in cylindrical coordinates, as the interface approaches the radial pole, the time step must be set to zero for the velocities through the pole and oscillations are introduced in the phase interface if this is not satisfied [3]. In order to avoid these problems, this thesis aims to work towards a method that is innately mass conserving but is not dependent on donating regions via the use of an optimisation problem. The idea behind this step is supported by the results from [13, 14], where an optimisation method has already been applied to a pure level set advection method. In section 2, the existing MCLS method is described in more detail. Then in 3, the layout of the new method is given and the various building blocks are specified. The analysis in section 4 shows that the resulting method will indeed be generally applicable in three dimensions and on unstructured grids, and finally the performance is assessed and conclusions are drawn in section 5.

2 Theory

2.1 Level set method

The *level set* (LS) interface capturing method [7] uses a marker function ϕ defined on the domain Ω that changes sign at the interface. The sign of ϕ defines the fluid present at that location, so the interface is located where ϕ equals 0:

$$\Gamma(t) = \{\boldsymbol{x} \in \Omega : \ \phi(\boldsymbol{x}, t) = 0\}. \tag{2.1}$$

The interface is evolved in the flow field $\boldsymbol{u}$ by applying the advection equation to the level set function as if it were a material property,

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\boldsymbol{u}\phi) = 0. \tag{2.2}$$

Since the level set function is typically smooth near the interface, accurate discretisation can be obtained with a simple scheme. The level set function is usually chosen to be the signed distance to the nearest interface surface. The signed distance property is useful for solving the two phase Navier-Stokes equations, as it allows for easy extracting of certain geometrical properties like the unit normal $\boldsymbol{n}$ to the interface and the curvature κ of the interface:

$$\boldsymbol{n} = \frac{\nabla \phi}{|\nabla \phi|}, \qquad \kappa = \nabla \cdot \boldsymbol{n} = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}, \tag{2.3}$$

where $|\nabla \phi| = 1$ everywhere near the interface if ϕ is a signed distance function. In general this property is not conserved when advecting the interface, and would require so-called level set reinitialisation steps to maintain the usefulness of this initial choice [3, 7].

The major drawback of the level set method is that while the level set function itself is conserved, the area enclosed by the interface does not have to be. While higher order discretisation schemes improve mass conservation, the loss of mass is not strictly a result of numerical errors, but is caused by the advection equation not enforcing mass conservation.

2.2 Volume of fluid method

The *volume of fluid* (VoF) method [1, 10] is an interface capturing method that differentiates between the materials using a colour function

$$\chi(\boldsymbol{x}, t) = H(\phi(\boldsymbol{x}, t)), \tag{2.4}$$

where $H(x)$ is the Heaviside step function with $H(0) = 1/2$. This defines the interface as the set of all locations where $\chi = 1/2$. For numerical modelling, the cell averaged value of χ is used instead, representing the fraction of the cell that is occupied by fluid. This defines the discrete volume of fluid function ψ_j in cell Ω_j:

$$\psi_j = \frac{1}{|\Omega_j|} \int_{\Omega_j} \chi(\boldsymbol{x}) \ d\boldsymbol{x}, \qquad |\Omega_j| = \int_{\Omega_j} d\boldsymbol{x}. \tag{2.5}$$

Since ψ is the cell-averaged version of χ and thus is a discrete function, the advection equation is not immediately applicable to ψ. Instead, the transport equation can be written as [1]:

$$\frac{\partial \psi_j}{\partial t} + F_{j,\text{net}} = \oint_{\Omega_j} \chi \nabla \cdot \boldsymbol{u} \ d\boldsymbol{x}, \tag{2.6}$$

where F_j is the net flux of the tracked fluid out of the cell Ω_j.

Since only divergence-free flows are considered, the boundary integral reduces to zero, and the total amount of ψ is thus conserved in the system if none crosses the domain boundary. This equation thus explicitly enforces mass conservation. However, χ is a step function, and the discrete ψ does not have derivatives, and thus cannot approximate this discontinuity. When applying a finite volume scheme directly to ψ, this results in the interface being smeared out over multiple layers of cells even when a simple shift in interface is expected (see figure 1). Additionally, the discontinuity at the interface in χ moves with the flow velocity, so the interface represents a linearly degenerate wave. When trying to apply a finite volume method directly to χ the method would thus have at most linear convergence [11]. To avoid being restricted to linear convergence, the discontinuity in χ must be taken into account when applying an evolution method to ψ.

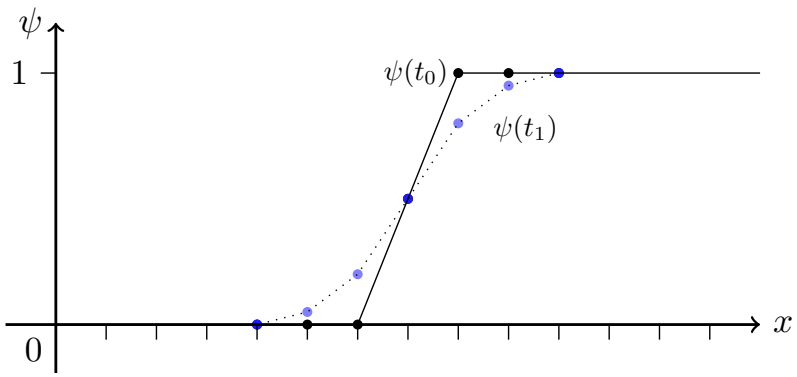


Figure 1: The effect of direct application of a FVM advection scheme on the volume of fluid function ψ. The discrete values of ψ are shown for the initial case by black circles, and after a few time steps by blue circles. In the initial case only one interface cell is present, while after a few time steps more interface cells have been created. This shows that the interface loses definition as it is now smeared out over different layers. This behaviour is caused by ignoring the discontinuity of χ when evolving the volume of fluid field. [7]

Instead of directly applying a finite volume scheme, advection methods for the volume of fluid explicitly define the regions inside the cells where fluid is present, and update the fractional volumes based on this information. Inside each cell, the interface is assumed to be linear, requiring only it's orientation to be approximated. In [1] this orientation is determined using data in neighbouring cells, and straight interfaces are perfectly reconstructed. This method appears to approximate the non-existent derivatives of ψ so it's use is not clearly justified, though the numerical results do indicate that it works. After the interface reconstruction steps, the fluid regions are evolved in time. The only mass conserving algorithm for the advection step is the dimensionally split method from [1]. At each cell boundary in this method, donating regions are constructed of length $u\Delta t$, defining the region that exits or enters a cell. The volume of fluid inside a cell is then updated by the net amount of fluid that enters that cell, with an additional correction accounting for the divergence of the 1D velocity field. Figure 2 shows the specific donating regions, and also illustrates why dimensional splitting is used for this kind of method.

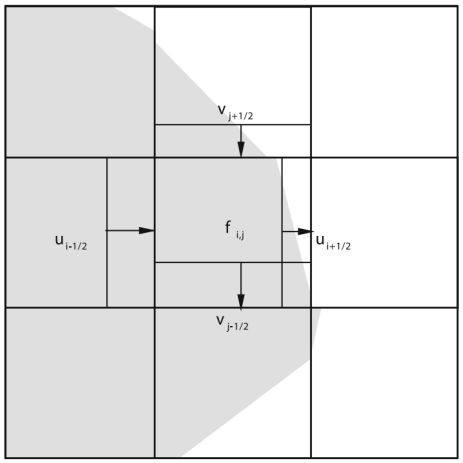


Figure 2: Illustration of the donating regions used for the horizontal and vertical sweep of the MCLS and VoF methods. Note that these donating regions cannot be used in a dimensionally unsplit advection method. The fluid in the lower right corner of the central cell is duplicated and advected both downwards and to the right, and fluid from the top right cell doesn't enter the central cell even though it should. [1]

To ensure mass conservation, a Courant restriction is required on the time step [1], so

$$\Delta t \sum_{d=1}^{\mathcal{N}} \left| \frac{u_d}{\Delta x_d} \right| < \frac{1}{2} \tag{2.7}$$

where $\mathcal{N}$ indicates the dimensionality of the problem. This restriction requires that the donating regions cover at most half a cell, thus ensuring that no overlap between donating regions is possible.

Other advection methods may rely on the Lagrangian pre- or post-image of a cell to evolve the volume fractions [6], as it is difficult to use a donating regions method on anything except Cartesian grids. Since the edges of the pre-image do not have to be straight in nonlinear velocity fields, methods that do not account for this are not innately mass conserving and require mass correction steps to arbitrarily distribute the change in mass among the layer of interface cells. This effect is illustrated in figure 3. While it is possible for these methods to have mass conservation [12], the additional work required to detect the curvature of the edges makes these methods infeasible for larger problems, especially when moving towards three-dimensional problems.

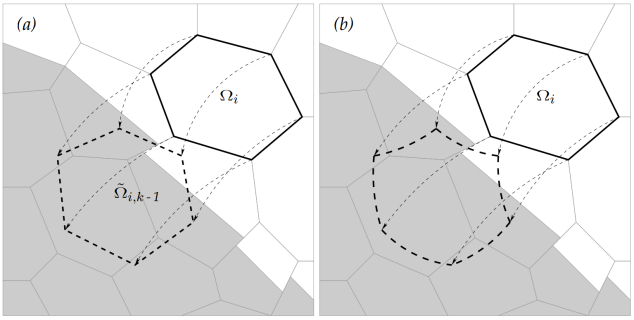


Figure 3: Illustration of the Lagrangian pre-image method. This method relies on advecting the vertices of a cell backwards in time to see which volume flows into this cell during one time step. The amount of fluid in the pre-image then defines the volume of fluid in the cell at the next iteration. If only the vertices are used for this pre-image, then this image will have straight edges (a). In nonlinear flows the edges do not have to remain straight (b), which may induce mass errors. The curvature of the edges can be estimated, but this would require advecting many intermediate points, which makes this method infeasible. [6]

2.3 Dual interface method

In the interface reconstruction step of the volume of fluid method, interface data was extracted from regional data. This has as a consequence that linear interfaces are only exactly reconstructed in a cell if it is linear in the entire 3×3 grid around this cell. Additionally, the reconstruction method seems to require approximations of the (undefined) derivatives of ψ, which cannot be justified as these approximations would not converge. To circumvent these issues, one can utilise the fluid centroids in each cell to improve reconstruction (Moment of Fluid, see [6]) or can use the local gradient of a level set field. The latter option results in *dual interface methods* such as the MCLS method [3, 7], whose steps are illustrated in figure 4.

In the MLCS method the level set and volume of fluid methods are advected according to the scheme described in [1], using the level set to provide the interface orientations and the volume of fluid to keep the level set mass conserving. Using an approximation $\nabla\phi$ of the gradient of the level set, the interface intersection points can be determined, and thus a volume of fluid field can be created from the corresponding level set. This defines a cheap to evaluate function f such that in each cell Ω_i,

$$\psi_i = f(\phi(\boldsymbol{x}_i), \nabla\phi(\boldsymbol{x}_i)),$$

where $\boldsymbol{x}_i$ denotes the cell centre of the cell Ω_i. Note that, unlike the derivatives of ψ, the gradient of ϕ exists at least in a neighbourhood of the interface, so these interface orientations are justified.

Finding the inverse relation is not trivial, but since f is cheap, the inverse function g such that

$$g(f(\phi(\boldsymbol{x}_i), \nabla\phi(\boldsymbol{x}_i)), \nabla\phi(\boldsymbol{x}_i)) = \phi(\boldsymbol{x}_i) \tag{2.8}$$

can be found iteratively, e.g. using a nonlinear solver. With these two functions f and g, from the advected volume of fluid field a corresponding level set function can be created using the advected level set as starting point for the process. If the level set matches the volume of fluid within reason, e.g.

$$||\psi_i^{n+\frac{1}{2}} - f(\widetilde{\phi}(\boldsymbol{x}_i), \nabla\widetilde{\phi}(\boldsymbol{x}_i))|| < \varepsilon \tag{2.9}$$

in each cell, then the corrected level set $\widetilde{\phi}$ is taken to be $\phi^{n+\frac{1}{2}}$ and used for the next time step. Since this field matches the volume of fluid up to a tolerance ε, a smaller tolerance means better mass conservation in the level set.

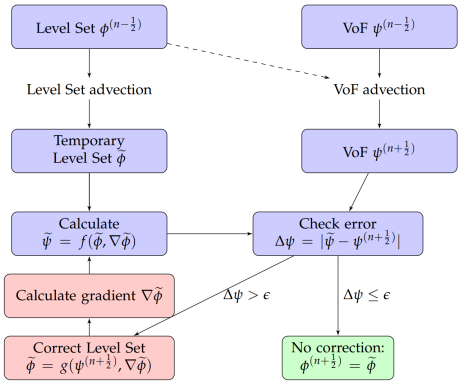


Figure 4: Flowchart showing the steps involved in the MCLS method. Note that correcting the level set also changes the local gradient, thus requiring the correction to be done iteratively. [3]

This method improves upon the previously described methods by ensuring mass conservation in the level set, while simultaneously giving justification to the interface orientations required for the level set. Some downsides of the individual methods are still there, however. Since volume of fluid is the main quantity for advection, the Courant condition (2.7) still must be satisfied to ensure mass conservation. Additionally, the use of donating regions for VoF advection requires a dimensionally split method to work. Using the donating regions of MCLS in an unsplit method results in some areas of fluid that reside in overlapping donating regions, thus being donated twice. This effectively duplicates any fluid inside this overlapping regions, making mass conservation no longer guaranteed. On the other hand, using non-overlapping donating regions based on the local flow velocity effectively creates the same method as the Lagrangian pre- or post-image method used in [6], which does not conserve mass as edge curvature cannot be easily taken into account. Thus, the method must be dimensionally split, which required a rectangular grid in some coordinate system. This severely impacts the domains that can be simulated. This shows that there is a need for a method which does not have the same limitations in time step and possible grid choice, to replace the MCLS method in cases where it is not applicable. The analysis in this thesis is a step towards such a method. For comparison purposes, the results of the MCLS method are shown in figure 31.

3 Optimisation based dual interface method

The analysis of section 2 forms the basis for the need of a new interface capturing method. The MCLS method is not sufficient to model certain flows due to the time step condition and grid requirements. As already mentioned in section 1, in a pipe flow where cylindrical coordinates are used, the method cannot effectively deal with an interface that passes through the radial pole of a pipe [3]. This occurs because a rectangular grid is used in cylindrical coordinates, which means that as the interface gets closer to $r = 0$, the width of the cell also goes to zero. If these cells are completely full or empty, this has no real impact on the solution, but when these cells become interface cells, the CFL condition (2.7) becomes important but cannot be satisfied as the contravariant velocity of the interface approaches infinity. This is merely one example of a situation where the constraints of the MCLS method have to be ignored, resulting in inaccuracies in the solution, but it shows that the method proposed in this thesis should not have issues handling these conditions.

The core concept of the MCLS method is that the level set function is being used to provide justified interface orientations for the volume of fluid evolution method. To this end, the method contains a level set advection method, a volume of fluid advection method that uses this level set, and a correction step that keeps the level set mass conserving to ensure that the information from the level set corresponds to the volume of fluid field. However, in section 2.2 it was noted that the only way to have a VoF advection method that conserves mass and requires reasonable computation time is obtained by using donating regions. The issue with this is that the use of donating regions automatically limits the method due to the requirement on time step and grid choice (figure 2). This shows that an unrestricted method must not rely on VoF advection for mass conservation, instead using the level set function as main variable. Additionally, the complete method must be applicable on unstructured grids using polygonal cells but remain easily expandable to more dimensions. Since volume of fluid advection methods are incapable of this application, as mentioned in section 2.2, a replacement must be found.

Using the level set function as the method's main variable returns the problem described in section 2.1; applying the advection equation to the level set field does not guarantee mass conservation. On the other hand, the finite volume advection scheme used in the MCLS method is second order accurate, so it does give a good approximation of the exact level set function. Additionally, the smoothness of the level set field near the interface allows for even higher orders of convergence with properly chosen discretisation methods. Thus, the volume of fluid field should be used to give a (probably small) correction to the advected level set field in order to keep mass conserved. The mass error must be distributed among the cells to obtain a conservative method. The problem with mandatory mass correction steps is that the exact formulation is often quite arbitrary. For example, the overfilled or underfilled cells might overflow to or borrow from neigbouring cells, redistributing the mass layer by layer until a valid VoF field is obtained [6] or mass errors may be transported to the interface, where they adjust the volume fractions of interface cells [7]. In the redistribution step of [7], it is mentioned that the conditions on the correction "would suggest to formulate a constrained minimisation problem". While they opted for another method, this idea forms the basis for the mass correction step used in this thesis. Additionally, in [13, 14], a minimisation problem is applied to the level set method to ensure mass conservation, which supports the idea of using an optimisation problem for a dual interface method.

