

A Space-Time Discontinuous Galerkin Finite Element Method for Two-Fluid Problems

W.E.H. Sollie, J.J.W. van der Vegt* and O. Bokhove

Department of Applied Mathematics, Institute of Mechanics, Processes and Control Twente, University of Twente, P.O.Box 217, 7500 AE, Enschede, The Netherlands

Abstract

A space-time discontinuous Galerkin finite element method for two fluid flow problems is presented. By using a combination of level set and cut-cell methods the interface between two fluids is tracked in space-time. The movement of the interface in space-time is calculated by solving the level set equation, where the interface geometry is identified with the 0-level set. To enhance the accuracy of the interface approximation the level set function is advected with the interface velocity, which for this purpose is extended into the domain. Close to the interface the mesh is locally refined in such a way that the 0-level set coincides with a set of faces in the mesh. The two fluid flow equations are solved on this refined mesh. The procedure is repeated until both the mesh and the flow solution have converged to a reasonable accuracy. The method is tested on linear advection and Euler shock tube problems involving ideal gas and compressible bubbly magma. Oscillations around the interface are eliminated by choosing a suitable interface flux.

Key words: cut-cell method, discontinuous Galerkin finite element method, interface tracking, level set method, space-time, two fluid flows.

1 Introduction

Moving interfaces are important in many fluid problems including free surface, multifluid and multiphase flows. Frequently the interface topology and flow fields are coupled and need to be solved simultaneously. For example, interface

* Corresponding author.

Email addresses: w.e.h.sollie@math.utwente.nl (W.E.H. Sollie),
j.j.w.vandervegt@math.utwente.nl (J.J.W. van der Vegt),
o.bokhove@math.utwente.nl (O. Bokhove).

curvature and tension can cause a pressure jump across an interface, which influences the flow field. Also, pressure variations in the bulk flow fields can bring about changes in the interface shape as well. In addition, topological changes like breakup and coalescence can be of importance.

To solve problems involving moving interfaces numerically often a macroscopic view is adopted, where the interface is modelled as a hypersurface separating two fluids, together with certain interface conditions that need to be satisfied. In the literature many methods have been proposed for computing flows with interfaces or, to be more general, fronts. One way to classify these methods is by looking at the front representation in the mesh. Firstly, in front capturing methods a regular stationary mesh is used and there is no explicit front representation. Instead, the front is either described by means of marker particles, like in the marker and cell method, or by means of functions defined on the mesh, such as in the volume of fluid and level set method. Secondly, in front tracking and Lagrangian methods the front is tracked explicitly in the mesh. Other methods include particle methods and boundary integral methods.

The method presented in this article combines front capturing and front tracking methods using the space-time framework together with a discontinuous Galerkin (DG) discretization. This new approach provides an accurate and versatile scheme for dealing with interfaces in two fluid flow problems which can alleviate some of the problems encountered with front tracking and front capturing methods. In order to motivate the choices made in this algorithm, first a summary is given of the main aspects of the most important existing techniques to deal with interfaces.

Front capturing methods have the advantage of a relatively simple formulation. The main drawback of these methods lies in the need for complex interface shape restoration techniques, which often have problems in restoring the smooth and continuous interface shape, particularly in higher dimensions. Front tracking methods can reach high accuracy when the interface representation is detailed enough. One drawback of front tracking methods is, however, that they are hard to implement in higher dimensions due to the complexity of the geometric refinement. Also, topological changes typically cannot be handled. Another drawback is the occurrence of small elements which can give problems with the stiffness of the equations and numerical stability. Lagrangian front tracking methods typically also have difficulty with mesh deformation and may therefore require frequent remeshing.

The earliest numerical method for time dependent free surface flow problems was the marker and cell (MAC) method ([2], [3]). Being a volume marker method it uses tracers or marker particles defined in a fixed mesh to locate the phases. However, the large number of markers required make the method expensive.

In the Volume Of Fluid (VOF) method ([4], [5], [6], [7]) a fractional volume or color function is defined to indicate the fraction of a mesh element that covers a particular type of fluid. Algorithms for volume tracking are designed to solve the equation $\partial c/\partial t + \bar{\nabla} \cdot (c \mathbf{u}) = 0$, where c is the color function, \mathbf{u} the velocity, t the time and $\bar{\nabla} = (\partial/\partial x_1, \dots, \partial/\partial x_d)$ the spatial gradient operator in d -dimensional space. In the VOF method typically a reconstruction step is necessary to reproduce the interface geometry from the color function. Higher accuracy VOF techniques like the Piecewise Linear Interface Construction (PLIC) method attempt to fit the interface by means of piecewise linear segments. VOF methods are easy to extend to higher dimensions and can be parallelized readily due to the local nature of the scheme. Also, they can automatically handle reconnection and breakup. However, VOF methods have difficulty in maintaining sharp boundaries between different fluids, and interfaces tend to smear. In addition, these methods can give inaccurate results when high interface curvatures occur. Also, the computation of surface tension is not straightforward. While VOF methods conserve mass well, spurious bubbles and drops may be created. Recent developments include the combination of the VOF method with the Level Set Method [8].

The Level Set Method (LSM) was introduced by Osher and Sethian in [9] and further developed in [10], [11], [12]. For a survey, see [13]. The LSM uses an implicit representation of the interface by means of a level set function $\psi(\mathbf{x}, t)$, where the interface is represented by the 0-level $\psi(\mathbf{x}, t) = 0$. The evolution of the interface is found by solving the level set equation $\partial\psi/\partial t + \mathbf{u}_{ext} \cdot \bar{\nabla}\psi = 0$. The velocity \mathbf{u}_{ext} is an extension of the interface velocity into the domain. It is constructed every time step by solving $\bar{\nabla}\mathbf{u}_{ext} \cdot \bar{\nabla}\psi = 0$ outward from the interface on which \mathbf{u}_{ext} equals the known interface velocity. To reduce the computational costs a narrow band approach, which limits the computations to a thin region around the interface, can be used. Over time the level set can become distorted and reinitialization may be required. Although the choice of the level set function is somewhat arbitrary, the signed distance to the interface seems to give the best accuracy in computing the curvature. Also, the LSM is easy to extend to higher dimensions and can automatically handle reconnection and breakup. However, the LSM is not conservative in itself.

Front tracking was initially proposed in [14] and further developed in [15], [16], [17], [18], [19], [20], [21] and [22]. For a survey, see [23] and [24]. In front tracking methods the evolution of the front is calculated by solving the equation $\partial\mathbf{x}/\partial t = \mathbf{u}$ at the front, where \mathbf{x} is a coordinate at the front and \mathbf{u} is the velocity. Front tracking methods are often combined with surface markers to define the location of the front. Recently, the front tracking method has been combined with the cut-cell method ([25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38]), also referred to as the embedded boundary method or the Cartesian mesh method. In the cut-cell method a Cartesian mesh is used for all elements except those which are intersected by the front.

These elements are refined in such a way that the front coincides with the mesh. Away from the front the mesh remains Cartesian and computations are less expensive. A common problem with cut-cell methods is the creation of very small elements which leads to problems with the stiffness of the equations and numerical instability. One way to solve this problem is by element merging as proposed in [39] and [40]. Because of the explicit interface representation front tracking methods are good candidates for solving problems that involve complex interface physics. They are robust and can reach high accuracy when the interface is represented using higher order polynomials, even on coarse meshes. Also, topological changes do not occur without explicit action. A drawback of front tracking methods is that they require a significant effort to implement, especially in higher dimensions.

In Lagrangian or moving mesh methods ([41], [42], [43], [44], [45], [46], [47]) the mesh is modified to follow the fluid. In these methods the mesh can become deformed considerably, which gives problems with the mesh topology and stretched elements. In the worst case, frequent remeshing may be necessary. In cases of breakup and coalescence, where the interface topology changes, these methods tend to fail.

In this paper a novel method is presented for numerically solving two fluid flow problems, which combines the LSM with front tracking and a cut-cell approach. The interface is represented explicitly in both space and time allowing for high accuracy to be achieved for the interface position and shape and also for the flow field approximation. The method uses a space-time Cartesian background mesh, that is refined near the interface. Firstly, this has the advantage that away from the interface the elements are shaped regularly and computations are cheaper. Secondly, if the accuracy of the interface representation in the mesh is good enough, the accuracy of the flow solutions will also improve. In addition, computing in space-time allows some control over topology changes, which can be dealt with by means of mesh refinement. The interface evolution is computed by means of the LSM, which uses an implicit description of the interface in space-time. The LSM can handle topology changes and also allows for an easy calculation of the interface curvature. For numerically solving the level set and the flow equations the Space-Time Discontinuous Galerkin (STDG) finite element method is used ([48], [49], [50], [51], [52], [53]). The discontinuous Galerkin (DG) finite element method was first proposed in [54] and further developed for systems of hyperbolic conservation laws (RKDG) in [55], [56], [57], [58], [59]. Also, see [60], [61], [62], [63], [48], [Tassi et.al.] and for a survey [64]. Recently [65] combined the DG finite element method with an advancing front strategy in space-time. In the STDG method the solution is allowed to be discontinuous at the element faces and hence jumps in the flow variables that occur at the interface are handled naturally. The DG finite element method provides a conservative discretization which means that artificial mass loss or gain can not occur. In addition the DG

finite element method can easily be used in combination with *hp*-refinement and parallelized. The STDG finite element method is well suited for dealing with interface problems, since it allows the solution to be discontinuous at the interface and also because the scheme is locally conservative. In addition, the STDG finite element method is unconditionally stable which is an advantage when dealing with very small cut-cells. The interface conditions are dealt with by incorporating them in a suitable interface flux. Since both the LSM and the STDG finite element method can be formulated independent of the dimension, a large part of the method presented is dimension independent, except for the refinement strategy near the interface.

The outline of this article is as follows. In Section 2 the flow, level set and extension velocity equations are introduced. The STDG discretizations are presented in Section 3, followed in Section 4 by the two fluid mesh refinement. In Section 5 the results of a number of model problems in one spatial dimension are presented and in Section 6 various aspects of the two fluid method are discussed based on the test results.

2 Level set and two fluid flow equations

2.1 Two fluid flow equations

Considered are two fluid flow problems on an open domain $\mathcal{E} \subset \mathbb{R}^{d+1}$ in space-time, with d the spatial dimension. The flow domain at any time $t \in [t_0, T]$ is defined as $\Omega(t) = \{\bar{\mathbf{x}} \in \mathbb{R}^d | (t, \bar{\mathbf{x}}) \in \mathcal{E}\}$ with t_0 the initial time, T the end time and $\bar{\mathbf{x}} = (x_1, \dots, x_d)$ the spatial coordinates. Let $\mathbf{x} = (t, \bar{\mathbf{x}}) = (x_0, \dots, x_d)$ denote the space-time coordinates. The space-time domain boundary $\partial\mathcal{E}$ is composed of the initial and final flow domains $\Omega(t_0)$ and $\Omega(T)$ and $\mathcal{Q} = \{\mathbf{x} \in \partial\mathcal{E} | t_0 < t < T\}$. Let the two fluids be separated in space-time by an interface S . The vector of conserved variables will be denoted by \mathbf{w}^i , where $i = 1, 2$ is the fluid index. Furthermore, let the bulk fluid dynamics be given as a system of conservation laws:

$$\frac{\partial \mathbf{w}^i}{\partial t} + \bar{\nabla} \cdot F^{F,i}(\mathbf{w}^i) = 0, \quad i = 1, 2, \quad (1)$$

where $\bar{\nabla} = (\frac{\partial}{\partial x_1}, \dots, \frac{\partial}{\partial x_d})$ denotes the spatial gradient operator and $F^{F,i}(\mathbf{w}^i) = (F_1^{F,i}, \dots, F_d^{F,i})$ the spatial flux tensor for fluid i with $F_j^{F,i}$ the j -th flux vector, $j = 1, \dots, d$. Equation (1) can be reformulated in space-time as:

$$\begin{aligned} \nabla \cdot \mathcal{F}^{F,i}(\mathbf{w}^i) &= 0, \\ \mathcal{F}^{F,i}(\mathbf{w}^i) &= (\mathbf{w}^i, F^{F,i}(\mathbf{w}^i)), \quad i = 1, 2, \end{aligned} \quad (2)$$

with $\nabla = (\frac{\partial}{\partial t}, \bar{\nabla})$ denoting the space-time gradient operator and $\mathcal{F}^{F,i}(\mathbf{w}^i)$ the space-time flux tensor. The flow variables are subject to initial conditions:

$$\mathbf{w}^i(0, \bar{\mathbf{x}}) = \mathbf{w}_0^i(\bar{\mathbf{x}}), \quad (3)$$

boundary conditions:

$$\mathbf{w}^i(t, \bar{\mathbf{x}}) = \mathcal{B}(\mathbf{w}^i, \mathbf{w}_w^i) \text{ on } \mathcal{Q} \quad (4)$$

with \mathbf{w}_w^i the prescribed boundary data, and interface conditions. The actual flow variables, fluxes and initial, boundary and interface conditions are problem specific and shall be provided when the test cases are discussed.

2.2 Level set equation

To distinguish between the two fluids a level set function $\psi(\mathbf{x})$ is used:

$$\psi(t, \bar{\mathbf{x}}) = \begin{cases} < 0 & \text{in Fluid 1} \\ > 0 & \text{in Fluid 2} \\ = 0 & \text{on the interface.} \end{cases} \quad (5)$$

The level set function is initially defined as the minimum signed distance to the interface:

$$\psi(t, \bar{\mathbf{x}}) = \alpha \inf_{\forall \bar{\mathbf{x}}_S \in S(t)} \|\bar{\mathbf{x}} - \bar{\mathbf{x}}_S\|, \quad (6)$$

where $\alpha = -1$ in Fluid 1 and $\alpha = +1$ in Fluid 2, $\bar{\mathbf{x}}_S$ denotes a point on the interface $S(t)$ and $\|\cdot\|$ is the Euclidian distance. The evolution of the level set is determined by an advection equation:

$$\frac{\partial \psi}{\partial t} + \bar{\mathbf{a}} \cdot \bar{\nabla} \psi = 0, \quad (7)$$

where $\bar{\mathbf{a}} = (a_1, \dots, a_d)$ is a vector containing the extension velocity. At the interface $\psi(\mathbf{x}) = 0$, hence $\bar{\nabla} \cdot (\bar{\mathbf{a}}\psi) = \psi \bar{\nabla} \cdot \bar{\mathbf{a}} + \bar{\mathbf{a}} \cdot \bar{\nabla} \psi = \bar{\mathbf{a}} \cdot \bar{\nabla} \psi$ holds. Therefore instead of an advection equation also a conservative formulation can be used, which results in a simpler discontinuous Galerkin discretization:

$$\frac{\partial \psi}{\partial t} + \bar{\nabla} \cdot (\bar{\mathbf{a}}\psi) = 0. \quad (8)$$

The level set formulation in space-time can now be stated as:

$$\begin{aligned} \nabla \cdot \mathcal{F}^L(\psi, \mathbf{a}) &= 0 \\ \mathcal{F}^L(\psi, \mathbf{a}) &= (\psi, F^L(\psi)) = \mathbf{a}\psi \end{aligned} \quad (9)$$

with \mathcal{F}^L the space-time flux for the level set and $\mathbf{a} = (1, \bar{\mathbf{a}})$. The level set function is subject to initial and boundary conditions:

$$\begin{aligned}\psi(0, \bar{\mathbf{x}}) &= \psi_0(\bar{\mathbf{x}}), \text{ for } \bar{\mathbf{x}} \in \Omega(t_0) \\ \psi(t, \bar{\mathbf{x}}) &= \psi^-(t, \bar{\mathbf{x}}) \text{ for } \bar{\mathbf{x}} \in \mathcal{Q}\end{aligned}\quad (10)$$

with $\psi^-(t, \bar{\mathbf{x}})$ the limit taken from the inside of the space-time domain. The velocity vector $\bar{\mathbf{a}}$ is an extension of the interface velocity into the domain and is found by solving

$$\text{sign}(\psi) \frac{\bar{\nabla}\psi}{|\bar{\nabla}\psi|} \cdot \bar{\nabla}a_i = 0, \text{ for } i = 0, \dots, d-1, \quad (11)$$

where the level set sign is added to ensure that the equation is solved in the direction away from the interface. Now, since $\bar{\nabla}\psi \cdot \bar{\nabla}a_i = \bar{\nabla} \cdot (a_i \bar{\nabla}\psi) - a_i \bar{\nabla} \cdot \bar{\nabla}\psi$ and using that because near the interface the level set ψ is a linear function, $\bar{\nabla} \cdot \bar{\nabla}\psi = 0$, hence equation (11) can be rewritten as:

$$\text{sign}(\psi) \frac{1}{|\bar{\nabla}\psi|} \bar{\nabla} \cdot (a_i \bar{\nabla}\psi) = 0, \text{ for } i = 0, \dots, d-1. \quad (12)$$

The sign of the level set $\text{sign}(\psi)$ is smooth everywhere except at the interface. Also, $|\bar{\nabla}\psi| = 1$ everywhere except at the points where $\bar{\nabla}\psi$ changes sign, which is typically at some distance away from the interface where the exact shape of the level set is of less importance. Therefore instead of (12) a conservative form is used for the extension velocity:

$$\bar{\nabla} \cdot \left(\text{sign}(\psi) a_i \frac{\bar{\nabla}\psi}{|\bar{\nabla}\psi|} \right) = 0, \text{ for } i = 0, \dots, d-1, \quad (13)$$

which can be reformulated in space-time as:

$$\begin{aligned}\nabla \cdot \mathcal{F}_i^A(\psi, a_i) &= 0 \\ \mathcal{F}_i^A(\psi, a_i) &= \left(0, \text{sign}(\psi) a_i \frac{\bar{\nabla}\psi}{|\bar{\nabla}\psi|} \right), \text{ for } i = 0, \dots, d-1,\end{aligned}\quad (14)$$

with \mathcal{F}_i^A the space-time flux for the extension velocity. The extension velocity is subject to initial and boundary conditions:

$$\begin{aligned}\bar{\mathbf{a}}(t, \bar{\mathbf{x}}) &= \bar{\mathbf{u}}^S(t, \bar{\mathbf{x}}) \text{ on } S(t, \bar{\mathbf{x}}), \\ \bar{\mathbf{a}}(t, \bar{\mathbf{x}}) &= \bar{\mathbf{a}}^-(t, \bar{\mathbf{x}}) \text{ on } \mathcal{Q}\end{aligned}\quad (15)$$

with $\bar{\mathbf{u}}^S$ the interface velocity. The level set, extension velocity and flow equations (9), (14) and (2) are coupled and are solved using the space-time discontinuous Galerkin finite element method discussed in Section 3.

3 Space-time discontinuous Galerkin discretization

In this section the space-time discontinuous Galerkin discretizations for the level set, extension velocity and fluid flow equations on one space-time slab are discussed for a given mesh refinement. The details of the mesh refinement and the two fluid algorithm will be explained in Section 4.

3.1 Computational mesh

To simplify the computations, the domain \mathcal{E} is subdivided into a number of time slabs on which the equations are solved consecutively. In order to define space-time slabs the time interval (t_0, T) is subdivided into N_t intervals $I_n = (t_n, t_{n+1})$, with $t_0 < t_1 < \dots < t_{N_t} = T$. The intervals are used to subdivide the domain \mathcal{E} into N_t space-time slabs $\mathcal{J}_n = \{\mathbf{x} \in \mathcal{E} | t \in I_n\}$. Let for the space-time slab \mathcal{J}_n a tessellation \mathcal{T}_h^n of space-time elements $\mathcal{K}_j^n \subset \mathbb{R}^{d+1}$ be defined as:

$$\mathcal{T}_h^n = \left\{ \mathcal{K}_j^n \mid \bigcup_{j=0}^{N_x-1} \bar{\mathcal{K}}_j^n = \bar{\mathcal{J}}_n \text{ and } \mathcal{K}_j^n \cap \mathcal{K}_{j'}^n = \emptyset \text{ if } j \neq j', 1 \leq j, j' \leq N_x - 1 \right\} \quad (16)$$

with N_x the number of space-time elements and the bar representing the element closure. It is assumed that every element in \mathcal{T}_h^n contains exactly one fluid.

3.2 Finite element basisfunctions

The finite element broken space $B_h^k(\mathcal{T}_h^n)$ associated with the tessellation \mathcal{T}_h^n is defined as:

$$B_h^k(\mathcal{T}_h^n) = \{U \in L^2(\mathcal{E}_h) : U|_{\mathcal{K}} \circ G_{\mathcal{K}} \in P^k(\hat{\mathcal{K}}), \forall \mathcal{K} \in \mathcal{T}_h\} \quad (17)$$

with \mathcal{E}_h the discrete flow domain, $L^2(\mathcal{E}_h)$ the space of square integrable functions on \mathcal{E}_h and $P^k(\hat{\mathcal{K}})$ the space of polynomials of degree at most k in element \mathcal{K} . The mapping $G_{\mathcal{K}_j^n}$ relates every element \mathcal{K}_j^n to a reference element $\hat{\mathcal{K}} \subset \mathbb{R}^{d+1}$:

$$G_{\mathcal{K}_j^n} : \hat{\mathcal{K}} \rightarrow \mathcal{K}_j^n : \xi \mapsto \mathbf{x} = \sum_{i=0}^{N_F-1} x_i(\mathcal{K}_j^n) \chi_i(\xi) \quad (18)$$

with N_F the number of vertices, $x_i(\mathcal{K}_j^n)$ the coordinates of the vertices of the space-time element \mathcal{K}_j^n and $\chi_i(\xi)$ a set of finite element shape functions

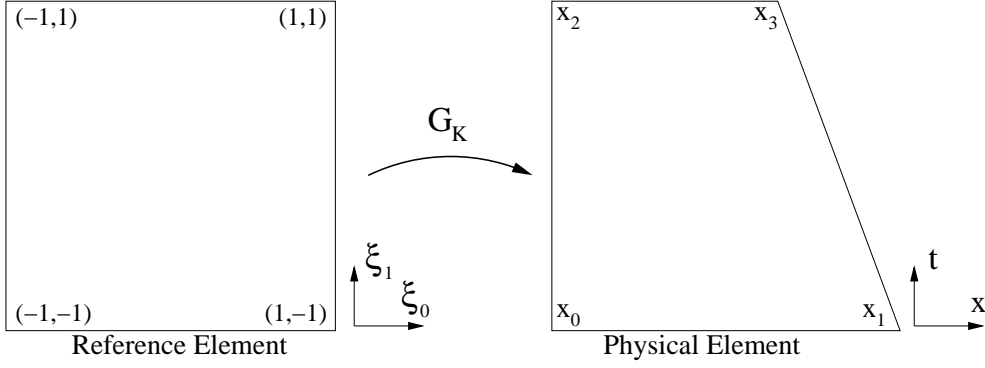


Figure 1. Every physical element \mathcal{K}_j^n is related to a reference element $\hat{\mathcal{K}}$ by means of a mapping $G_{\mathcal{K}_j^n}$.

defined on $\hat{\mathcal{K}}$, with $\xi = (\xi_0, \dots, \xi_d)$ the coordinates in the reference element. The mapping $G_{\mathcal{K}_j^n}$ is illustrated in Figure 1. Given a set of basis functions $\hat{\phi}_m$ defined on the reference element, basis functions $\phi_m : \mathcal{K}_j^n \rightarrow \mathbb{R}$ are defined on the space-time elements $\mathcal{K}_j^n \in \mathcal{T}_h^n$ by means of the mapping $G_{\mathcal{K}_j^n}$:

$$\phi_m = \hat{\phi}_m \circ G_{\mathcal{K}_j^n}^{-1}, \quad (19)$$

The approximated level set in space-time element \mathcal{K}_j^n is now defined as:

$$\psi_h(t, \bar{\mathbf{x}})|_{\mathcal{K}_j^n} = \sum_m \hat{\Psi}_m(\mathcal{K}_j^n) \phi_m(t, \bar{\mathbf{x}}), \quad (20)$$

the approximated extension velocity as:

$$\mathbf{a}_h(t, \bar{\mathbf{x}})|_{\mathcal{K}_j^n} = \sum_m \hat{\mathbf{A}}_m(\mathcal{K}_j^n) \phi_m(t, \bar{\mathbf{x}}), \quad (21)$$

and the approximated flow variables for the two fluids are defined as:

$$\mathbf{w}_h(t, \bar{\mathbf{x}})|_{\mathcal{K}_j^n} = \begin{cases} \mathbf{w}_h^1(t, \bar{\mathbf{x}})|_{\mathcal{K}_j^n} = \sum_m \hat{\mathbf{W}}_m^1(\mathcal{K}_j^n) \phi_m(t, \bar{\mathbf{x}}) & \text{in Fluid 1 elements} \\ \mathbf{w}_h^2(t, \bar{\mathbf{x}})|_{\mathcal{K}_j^n} = \sum_m \hat{\mathbf{W}}_m^2(\mathcal{K}_j^n) \phi_m(t, \bar{\mathbf{x}}) & \text{in Fluid 2 elements} \\ \text{undefined} & \text{otherwise} \end{cases} \quad (22)$$

with $\hat{\Psi}_m$, $\hat{\mathbf{A}}_m$ and $\hat{\mathbf{W}}_m^i$ for $i = 1, 2$ the approximation coefficients for the level set, the extension velocity and the flow field approximations. In every element at most one of the flow variables can be defined. While the level set and the extension velocity are approximated as piecewise linear functions, the order of the approximation for the flow variables is not restricted. Note, because the basis functions are defined locally in every element the solutions can be discontinuous in space-time at element faces.

Since the equations for the level set, extension velocity and the flow variables can all be written as systems of conservation laws the space-time discontinuous

Galerkin discretization will be introduced using a general conservation law:

$$\nabla \cdot \mathcal{F}(\mathbf{U}) = 0, \quad (23)$$

with \mathbf{U} the variable and \mathcal{F} the space-time flux. The approximated variable \mathbf{U}_h is defined as:

$$\mathbf{U}_h(t, \bar{\mathbf{x}})|_{\mathcal{K}} = \sum_m \hat{\mathbf{U}}_m(\mathcal{K}) \phi_m(t, \bar{\mathbf{x}}), \quad (24)$$

and the test function as:

$$\mathbf{V}_h(t, \bar{\mathbf{x}})|_{\mathcal{K}} = \sum_m \hat{\mathbf{V}}_m(\mathcal{K}) \phi_m(t, \bar{\mathbf{x}}) \quad (25)$$

with $\hat{\mathbf{U}}_m$ and $\hat{\mathbf{V}}_m$ the approximation coefficients for the trial and test functions. The trace \mathbf{V}_h^K of a function \mathbf{V}_h on a face \mathcal{S}_m with respect to the element \mathcal{K}^K , $K = l, r$ is defined as:

$$\mathbf{V}_h^K = \lim_{\epsilon \downarrow 0} \mathbf{V}_h(\mathbf{x} - \epsilon \mathbf{n}_{\mathcal{K}}^K), \quad (26)$$

where $\mathbf{n}_{\mathcal{K}}^K = (n_0, \dots, n_d)$ is the space-time outward unit normal vector at the face \mathcal{S}_m with respect to element \mathcal{K}^K . The left and right normal vectors of a face are related as $\mathbf{n}_{\mathcal{K}}^l = -\mathbf{n}_{\mathcal{K}}^r$. The element local traces \mathbf{V}_h^\pm of a function \mathbf{V}_h on a face \mathcal{S}_m are defined as:

$$\mathbf{V}_h^\pm = \lim_{\epsilon \downarrow 0} \mathbf{V}_h(\mathbf{x} \pm \epsilon \mathbf{n}_{\mathcal{K}}). \quad (27)$$

3.3 Space-time weak formulation

Let $\Gamma = \Gamma_{int} \cup \Gamma_{bou}$ denote the set of all faces \mathcal{S}_m , with Γ_{int} the set of all internal and Γ_{bou} the set of all boundary faces. Every internal face connects to exactly two elements, denoted as the left element \mathcal{K}^l and the right element \mathcal{K}^r . Every boundary face bounds exactly one element, denoted as the element \mathcal{K}^l . The weighted average $\{\{F\}\}_{\alpha, \beta}$ of a scalar function F on the face $\mathcal{S}_m \in \Gamma_{int}$ is defined as:

$$\{\{F\}\}_{\alpha, \beta} := \alpha F^l + \beta F^r \quad (28)$$

and the weighted average $\{\{\mathbf{G}\}\}_{\alpha, \beta}$ of a vector function \mathbf{G} on the face $\mathcal{S}_m \in \Gamma_{int}$ is defined as:

$$\{\{\mathbf{G}\}\}_{\alpha, \beta} := \alpha \mathbf{G}^l + \beta \mathbf{G}^r \quad (29)$$

with $\alpha + \beta = 1$. The jump $[[F]]$ of a scalar function F on the face $\mathcal{S}_m \in \Gamma_{int}$ is defined as:

$$[[F]] := F^l \mathbf{n}^l + F^r \mathbf{n}^r \quad (30)$$

and the jump $[[\mathbf{G}]]$ of a vector function \mathbf{G} on the face $\mathcal{S}_m \in \Gamma_{int}$ is defined as:

$$[[\mathbf{G}]] := \mathbf{G}^l \cdot \mathbf{n}^l + \mathbf{G}^r \cdot \mathbf{n}^r. \quad (31)$$

If $[[\mathbf{G}]] = 0$ then the following relation holds:

$$F^l \mathbf{G}^l \cdot \mathbf{n}^l + F^r \mathbf{G}^r \cdot \mathbf{n}^r = [[F]] \cdot \{\{\mathbf{G}\}\}_{\alpha,\beta}. \quad (32)$$

The discontinuous Galerkin finite element approximation is found by multiplying (23) with a test function V and integrating over all elements in the domain \mathcal{E} :

$$\sum_{\mathcal{K}} \left\{ \int_{\mathcal{K}_j^n} V \nabla \cdot \mathcal{F}(\mathbf{U}) d\mathcal{K} \right\} = 0. \quad (33)$$