The specific formulations of the various building blocks of this method as described in the remainder of this chapter are based on the assumption of a two-dimensional rectangular grid in Cartesian coordinates. The layout of the method is shown in figure 5. The various building blocks that are described here are all expandable to general unstructured grids and three dimensions, but the concepts are simpler to analyse and implement in the Cartesian case. The expansion to three dimensions and general grids is discussed in more detail in section 4.

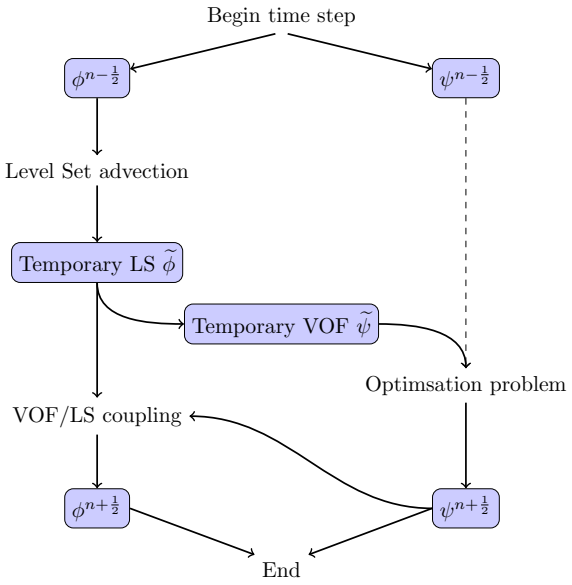


Figure 5: Flowchart detailing the steps in the optimisation based method. Since no volume of fluid advection method is used, the initial volume fractions only supply data for use in the optimisation problem.

3.1 Discontinuous Galerkin level set advection

The level set advection scheme used in this interface method must be applicable on unstructured grids. This means that the finite volume advection scheme of [3] is unusable for this method. For standard elliptic problems, a finite element method would be used, but for hyperbolic conservation laws such as the advection equation, these schemes require special stabilisation techniques. This is caused by the spatial symmetry of the basis functions, which can cause problems when modelling conservation laws as information flows in specific directions, and makes finite elements a less natural choice for this kind of problem [15]. Instead, for this problem, a *discontinuous Galerkin* (DG) method [2, 15] will be used for LS advection. This method not only allows for advection on grids with polygonal cells, but due to it's discontinuous nature, also allows the correction step to be applied locally. The result of this is that in the VOF/LS coupling step, the level set function only has to be altered once per interface cell, while the MCLS method requires multiple correction to account for the change in derivatives. In a worst-case scenario, the MCLS method may need to correct up to 30 times, depending on the method and tolerance, so only requiring one alteration per cell can be a significant speed-up for this step.

The discontinuous Galerkin method as described in [15] has a similar derivation to the standard finite element schemes, but uses purely local basis functions to allow discontinuities on element boundaries. The only interaction between elements is through a *numerical flux* $\boldsymbol{f}^*$ on each element boundary, whose value depends on the multiply-defined level set value at this boundary. Since the method should work on general polygonal grids, the DG method of [2] is used. This method not only has a *modal* expansion $\widetilde{\boldsymbol{\phi}}$ in the local basis functions on an element, but also a *nodal* solution $\boldsymbol{\phi}$, which consists of the level set value at certain nodal points. The number of basis functions and nodal points are dependent on the interpolation order p. The two expansions, as well as their basis functions, are coupled by a generalised Vandermonde matrix $\boldsymbol{V}$, with entries

$$V_{m,n} = \widetilde{v}_n(\boldsymbol{x}_m), \quad m = 1, ..., M_p, \ n = 1, ..., N_p, \tag{3.1}$$

where $\widetilde{\boldsymbol{v}} = \{\widetilde{v}_1, ..., \widetilde{v}_{N_p}\}^T$ contains the modal basis functions, and $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_{M_p}\}$ is the nodal set corresponding to the nodal basis expansion with basis functions $\boldsymbol{v} = \{v_1, ..., v_{M_p}\}$. Now, the two

bases can be coupled using

$$\boldsymbol{\phi} = \boldsymbol{V}\widetilde{\boldsymbol{\phi}}, \qquad \widetilde{\boldsymbol{v}} = \boldsymbol{V}^T\boldsymbol{v}. \tag{3.2}$$

In general, the number of modal basis functions and the number of nodal points are unequal, which makes that only a pseudo-inverse of the Vandermonde matrix is available. However, on rectangular elements it is simple to have an equal amount of modal and nodal points, which means that the inverse transformations can be applied in a straightforward manner. For these rectangular cells, all calculations are done on a unit square, and the modal basis functions are chosen to be

$$\widetilde{v}_{(p+1)j+i+1}(\boldsymbol{x}) = P_i(x)P_j(y), \quad 0 \leq i,j \leq p, \tag{3.3}$$

with $\{P_i(x)\}_{i=0,...,p}$ the set of one-dimensional normalized Legendre polynomials on [-1,1]. The nodal set is chosen in a similar way, with

$$\boldsymbol{x}_{(p+1)j+i+1} = (\xi_i, \xi_j), \quad 0 \leq i,j \leq p, \tag{3.4}$$

where $\{\xi_i\}_{i=0,...,p}$ are chosen according to the Legendre-Gauss-Lobatto distribution. The relation between the interpolation order and the amount of basis functions and nodal points is now given by $N_p = M_p = (p+1)^2$.

Once the choice for basis function and nodal set is made, the time evolution of the modal solution on one element K is given by

$$\sum_{n=1}^{N_p} \mathcal{M}_{m,n}\frac{\mathrm{d}\widetilde{\phi}_n}{\mathrm{d}t} - \sum_{n=1}^{M_p}(\mathcal{S}_{x,(m,n)}f_{x,n} + \mathcal{S}_{y,(m,n)}f_{y,n}) + \int_{\partial K}\boldsymbol{f}^*\cdot\boldsymbol{n}\widetilde{v}_m = 0, \quad m = 1,...,N_p. \tag{3.5}$$

The general mass and stiffness matrices for this element are then

$$\begin{aligned}
\mathcal{M}_{m,n} &= \int_K \widetilde{v}_m(\boldsymbol{x})\widetilde{v}_n(\boldsymbol{x})\ d\boldsymbol{x}, \\
\mathcal{S}_{x,(m,n)} &= \int_K \frac{\partial\widetilde{v}_m}{\partial x}v_n\ d\boldsymbol{x}, \\
\mathcal{S}_{y,(m,n)} &= \int_K \frac{\partial\widetilde{v}_m}{\partial y}v_n\ d\boldsymbol{x}.
\end{aligned} \tag{3.6}$$

Due to the choice in basis function, the mass matrix $\mathcal{M}$ equals $\left(\frac{\Delta x}{2}\right)^2 I$, with I the identity matrix of the appropriate size. The flux vector is simply given by $(f_x, f_y)_m = \boldsymbol{u}(\boldsymbol{x}_m)\phi(\boldsymbol{x}_m)$. Since the velocities are only given on the cell boundaries, the flux in the interior is derived by linearly interpolating the velocities between the edges. The only unknown is now the choice of the numerical flux. Similar to [2], the Local Lax-Friedrichs flux is used as the numerical flux for this discretisation. Since only one velocity is given per boundary, this method is identical to a first order upwind scheme. The continuous integral in (3.5) on a boundary is now calculated using Lagrange interpolation on the discrete $\boldsymbol{f}^*$ along this edge.

This defines the level set advection method used. As an intermediate result, the DG method can replace the finite volume advection scheme used in the baseline MCLS method. Since volume of fluid advection can only be done using a dimensionally split method, while DG is dimensionally unsplit, the level set correction step must be changed somewhat. Before any correction steps have taken place, both the original and advected level set functions are stored. After the x-direction sweep, the correction is applied to the original level set, which then supplies the interface orientations for the y-direction sweep. After the volume of fluid has finished advection in all coordinates, the final correction is applied to the advected level set and the method starts advection for the next time step.

3.2 Coupling between VoF and LS

Since the volume of fluid field must be used to preserve mass conservation in the level set function, the coupling between the level set and volume of fluid fields must be described. More specifically, it is required to know how to generate the temporary volume of fluid from the advected level set, and how to correct the level set after the correction step is finished. On one element K, a linear expansion is made from the local modal solution $\widetilde{\phi}^K$. This local expansion can be written as

$$\widetilde{\phi}^K \approx c_0 + c_1(x - x_i) + c_2(y - y_i) \tag{3.7}$$

with (x_i, y_i) the cell centre of element K. These coefficients are then taken to be

$$\begin{aligned} c_0 &= \frac{1}{2}\widetilde{\phi}_0 \\ c_1 &= \frac{\sqrt{3}}{\Delta x}\widetilde{\phi}_1 \\ c_2 &= \frac{\sqrt{3}}{\Delta x}\widetilde{\phi}_{p+1} \end{aligned} \tag{3.8}$$

since the relevant basis functions for these modes on the reference element are $\frac{1}{2}, \frac{1}{2}\sqrt{3}x, \frac{1}{2}\sqrt{3}y$ respectively. If this linear expansion has two edges where the it changes sign, an interface can be constructed between the roots. From this interface, the volume fraction is constructed. By doing this for every cell, a volume of fluid field can be constructed from the level set values.

The reverse coupling is not as straightforward. After the optimisation problem is finished, a level set must be constructed that corresponds to the volume of fluid field by the coupling described above. To do so, the advected level set is taken as baseline for the corrected field, and the corresponding volume of fluid field is constructed. In every cell where a volume mismatch is present between the level set and optimised volume of fluid, the constant mode of the level set is altered using the bisection method until this mismatch vanishes. The gradient of the level set is kept constant through this entire process, and since the level set is allowed to be discontinuous, this reconstruction only needs to be done once per correction.

3.3 Optimisation problem for mass conservation

The DG method allows unrestricted level set advection on general polygonal grids, but it does not conserve mass for the same reason as the finite volume scheme. A constrained minimisation problem is now formulated to distribute the mass error in a logical way such that mass is conserved without the use of unfounded correction steps. Via the coupling described in section 3.2, from the advected, non-conservative level set function $\widetilde{\phi}$ a volume of fluid field $\widetilde{\psi}$ can be constructed. Since the uncorrected level set is nearly mass conserving, due to the level set advection method being asymptotically mass conserving, the same can be said for the corresponding fractional volumes. It therefore must hold that to obtain the actual advected volume of fluid field $\psi^{n+\frac{1}{2}}$, only a small correction on $\widetilde{\psi}$ is required. Thus,

$$\psi^{n+\frac{1}{2}} = \widetilde{\psi} + \Delta\psi \tag{3.9}$$

for some small correction $\Delta\psi$. The mentioned minimisation problem will attempt to find the $\Delta\psi$ that best fits this description.

To formulate the minimisation problem, an objective function must be chosen and constraints must be placed on this correction $\Delta\psi$. The analysis in this thesis is done to gain more insight on which choices should be made for these constraints and objective function. It is obvious that $\Delta\psi$ must at least be chosen such that

- $\psi^{n+\frac{1}{2}}$ must be a valid volume of fluid field, so $0 \leq \psi^{n+\frac{1}{2}} \leq 1$,

- mass must be conserved, so $\sum \psi^{n+\frac{1}{2}} = \sum \psi^{n-\frac{1}{2}}$,
- the correction $||\Delta\psi||$ must be small in some sense to preserve the accuracy of the DG method.

However, only requiring these conditions will not make the solution $\Delta\psi$ unique by any means, as can be seen by the simple example in figure 6. The situation is symmetric, so both interface cells will have equal weights in the objective function of the problem. This means that the mass correction can be applied to either cell, resulting in multiple valid solutions to the problem. A well-posed problem, however, should have a unique solution, so additional constraints are required to ensure that the optimisation problem is well-posed. While there are multiple ways to further constrain the problem, it is opted to use a flux condition as a first additional constraint.

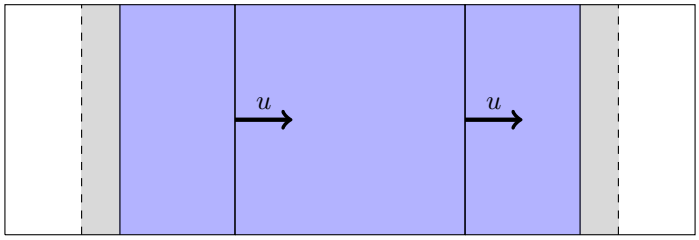


Figure 6: Simple situation to indicate that constraints are required to ensure the uniqueness of the solution. In this case, the mass difference can be fixed on the left or on the right side of the fluid column (or a linear combination of the two). Both choices result in a valid, mass conserving volume of fluid field, and due to symmetry, both corrections are equally 'small'. This means that both corrections are valid for the minimisation problem, so the solution is not unique.

3.3.1 Flux condition

The flux condition is based on the comparison of multiple formulations of the local rate of change of the discrete volume fractions $\frac{\partial \psi}{\partial t}$. More specifically, one such formulation is given by the total volume of fluid field, while the other can be calculated using only local values. It is then logical that for a correct volume of fluid field, both of these formulations are equal to each other. This gives a way to couple the volume fraction in one cell to the neighbouring values and orientations, and allows for the construction of a new volume of fluid field.

The first formulation is quite straightforward, as it has already been described. In mass conserving volume of fluid advection methods, the time derivative of the volume fraction is dependent on the net flux through the cell boundaries. More specifically, from equation (2.6), in a divergence-free velocity field the flux is the only contribution to the local rate of change. But while advection methods require the approximation of this flux during a time step Δt, since in this method no time steps are done for the volume of fluid, such an approximation is not required. This is ideal since requiring this approximation over a time step would lead directly to a donating regions method, which is specifically to be avoided. Instead, only the rate of change at the current time is needed, which requires only the calculation of the momentary net flux through the cell boundaries.

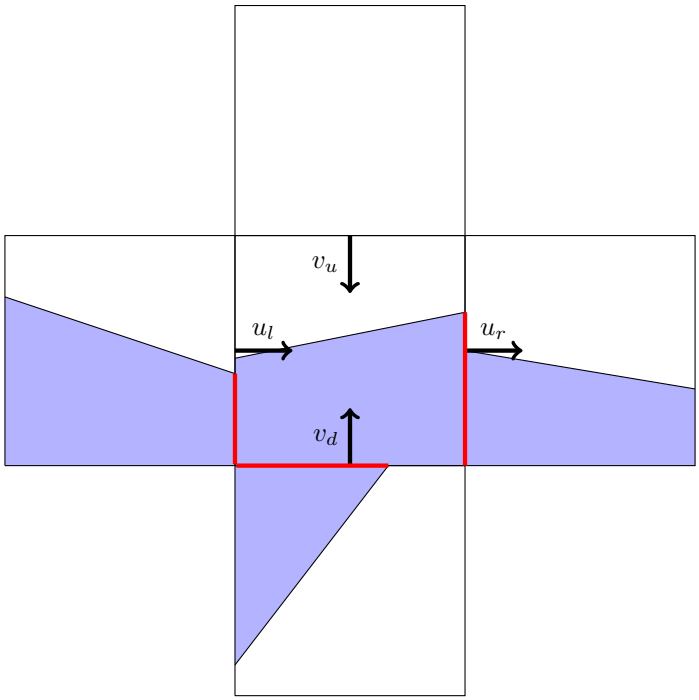


Figure 7: A scenario to illustrate the workings of the flux condition. At each edge, the momentary flux can be described using an infinitely thin donating region. This means that the flux equals the length of the edge that is covered in fluid (red line), multiplied by the velocity at this edge. As shown, only the fluid in the donating cell is relevant, so on the inflow boundaries the data from neighbouring cells is required.

The calculation of this net flux can be done in a simple way. Since the required value is effectively a momentary version of the donating regions flux of [1], the net flux can be calculated using 'donating regions' as well. The only difference is that no time step is taken, so the donating regions only approximate the flux in an infinitesimally small time step dt. This means that these regions effectively have no thickness, so regions corresponding to different edges cannot overlap. To illustrate this, consider the scenario in figure 7. In this case, the central cell has three inflow edges and only one outflow boundary, though the top boundary is devoid of any fluid and thus does not contribute any flux. The total flux (in volume) going into the cell is then equal to $u_l \Delta x_l + v_d \Delta x_d$, while the outgoing flux would be $u_r \Delta x_r$. Here, Δx_l denotes the length of the donating region corresponding to the left boundary, and analogous for the other boundaries. Note that the length of this region is only the result of the fluid in the donating cell. Since at each edge, the fluid moves from the donating cell to the accepting cell, any part of the edge that is not covered by fluid on the donating side does not contribute any flux, even if it is covered on the accepting side. This means that on any inflow boundary for the central cell, the donating region is governed by the neighbouring cell, while the outflow boundary is controlled by itself.