Applying Gauss' theorem results in:

$$\begin{aligned} & - \sum_{\mathcal{K}} \int_{\mathcal{K}_j^n} \nabla V \cdot \mathcal{F}(\mathbf{U}) d\mathcal{K} + \sum_{\mathcal{S}_m \in \Gamma_{int}} \int_{\mathcal{S}_m} \mathcal{F}^l(\mathbf{U}^l) \cdot \mathbf{n}_{\mathcal{K}}^l V^l + \mathcal{F}^r(\mathbf{U}^r) \cdot \mathbf{n}_{\mathcal{K}}^r V^r d\mathcal{S} \\ & + \sum_{\mathcal{S}_m \in \Gamma_{bou}} \int_{\mathcal{S}_m} \mathcal{F}^l(\mathbf{U}^l) \cdot \mathbf{n}_{\mathcal{K}}^l V^l d\mathcal{S} = 0, \end{aligned} \quad (34)$$

where \mathcal{F}^K and \mathbf{U}^K are the limiting trace values on the face \mathcal{S}_m of element \mathcal{K}^K , $K = l, r$. By using a conservative flux, $\mathcal{F}^l(\mathbf{U}^l) \cdot \mathbf{n}_{\mathcal{K}}^l = -\mathcal{F}^r(\mathbf{U}^r) \cdot \mathbf{n}_{\mathcal{K}}^r$ and hence $[[\mathcal{F}(\mathbf{U})]] = 0$, and definitions (28)-(31), equation (34) can be rewritten as:

$$\begin{aligned} & - \sum_{\mathcal{K}} \int_{\mathcal{K}_j^n} \nabla V \cdot \mathcal{F}(\mathbf{U}) d\mathcal{K} + \sum_{\mathcal{S}_m \in \Gamma_{int}} \int_{\mathcal{S}_m} \{\{\mathcal{F}(\mathbf{U})\}\}_{\alpha,\beta} \cdot [[V]] d\mathcal{S} \\ & + \sum_{\mathcal{S}_m \in \Gamma_{bou}} \int_{\mathcal{S}_m} \mathcal{F}^l(\mathbf{U}^l) \cdot \mathbf{n}_{\mathcal{K}}^l V^l d\mathcal{S} = 0. \end{aligned} \quad (35)$$

At both the internal and boundary faces the flux is replaced by a numerical flux $\mathcal{H}(\mathbf{U}^l, \mathbf{U}^r, \mathbf{n}_{\mathcal{K}})$, which is consistent: $\mathcal{H}(\mathbf{U}, \mathbf{U}, \mathbf{n}_{\mathcal{K}}) = \mathcal{F}(\mathbf{U}) \cdot \mathbf{n}_{\mathcal{K}}^l$ and conservative. Using the fact that for a conservative flux $\{\{\mathcal{H}(\mathbf{U}^l, \mathbf{U}^r, \mathbf{n}_{\mathcal{K}})\}\}_{\alpha,\beta} = \mathcal{H}(\mathbf{U}^l, \mathbf{U}^r, \mathbf{n}_{\mathcal{K}})$, equation (35) becomes:

$$\begin{aligned} & - \sum_{\mathcal{K}} \int_{\mathcal{K}_j^n} \nabla V \cdot \mathcal{F}(\mathbf{U}) d\mathcal{K} + \sum_{\mathcal{S}_m \in \Gamma_{int}} \int_{\mathcal{S}_m} \mathcal{H}(\mathbf{U}^l, \mathbf{U}^r, \mathbf{n}_{\mathcal{K}}) (V^l - V^r) d\mathcal{S} \\ & + \sum_{\mathcal{S}_m \in \Gamma_{bou}} \int_{\mathcal{S}_m} \mathcal{H}(\mathbf{U}^l, \mathbf{U}_b, \mathbf{n}_{\mathcal{K}}) V^l d\mathcal{S} = 0. \end{aligned} \quad (36)$$

After replacing the trial and test functions by their approximations the weak formulation can be defined as:

Find a $\mathbf{U}_h \in B_h^k(\mathcal{T}_h^n)$ such that for all $\mathbf{V}_h \in B_h^k(\mathcal{T}_h^n)$:

$$\begin{aligned} & - \sum_{\mathcal{K}} \int_{\mathcal{K}_j^n} \nabla V_h \cdot \mathcal{F}(\mathbf{U}_h) d\mathcal{K} + \sum_{S_m \in \Gamma_{int}} \int_{S_m} \mathcal{H}(\mathbf{U}_h^l, \mathbf{U}_h^r, \mathbf{n}_{\mathcal{K}}) (V_h^l - V_h^r) dS \\ & + \sum_{S_m \in \Gamma_{bou}} \int_{S_m} \mathcal{H}(\mathbf{U}_h^l, \mathbf{U}_b, \mathbf{n}_{\mathcal{K}}) V_h^l dS = 0. \end{aligned} \quad (37)$$

The weak formulation (37) can be rewritten as an element local formulation: For all elements \mathcal{K}_j^n find a \mathbf{U}_h such that for all V_h :

$$- \int_{\mathcal{K}_j^n} \nabla V_h \cdot \mathcal{F}(\mathbf{U}_h) d\mathcal{K} + \sum_{S_m \in \partial \mathcal{K}_j^n} \int_{S_m} \mathcal{H}(\mathbf{U}_h^-, \mathbf{U}_h^+, \mathbf{n}_{\mathcal{K}}) V_h^- dS = 0. \quad (38)$$

3.4 Discretization

Introduction of the polynomial expansions (24), (25) in the weak formulation (38) gives the following discretization in each space-time element:

$$\begin{aligned} & - \int_{\mathcal{K}_j^n} \nabla \phi_l \cdot \mathcal{F}(\mathbf{U}_h) d\mathcal{K} \\ & + \sum_{S_m \in \partial \mathcal{K}_j^n} \int_{S_m} \mathcal{H}(\mathbf{U}_h^-, \mathbf{U}_h^+, \mathbf{n}_{\mathcal{K}}) \phi_l^- dS = 0 \text{ with } l = 0, \dots, N_B - 1 \end{aligned} \quad (39)$$

with N_B the number of basis functions in the element. The expansion coefficients for the test functions have been chosen as $\hat{V}_{lm} = \delta_{lm}$, $l, m = 0, \dots, N_B - 1$, meaning that the test functions are just the basisfunctions used in the element. The space-time discretization can be written in each element as

$$\mathcal{L}(\hat{\mathbf{U}}^n; \hat{\mathbf{U}}^{n-1}) = 0, \quad (40)$$

where the operator \mathcal{L} is given by:

$$\mathcal{L}_{il}(\hat{\mathbf{U}}^n, \hat{\mathbf{U}}^{n-1}) = A_{il} + R_{il}, \quad i = 0 \dots N_U, \quad l = 0 \dots N_B - 1 \quad (41)$$

with N_U the total number of variables.

The discretization of the level set equation is found by replacing \mathbf{U} with ψ_h and \mathcal{F} by \mathcal{F}^L as defined in equations (20) and (9):

$$\mathcal{L}^L(\hat{\Psi}^n, \hat{\Psi}^{n-1}) = A_l^L + R_l^L = 0, \text{ for } l = 0 \dots N_B - 1, \quad (42)$$

where the matrices for the level set discretization are defined as:

$$\begin{aligned} A_l^L &= - \int_{\mathcal{K}_j^n} (\nabla \phi_l)_j \mathcal{F}_j^L(\psi_h) d\mathcal{K}, \\ R_l^L &= \sum_{\mathcal{S}_m \in \partial \mathcal{K}_j^n} \int_{\mathcal{S}_m} \mathcal{H}^L(\psi_h^-, \psi_h^+, \mathbf{a}_h^-, \mathbf{a}_h^+, \mathbf{n}_{\mathcal{K}}) \phi_l^- d\mathcal{S}. \end{aligned} \quad (43)$$

The discretization of the extension velocity equations is found by replacing \mathbf{U} with \mathbf{a}_h and \mathcal{F} by \mathcal{F}^A as defined in equations (21) and (14):

$$\mathcal{L}^A(\hat{\mathbf{A}}^n, \hat{\mathbf{A}}^{n-1}) = A_{il}^A + R_{il}^A = 0, \text{ for } i = 0 \dots d-1, l = 0 \dots N_B - 1, \quad (44)$$

where the matrices for the extension velocity discretization are defined as:

$$\begin{aligned} A_{il}^A &= - \int_{\mathcal{K}_j^n} (\nabla \phi_l)_j \cdot \mathcal{F}_{ij}^A(\mathbf{a}_h) d\mathcal{K}, \\ R_{il}^A &= \sum_{\mathcal{S}_m \in \partial \mathcal{K}_j^n} \int_{\mathcal{S}_m} \mathcal{H}_i^A(\mathbf{a}_h^-, \mathbf{a}_h^+, \mathbf{w}_h^-, \mathbf{w}_h^+, \mathbf{n}_{\mathcal{K}}) \phi_l^- d\mathcal{S}. \end{aligned} \quad (45)$$

The discretization of the flow equations is found by replacing \mathbf{U} with \mathbf{w}_h and \mathcal{F} by \mathcal{F}^F as defined in equations (22) and (2):

$$\mathcal{L}^F(\hat{\mathbf{W}}^n, \hat{\mathbf{W}}^{n-1}) = A_{il}^F + R_{il}^F = 0, \text{ for } i = 0 \dots N_F, l = 0 \dots N_B - 1, \quad (46)$$

where the matrices for the flow equation discretization are defined as:

$$\begin{aligned} A_{il}^F &= - \int_{\mathcal{K}_j^n} (\nabla \phi_l)_j \cdot \mathcal{F}_{ij}^F(\mathbf{w}_h) d\mathcal{K}, \\ R_{il}^F &= \sum_{\mathcal{S}_m \in \partial \mathcal{K}_j^n} \int_{\mathcal{S}_m} \mathcal{H}_i^F(\mathbf{w}_h^-, \mathbf{w}_h^+, \mathbf{n}_{\mathcal{K}}) \phi_l^- d\mathcal{S}. \end{aligned} \quad (47)$$

3.5 Numerical flux

The numerical flux \mathcal{H}^L for the level set is defined as an upwind flux:

$$\mathcal{H}^L(\psi_h^-, \psi_h^+, \mathbf{a}_h^-, \mathbf{a}_h^+, \mathbf{n}_{\mathcal{K}}) = \begin{cases} \psi_h^- \mathbf{a} \cdot \mathbf{n}_{\mathcal{K}} & \text{if } \mathbf{a} \cdot \mathbf{n}_{\mathcal{K}} > 0, \\ \psi_h^+ \mathbf{a} \cdot \mathbf{n}_{\mathcal{K}} & \text{if } \mathbf{a} \cdot \mathbf{n}_{\mathcal{K}} \leq 0. \end{cases} \quad (48)$$

The numerical flux \mathcal{H}^A for the extension velocity is defined at the interface as:

$$\mathcal{H}_i^A(\mathbf{a}_h^-, \mathbf{a}_h^+, \mathbf{w}_h^-, \mathbf{w}_h^+, \mathbf{n}_{\mathcal{K}}) = \left(0, \text{sign}(\psi^-) \bar{\mathbf{u}}_{i,h} \frac{\bar{\nabla} \psi^-}{|\bar{\nabla} \psi^-|} \right) \quad (49)$$

and at the other faces as upwind flux:

$$\mathcal{H}_i^A(\mathbf{a}_h^-, \mathbf{a}_h^+, \mathbf{w}_h^-, \mathbf{w}_h^+, \mathbf{n}_{\mathcal{K}}) = \begin{cases} (0, \text{sign}(\psi^-) a_{i,h}^- \frac{\bar{\nabla} \psi^-}{|\bar{\nabla} \psi^-|} \cdot \bar{\mathbf{n}}_{\mathcal{K}}) \\ \quad \text{if } \text{sign}(\psi^-) \bar{\nabla} \psi^- \cdot \bar{\mathbf{n}}_{\mathcal{K}} > 0, \\ (0, \text{sign}(\psi^-) a_{i,h}^+ \frac{\bar{\nabla} \psi^-}{|\bar{\nabla} \psi^-|} \cdot \bar{\mathbf{n}}_{\mathcal{K}}) \\ \quad \text{if } \text{sign}(\psi^-) \bar{\nabla} \psi^- \cdot \bar{\mathbf{n}}_{\mathcal{K}} \leq 0. \end{cases} \quad (50)$$

The numerical fluxes used for the flow equations will be specified for various test cases in Section 5.

3.6 Stabilization operator

In the STDG finite element method discontinuities can only be represented exactly on element boundaries. These discontinuities can cause large gradients in the numerical solution, resulting in overshoots which could make the scheme unstable. In order to remove these spurious oscillations and ensure monotonicity of the solution near discontinuities a stabilization operator is added to the discretization. The stabilization operator is defined as follows:

$$D(\mathbf{U}_h) = | \nabla \cdot \mathcal{F}(\mathbf{U}_h(G_K(p_m))) | \int_{\mathcal{K}_j^n} \epsilon_0 \bar{\nabla} \mathbf{V}_h \bar{\nabla} \mathbf{U}_h^T + \epsilon_1 \frac{\partial \mathbf{V}_h}{\partial t} \frac{\partial \mathbf{U}_h}{\partial t} dx dt, \quad (51)$$

where T denotes the transpose of a vector and ϵ_0 and ϵ_1 are dissipation constants which are chosen depending on the desired amount of dissipation. The operator distinguishes between smooth and discontinuous parts of the solution by means of the value of the residual in the reference element midpoint p_m . Near discontinuities the differential form of the conservation law does not hold and a large amount of dissipation will be added to stabilize the solution. However, if the interface position is being tracked in the mesh, the interface will approximately coincide with a face and hence the amount of added dissipation near the interface will be much smaller.

3.7 Pseudo-time integration

To solve the discretized equations (42), (44) and (46) on a given two fluid mesh a five stage semi-implicit Runge-Kutta iterative scheme is used ([49], [68]). Starting from a guess for the initial solution, the solution is iterated in pseudo-time until a steady state is reached. The steady state solution is also the real time solution of the space-time discretization. Using the general discretization (40) the scheme is given as: Here $\lambda = \Delta\tau/\Delta t$ and the coefficients α_s are defined as: $\alpha_1 = 0.0791451$, $\alpha_2 = 0.163551$, $\alpha_3 = 0.283663$, $\alpha_4 = 0.5$, $\alpha_5 = 1.0$ (optimization based on a space-time discretization of the linear

1. Initialize first Runge-Kutta stage:

$$\bar{\mathbf{U}}^{(0)} = \hat{\mathbf{U}}^n.$$

2. Calculate $\bar{\mathbf{U}}^{(s)}, s = 1, \dots, 5$:

$$\begin{aligned} \left(I + \frac{\alpha_s \lambda}{|K^n|} (|K^n| I + D^{(s-1)}) \right) \bar{\mathbf{U}}^{(s)} = \\ \bar{\mathbf{U}}^{(0)} + \frac{\alpha_s \lambda}{|K^n|} \left(|K^n| \bar{\mathbf{U}}^{(s-1)} - \mathcal{L}(\bar{\mathbf{U}}^{(s-1)}, \bar{\mathbf{U}}^{n-1}) \right) \end{aligned}$$

3. Update solution:

$$\hat{\mathbf{U}}^n = \bar{\mathbf{U}}^{(5)}.$$

Algorithm 1. Pseudo-time integration method for solving the non-linear algebraic equations in the space-time discretization.

advection equation). The factor K^n represents the spatial size of the element at time $t = t_n$. To get a measure of the size of the element in space and time the factors $h_i = (\sum_{j=0}^{d-1} (\partial x_i / \partial \xi_j)^2)^{1/2}, j = 0, \dots, d-1$ are defined. The physical time step Δt is defined globally by using a Courant-Friedrichs-Levy (*CFL*) condition:

$$\Delta t = \frac{CFL_{\Delta t} \Delta x}{S_{max}}, \quad (52)$$

with $CFL_{\Delta t}$ the physical *CFL* number, Δx the diameter of the sphere enclosing the element and S_{max} the maximum value of the wave speed on the faces. In the pseudo-time iteration λ is determined as $\lambda = CFL_{\Delta \tau} S_g / S_{max}$, where $S_g = (\min_{j=1, \dots, d-1} h_j) / h_0$ is a measure of the maximum allowed velocity in the element. In the cut-cells the physical *CFL* number can be larger than in the background elements hence for the pseudo-time *CFL* number $CFL_{\Delta \tau} = 1.0$ is taken, which makes the pseudo-time iteration stable for any physical *CFL* number.

4 Two fluid mesh refinement

In this section the details of the two fluid mesh refinement are discussed based on a given state of the level set, extension velocity and flow approximations.

4.1 Two fluid mesh

The construction of the two fluid mesh starts with the definition of a background mesh. The approximated flow variables (22) cannot be defined directly

on the background mesh since some background elements may contain two separate fluids. For this purpose the background elements containing two fluids (two fluid elements) are refined into a number of non overlapping elements containing only one fluid (one fluid elements). Let \mathcal{T}_b^n denote the set of background elements in timeslab \mathcal{J}_n and let $\mathcal{T}_{b,1}^n$ and $\mathcal{T}_{b,2}^n$ denote the sets of background elements containing one and two fluids. The refinement procedure results in a triangulation composed of only one fluid elements, which will be referred to as the computational or two fluid mesh:

$$\mathcal{T}_h^n = \mathcal{T}_{b,1}^n \cup \mathcal{T}_c^n, \quad (53)$$

where \mathcal{T}_c^n is the set of single fluid child elements created in the refinement of the elements in $\mathcal{T}_{b,2}^n$. In the two fluid mesh all elements have exactly one set of approximated flow variables corresponding to the type of fluid contained in the element. In Figures 2 and 3 an example two fluid problem defined on a two-dimensional background mesh and the corresponding computational mesh are shown.

4.2 Two fluid mesh refinement

The two fluid mesh is refined based on the approximate description of the interface position and shape given by the 0-level set. However, in the space-time discontinuous Galerkin discretization the level set is allowed to be discontinuous at the element faces, which is not desirable for the two fluid mesh refinement, since it can result in holes in the mesh. For this purpose the level set is redefined as a continuous function before performing the two fluid mesh refinement. Assuming computations have reached time slab \mathcal{J}_n the approximated level set function ψ_h is made continuous by first looping over all elements in \mathcal{J}_n and while storing the multiplicity and the sum of the values of ψ_h in that vertex. For every vertex i in \mathcal{J}_n the continuous level set value $\psi_{h,i}^c$ is then calculated by dividing the sum of the ψ_h values by the vertex multiplicity. In every background element in \mathcal{J}_n , ψ_h is then reinitialized using the $\psi_{h,i}^c$ values in the element vertices. To ensure continuity of the mesh at the time slab faces only the values of the level set in the background elements belonging to \mathcal{J}_{n-1} are used at the faces between the previous and the current time slab. The mesh update step can be performed locally in every element.

To check if a background element contains more than one fluid the continuous level set is evaluated at the edge vertices of the element. For this purpose the element edges are numbered using a local edge index. If there is no sign change in the evaluations for any edge, the element holds only one fluid. Otherwise the level set function is zero somewhere in the element and two fluids must be

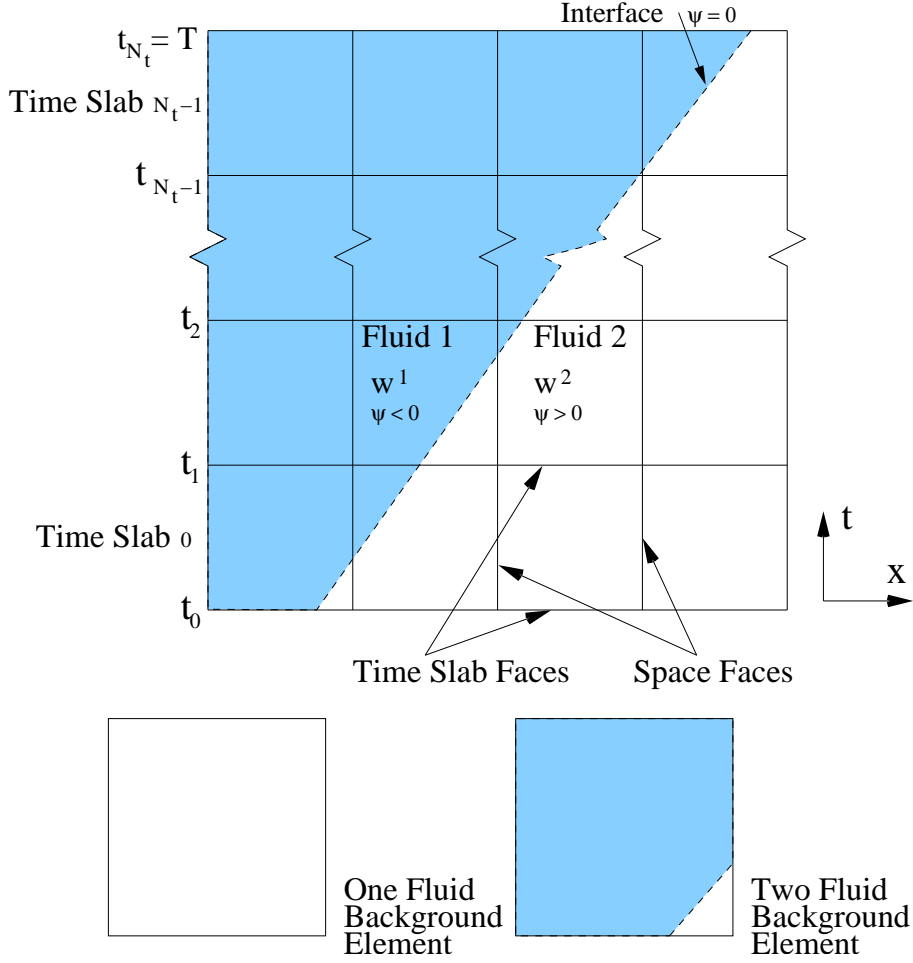


Figure 2. Space-time background mesh in two dimensions with two fluids. A level set function ψ is used to distinguish between the two fluids, with the 0-level set representing the interface. Individual background elements can contain either one or two fluids.

present. The interface position \mathbf{x}_I is then calculated at the relevant edges as:

$$\mathbf{x}_I = \frac{\mathbf{x}_A \psi_h(\mathbf{x}_B) - \mathbf{x}_B \psi_h(\mathbf{x}_A)}{\psi_h(\mathbf{x}_A) - \psi_h(\mathbf{x}_B)}, \quad (54)$$

where \mathbf{x}_A and \mathbf{x}_B denote the coordinates of the edge vertices. Based on the local edge indices of the edges cut by the interface a refinement type is selected. In two-dimensional space-time six such refinement types have been defined which are illustrated in Figure 4.

4.3 Two fluid algorithm

The two fluid algorithm is as follows:

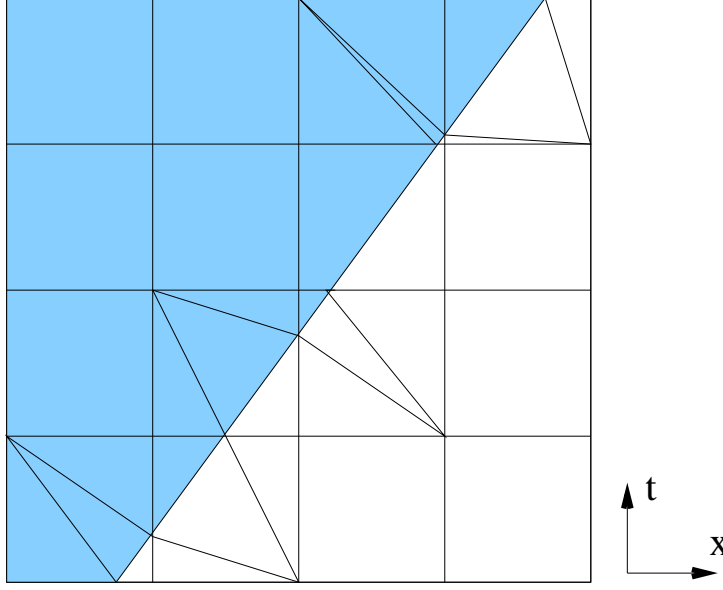


Figure 3. Space-time two fluid mesh in two dimensions.

```

Initialize two fluid mesh:  $\mathcal{T}_{h,0}^0$ 
 $n = 0$ 
WHILE  $n < N_t$  DO
   $j = 0$ 
  WHILE two fluid mesh has not converged:  $|e_j - e_{j-1}| > \epsilon_{IF}$  DO
    Solve  $\psi_h, \mathbf{a}_h$  on  $\mathcal{T}_{h,j}^n$ 
    Calculate level set error  $e_j = \|\psi_h\|_2^{IF}$  at the interface for  $\mathcal{T}_{h,j}^n$ 
    Update two fluid mesh:  $\mathcal{T}_{h,j}^n \rightarrow \mathcal{T}_{h,j+1}^n$ 
    Solve  $\mathbf{w}_h$  on  $\mathcal{T}_{h,j+1}^n$ 
     $j = j + 1$ 
  ENDDO
  Update two fluid mesh for next time slab:  $\mathcal{T}_{h,j+1}^n \rightarrow \mathcal{T}_{h,0}^{n+1}$ 
   $n = n + 1$ 
ENDDO

```

Algorithm 2. Computational and mesh update steps in the two fluid method.

The initialization is illustrated in Figure 5. First the level set function ψ_h is initialized in every element of the background mesh \mathcal{T}_b^n (Figure 5a). The initial level set used here is constant in time in the space-time slab, but this is not obligatory. The initial level set is also assumed to be continuous and hence

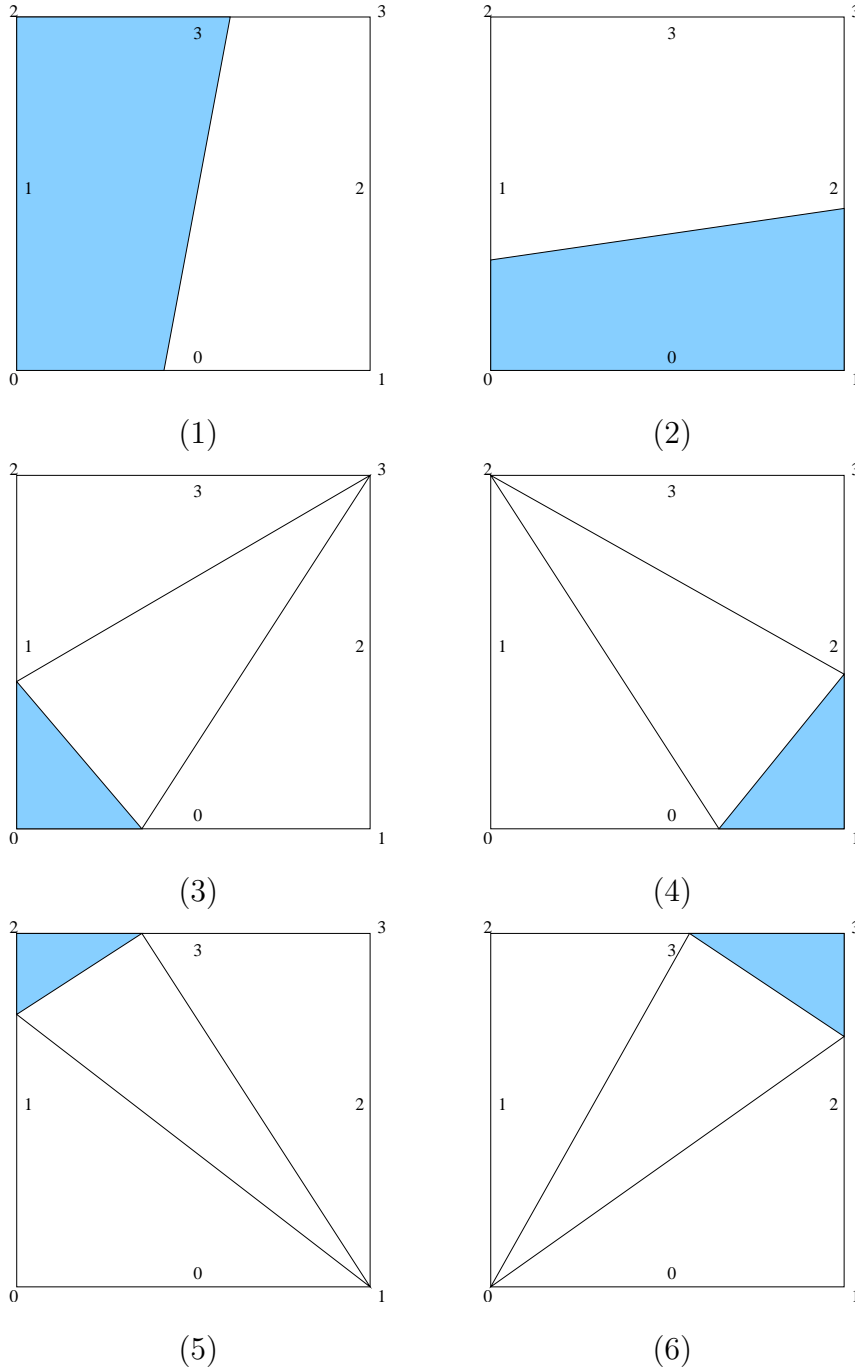


Figure 4. The six two fluid element refinement types for a square.

$\psi_h^c = \psi_h$ initially. From the initial level set the two fluid mesh is then created by first finding the 0-level set (Figure 5b) and then performing the refinement (Figure 5c). Here refinement type 1 is used for both the previous and the current time slabs. Next the level set, extension velocity and flow variables are initialized in all one fluid elements (Figure 5d).