On the other hand, the interface in a cell also supplies unique knowledge about the local rate of change in a cell. Given that the flow velocity is known everywhere, though it must be interpolated in the interior of a cell, the velocity of the interface can also be determined. Since the interface moves according to the local flow velocity, this speed must also influence the rate of change of the volume fractions. Assuming that the local flow velocity is constant everywhere along the interface, and that the speed of the interface is then u_F in it's normal direction, it can be deduced that the rate of change equals

$$\frac{\partial \psi}{\partial t} = \frac{u_F l_F}{|\Omega|}, \tag{3.10}$$

where l_F indicates the length of the interface. Due to a time step dt, the interface shifts by a distance $u_F dt$, so the total volume added is $u_F l_F dt + O(dt^2)$. Converting this to volume fractions gives the time derivative of (3.10).

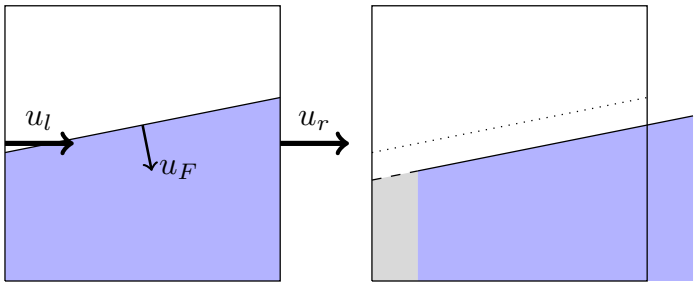


Figure 8: Illustration of the volume change due to a shift in the interface. Note that this shift can also be explained by a flux through the interface. The shift from the original interface (dotted line) to the current one (dashed/solid line) due to a horizontal velocity is effectively the same as using donating regions, where the gray volume is fluxed from 'outside'.

The formulation from (3.10) only works in linear velocity fields, however. In nonlinear flows, the velocity change due to the interface movement possibly cannot be calculated in the same way due to it's deformation. This would cause problems, since it means that the time derivative due to the interface velocity can no longer be accurately calculated. However, there is another method to use for this calculation that does remain accurate in nonlinear velocity fields. Consider an interface in any velocity field. During an infinitesimally small time step dt, any fluid that lies in the interior of the cell is going to remain in this cell. Therefore, it can be concluded that the momentary rate of change must be caused by a change of mass at the cell boundaries. Visualising the situation (figure 8) shows that the change in volume can effectively also be described using donating regions, the same way as with the net flux. The only difference in this case, is that the local interface must be used. This formulation is then the same as the momentary net flux, if the local interface orientation is extended to the entire domain.

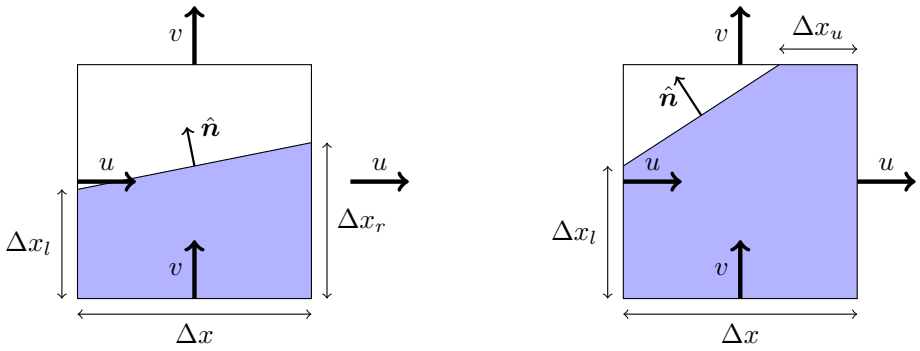


Figure 9: Illustration for the proof that the edge flux and interface shift produce identical results.

To prove that the local momentary net flux is identical to (3.10) for linear velocity fields is straightforward to show, and will be done for the two cases illustrated in figure 9. First, consider the case where two cell vertices are covered in fluid. In this case, the net volume flux is simply

$$F_{\text{net}} = u(\Delta x_l - \Delta x_r) + \Delta x v. \tag{3.11}$$

It must then be shown that this is equal to $u_F l_F$, in order to comply with (3.10). Assume that the interface orientation is $\hat{\boldsymbol{n}} = (n_x, n_y)$, then it follows that

$$\Delta x_r - \Delta x_l = -\frac{n_x}{n_y}\Delta x, \tag{3.12}$$

since the interface tangent has direction $(n_y, -n_x)$. Using the same information, it can also be concluded that

$$l_F = \sqrt{\Delta x^2 + (\Delta x_l - \Delta x_r)^2} = \sqrt{\Delta x^2 \frac{n_y^2}{n_y^2} + \Delta x^2 \frac{n_x^2}{n_y^2}} = \frac{\Delta x}{|n_y|}\sqrt{n_x^2 + n_y^2} = \frac{\Delta x}{|n_y|}. \tag{3.13}$$

Now, taking that the normal velocity of the interface is $u_F = un_x + vn_y$, it can be concluded that

$$u_F l_F = u\frac{n_x}{|n_y|}\Delta x + v\frac{n_y}{|n_y|}\Delta x = u\frac{n_x}{n_y}\Delta x + v\Delta x = F_{\text{net}}, \tag{3.14}$$

as $n_y > 0$. Now note that, would n_y be negative, the conclusion still holds. If $n_y < 0$, then not only does the interface length utilise $-n_y$, but in the net flux, the vertical part will become $-v\Delta x$ as well since the fluid is then at the top of the cell. Additionally, the tangent $(n_y, -n_x)$ now points to the left, so now $-\frac{n_x}{n_y}\Delta x$ denotes the difference $\Delta x_l - \Delta x_r$ rather than vice versa. This means that if n_y becomes negative, not only the interface shift contribution is mirrored, but also all flux terms change sign. Therefore, the conclusion is valid for all cases where two vertices are covered by fluid.

When three vertices contain fluid, the analysis becomes slightly more complex. The net flux is now written as

$$F_{\text{net}} = u(\Delta x_l - \Delta x) + v(\Delta x - \Delta x_u). \tag{3.15}$$

Defining $a = \Delta x - \Delta x_u$ and $b = \Delta x - \Delta x_l$ to be the sides of the triangle not covered in fluid, these lengths can be linked in the same way as before

$$b = -\frac{n_x}{n_y}a, \tag{3.16}$$

which means that the interface length becomes $l_F = \frac{a}{|n_y|}$. Additionally, note that $\Delta x_l - \Delta x$ can be written as

$$\Delta x_l - \Delta x = -b = \frac{n_x}{n_y}a = \frac{n_x}{n_y}(\Delta x - \Delta x_u). \tag{3.17}$$

Now, it again follows that

$$\begin{aligned} u_F l_F = un_x\frac{\Delta x - \Delta x_u}{|n_y|} + vn_y\frac{\Delta x - \Delta x_u}{|n_y|} &= u\frac{n_x}{n_y}(\Delta x - \Delta x_u) + v(\Delta x - \Delta x_u) \\ &= u(\Delta x_l - \Delta x) + v(\Delta x - \Delta x_u) = F_{\text{net}}. \end{aligned} \tag{3.18}$$

Again, if n_y becomes negative, the interface shift term starts using $-n_y$, but like before the net flux terms also flip sign for the same reason as in the previous case. Now, if n_x flips sign, the sign of b changes as well, while b should be a positive length. However, this only affects the horizontal part of the interface shift, and in the net flux, the b term moves from the inflow boundary to the outflow one or vice versa, cancelling this sign change. Additionally, if only the light fluid would be considered (so only one vertex is covered) the net inflow becomes $ub + va$, which is (as expected) the outflow of dark fluid, and the interface shift changes sign due to the outward normal being flipped. Therefore, for all interface orientations and volume fractions, the edge flux and interface shift terms are identical in linear velocity fields. In figure 10, the scenario of figure 7 is shown, but the donating regions are now the ones required for the interface shift term.

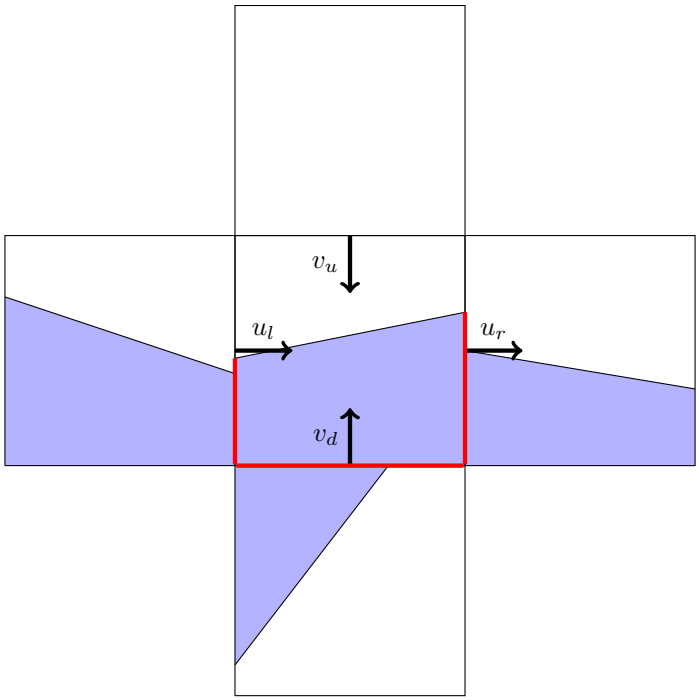


Figure 10: The scenario of figure 7, but the donating regions (in red) are now defined solely by the interface of the central cell, since they contribute to the rate of change due to an interface shift.

The goal of the flux condition is now to ensure that the total flux for both these formulations are identical in all cells, so that the length of the inflow regions minus the length of the outflow regions is the same for both cases. Note, now, that for the net flux, the donating regions for the outflow boundaries are also defined by the cell in question. Therefore, the only issue is the combined length of regions on the inflow boundaries. The most trivial solution to this would be to simply match edge length on all cell boundaries, thereby ensuring that the inflow length is the same for both cases. Such a configuration is shown in figure 11. Though only the top, left, and bottom cell provide information to the central cell, the interface of the central cell requires the rightmost cell to change too.

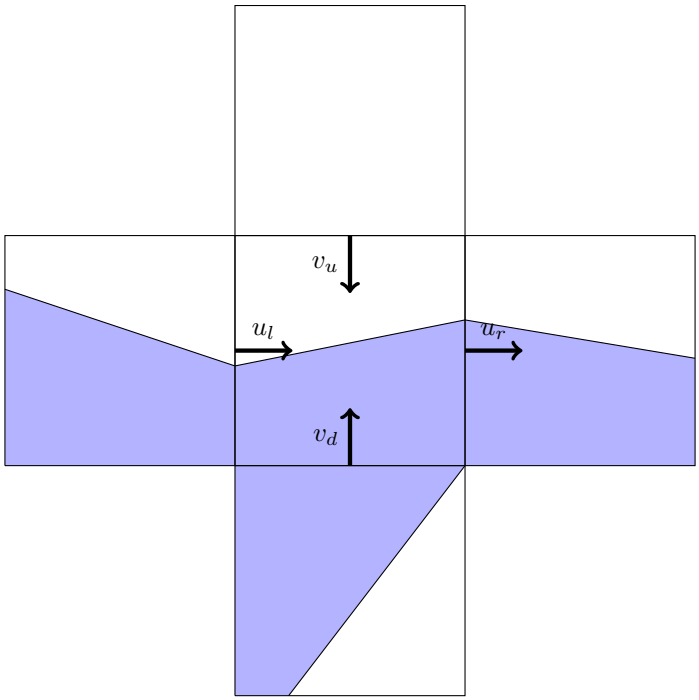


Figure 11: The scenario of figure 7, shifted in such a way that the edge flux and interface shift scenarios provide the same rate of change for all cells. The right cell has been shifted, as it must match lengths on it's inflow boundary, which is the outflow boundary of the central cell.

This condition is the basis for the mass correction routine. Note that, in this thesis, a linear minimisation problem will be used to enforce this condition. This makes the correction from figure 11 problematic, since the bottom cell changes states (1 vertex in fluid $\rightarrow$ two vertices in fluid). This means that the linearised condition will not recognise the interface intersection with the lower boundary of the bottom cell, and will see an incorrect flux through it's left boundary. Using nonlinear optimisation would allow dealing with these state changes. Additionally, note that only the length of the fluid on the boundary matters to the flux condition. With a linear problem, this only requires a linear response on each cell edge. So even if an interface would be nonlinear, as long as the edge profile of the level set remains linear, the linearised flux condition can handle it. Though this is the case, interfaces are linearised to ensure that this linearity exists.

The flux condition allows for correcting the mass of the level set function while taking the discontinuity of the color function χ into account, which is essential for accuracy. It is logical to attempt to make the expected rate of change due to the interface in a cell consistent with the neighbouring cells, especially if it smooths the interface transitions on cell edges. The flux condition should thus be satisfied first, with any additional conditions being applied afterwards. Ideally, the minimisation problem would use other physical constraints and objectives, such as requiring the centre of mass in a certain position or minimising the energy requirement of the fluid region (lowest curvature of the entire interface). These conditions, however, would require a nonlinear optimisation problem. Since this thesis is an early step towards optimisation based methods, linear methods are preferred for now. However, should nonlinear optimisation methods be required and the physical constraints be implemented, the flux condition at least helps minimising the energy by smoothing the interface transitions. Therefore, the flux condition will be used here, and should also be considered in a nonlinear minimisation problem.

To be able to use the flux condition, the exact dependence on the volume of a cell must be described. Since a linear optimisation problem is used, the fluxes are found using a linear approxi-

mation around the base state $\widetilde{\psi}$:

$$\Delta F(\psi^{n+\frac{1}{2}}) = \Delta F(\widetilde{\psi}) + \Delta\psi \frac{\partial F}{\partial \psi} \tag{3.19}$$

with ΔF the net flux for either the momentary flux or interface shift scenario. The effect of the base state is simple to calculate, as this is given on every cell edge by the length of this cell boundary that is covered in fluid, multiplied by the normal velocity on this edge. For the dependence on the correction $\Delta\psi$, consider an interface that has been shifted in it's normal direction (n_1, n_2) by a small distance Δx. Using congruence of triangles (see figure 12), it is straightforward to show that the interface intersection with a horizontal boundary moves with distance $\frac{\Delta x}{n_1}$, whereas for a vertical boundary this becomes $\frac{\Delta x}{n_2}$.

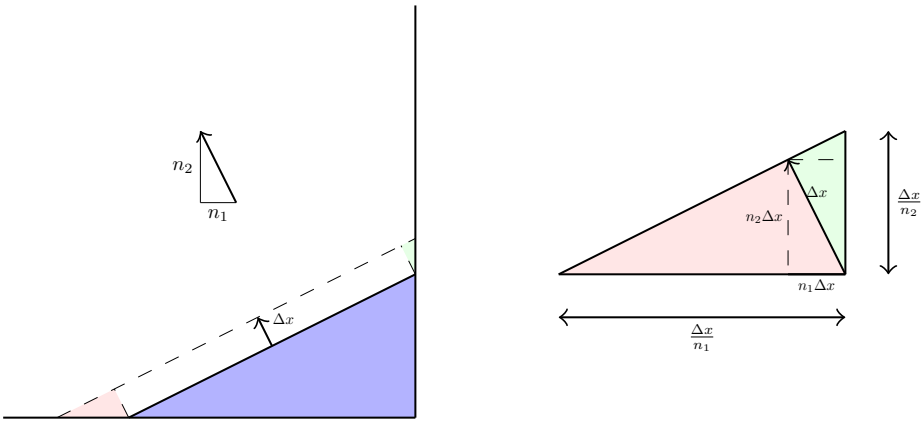


Figure 12: Shift of the horizontal and vertical interface intersections due to a shift in interface of Δx. The red and green triangles correspond to sections near the horizontal and vertical cell boundaries respectively. Both coloured triangles are congruent with the triangle created by the normal vector, making this shift easy to prove.

At the same time, the volume of the fluid inside the cell changes by $l_F \Delta x + O(\Delta x^2)$, where l_F denotes the length of the initial interface. Because of this, the dependence of the flux on the volume fraction for a horizontal edge with normal velocity u is given by

$$\frac{\partial F}{\partial \psi} = \frac{u|\Omega|}{l_f n_1}, \tag{3.20}$$

where the factor $|\Omega|$ is caused by the conversion between volume and volume fraction. This applies to both formulations of the rate of change calculation, as the only difference between them is the interface that is considered.