After the level set and extension velocity have been computed on a given mesh,

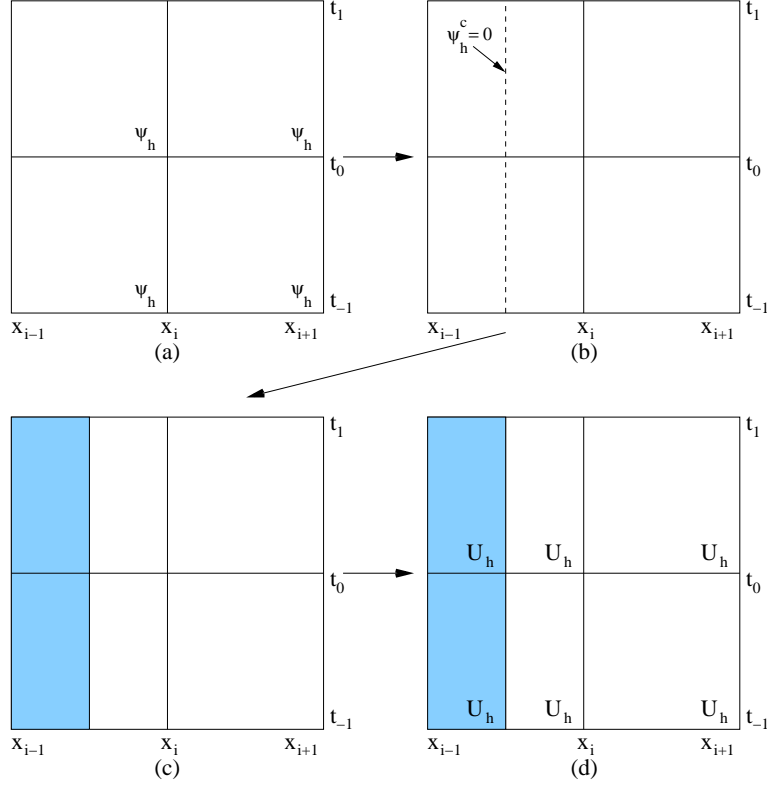


Figure 5. At the start of the computations, first the level set is initialized on the background mesh, both in the current and previous time slab (a). A check is made for every element to see if it contains a part of the 0-level set (b). These background elements are refined and the resulting child elements combined with the unrefined background elements are used to define the two fluid mesh (c). Finally, in all elements of the two fluid mesh the level set function, extension velocity and flow variables are initialized (d).

the mesh is updated as illustrated in Figure 6 for two space-time dimensions. The converged level set ψ_h can be discontinuous (Figure 6a) and therefore first a continuous level set ψ_h^c is constructed by using the averaging procedure described in Section 3.3.2 (see also Figure 6b). Note that ψ_h^c is defined on the background element. Based on the continuous level set the two fluid mesh refinement is performed for the current time slab (Figure 6c), where in this case the refinement types 4 and 5 are used. Finally the level set, extension velocity and the flow variables are initialized in the one fluid elements (Figure 6d). The initialization of the level set function is done by using the values of the continuous level set ψ_h^c in the parent vertices and using that $\psi_h^c = 0$ on the interface vertices. The initialization of the flow functions is done by using globally defined constants for every fluid type. To test whether or not the two fluid mesh has converged, the error in the level set at the interface is calculated and compared with the error from the previous iteration step. When the value is small enough, it is assumed that the two fluid mesh has converged, otherwise the procedure is continued using the most recently computed level

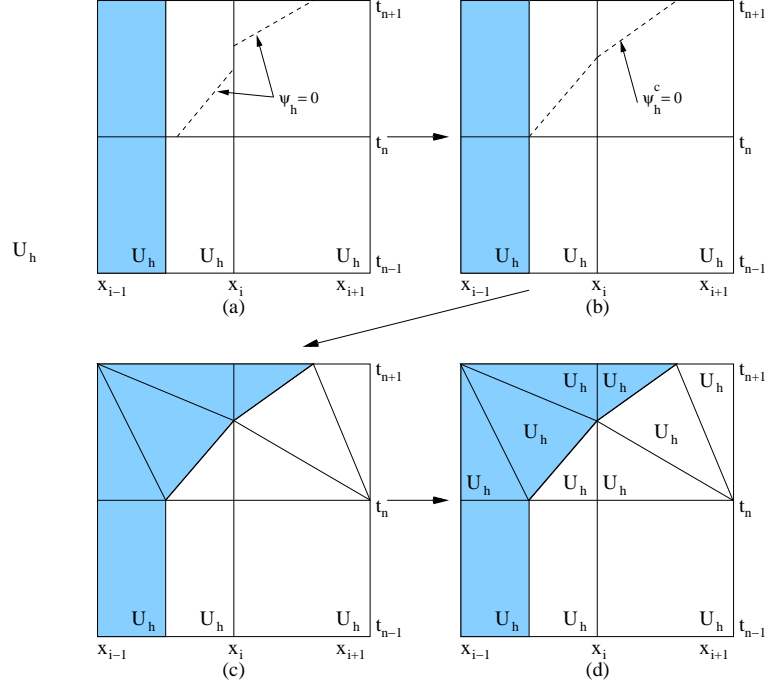


Figure 6. When performing a two fluid mesh update, first the (discontinuous) level set function ψ_h (a) is used to define a continuous level set function ψ_h^c in all background elements of the current time slab (b), where at the time slab faces with $t = t_n$ the level set function from the previous time slab is used. The background elements which contain a part of the 0-level set of ψ_h^c are then refined and the resulting child elements, combined with the unrefined background elements, are used to define the new two fluid mesh (c). Finally, in all elements of the new two fluid mesh the level set function, extension velocity and flow variables are initialized (d).

set and refined one fluid mesh as initial condition.

When the flow equations have been solved on the converged mesh the computations are moved to the next time slab J_{n+1} . For this purpose a time slab update, as illustrated in Figure 7, is performed. First, ψ_h is initialized in the background elements of the new time slab as $\psi_h(t, \bar{x}) = \psi(t_{n+1}^+, 0)$ (Figure 7a). Based on ψ_h^c the two fluid mesh is constructed (Figure 7b,c) where refinement type 1 is used and finally the approximations are initialized in the one fluid elements of the new time slab (Figure 7d).

5 Test problems

The STDG finite element method for two fluid flows presented in Section 3 is tested on a number of different model problems in two space-time dimensions. In these tests the accuracy of the interface position and the accuracy of the flow solution are considered. The implementations were created based

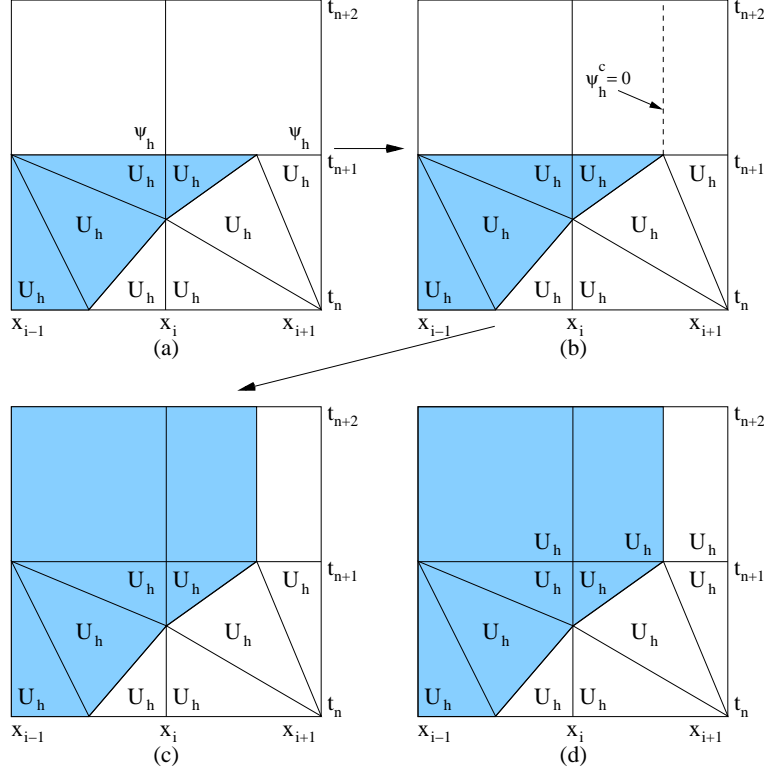


Figure 7. The two fluid mesh update in a time slab starts with first copying both the mesh and data from the current to the previous time slab (a). Then the continuous level set function ψ_h^c is initialized in all background elements of the new time slab based on the level set values at the time slab faces (b). Based on ψ_h^c the background mesh is refined (c). Finally, the level set function, extension velocity and flow variables are initialized in all elements of the resulting two fluid mesh belonging to the new time slab (d).

on the hpGEM software framework for Discontinuous Galerkin finite element methods (See [Pesch et.al.] for further information).

5.1 Linear advection tests

The method is first tested for the linear advection equation using a constant advection velocity. The interface movement is linear in space-time hence the interface representation in the mesh can be exact. The test also illustrates some aspects of the STDG numerical scheme. The linear advection equation is given as:

$$\frac{\partial \rho}{\partial t} + a \frac{\partial \rho}{\partial x} = 0 \quad (55)$$

with ρ the advection variable and $a = 5.0$ the advection velocity. Both continuous and discontinuous initial conditions are used:

$$\begin{aligned} \rho_0(x) &= \begin{cases} 1.5 + 0.5 \cos(\pi(x + 2.5)) & \text{for } |x + 2.5| \leq 1.0 \\ 1.0 & \text{for } |x + 2.5| > 1.0, \end{cases} \\ \rho_0(x) &= \begin{cases} 2.0 & \text{for } x \leq 0.0 \\ 1.0 & \text{for } x > 0.0. \end{cases} \end{aligned} \quad (56)$$

Extrapolating boundary conditions are used:

$$\rho(x, t) = \rho^-(x, t) \text{ at the boundary}, \quad (57)$$

where $\rho^-(x, t)$ denotes the limit of $\rho(x, t)$ taken from the inside of the domain. Without boundaries, the exact solution to (55) is:

$$\rho(x, t) = \rho_0(x - at). \quad (58)$$

All linear advection simulations are performed on a spatial domain $[-5.0, 5.0]$ and run from time $t = 0.0$ to $t = 1.0$ at $CF L_{\Delta t} = 0.4$ while using linear basis functions. The tolerance for the flow residual is 1.0×10^{-12} to ensure full convergence of the numerical solutions. For the test with interface tracking the tolerances for the extension velocity and the level set residuals are both 1.0×10^{-6} and the tolerance for the mesh convergence is 1.0×10^{-3} . For the linear advection tests no dissipation is used.

5.1.1 Interface integral

At the interface \mathcal{S}_{IF} the face contribution appearing in the weak formulation of the flow equations (47) is modified as follows:

$$\begin{aligned} R_{il} &= \int_{\mathcal{S}_{IF}} \mathcal{H}_E^F(\mathbf{U}_h^-, \mathbf{U}_h^+, \mathbf{n}_{\mathcal{K}})(\phi_l^- - (\hat{\phi}_{l,av}^- \circ G_{\mathcal{K}_j}^{-1})) \\ &\quad + \mathcal{H}_I^F(\mathbf{U}_h^-, \mathbf{U}_h^+, \mathbf{n}_{\mathcal{K}})(\hat{\phi}_{l,av}^- \circ G_{\mathcal{K}_j}^{-1}) d\mathcal{S}, \quad K = l, r \end{aligned} \quad (59)$$

with $\hat{\phi}_{l,av}^-$ the average of the basisfunction $\hat{\phi}_l$ over the reference element $\hat{\mathcal{K}}^-$. The flux \mathcal{H}_I^F is defined as the solid wall flux for a wall moving in space-time:

$$\mathcal{H}_I^F(\mathbf{U}_h^-, \mathbf{U}_h^+, \mathbf{n}_{\mathcal{K}}) = 0 \quad (60)$$

and the flux \mathcal{H}_E^F is defined as the extrapolating flux:

$$\mathcal{H}_E^F(\mathbf{U}_h^-, \mathbf{U}_h^+, \mathbf{n}_{\mathcal{K}}) = (1, a)\rho_h^-. \quad (61)$$

The purpose of using this modified formulation is to damp any numerical oscillations originating from the interface without compromising the conservative

Table 1

Test results 5.1.2: Accuracy and order of convergence of the advection variable in the linear advection test with smooth initial condition given by (56a) without interface tracking.

$N_x \times N_t$	L^2 error	order
20 x 25	2.305e-1	-
40 x 50	6.510e-2	1.82
80 x 100	1.374e-2	2.24
160 x 200	3.129e-3	2.13

properties of the STDG scheme. For any interface element, the approximation mean will only be affected by the flux \mathcal{H}_I^F while the approximation slopes will only be affected by the flux \mathcal{H}_E^F .

5.1.2 Linear advection with continuous initial conditions and without interface tracking

In the first linear advection test the accuracy of the STDG finite element method is checked for a smooth initial solution (56a). The results are presented in Table 1, where the L^2 error for various mesh resolutions and the order of accuracy are given. Orders of accuracy of around 2 are observed, which is as expected since the STDG finite element method is of order $O(h^{p+1})$ for smooth solutions. The solution at time $t = 1.0$ is illustrated in Figure 8a. On this coarse mesh the solution is dissipative and also shows some spurious oscillations around the steep slope. Note that since no stabilization operator or limiter has been used for this test case and some small oscillations can be expected. Also, the solution is plotted as discontinuous data, without any postprocessing to enhance the solution accuracy, for the purpose of giving a clear illustration of the behavior of the STDG numerical scheme in each individual element.

5.1.3 Linear advection with discontinuous initial conditions and without interface tracking

In the second linear advection test the performance of the STDG finite element method is checked for a discontinuous initial solution (56b). The results are presented in Table 2, where the L^2 error and the order of accuracy are given for various mesh resolutions. The solution at time $t = 1.0$ is illustrated in Figure 8b. Like in the first test, the solution is dissipative and shows spurious oscillations. As can be seen from Table 2, the orders of accuracy are much smaller than those found in the first test. However, for discontinuous solutions computed on a static mesh the order of accuracy will typically not exceed

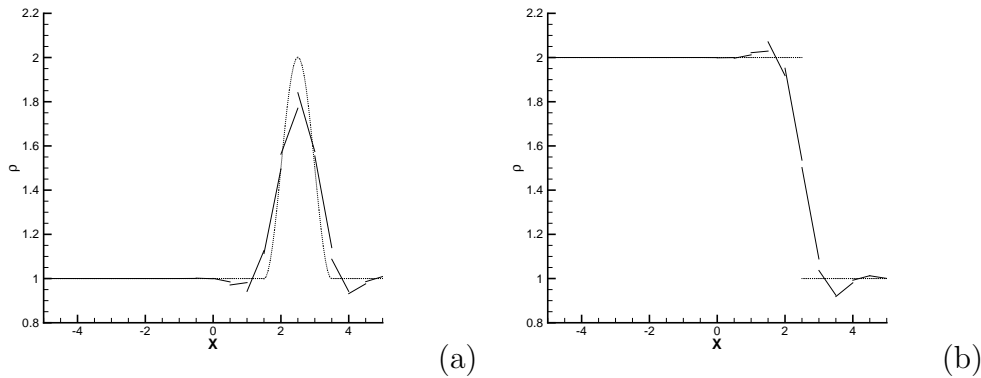


Figure 8. The exact (dotted) and numerical (solid) solutions of the linear advection tests without mesh refinement (5.1.1 (a) and 5.1.2 (b)) at time $t = 1.0$ using 20 elements.

Table 2

Test results 5.1.3: Accuracy and order of convergence of the advection variable for the linear advection test with discontinuous initial condition given by (56b) and without interface tracking.

$N_x \times N_t$	L^2 error	order
20 x 25	3.072e-1	-
40 x 50	2.385e-1	0.37
80 x 100	1.846e-1	0.37
160 x 200	1.428e-1	0.37

$O(h^{1/2})$ ([67]).

5.1.4 Linear advection with discontinuous initial conditions and interface tracking

In the third linear advection test the STDG finite element method for two fluid flows is used to solve the linear advection problem with the discontinuous initial solution (56b). The performance of the two fluid scheme is optimal for this test, with the accuracy of the solution and the interface position both at machine precision, even without dissipation. Because the advection speed is given and constant, the interface movement in space-time is linear and can be represented exactly on the refined mesh. The numerical solution and the refined space-time mesh are shown in Figures 9 and 10. In Figure 11 the mesh refinement steps for the first four time steps are shown to illustrate the mesh refinement procedures discussed in Section 4.