3.3.2 Objective function

To finish the formulation of this minimisation problem, an objective function is still required. The method must not be allowed to create interfaces where there weren't any, and must attempt to keep the shape of the temporary solution somewhat intact. The problem with this is that a linear minimisation problem cannot, in general, be forced to spread out the remaining mass among the various interface cells. Even when multiple cells are available once the solution complies with the flux condition, the method simply finds one with the lowest impact on the objective function and adds or removes the remaining mass difference there. Even if multiple cells have the lowest contribution to the objective, the method has no incentive to smear out the remaining mass, and cannot be forced to do so without the use of nonlinear optimisation. Because of this, the minimisation problem

cannot apply the mass conservation condition, which must be done separately. Additionally, there are no real physical objective functions that can be applied here, so the objective function is taken to minimise the size of the correction $\Delta\psi$.

After the minimisation method has applied the flux condition to the solution $\widetilde{\psi}$, the mass correction method can ensure that mass remains conserved during the time step. For this correction, the cells that are independent of the flux condition are considered. This means that the size of both formulations of the rate of change for this cell have no dependence on the volume, and that no neigbouring cell depends on the volume inside this cell either. This ensures that the flux condition remains in effect, even when this mass correction is applied. Among the cells that are being altered, the mass is distributed using Moment of Fluid-like weights; the weight in each cell is given by

$$w_i = \frac{||\boldsymbol{x}_{c,i} - \boldsymbol{c}_{F,i}||}{\psi_i}, \tag{3.21}$$

where $\boldsymbol{x}_c$ is the centre of mass inside the cell, and $\boldsymbol{c}_F$ is the centre of the interface. The idea of these weight is that cells whose centre of mass would move more due to a change in volume (and thus have a higher weight) get a smaller volume change. The remaining mass is then distributed among the cells proportionally to the reciprocal of their weights, so that lower weight means more change. If the change in any cell would be more than it can provide, the debt is distributed among the other cells using the same weights.

The entire optimisation based mass correction method now looks as follows: First, define the correction

$$\Delta\psi_i = p_i - q_i, \quad p_i, q_i \geq 0 \tag{3.22}$$

as having a positive and negative part, to enforce the objective function. Then, the flux condition is applied by the optimisation problem

$$\begin{aligned}
&\text{minimise} \quad \sum_i p_i + q_i \\
&\text{such that} \\
&\qquad p_i = q_i = 0 && \text{for completely full or empty cells} \\
&\qquad p_i - q_i \leq 1 - \widetilde{\psi}_i && \text{avoid overfilling} \\
&\qquad p_i - q_i \geq -\widetilde{\psi}_i && \text{avoid overemptying} \\
&\qquad \sum_{j \in \text{neighbourhood}} c_j(p_j - q_j) = \Delta F_{\text{net}}(\widetilde{\psi}) - \Delta F_{\text{shift}}(\widetilde{\psi}) && \text{flux condition}
\end{aligned} \tag{3.23}$$

This optimisation problem effectively contains two variables for each interface cell as non-interface cells are not altered, and has four inequality conditions per such cell, if the equality is interpreted as two inequality equations.

Then, the weight w_i are calculated in each cell, and in each relevant cell, the volume is updated by

$$\Delta m \frac{\frac{1}{w_i}}{\sum_j \frac{1}{w_j}} \tag{3.24}$$

with Δm the total mass difference. This defines the total mass correction $\Delta\psi$, which allows calculation of $\psi^{n+\frac{1}{2}}$. When the volume of fluid field at the new time step is known, this can be used to reconstruct the corresponding level set function according to the VOF-LS coupling described in section 3.2. Using this, the new level set field $\phi^{n+\frac{1}{2}}$ can be constructed, and all information is known for the next time step.

The method as described here is designed to be applicable to general unstructured grids, and does not have the time step restriction of the MCLS. Larger time steps can be taken with the DG

method [15], and the method should remain stable for much larger time steps than any method with VoF advection, though more accurate time integration methods may be required to avoid losing too much accuracy. The expansion to general grids will be described in section 4. Thus, the method fulfils the criteria that were set.

4 Expansion to polygonal cells

The method described in section 3 has been created on a two-dimensional Cartesian grid. This was done to simplify the analysis of the methods, and allow comparison to the MCLS method or the method that combines volume of fluid advection with a discontinuous Galerkin scheme for the level set function. For general use, however, the method must be applicable in three dimensions and must be usable on unstructured grids. If a method cannot be extended to three dimensions, it is not useful for general purposes.

First, the extension to unstructured two-dimensional grids is considered. Here, a discontinuous Galerkin scheme was implemented on a Cartesian grid, but according to [2] it can also be applied on general polygons. The problem with this application on unstructured grids, is that as the amount of unique cells increases, so do the storage requirements for this method. On a Cartesian grid, all calculations could be done on a reference square, but if a grid is filled with unique cells that cannot be easily transformed into another, this is not possible. This would require each cell or group of similar cells to have their own reference element with its own Vandermonde matrix and mass- and stiffness matrices. Additionally, the set of nodal points would be different for each unique cell, so the location of these points would need to be tracked as well. The fact that the method requires storing several matrices for each unique cell makes the computational requirement infeasible for larger grids. Of course, there are ways to fix this quite severe issue. In [2], it is proposed that for quadrilateral cells, calculations are done on a reference square anyway and then transformed to the original cell. Though this transformation does not have constant Jacobian for general quadrilaterals, an approximate conversion is done instead. This would allow any number of cells without significantly increasing storage, since now only the conversion constants are stored, but the effect of this approximation is unknown, especially for polygons with more vertices. On the other hand, if polygonal grids are required, in most cases quadrilaterals or rectangles can be used for the interior of the grid, using triangles or more unique quadrilaterals to join the others to the domain boundary.

If the DG method works for a given grid, then the optimisation problem can work as well. The conversion between the level set and the volume of fluid is unchanged, and with a linear interface the two interface intersections on the boundary will remain, at least if the elements are convex. Additionally, the 'donating regions' defining the flux for the temporary solution $\widetilde{\psi}$ will also be applicable. With normal donating regions, the use on unstructured grids is hindered by overlapping regions causing errors in mass conservation. For the optimisation problem, the regions are infinitely thin, so no overlap is possible. The only difficulty is in the calculation of the volume dependence. This quantity is solely defined by the angle between the cell boundary and the interface, but since the boundary no longer matches the governing Cartesian coordinates, calculating this angle takes more work. Even though it is no longer as simple as taking the x- or y-component of the normal direction, this dependence can still be used. Thus, the method of section 3 works for general grids.

The extension to three dimensions for the optimisation problem is also not going to cause problems. It's extension is straightforward, with the only difference being that now the fluid-covered parts of each cell boundary are now areas rather than lengths. The net flux and its volume dependence can still be calculated in a similar way as in two dimensions. The problem with the extension of the entire method is the level set. While the DG method definitely works in three dimensions on a tetrahedral grid [15], the hybrid nodal-modal method of [2] may not be so easy to convert. The third dimension is probably simple to incorporate in the equation itself, but the calculation of the boundary integral and the choice for nodal set may pose some problems. If this can be done easily, then this method will be able to be implemented on three dimensional unstructured grids as well.

5 Results

To analyse the performance of the model detailed in section 3, the method is used to evolve the fluid interface in four different test cases.

5.1 Solid body translation

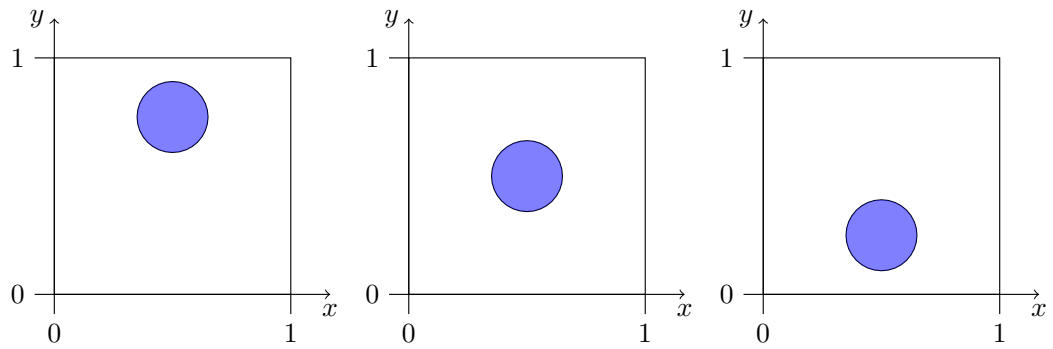
In the solid body translation test, the movement of the fluid is governed by the velocity field

$$u(x, y) = 0, \qquad v(x, y) = -1. \tag{5.1}$$

This field is linear and constant, so the exact interface $\Gamma(t)$ is known:

$$\Gamma(t) = \{(x, y) \in \Omega : (x, y + t) \in \Gamma(0)\}. \tag{5.2}$$

The interface simply shifts in the $-y$ direction with a speed of 1 unit length per unit time. The interface is initialised as a circle of radius $r = 0.15$ around the point $(x, y) = (0.5, 0.75)$, and the end time $T = 0.5$. This means that the final interface is a circle of radius r around the point $(x, y) = (0.5, 0.25)$, as is shown in figure 13.



(a) The interface at $t = 0$. (b) The interface at $t = 0.25$. (c) The interface at $t = 0.5$.

Figure 13: The exact movement of the interface in the solid body translation velocity field.

5.2 Solid body rotation

Similar to the solid body translation, the solid body rotation test uses a velocity field that does not deform the interface:

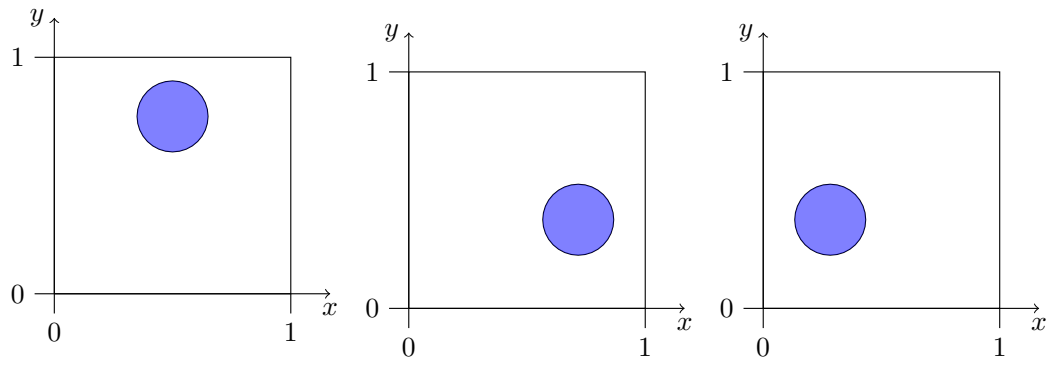
$$u(x, y) = 2\pi(y - 0.5), \qquad v(x, y) = -2\pi(x - 0.5). \tag{5.3}$$

This is a nonlinear velocity field which rotates the interface around $(0.5, 0.5)$ with a full revolution taking exactly 1 unit of time. In polar coordinates, the exact interface is

$$\Gamma(t) = \{(r, \theta) \in \Omega : (r, \theta + 2\pi t) \in \Gamma(0)\} \tag{5.4}$$

with the positive unit vector of θ defined in the counterclockwise direction.

The initial condition is chosen the same as in the solid body translation test, and the end time is taken as $T = 1$, to ensure that the interface returns to it's original position at the end time, as shown in figure 14.



(a) The interface at $t = 0$ and $t = 1$.

(b) The interface at $t = 0.33$.

(c) The interface at $t = 0.67$.

Figure 14: The exact movement of the interface in the solid body rotation velocity field.

5.3 Corner flow

The corner flow test is defined by the velocity field

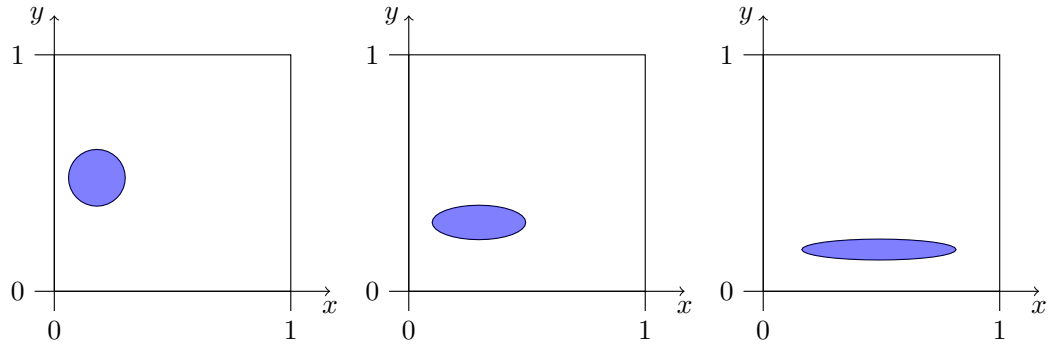
$$u(x, y) = x, \qquad v(x, y) = -y. \tag{5.5}$$

This velocity field deforms the interface; if the interface is initialised as a circle, it transitions to an elliptical shape as it is exponentially compressed in the x direction, and exponentially widened in the y-direction.

$$\Gamma(t) = \{(x, y) \in \Omega : (xe^{-t}, ye^{t}) \in \Gamma(0)\}. \tag{5.6}$$

This velocity field was used by [1] as a test case for the dimensionally split method, which shows the importance of taking the one-dimensional dilatation into account. This test case would also be relevant for this method for testing it's CFL condition number, as the (relatively) high shear stress would quickly limit the maximum time step for volume of fluid methods.

The initial condition is a circular interface chosen with the same ratio as in [1], in this case with $r = 0.12$. It is thus centered around the point $(x, y) = (0.18, 0.48)$. The deformation of the interface is shown in figure 15.



(a) The interface at $t = 0$.

(b) The interface at $t = 0.5$.

(c) The interface at $t = 1$.

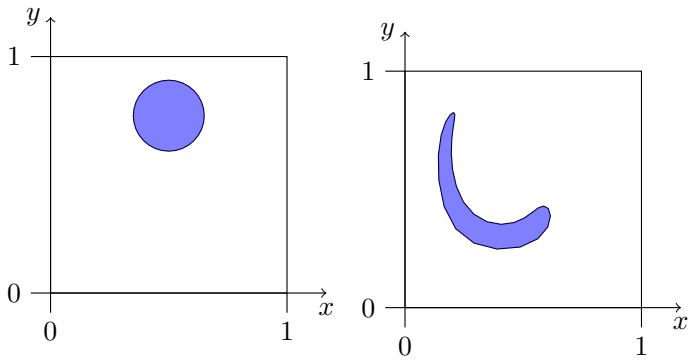
Figure 15: The exact movement of the interface in the corner flow test.

5.4 Vortex deformation

This test evolves the interface in the shear deformation vortex field [6]

$$u(x, y, t) = \sin^2(\pi x)\sin(2\pi y)\cos(\pi t/T), \qquad v(x, y, t) = -\sin^2(\pi y)\sin(2\pi x)\cos(\pi t/T). \tag{5.7}$$

The interface is initialised at the same position as in the solid body translation and rotation tests. The interface is then gradually deformed by the vortex until $t = T/2$, when the cosine term causes the velocity field to reverse, from which point the exact interface returns to its original position at $T = 2$. In figure 16, the state of the interface is shown for different times.



(a) The interface at $t = 0$ and $t = 2$.

(b) The interface at $t = 1$.

Figure 16: The exact movement of the interface in the vortex deformation test.

5.5 Error analysis

The given test cases are used to verify the accuracy and convergence of the volume of fluid error in the L^2 norm. For these test, the error metric used for an $N \times N$ grid is

$$E_2 = \frac{1}{N}\sqrt{\sum_i (\psi_i - \bar{\psi}_i)^2}, \tag{5.8}$$

where ψ indicates the simulated volume fractions, and $\bar{\psi}$ is the exact volume of fluid solution at the end time. The results of the test cases for various mesh widths and time stepping methods are given in table 1. All results in this thesis are done with DG interpolation order $p = 1$.

		Euler Forward	Modified Euler	Runge Kutta 4
Solid body translation	$N = 8$	0.066439	0.045930	0.048199
	$N = 16$	0.084753	0.057593	0.060916
	$N = 32$	0.048526	0.036671	0.036643
Solid body rotation	$N = 8$	0.223931	0.136772	0.155901
	$N = 16$	-	0.020412	0.017753
	$N = 32$	-	0.004686	0.004508
Corner flow	$N = 8$	0.186021	0.181482	0.176793
	$N = 16$	0.028024	0.093216	0.072679
	$N = 32$	0.013075	0.015224	0.016124
Vortex deformation	$N = 8$	0.129606	0.123719	0.140808
	$N = 16$	0.051198	0.045461	0.045189
	$N = 32$	0.029662	0.030533	0.030192

Table 1: Volume of fluid error in the L^2 norm for the different flows on an $N \times N$ grid with various time stepping methods. Time steps are $\Delta t = \frac{1}{64}$ for $N = 8$, $\frac{1}{128}$ for $N = 16$, and $\frac{1}{256}$ for $N = 32$. For the solid body rotation flow, the time steps are twice as small as the given values. The dashed scenarios give results that are not stable. In general, a smaller mesh width corresponds to a lower error, but the apparent order of convergence fluctuates wildly between test cases. Additionally, more accurate time stepping methods do not equal a lower error in many cases.