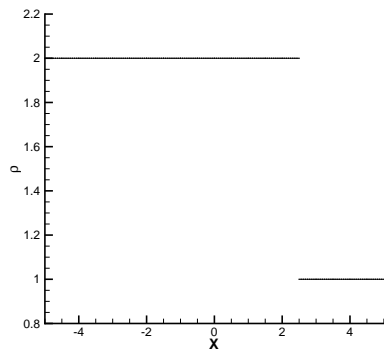


Figure 9. The numerical solution of the linear advection test with mesh refinement (5.1.3) at time $t = 1.0$ using 20 elements.

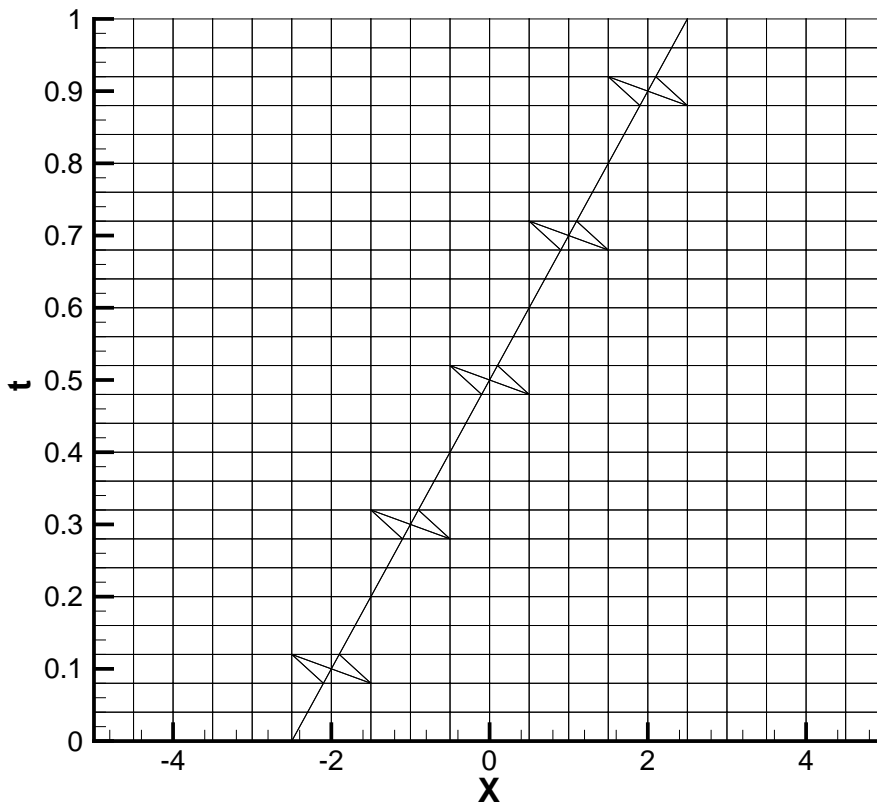


Figure 10. The refined space-time mesh of the linear advection test with mesh refinement (5.1.3) at time $t = 1.0$ using 20 elements.

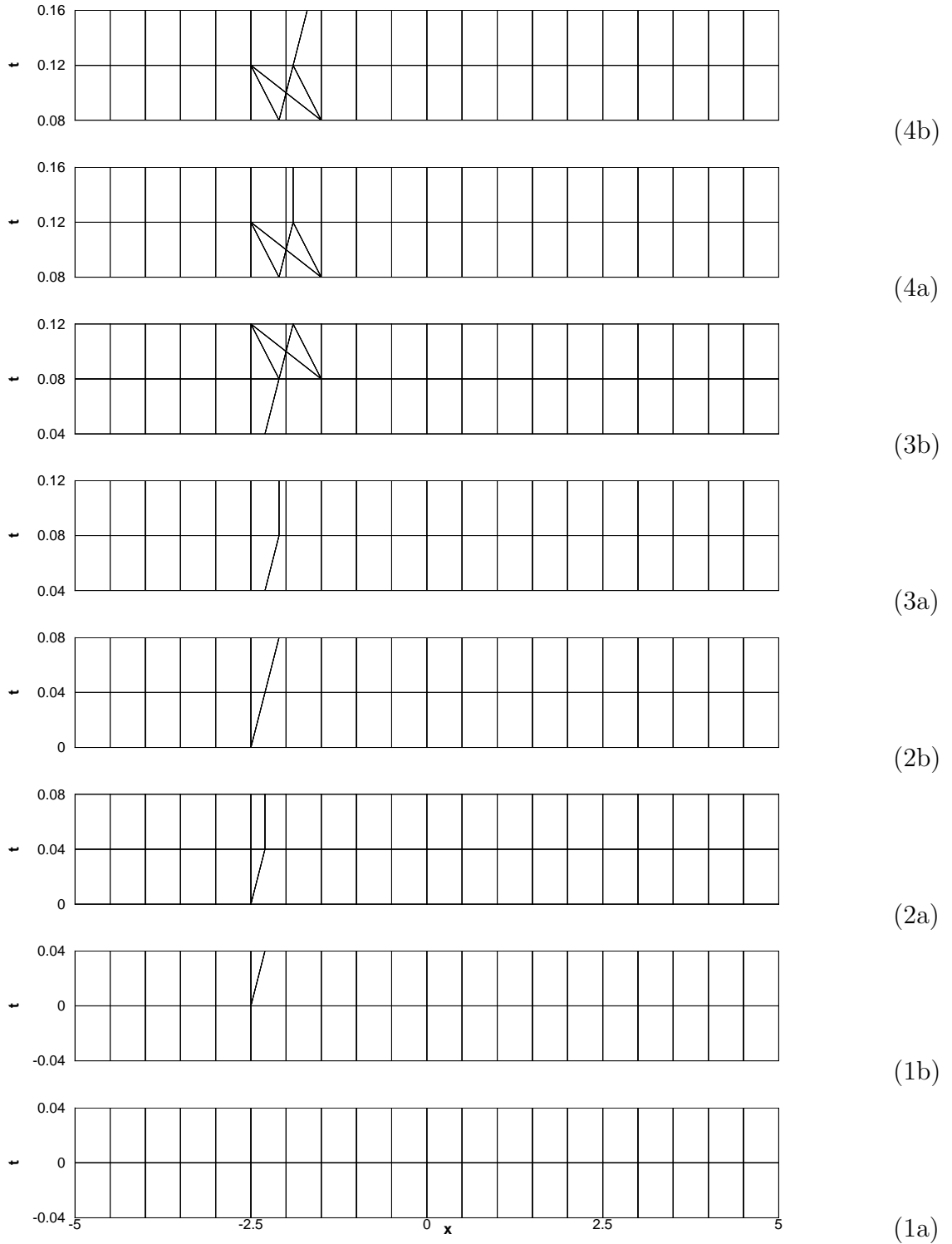


Figure 11. The various mesh updates performed in the linear advection test with interface (5.1.3) for four time slabs. After initialization the mesh looks like (1a), with the interface placed at $x = -2.5$. The level set is solved and the mesh is updated as in (1b). On the updated mesh the linear advection equation then is solved. Next, the computations move to the next time slab. A number of the subsequent mesh reinitializations and updates are shown in (2a)-(2b), (3a)-(3b) and (4a)-(4b).

5.2 Conservation of mass test

To test what happens when two interfaces move close towards collision, a test case is considered based on the conservation of mass equation:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho a(x))}{\partial x} = 0, \quad (62)$$

with ρ the density and $a(x)$ a given velocity. The initial condition is:

$$\rho(x, 0) = \rho_0(x) = \begin{cases} 1.0 & \text{for } |x| \leq 2.5 \\ 2.0 & \text{for } |x| > 2.5. \end{cases} \quad (63)$$

Extrapolating boundary conditions are used:

$$\rho(x, t) = \rho^-(x, t) \text{ at the boundary.} \quad (64)$$

The velocity is defined as $a(x) = -\alpha x$, with $\alpha = 1.0 > 0$ to ensure that the characteristics converge to $x = 0.0$. The exact solution is given as:

$$\rho(x, t) = \rho_0(xe^{\alpha t})e^{\alpha t}. \quad (65)$$

The simulations are performed on a spatial domain $[-5.0, 5.0]$ and run from time $t = 0.0$ to $t = 2.0$ at $CFL_{\Delta t} = 1.0$ while using linear basis functions. The tolerance for the flow residual is 1.0×10^{-12} . For the test with interface tracking the tolerances for the extension velocity and the level set residuals are both 1.0×10^{-6} and the tolerance for the mesh convergence is 1.0×10^{-3} . To be able to deal with the two interfaces, two level set functions and corresponding extension velocities are used. Presently, the method cannot deal with the case when the two interfaces are in the same element, since no rules have been defined to refine an element based on two level sets. However, in principle such rules can be added. Alternatively, when such a case occurs the background mesh can be h -refined. In the current tests this problem does not occur because neither of the two interfaces will ever pass the point $x = 0$. Since the solution is symmetric around $x = 0$, the interface positions only differ by their sign and so in the following discussion only the left interface position $x_{IF}^L(t)$ is considered. At time $t = 2.0$, the exact left interface position is $x_{IF}^L(t) = -2.5e^{-t}$. In Figures 12 and 13 the numerical solution and the space-time mesh are shown. The results are presented in Table 3, where the L^2 error, the order of accuracy, the error in the left interface position x_{IF}^L and its order at the final time $t = 2.0$ are shown for various mesh resolutions.

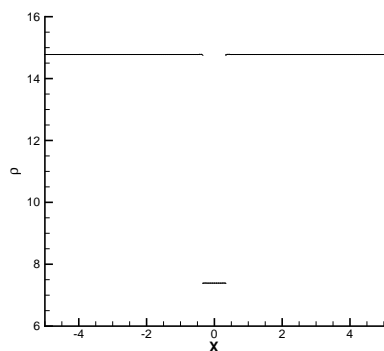


Figure 12. The numerical solution of the collision test with mesh refinement (5.2) at time $t = 2.0$ using 160 background elements.

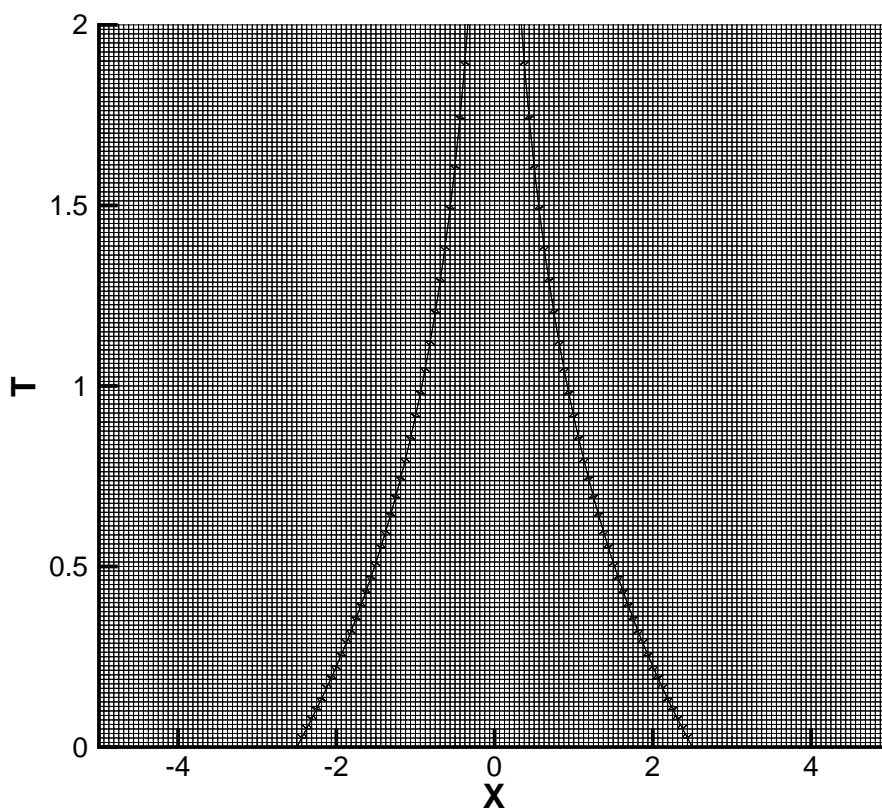


Figure 13. The refined space-time mesh for the collision test (5.2) using 160 background elements.

Table 3

Test results 5.2: Accuracy and order of convergence of the density and left interface position for the near collision test.

$N_x \times N_t$	L^2 error	order	x_{IF}^L error	x_{IF}^L order
20 x 20	2.790e-1	-	7.539e-4	-
40 x 40	1.370e-1	1.03	1.798e-4	2.07
80 x 80	6.667e-2	1.04	4.267e-5	2.08
160 x 160	3.361e-2	0.99	8.828e-6	2.27

5.3 Ideal gas Euler shock tube tests

To test the method for a case when the velocity is not predefined but is part of the flow variables, the method is applied to a one fluid Euler shock tube problem with ideal gas on both sides. An additional purpose of this test is to see how well the movement of the interface is captured by the method in a more practical and complex situation. Also, because the velocity is part of the flow variables in this case, the level set and flow equations are coupled, and hence the convergence of the coupled equations can be examined. The Euler equations express conservation of mass, momentum and energy and are given as:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} &= 0 \\
\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + p)}{\partial x} &= 0 \\
\frac{\partial(\rho E)}{\partial t} + \frac{\partial(u(\rho E + p))}{\partial x} &= 0,
\end{aligned} \tag{66}$$

where ρ is the density, u the fluid velocity, p the pressure and $\rho E = \rho u^2/2 + \rho e$ the total energy, with ρe the internal energy. In addition to these equations an equation of state (EOS) is required to account for the thermodynamic properties of the ideal gas:

$$e = p/\rho(\gamma - 1), \tag{67}$$

with $\gamma = 1.4$. The initial shock tube conditions are given by two constant states around an interface at $x = 0$. A left state $(\rho_L, u_L, p_L) = (2.37804, 0.0, 2.0 \times 10^5)$ for $x < 0$ and a right state $(\rho_R, u_R, p_R) = (1.18902, 0.0, 1.0 \times 10^5)$ for $x > 0$. The solution to (66) with these initial conditions has an expansion wave moving to the left with head speed $S_{LH} = -343.138$ and tail speed $S_{LT} = -241.218$, a contact wave moving to the right with speed $S_C = 84.9331$ and a shock wave also moving to the right with speed $S_R = 397.861$. Between the expansion and the contact wave the solution is constant and equal to the left star state $(\rho_L^*, u^*, p^*) = (1.84490, 84.9331, 1.40179 \times 10^5)$, and between the contact and

the shock wave the solution is also constant and equal to the right star state $(\rho_R^*, u^*, p^*) = (1.51174, 84.9331, 1.40179 \times 10^5)$ (see [69]). The contact wave can be considered an interface and is tracked using the two fluid method. Since this is a one fluid problem, at the interface the same flux as in the bulk fluids is used. The simulations are performed on a spatial domain $[-5.0, 5.0]$ and run from time $t = 0.0$ to $t = 0.005$ at $CFL_{\Delta t} = 1.0$ while using linear basis functions. The tolerance for the flow residual is 1.0×10^{-12} . For the test with interface tracking the tolerances for the extension velocity and the level set residuals are both 1.0×10^{-6} and the tolerance for the mesh convergence is 1.0×10^{-3} . A small amount of dissipation in space was used by setting the dissipation constants to $\epsilon_0 = 5.0 \times 10^{-2} \Delta x$, $\epsilon_1 = 0$, with Δx the background element spatial length.