While the results from table 1 are generated using the method from section 3, the solid body translation test requires one additional step. After every minimisation problem the level set function is 'reinitialised', where the value of this function at each point is set to be the distance to an interface segment. This is done only for the solid body translation test, since in this test the different vertical 'strips' are independent from each other with respect to the flux condition. Because of this, simulation solid body translation without reinitialisation can cause a build-up of mass near the outer layers of the fluid area, which is an unwanted result, as can be seen in figure 17. Since reinitialisation restores the level set to a signed distance function, the 'ridges' caused by the optimisation method are smoothed somewhat, as these areas are enclosed on two sides by interface segments. Due to this, these ridges are quick to disappear in the subsequent time steps. Even then, since the optimisation method consistently adds mass to the top side of the outer layers, this effect is still visible with reinitialisation, and is the reason why the outside layers have travelled less in the final result.

From table 1, it is clear that, in general, taking a more refined grid results in a lower accuracy. The only exception to this is the solid body translation case, but this result can be attributed to the test case's special nature compared to the other ones. Additionally, the solid body rotation test stands out since it is the only test that produces unstable results when using Euler Forward as time integration method.

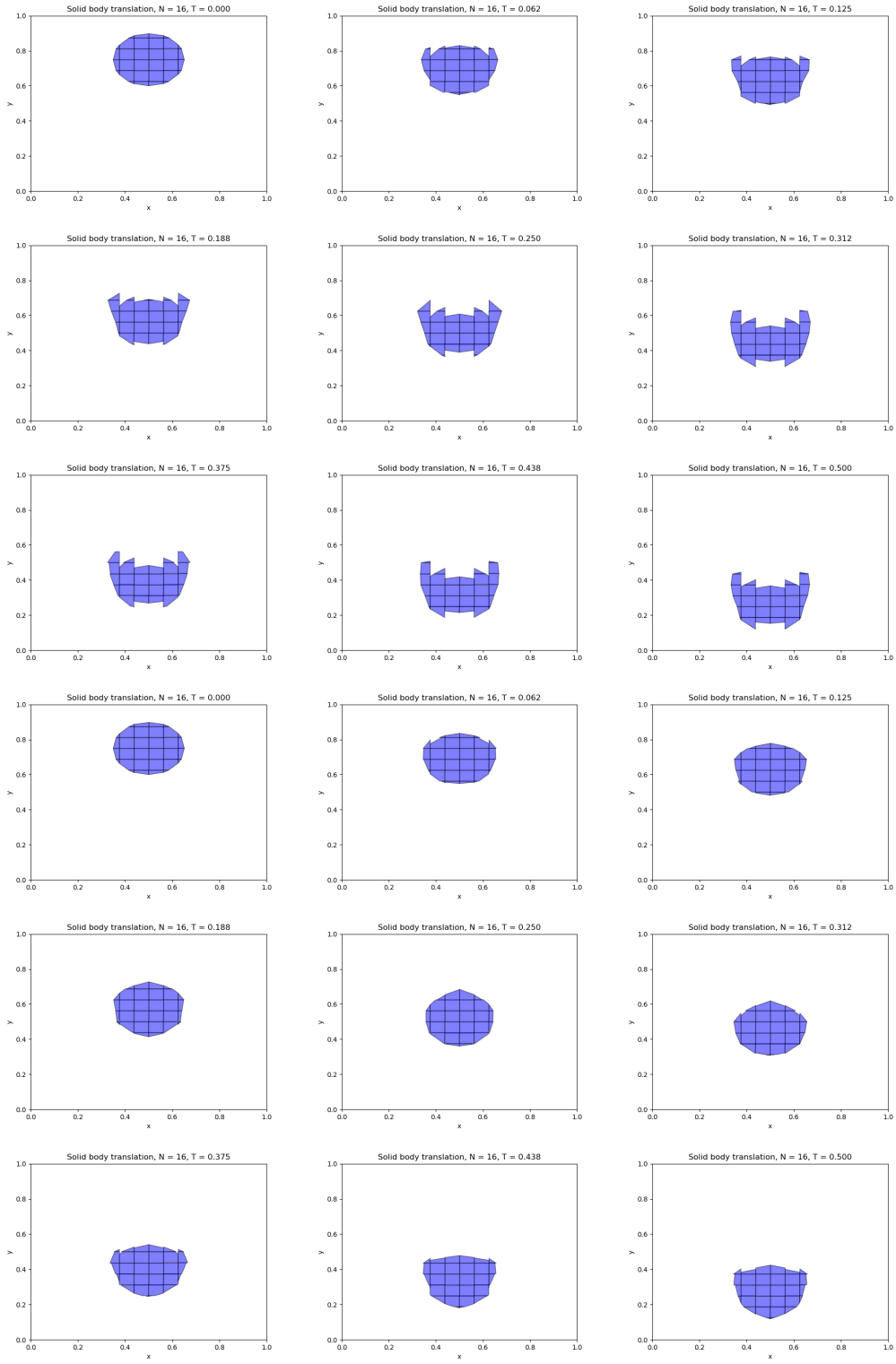


Figure 17: Solid body translation test on a 16×16 grid and time step $\Delta t = \frac{1}{128}$. The top nine figures are from the test without reinitialisation, and the bottom nine are with this extra step. The relative independence of the vertical layers mean that the optimisation method can arbitrarily add or remove mass in these layers, which causes the mass to be added in the incorrect locations. With reinitialisation, the shape is smoothed somewhat by the signed distance property, but the effect of the optimisation method is still present. This causes the outside layers to lag behind, though the solution better represents the exact interface.

The instability in the solid body rotation test is most likely an issue propagated by incorrect boundary conditions. The rotation test is the only test case that has relevant boundary conditions; the vortex deformation test has zero velocity at all boundaries, and in the translation and corner flow tests the solution propagates away from any inflow boundary. In the rotation test, however, every domain boundary is an inflow boundary on one half, and an outflow boundary on the other half, so due to the movement of the interface the boundary conditions now become relevant. In general for this kind of problem, a homogeneous Neumann condition is applied on any outflow boundary, while a Dirichlet condition is imposed on the inflow boundary. The exact implementation on the outflow boundary is irrelevant for the solution, though, since it has no impact on the level set in the interior of the domain. The Dirichlet condition is more problematic, since it essentially requires knowledge of the solution outside the domain, which is not available. In many examples in [15], the condition is found using the value of the exact solution at the boundary, which is thus also used here. It appears, however, that this is not enough to suppress the instabilities when using Euler Forward. Higher order integration methods seem to relieve this issue.

Aside from these obvious deviations, an interesting result is the effect of the time stepping method on the accuracy of the solution. While more accurate methods are required for solid body rotation to keep stability, for the other methods it may have little effect on the solution (vortex deformation), or it may even increase the error (corner flow, especially $N = 16$). The cause of this disparity is unknown, but it is likely due to the conversion from the level set to straight interfaces. Since the optimisation method attempts to join interfaces at certain cell edges, a shift in interface orientation due to a different time stepping method may result in a much larger change in mass in various regions. This could have a cumulative effect, eventually resulting in a build-up of mass in the wrong locations.

While the accuracy of the method seems to increase as the grid becomes more refined, the order of convergence is not consistent between refinements and is definitely not the order of convergence that is expected, perhaps aside from the solid body rotation test. Since the DG method with only Euler Forward is second order accurate (see table 3), this must be a result of the optimisation method. To further analyse the effect of the optimisation procedure, multiple level set advection steps are allowed between optimisation steps. This means that fewer correction steps are done, which should better preserve the shape of the levl set and improve accuracy, but each correction will then be larger than in the case where optimisation is done every advection step. The results of this are shown in table 2.

	Amount of time steps between optimisations			
	1	2	4	8
Solid body translation	0.060916	0.054398	0.058675	0.025988
Solid body rotation	0.017753	0.015282	0.013360	0.013159
Corner flow	0.072679	0.039329	0.054960	0.035148
Vortex deformation	0.045189	0.045493	0.044815	0.042152

Table 2: The effect of 'skipping' optimisation steps on the error in the volume of fluid field. This simulation was done on a 16×16 grid using the Runge Kutta 4 method and a time step size of $\Delta t = 1/128$, or half as large for the solid body rotation test. It appears that taking more time steps in between optimisation steps results in a slightly lower error, although this effect is only significant in the solid body translation and corner flow tests.

Decreasing the amount of optimisation steps does little to decrease the error for the solid body rotation and vortex deformation tests. The significant decrease for the solid body translation test is as expected, since it is the optimisation step that causes the need for level set reinitialisation. The corner flow produces the most significant result, as the solid body translation test is quite a special case, although this increase in error is still inconsistent. Since having only one optimisation step per 8 level set advection steps consistently produces the best results for each test case, the tests of table 1 are repeated with this added property. The tests are only done using Modified Euler, since

using RK4 instead gives only a very small decrease in the error and is not required for stability. These results are then compared to a pure discontinuous Galerkin advection method. This gives the results shown in table 3.

		MCLS	Pure DG	Optimisation based method
Solid body translation	$N = 8$	0.020084	0.019696	0.027837
	$N = 16$	0.005400	0.006652	0.025717
	$N = 32$	0.001928	0.002491	0.032888
Solid body rotation	$N = 8$	0.054585	0.035155	0.043412
	$N = 16$	0.014835	0.012777	0.013617
	$N = 32$	0.004397	0.003444	0.004295
Corner flow	$N = 8$	0.048457	0.014863	0.109794
	$N = 16$	0.011618	0.160412	0.035136
	$N = 32$	0.003026	0.194844	0.014572
Vortex deformation	$N = 8$	0.195660	0.100717	0.075536
	$N = 16$	0.054939	0.044733	0.042269
	$N = 32$	0.018058	0.025891	0.025870

Table 3: Volume of fluid error in the L^2 norm for the different flows on an $N \times N$ grid for the different interface methods. Time steps are $\Delta t = \frac{1}{64}$ for $N = 8$, $\frac{1}{128}$ for $N = 16$, and $\frac{1}{256}$ for $N = 32$, or half as large for the solid body rotation test. An optimisation step is done every 8 level set advection steps. The optimisation method leads to roughly second order convergence in the solid body translation and corner flow tests, though no increase in accuracy is found in the vortex deformation case.

Comparing the optimisation based method to a pure DG level set method allows for a better understanding of the effect of the optimisation method. The discontinuous Galerkin method with Modified Euler as time stepping method can be seen to have an order of convergence of slightly above than 1.5 for the solid body translation and rotation test cases. This result is somewhat as expected, since using a fourth-fifth order Runge-Kutta-Fehlberg time integration method results in a order of convergence of roughly $p + 1$ [2]. The rate of convergence from the solid body tests is, however, not mimicked in the corner flow and vortex deformation test. Especially in the corner flow case, the DG method has difficulty dealing with the relatively low width of the solution, which leads to large mass errors and thus also a large deviation in the volume of fluid field. This is likely also the cause of the roughly first order convergence in the vortex deformation test, where the loss of convergence is probably the result of errors in the 'tail' of the solution. Aside from the solid body translation test, where the optimisation method has been found to hamper the method in any case, the optimisation based method reflects the results of the pure DG method, obviously now without mass errors. It is interesting to see that the optimisation method is able to obtain this same rate of convergence in the corner flow test (see figure 18), while the vortex deformation case remains very similar to the pure DG implementation. From figures 29 and 30, it is visible that a lot of the mass error caused by the DG method is put into the parts that were split of from the main body of fluid, which results in relatively large loose areas of fluid that should be a lot smaller.

The optimisation based method is thus capable of obtaining quadratic convergence in some test cases. However, as this does not happen for the vortex deformation test, and since it requires the fine-tuning of the amount of optimisation steps, the current form of this method is too inconsistent to be used in general practice. While the method may not yet be suited for two-phase simulations, the fact that the order of convergence of the DG method can be mimicked by this method shows that, with more work, optimisation based methods may become applicable with the addition of a few additional steps.

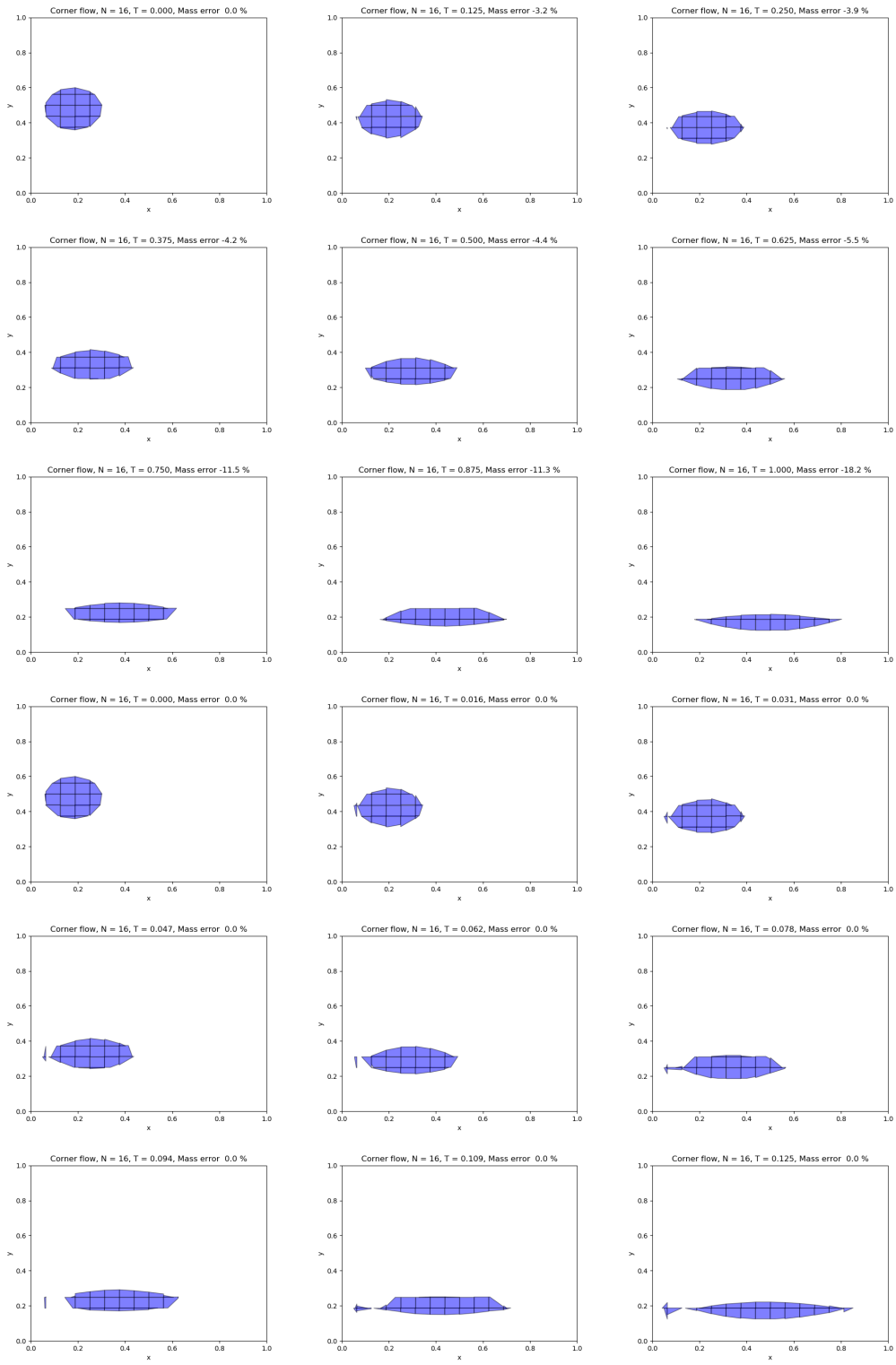


Figure 18: Corner flow test on a 16×16 grid and time step $\Delta t = \frac{1}{128}$. The top nine figures use only discontinuous Galerkin level set advection, and the bottom nine are with the optimisation based method. In the optimisation method, the shape of the DG result is clearly maintained. The mass error caused by DG is also mainly recovered near the short side of the solution.

6 Conclusion

In this thesis, an optimisation based dual interface method has been formulated and analysed, as a first step towards mass conserving interface methods on unstructured grids. The constructed method, ideally, would preserve the structure of the level set method and inherit it's accuracy, while remaining mass conserving due to a logical and justified mass correction step. Since the interface method is mainly governed by the level set advection method, the main goal is to analyse the effect of the optimisation method on the error and structure in the solutions. From the results, conclusions can be drawn about the optimisation method with regards to the mass distribution and error convergence.