5.3.1 HLLC flux for general meshes

For the numerical flux the HLLC flux (see [69]) is used, extended to general space-time meshes (for a full description see [49]), both in the bulk fluid and at the interface. Let $\mathbf{w} = (\rho, \rho u, \rho E)$ and $\mathbf{F}_F = (\rho u, \rho u^2 + p, u(\rho E + p))$. To simplify the notation in the description of the HLLC flux the subscript F will be omitted from the flux. The HLLC flux provides an accurate solution to the Riemann problem which is an initial value problem for the Euler equations, where the initial conditions consists of two constant states:

$$\mathbf{w}(x, 0) = \begin{cases} \mathbf{w}_L & \text{when } x < 0 \\ \mathbf{w}_R & \text{when } x > 0. \end{cases} \quad (68)$$

The formulation of the HLLC flux extended to general space-time meshes is given as:

$$\begin{aligned} \mathcal{H}_{HLLC} = & \frac{1}{2} \left(\mathbf{F}_L + \mathbf{F}_R \right. \\ & - (|S_L - v| - |S_M - v|) \mathbf{w}_L^* + (|S_R - v| - |S_M - v|) \mathbf{w}_R^* \\ & \left. + |S_L - v| \mathbf{w}_L - |S_R - v| \mathbf{w}_R - v(\mathbf{w}_L + \mathbf{w}_R) \right), \end{aligned} \quad (69)$$

with v the velocity of the interface. The HLLC flux is illustrated in Figure 14 for the case that $S_L \leq v < S_M$. It is assumed that the speeds are the same at both sides of the contact wave, so $S_M = u_L^* = u_R^* = u^*$. From the Rankine-Hugoniot relations $\mathbf{F}(\mathbf{w}_K) - \mathbf{F}(\mathbf{w}_K^*) = S_K(\mathbf{w}_K - \mathbf{w}_K^*)$ with $K = L$ or R for the left and the right waves, respectively, the following relations can be found for the star state variables:

$$\begin{aligned} \rho_K^* &= \rho_K \frac{S_K - u_K}{S_K - u^*} \\ \rho_K^* u^* (u^* - S_K) &= (p_K - p^*) + \rho_K u_K (u_K - S_K), \end{aligned} \quad (70)$$

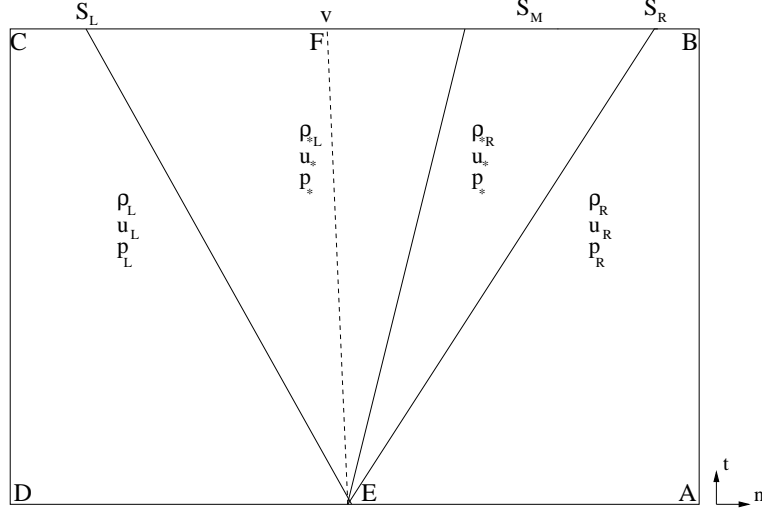


Figure 14. HLLC approximation of the solution of the Riemann problem for the Euler equations for ideal gas. The approximation consists of two shock waves with velocities S_L and S_R and a contact wave with velocity S_M separating four constant states. The HLLC flux gives an approximation for the (constant) flux at any point on EF with $\mathbf{n} = (n_n, n_t)$ the unit vector normal to EF with respect to the element considered.

and also an approximation for the speed $S_M = u^*$ of the contact wave can be obtained:

$$S_M = \frac{\rho_R u_R (S_R - u_R) - \rho_L u_L (S_L - u_L) + p_L - p_R}{\rho_R (S_R - u_R) - \rho_L (S_L - u_L)}. \quad (71)$$

The wave speeds S_L and S_R are estimated as:

$$S_L = u_L - a_L, \quad S_R = u_R + a_R. \quad (72)$$

By using the Rankine-Hugoniot relations for $\mathbf{w}_t + \mathbf{F}(\mathbf{w})_x = 0$ over the left wave and substituting the left and right states and wave speeds, the values of \mathbf{w}_L^* can be calculated as:

$$\mathbf{w}_L^* = \frac{S_L - u_L}{S_L - S_M} \mathbf{w}_L + \frac{1}{S_L - S_M} \begin{pmatrix} 0 \\ p^* - p_L \\ p^* S_M - p_L u_L \end{pmatrix}, \quad (73)$$

and likewise for \mathbf{w}_R^* by replacing L with R . By using the expression for ρ_K^* and u^* in the Rankine-Hugoniot relation for the momentum for the left and the right wave, two expressions for the intermediate pressure are found which are equal:

$$p^* = \rho_L (S_L - u_L) (S_M - u_L) + p_L = \rho_R (S_R - u_R) (S_M - u_R) + p_R. \quad (74)$$

Table 4

Test results 5.3.3: Accuracy and order of convergence of the density for the ideal gas Euler shock tube test without interface tracking.

$N_x \times N_t$	L^2 error	order
20 x 10	1.428e-1	-
40 x 20	1.057e-1	0.43
80 x 40	8.174e-2	0.37
160 x 80	6.622e-2	0.30

The HLLC solver is exact for a contact discontinuity. When $v = S_M$, which implies that the interface velocity is equal to the velocity of the contact wave, the HLLC flux becomes:

$$\mathcal{H}_{HLLC} = \frac{1}{2} \left(\mathbf{F}_L + \mathbf{F}_R + (S_M - S_L)(\mathbf{w}_L - \mathbf{w}_L^*) + (S_M - S_R)(\mathbf{w}_R - \mathbf{w}_R^*) - S_M(\mathbf{w}_L + \mathbf{w}_R) \right). \quad (75)$$

By inserting the expressions for \mathbf{w}_K^* , it follows that:

$$\mathcal{H}_{HLLC} = (0, p^*, p^* u^*)^T \quad (76)$$

which shows that there is no mass flux through the contact interface.

5.3.2 Interface integral

At the interface the modified face integral (59) is used. In the Euler test cases the flux \mathcal{H}_I^F is defined to be the contact flux (76). The flux \mathcal{H}_E^F is defined as the HLLC equivalent of the extrapolating flux:

$$\mathcal{H}_E^F(\mathbf{U}_h^-, \mathbf{U}_h^+, \mathbf{n}_{\mathcal{K}}) = \mathcal{H}_{HLLC}((\mathbf{U}_h^-, \mathbf{U}_h^-, \mathbf{n}_{\mathcal{K}})). \quad (77)$$

5.3.3 Ideal gas Euler shock tube without interface tracking

In the first test the performance of the STDG finite element method without interface tracking is tested for the Euler shock tube. In Table 4 the error and the order of accuracy of the solution are shown for various mesh resolutions at $t = 0.005$. Similar as in the linear advection test case 5.1.2, for this discontinuous solution the order of accuracy of the numerical solution is about $O(h^{1/2})$. The results are illustrated in Figure 15.

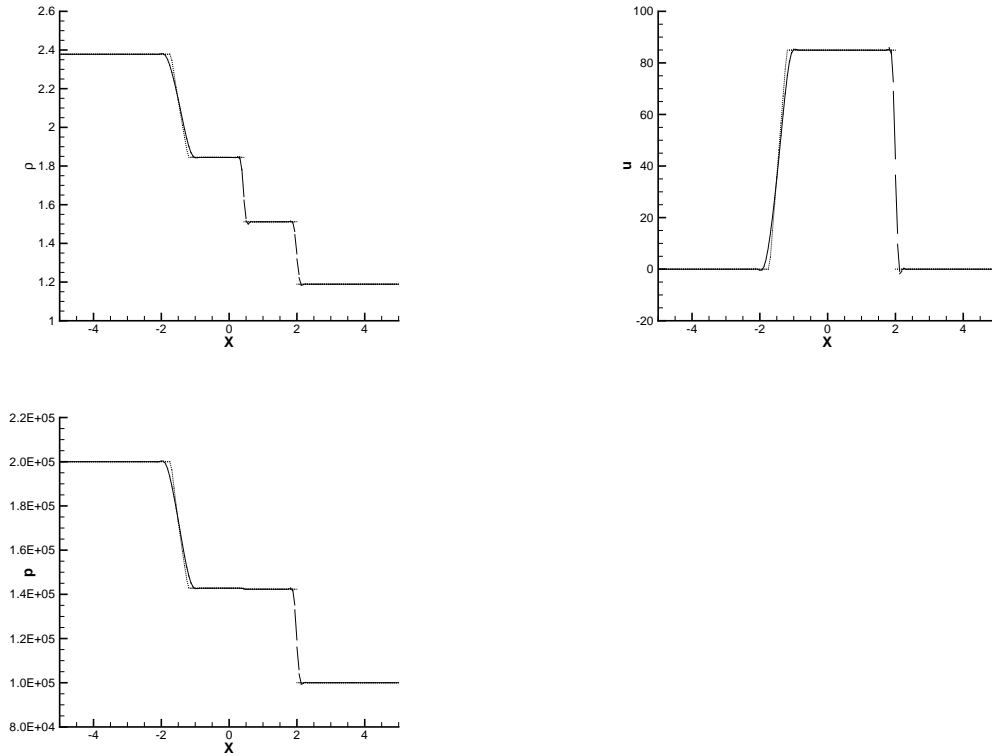


Figure 15. The exact (dotted) and numerical (solid) density, velocity and pressure for the Euler shock tube without mesh refinement using 160 background elements.

Table 5

Test results 5.3.4: Accuracy and order of convergence of the density and the interface position for the ideal gas Euler shock tube test with interface tracking.

$N_x \times N_t$	L^2 error	order	x_{IF} error	x_{IF} order
20 x 10	1.384e-1	-	4.257e-4	-
40 x 20	9.871e-2	0.49	1.142e-4	1.90
80 x 40	7.689e-2	0.36	1.308e-5	3.13
160 x 80	6.597e-2	0.22	5.164e-6	1.34

5.3.4 Ideal gas Euler shock tube with interface tracking

In the second test the performance of the STDG finite element method with two fluid refinement is tested for the Euler shock tube. In Table 5 the error and the order of accuracy of the solution as well as the error and order of accuracy in the position of the contact discontinuity are shown for various mesh resolutions at the end time $t = 0.005$. The results are illustrated in Figures 16 and 17. Compared to the case without mesh refinement the discontinuity at the contact discontinuity is captured much better. For the shock and rarefaction waves there is no difference in accuracy. The interface position corresponds

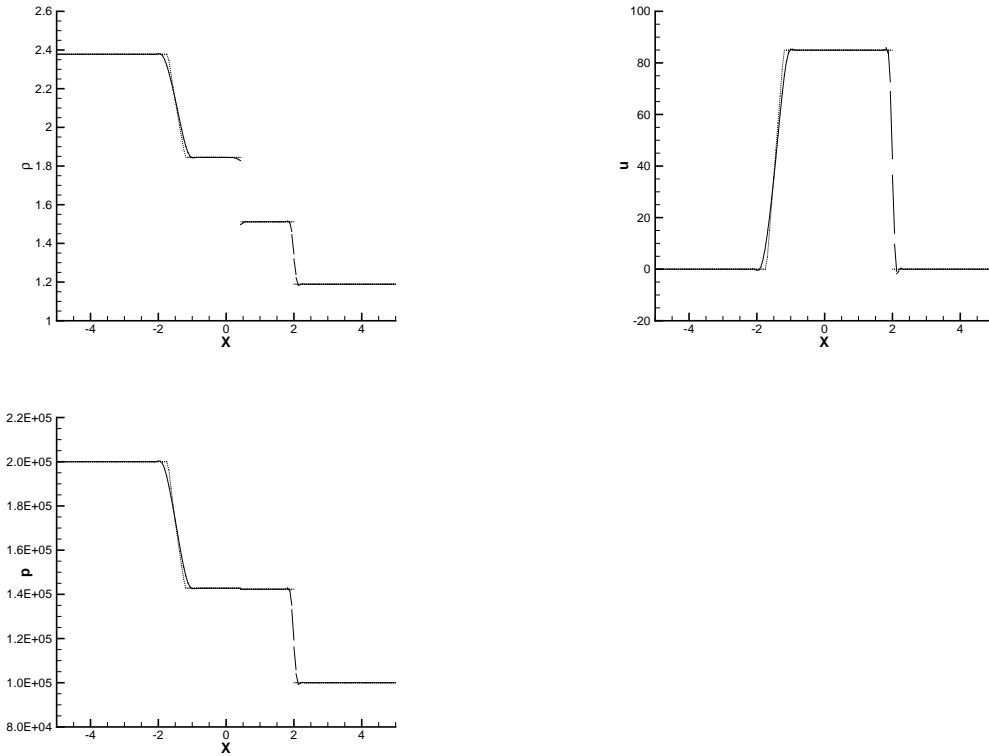


Figure 16. The exact (dotted) and numerical (solid) density, velocity and pressure for the Euler shock tube with mesh refinement using 160 background elements.

well to the exact interface position $x_{IF} = 0.4246655$ at $t = 0.005$.