Firstly, the mass distribution step of the optimisation method leaves some things to be desired. Most prominently, the lack of horizontal velocity in the solid body translation test causes the optimisation method to consistently add mass on the top few cells of the outside layer, thus destroying most of the structure present in the solution. Though this is largely an issue for the translation test, in other tests this is visible as well. Though not clearly reflected in the error analysis, visual inspection of the results show that the optimisation method, compared to the pure DG method, adds a relatively large amount of mass to the areas that are split of from the main fluid part. This is mainly an issue in the corner flow and vortex tests, where the solution becomes small enough to split apart, but is an unwanted result.

Secondly, the error analysis shows that the method appears to converge to the exact solution as the grid becomes more refined. However, the order of convergence is very inconsistent with respect to the grid refinement and the time integration method for level set advection. The only noticeable result of a more accurate time integration method is obtained in the rotation test, where these methods are required to keep stability. In order to better test the capabilities of this method, fewer optimisation steps were taken, and in this case the optimisation based method better reflected the order of convergence expected from the DG method, even for cases like the corner flow test where the DG method did not converge due to mass errors.

All together, the obtained results show that this type of method may become usable in cases where MCLS is not, but in it's current state it is too inconsistent and requires fine-tuning of the amount of optimisation steps, so it is not yet usable for general simulations.

7 Recommendations for future work

While the method detailed in this thesis did not quite obtain the convergence of the DG method (at least, not without modification), the results do show that it may be worthwhile further investigating optimisation based methods like this one. To this end, additional steps are proposed that could be incorporated into the optimisation method to improve its accuracy and reduce the likelihood of the problems that affected the current optimisation method. The recommendations given here are:

- Additional constraints for the minimisation problem
- Cluster based mass correction
- Fully nonlinear constrained optimisation problems.

Imposing additional constraints is an obvious expansion of the method, so will not be described in more detail here.

Cluster based mass correction would be an addition for the mass correction step of the algorithm. This step is mainly motivated by the solid body translation test and, when implemented, should allow this test to be done without reinitialising the level set. Since the solid body translation test as used in this thesis has a purely vertical velocity field, the flux condition only attempts to match interfaces at the top and bottom boundaries of each cell. This means that there are various vertical 'strips' of cells that can be altered independently of others. For general flows, there may be multiple clusters of cells that are connected in the same way, and are also independent of other clusters. This does mean that any change to one cell in such a cluster also must modify all other cells in the same cluster. Now, distributing the mass difference around clusters rather than isolated cells will allow the mass to be distributed more evenly among the interface cells, while still complying with the flux condition. Using nonlinear optimisation, the mass can then be distributed among the clusters in such a way that, for example, the exact centre of mass is best approximated or that some other physical constraint is applied. This would allow better control over the distribution of mass, and also ensures that the correction from the flux condition can be rectified if required. This mass correction step always has more places to put the mass difference, since the current method effectively uses only clusters of size one, but it is unknown if clusters like this are always available, especially if the flux condition creates an infeasible model and has no solution.

A fully nonlinear optimisation problem is the logical end step for this type of method. As mentioned in section 3.3, nonlinear optimisation allows for using constraints such as a specified centre of mass, or allows for minimising the curvature of an interface, which are physical constraints that should better preserve the shape of the level set solution. Additionally, using nonlinear optimisation may allow for curved interfaces inside each cell. When using higher order basis function for the DG advection method, the (possibly curved) zero level set could then be used for determining the volume of fluid, and since the flux condition only requires information from the cell edges, the added complexity should be manageable. This change, along with the fact that the flux condition joins interfaces at cell edges, should be able to give more accurate solutions, and even allow for actual circular interfaces. Additionally, this would allow for bubbles inside a cell, circular interfaces that do not have a boundary intersection, thus being able to better represent small details in the interface. Of course, such a big step introduces many questions, mainly about the possible time complexity of the resulting algorithm, as well as about the existence of a unique solution.

References

[1] Gabriel D Weymouth and Dick K-P Yue. Conservative Volume-of-Fluid method for free-surface simulations on Cartesian-grids. *Journal of Computational Physics*, 229(8):2853–2865, 2010.

[2] D Wirasaet, EJ Kubatko, CE Michoski, S Tanaka, JJ Westerink, and C Dawson. Discontinuous Galerkin methods with nodal and hybrid modal/nodal triangular, quadrilateral, and polygonal elements for nonlinear shallow water flow. *Comput. Methods Appl. Mech. Engrg*, 270:113–149, 2014.

[3] GT Oud. A dual interface method in cylindrical coordinates for two-phase pipe flows. 2017.

[4] Mark Sussman and Elbridge Gerry Puckett. A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of computational physics*, 162(2):301–337, 2000.

[5] Matthew Jemison, Mark Sussman, and Marco Arienti. Compressible, multiphase semi-implicit method with moment of fluid interface representation. *Journal of Computational Physics*, 279:182–217, 2014.

[6] Vadim Dyadechko and Mikhail Shashkov. Moment-of-fluid interface reconstruction. *Los Alamos Report LA-UR-05-7571*, 2005.

[7] Sander Pieter van der Pijl. *Computation of bubbly flows with a mass-conserving level-set method*. Citeseer, 2005.

[8] F Raees, DR Heul, and C Vuik. A mass-conserving level-set method for simulation of multiphase flow in geometrically complicated domains. *International Journal for Numerical Methods in Fluids*, 81(7):399–425, 2016.

[9] Matthew Jemison, Eva Loch, Mark Sussman, Mikhail Shashkov, Marco Arienti, Mitsuhiro Ohta, and Yaohong Wang. A coupled level set-moment of fluid method for incompressible two-phase flows. *Journal of Scientific Computing*, 54(2-3):454–491, 2013.

[10] Cyril W Hirt and Billy D Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of computational physics*, 39(1):201–225, 1981.

[11] Jeffrey W Banks, T Aslam, and William J Rider. On sub-linear convergence for linearly degenerate waves in capturing schemes. *Journal of Computational Physics*, 227(14):6985–7002, 2008.

[12] Qinghai Zhang and Aaron Fogelson. MARS: An analytic framework of interface tracking via mapping and adjusting regular semialgebraic sets. *SIAM Journal on Numerical Analysis*, 54(2):530–560, 2016.

[13] Dmitri Kuzmin. An optimization-based approach to enforcing mass conservation in level set methods. *Journal of Computational and Applied Mathematics*, 258:78–86, 2014.

[14] Christopher Basting and Dmitri Kuzmin. Optimal control for mass conservative level set methods. *Journal of Computational and Applied Mathematics*, 270:343–352, 2014.

[15] Jan S Hesthaven and Tim Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer Science & Business Media, 2007.

Appendix A: Result figures

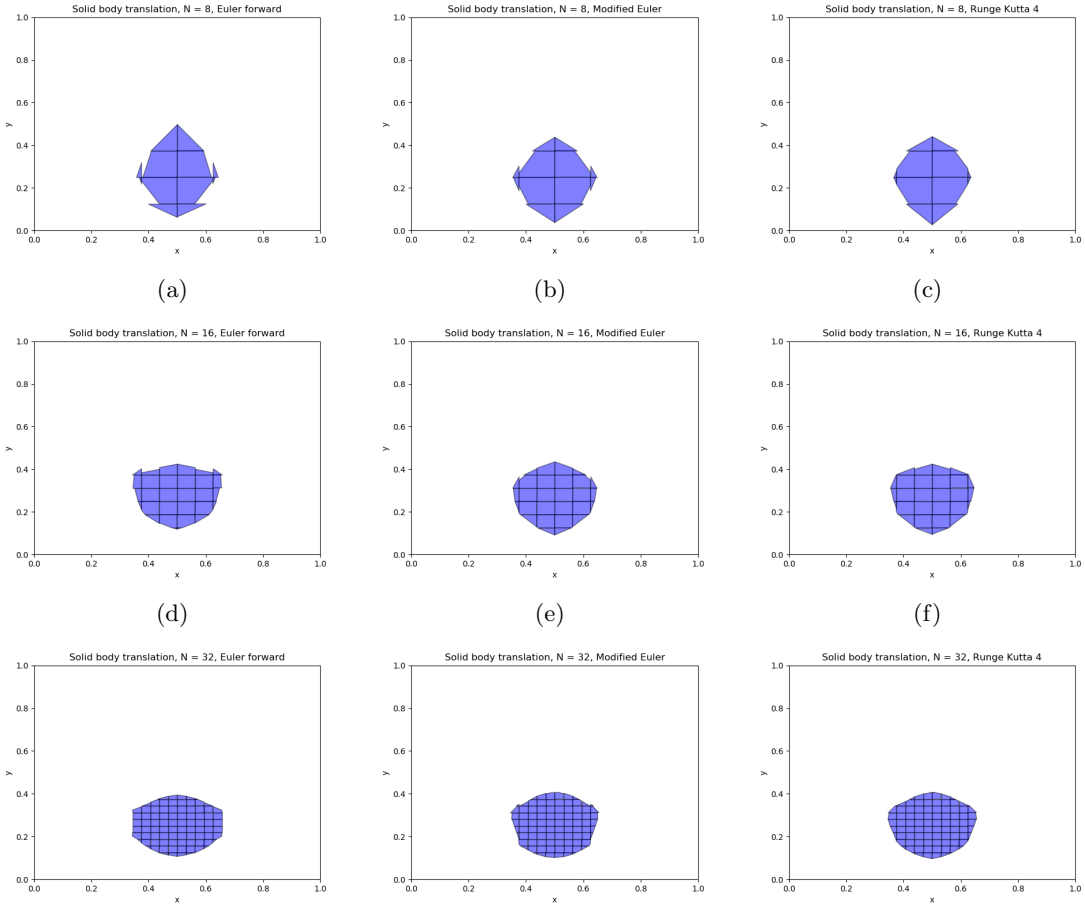


Figure 19: Volume of fluid fields for the test of table 1 using the solid body translation velocity field. This simulation is done with the use of level set reinitialisation.

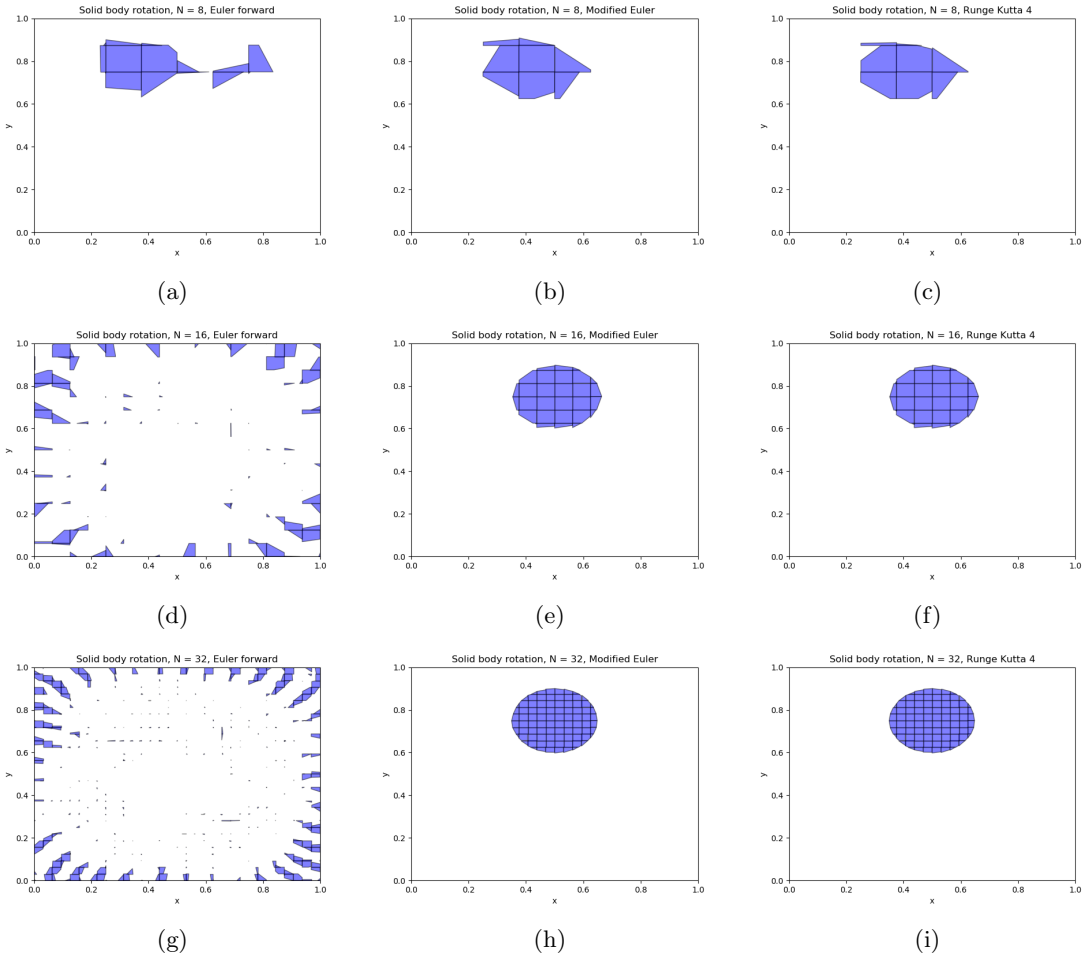


Figure 20: Volume of fluid fields for the test of table 1 using the solid body rotation velocity field. Higher order time stepping methods are required to make this test stable.

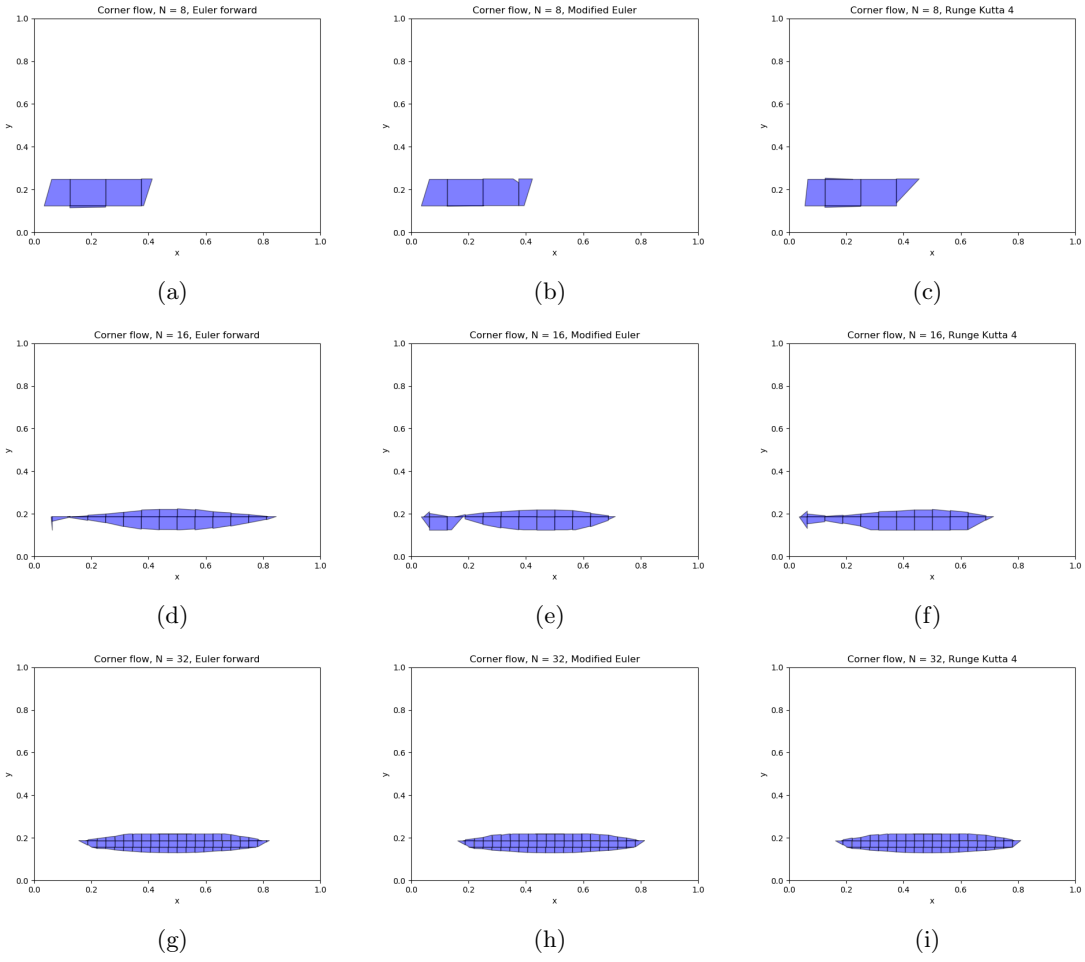


Figure 21: Volume of fluid fields for the test of table 1 for the corner flow test. The velocity field is handled much better than the solid body rotation test and no obvious instabilities are visible.

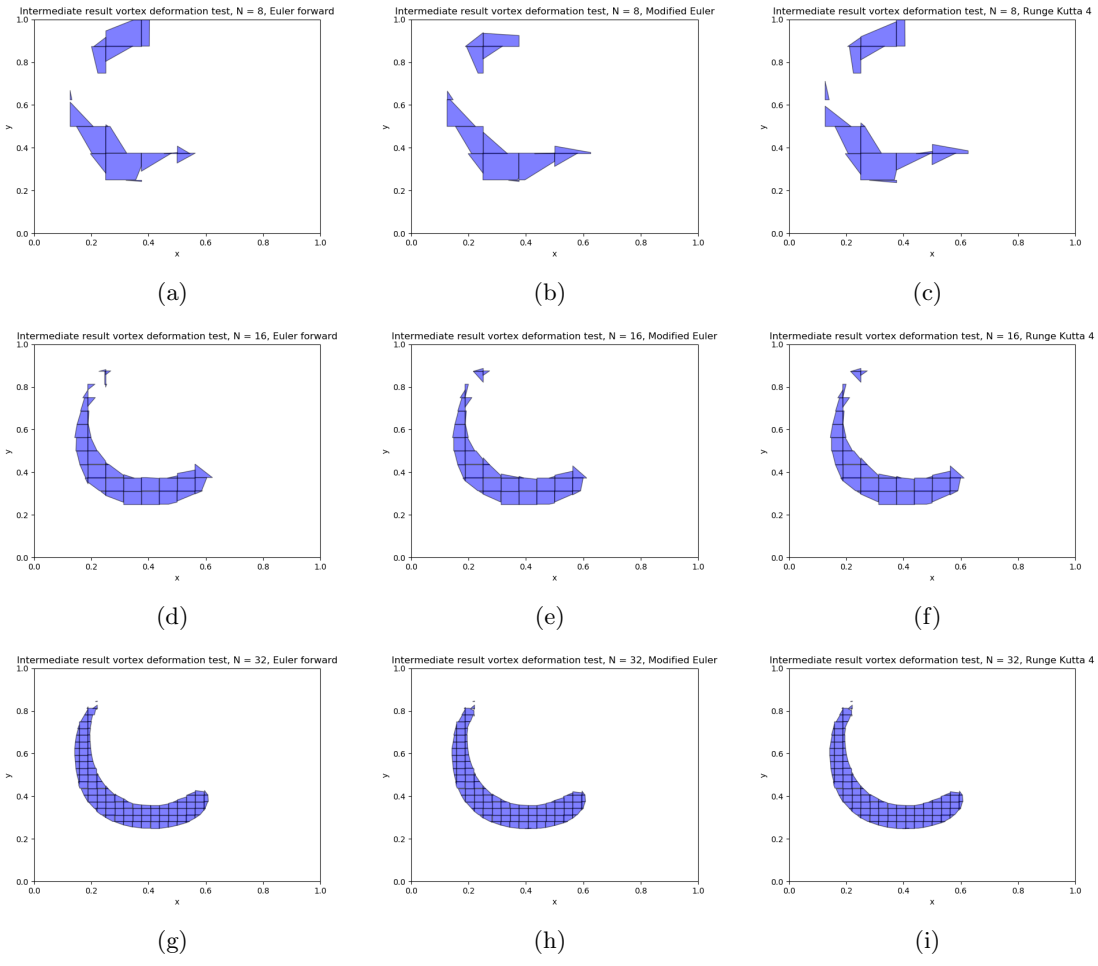
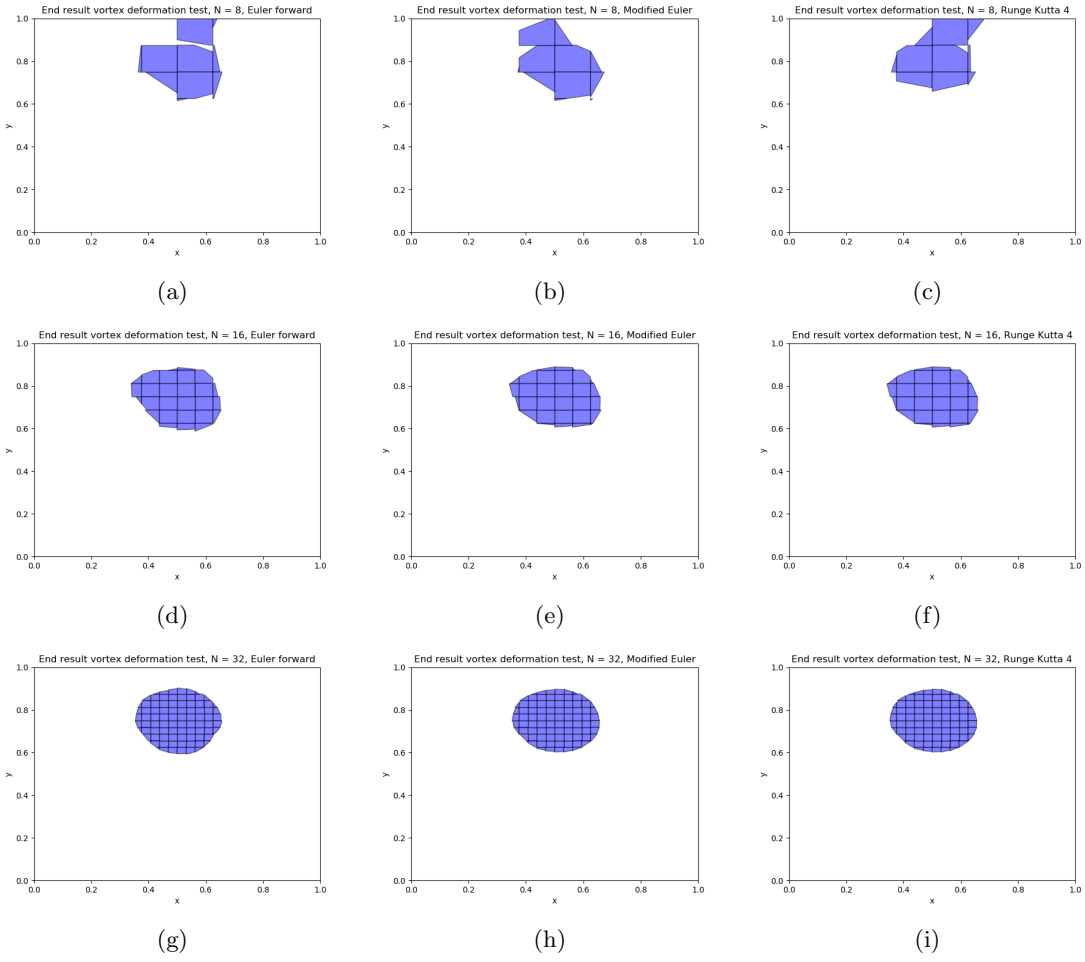


Figure 22: Intermediate volume of fluid fields for the test of table 1 for the vortex deformation test flow test. Using only an 8×8 grid will cause the solution to fall apart, but for more refined grids this is almost not the case.

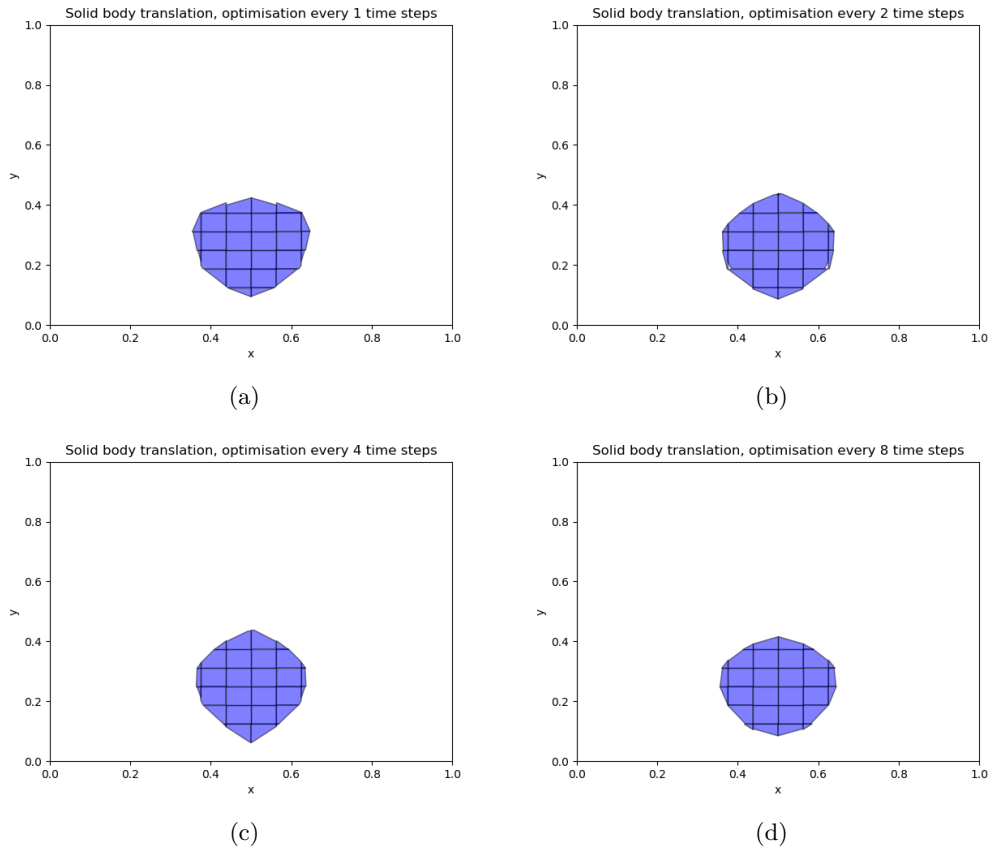


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

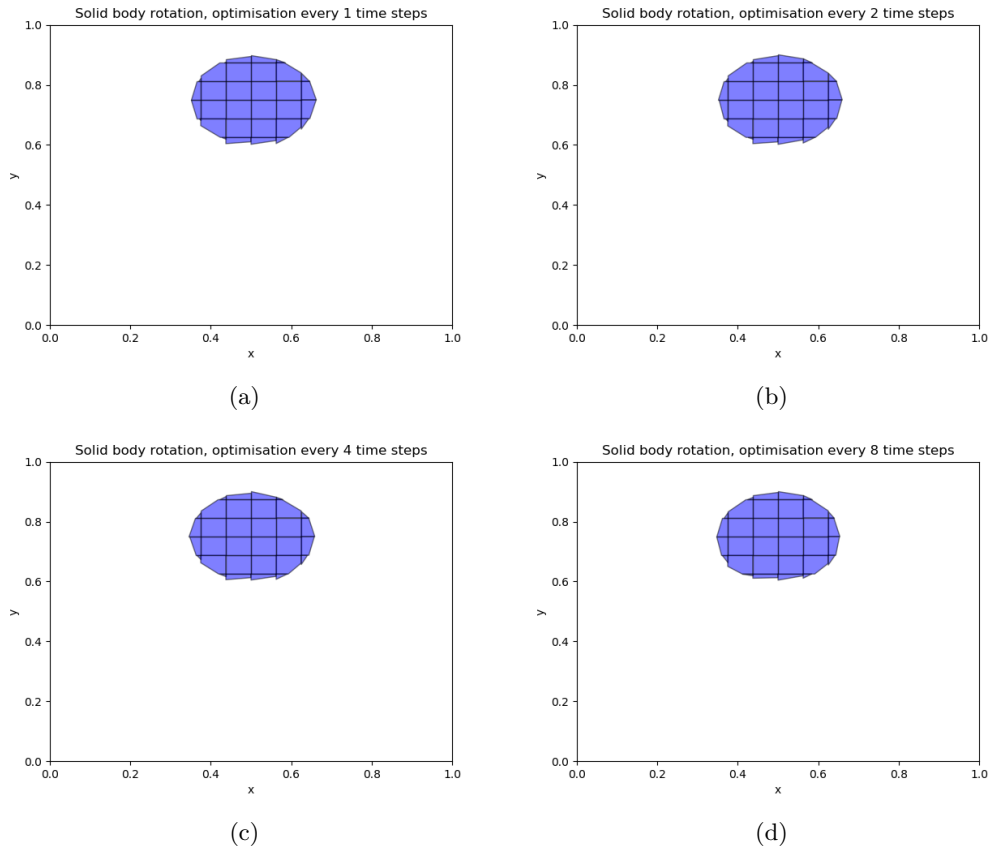
Figure 23: End results of the volume of fluid fields for the test of table 1 for the vortex deformation test flow test.



(a) (b)

(c) (d)

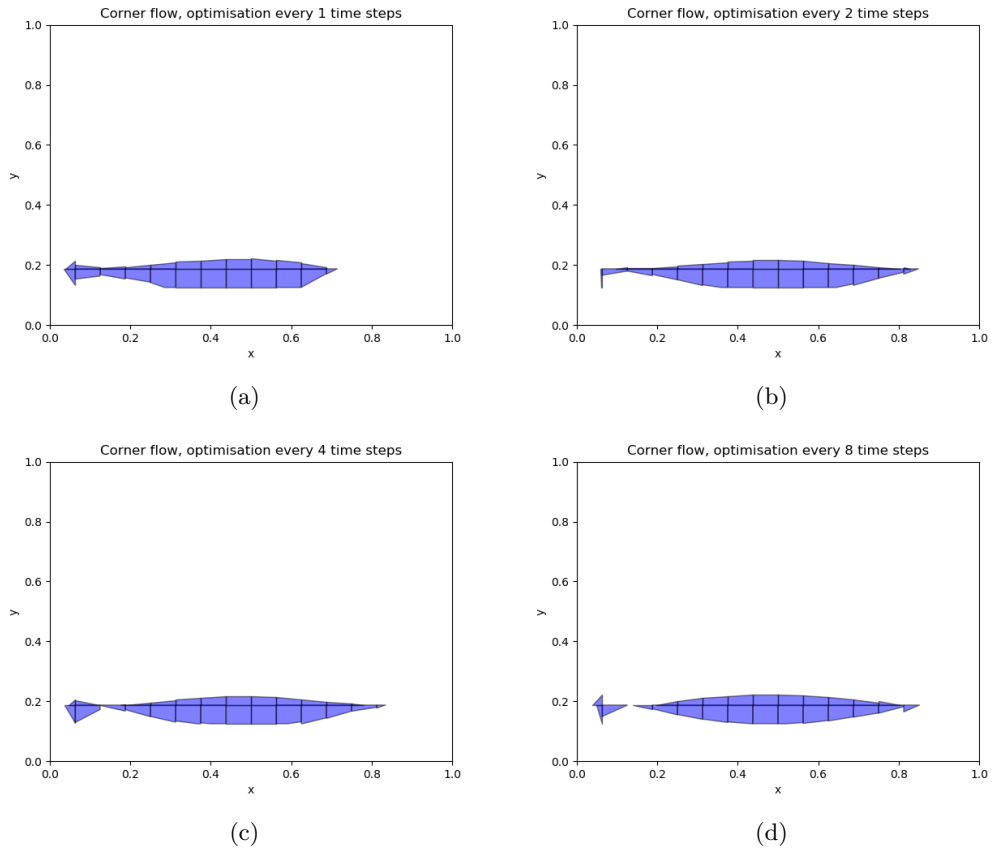
Figure 24: Volume of fluids field when taking more time steps per optimisation routine for solid body translation, table 2. Fewer optimisation steps result in a more rounded solution.



(a) (b)

(c) (d)

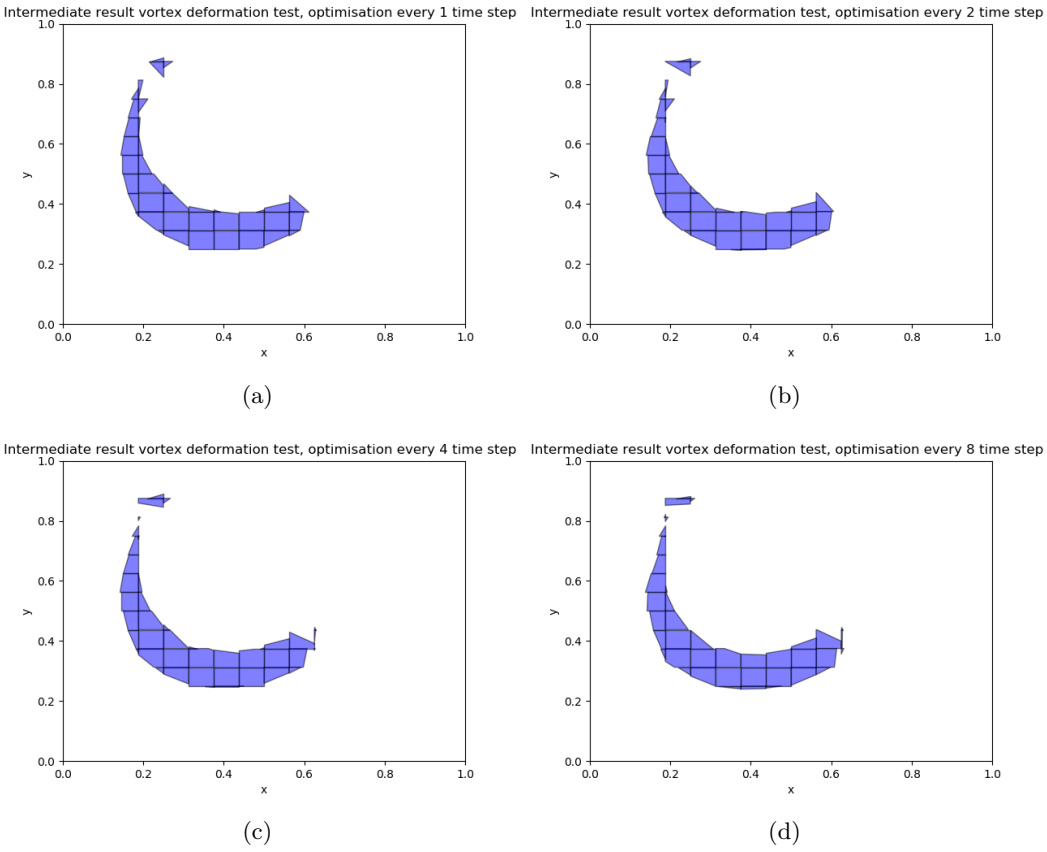
Figure 25: Volume of fluid fields when taking more time steps per optimisation routine for solid body rotation, table 2.



(a) (b)

(c) (d)

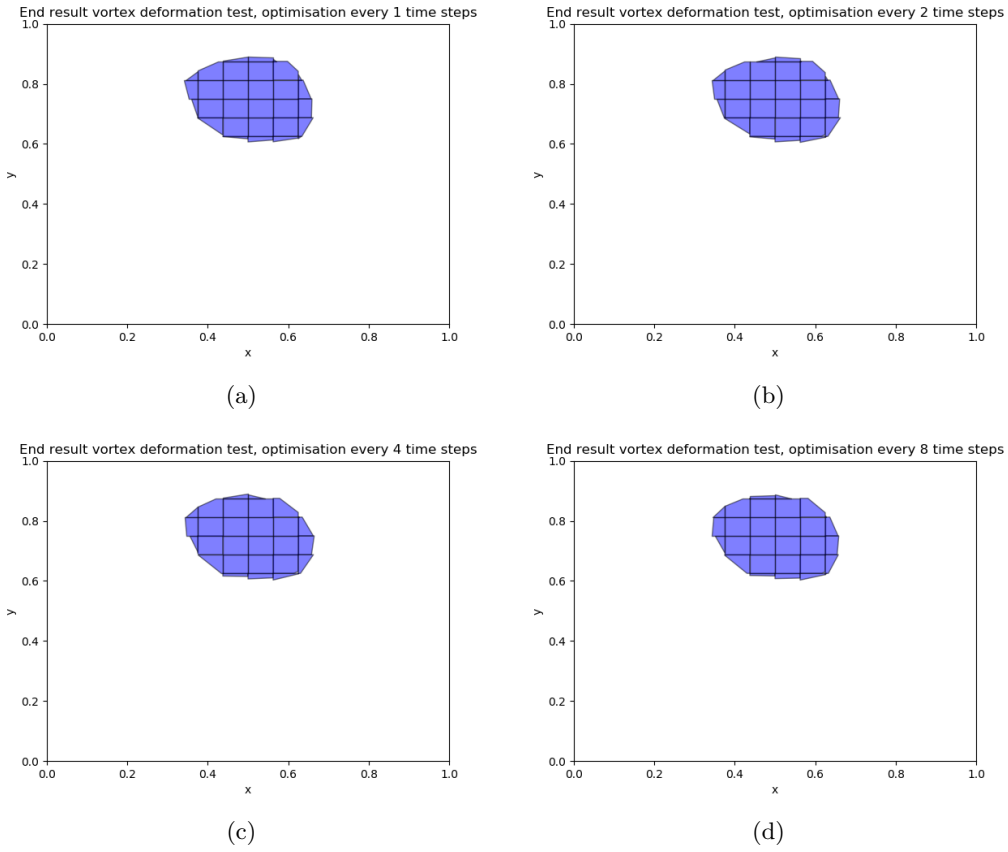
Figure 26: Volume of fluid fields when taking more time steps per optimisation routine for the corner flow, table 2. When taking fewer optimisation steps the solution becomes slightly more elongated.



(a) (b)

(c) (d)

Figure 27: Intermediate volume of fluid fields when taking more time steps per optimisation routine for the vortex deformation test, table 2. The shape of the tail of the solution fluctuates when taking fewer optimisation steps.



(a) (b)

(c) (d)

Figure 28: Final volume of fluid fields when taking more time steps per optimisation routine for the vortex deformation test, table 2. The general shape and location of the solution does not change much in this test.

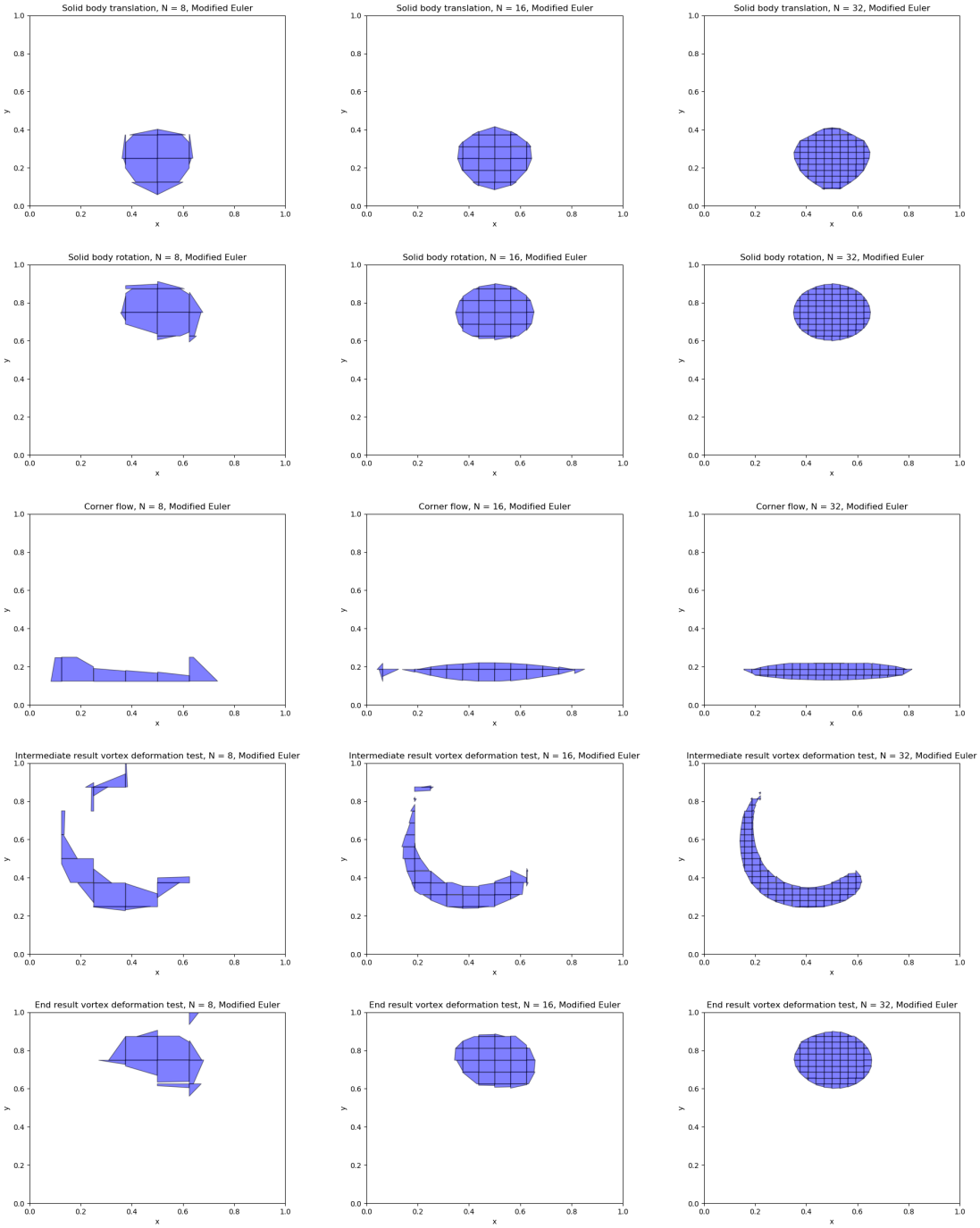


Figure 29: Results of the optimisation based method of the test cases for table 3.

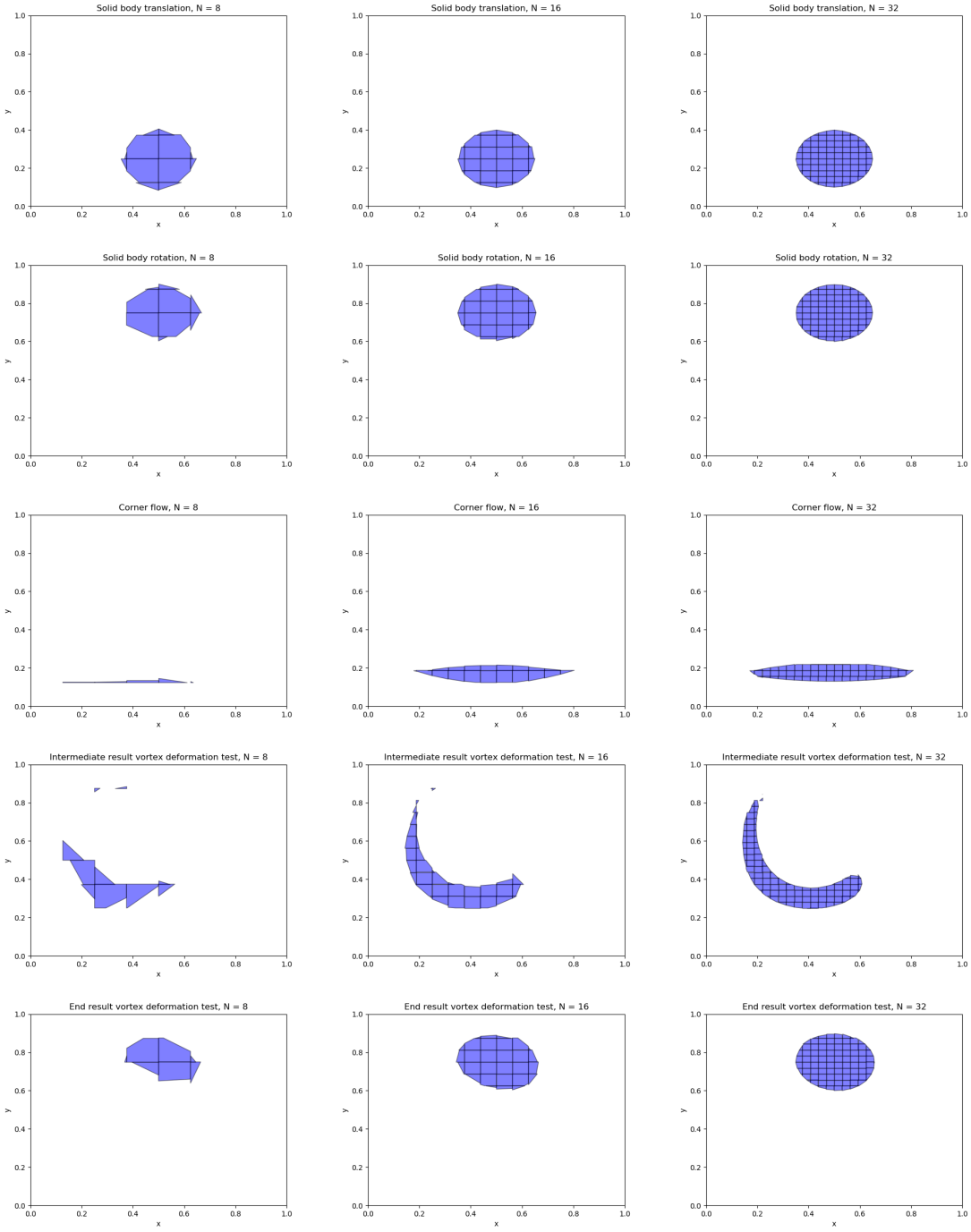


Figure 30: Results of the discontinuous Galerkin method of the test cases for table 3.

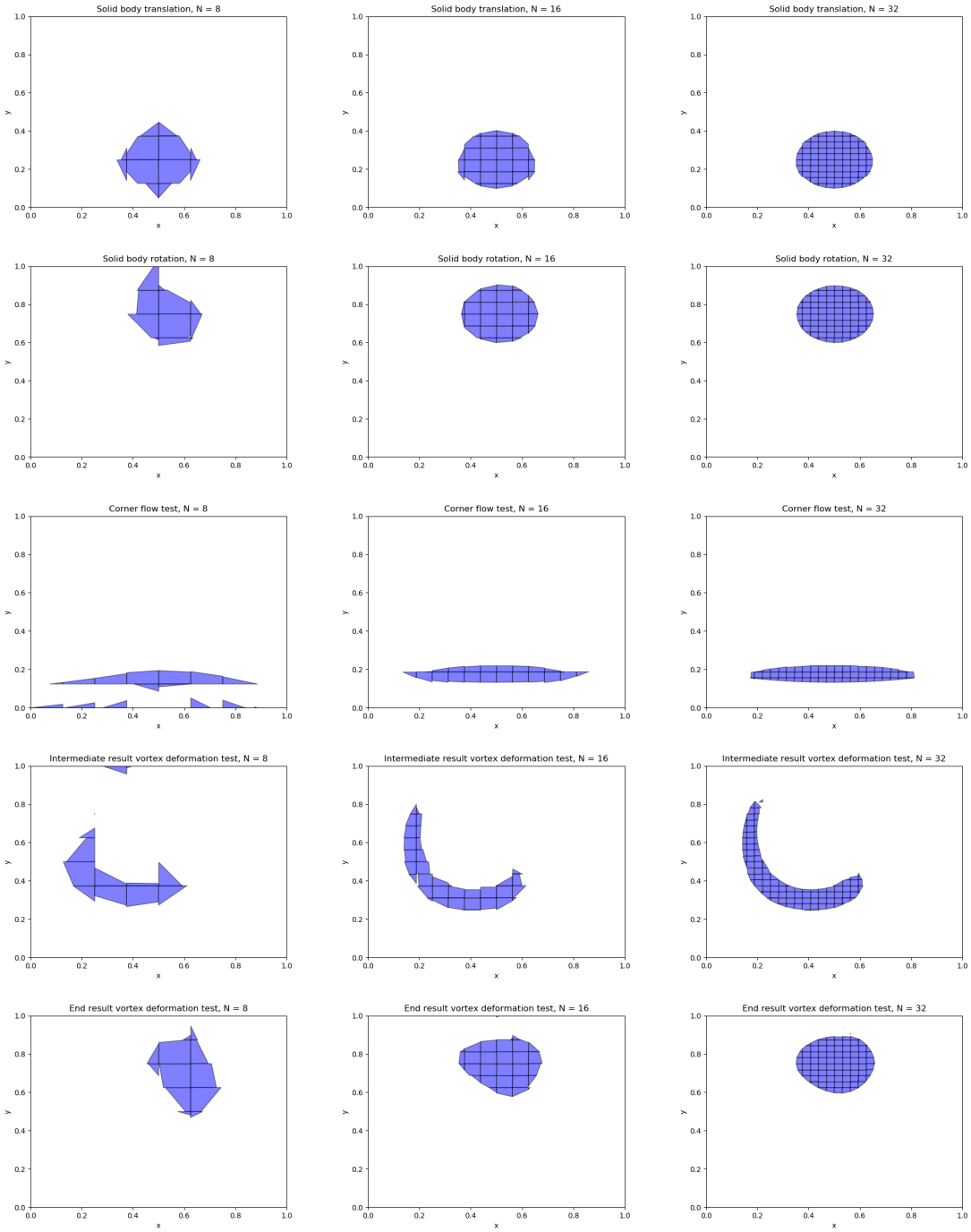


Figure 31: Results of the MCLS method of the test cases for table 3. The $N = 8$ grid is very coarse and can lead to strange interface orientations near the domain boundaries.