5.4 Isothermal magma and ideal gas Euler shock tube test

In the last test a magma-ideal gas shock tube is simulated motivated by the high speed geological event analyzed in [70], [71] and [72]. It is interesting, firstly, because it is truly a two fluid problem, unlike the previous tests problems. Secondly, it has the additional difficulty of featuring very high density and pressure ratios which cause strong oscillations around the interface between the gas and magma with standard shock capturing schemes. The governing equations for an effectively compressible magma are the Euler equations for mass and momentum using conservative variables:

$$\partial_t \mathbf{w} + \partial_x \mathbf{F}_F(\mathbf{w}) = 0, \quad (78)$$

with

$$\mathbf{w}^i = \begin{pmatrix} \rho \\ \rho u \end{pmatrix}, \quad \mathbf{F}_F = \begin{pmatrix} \rho u \\ \rho u^2 + p \end{pmatrix}. \quad (79)$$

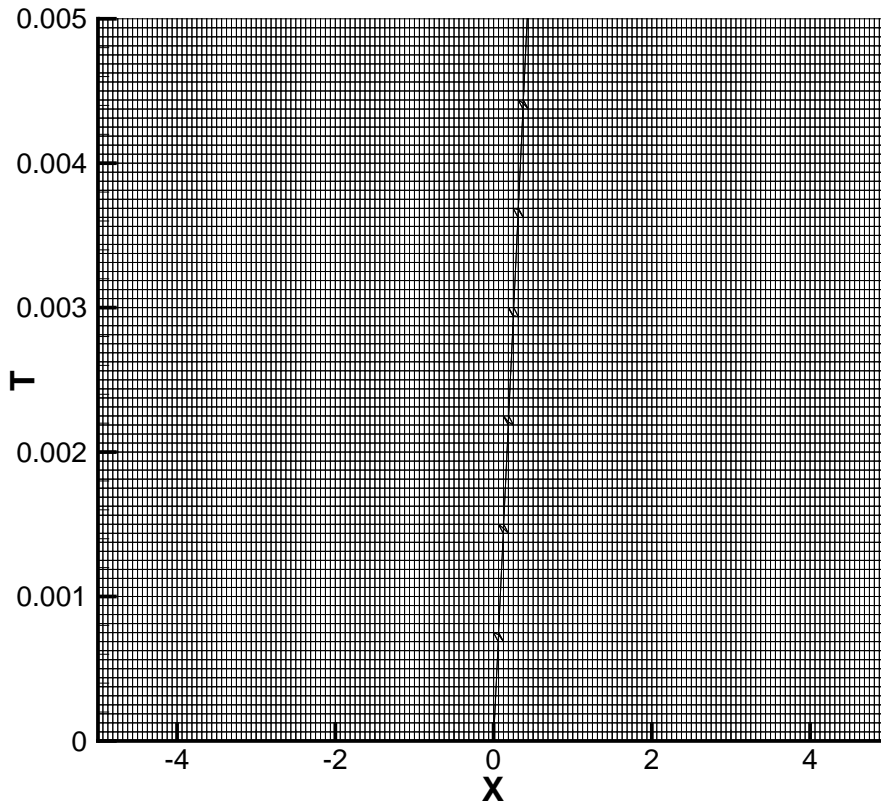


Figure 17. The mesh for the Euler shock tube with mesh refinement and 160 background elements.

The magma consists of a mixture of molten rock and 2 wt% (weight percentage) H_2O . At high pressure, the H_2O only has a liquid form. When the pressure decreases water vapor is formed within the mixture due to decompression effects. In this situation the magma effectively is a pseudo one-phase mixture. In explosive eruptions starting with a high pressure difference viscosity effects are negligible at leading order relative to the nonlinear inertial effects driven by the high bubble content ([70], [71]). The total mass fraction n_0 of H_2O in the magma consists of a fraction $n(p)$ which is exsolved in the magma as gas and a fraction $1 - n(p)$ which is dissolved in the magma as liquid. The mixture of magma and liquid H_2O has a density $\sigma = 2500 \text{ kg/m}^3$ and the water vapor has a density of ρ_g . The total void or bubble fraction of the mixture is given by $\alpha = n(p)\rho/\rho_g$. The density of the magma is defined as $\rho = \alpha\rho_g + (1 - \alpha)\sigma$. Using the relation for α and the ideal gas law $\rho_g = p/(RT)$ gives:

$$\rho = \left(\frac{n(p)R_m T}{p} + \frac{1 - n(p)}{\sigma} \right)^{-1}, \quad (80)$$

where $R_m = 462 \text{ J/kgK}$ is the mixtures gas constant. This relation is only valid when there are bubbles, i.e., $n(p) > 0$. The pressure at which there are no longer any bubbles in the mixture is called the critical pressure $p_c = (4/9) \times 10^8$. The magma considered will be assumed to be compressible, hence $p < p_c$. For $p \geq p_c$ the following relation can be used:

$$\rho = \sigma + c_m^{-2}(p - p_c), \quad (81)$$

with $c_m = 2000 \text{ m/s}$ the speed of sound in bubble free magma. The mass fraction $n(p)$ is assumed to satisfy Henry's law, which is valid when bubbles and melt are in equilibrium:

$$n(p) = n_0 - S_h p^\beta. \quad (82)$$

For basaltic high volatile magma, $n_0 = 0.02$, $\beta \approx 0.5$, $T = 1200 \text{ K}$ and $S_h = 3.0 \cdot 10^{-6} \text{ Pa}^{-\beta}$. The magma is assumed to be isothermal at a temperature of 1200 K . For isothermal magma the density depends only on the pressure, $\rho = \rho(p)$. The speed of sound a is defined for isothermal magma as:

$$\begin{aligned} 1/a^2 &\equiv \left(\frac{\partial \rho}{\partial p} \right)_T = -\rho^2 \frac{\partial(1/\rho)}{\partial p} \\ &= -\rho^2 \left[\frac{dn(p)}{dp} \left(\frac{R_m T}{p} + \frac{1}{\sigma} \right) - \frac{n(p) R_m T}{p^2} \right]. \end{aligned} \quad (83)$$

The initial state of the isothermal magma is $\rho_L = 535.195$, $u_L = 0.0$ and $p_L = 5.0 \times 10^6$, and the initial state of the ideal gas is $\rho_R = 1.18902$, $u_R = 0.0$ and $p_R = 1.0 \times 10^5$. The exact solution can be calculated by solving the Riemann problem and consists of a left moving expansion wave with head and tail speeds of $S_{LH} = -97.2861$, $S_{LT} = 186.409$ respectively, a contact wave which can be identified with the magma-air interface and moves with speed $S_C = 286.329$ and a right moving shock wave with speed $S_R = 287.821$. The left and right star states are defined as: $\rho_L^* = 28.0517$, $\rho_R^* = 2.45364$, $u^* = 286.329$, $p^* = 2.89134 \times 10^5$. At $t = 0.005$ the exact interface position is approximately 1.431645 . The simulations are performed on a spatial domain $[-5.0, 5.0]$ and run from time $t = 0.0$ to $t = 2.0$ at $CFL_{\Delta t} = 1.0$ while using linear basis functions. The tolerance for the flow residual is 1.0×10^{-12} . For the test with interface tracking the tolerances for the extension velocity and the level set residuals are both 1.0×10^{-6} and the tolerance for the mesh convergence is 1.0×10^{-3} . For the interface flux the flux defined in 5.3.2 is used. To deal with the strong discontinuities and the consequent problems caused by these some numerical dissipation has to be added, both in space and time. For the ideal gas the dissipation constants $\epsilon_0 = 5.0 \times 10^{-2} \Delta x$, $\epsilon_1 = 5.0 \times 10^{-3} \Delta t$ were used, with Δx the background element length in space and Δt the elements length in time. Since the values of flow variables of the magma are approximately a factor 1.0^2 larger than those of the ideal gas, the dissipation coefficients used for the magma were taken a factor 1.0^{-2} smaller than those

Table 6

Test results 5.4: Accuracy and order of convergence of the density and interface position for the isothermal magma and ideal gas Euler shock tube test.

$N_x \times N_t$	L^2 error	order	x_{IF} error	x_{IF} order
20 x 10	5.289e1	-	1.230e-1	-
40 x 20	4.066e1	0.38	1.433e-1	-0.22
80 x 40	3.111e1	0.39	1.188e-1	0.27
160 x 80	2.339e1	0.41	8.334e-2	0.51

for the ideal gas, resulting in $\epsilon_0 = 5.0 \times 10^{-4} \Delta x$, $\epsilon_1 = 5.0 \times 10^{-5} \Delta t$. The test results are presented in Table 6 and are illustrated in Figures 18 and 19.

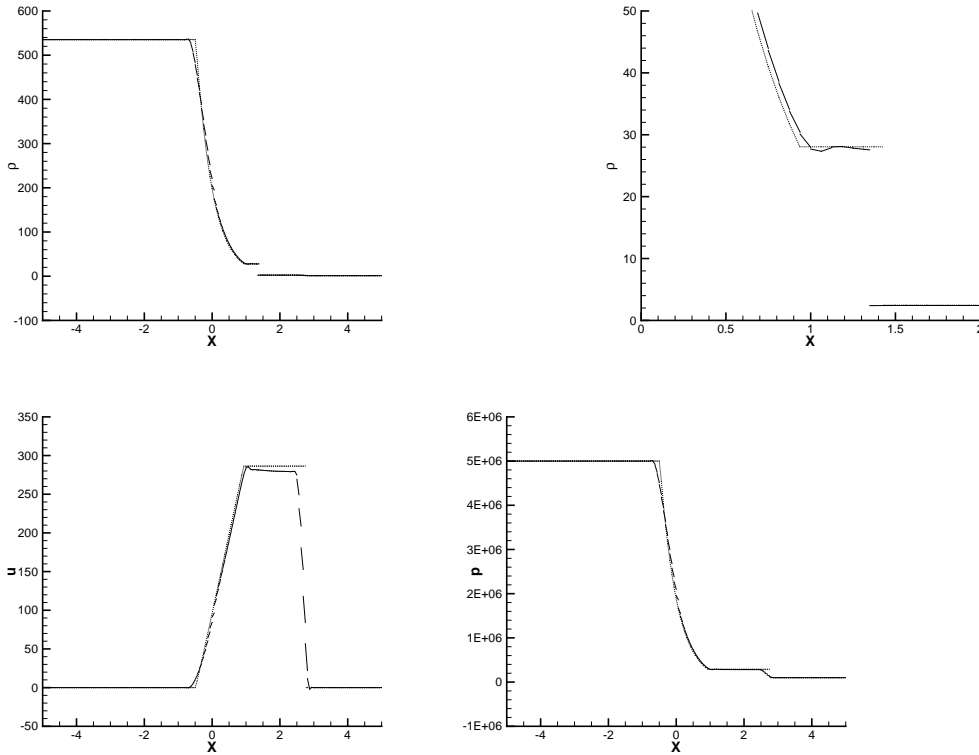


Figure 18. The exact (dotted) and numerical (solid) density, density zoom, velocity and pressure for the Euler magma air shock tube with mesh refinement using 160 background elements.

6 Discussion

A space-time discontinuous Galerkin finite element method for two fluid flows has been presented. The method combines the level set method with mesh

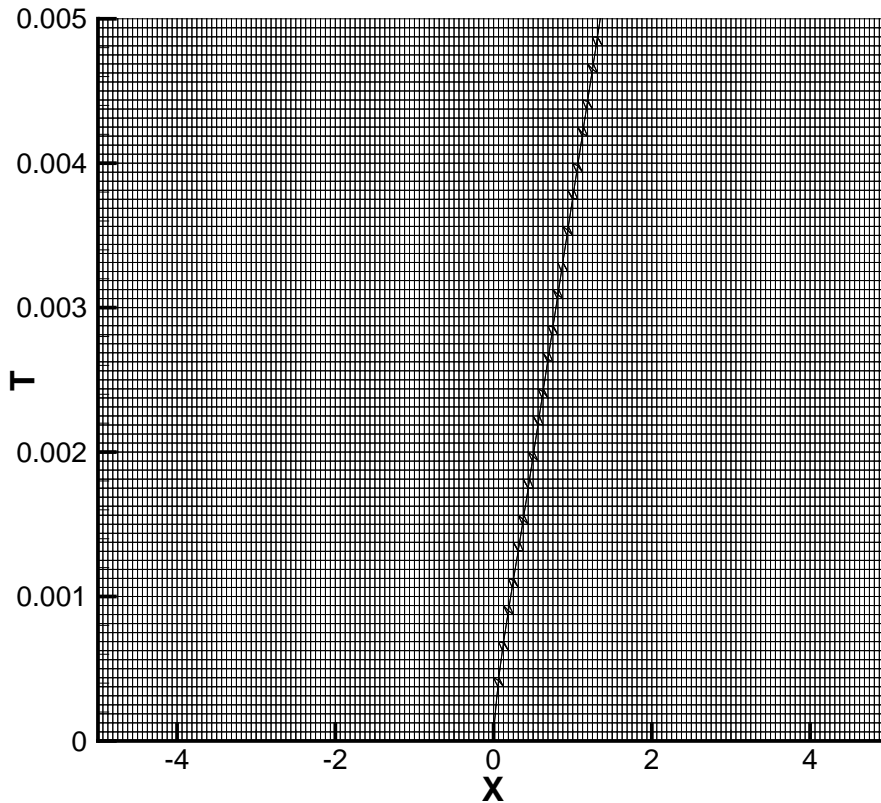


Figure 19. The mesh for the Euler magma air shock tube with mesh refinement using 160 background elements.

refinement to track the interfaces between fluids in space-time. The level set function is advected with the interface velocity which for this purpose is extended into the domain. Oscillations around the interface are eliminated by choosing a suitable interface flux. The method was tested using linear advection and Euler shock tube problems with promising results. The next efforts are directed towards making the method less expensive by firstly using a space-time narrow band approach for the level set and extension velocity and secondly enhancing the refinement strategy in order to avoid very small elements. The long term efforts are directed towards creating a three dimensional space-time extension of the method.

Acknowledgments

The author is kindly indebted to L.Pesch and A.Bell for the valuable discussions, suggestions and support. Special thanks are due to Vijaya Ambati who

devised the interface flux employed, and was strongly involved in discussions on modelling the extension velocity ([77]). O. Bokhove acknowledges a fellowship (2001–2006) from the Royal Netherlands Academy of Arts and Sciences (KNAW).

References

- [1] SCARDOVELLI, R. & ZALESKI, S. 1999 Direct numerical simulation of free-surface and interfacial flow, *Annu.Rev.Fluid Mech.* 31, **567–603**.
- [2] HARLOW, F.H. & WELCH, J.E. 1965 Numerical calculation of time-dependent viscous incompressible flow, *Phys.Fluids* 8, **2182**.
- [3] DALY, B.J. 1969 Numerical study of the effect of surface tension on interface instability, *Phys.Fluids* 12, **1340**.
- [4] YOUNG, S.S., WHITE, J.J., CLARK, E.S. & OYANAGI, Y. 1980 A basic experimental study of sandwich injection moulding with sequential injection, *Pol.Eng.Sci.* 20 **798–804**.
- [5] HIRT, C.V. & NICHOLS, B.D. 1981 Volume of fluid (VOF) methods for the dynamics of free boundaries, *J.Comput.Phys* 39, **201–255**.
- [6] PUCKETT, E.G. & SALTZMAN, J.S. 1992 A 3D adaptive mesh refinement algorithm for interfacial gas dynamics, *Physica D.* 60, **84–93**.
- [7] SAURELL, R. & ABGRALL, R. 1998 A simple method for compressible multifluid flows, *SIAM J.Sci.Comput.* 21, no 3,**1115-1145**.
- [8] SUSSMAN, M. & PUCKETT, E.G. 2000 A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows, *J.Comp.Phys.* 162, **301–337**.
- [9] OSHER, S.J. & SETHIAN, J.A. 1988 Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations, *J.Comp.Phys.* 79, **12–49**.
- [10] SUSSMAN, M., SMEREKA, P. & OSHER, S. 1994 A level set approach for computing solutions to incompressible two-phase flow, *J.Comput.Phys.* 114, **146–159**.
- [11] ADALSTEINSSON, D. & SETHIAN, J.A. 1995 A fast level set method for propagating interfaces, *J.Comp.Phys.* 118, **269–277**.
- [12] SETHIAN, J.A. 1996 *Level Set Methods*, Cambridge Univ.Press.
- [13] SETHIAN, J.A. & SMEREKA, P. 2003 Level set methods for fluid interfaces, *Annu.Rev.Fluid.Mech.* 35, **341–372**.
- [14] RICHTMYER, R.D. & MORTON, K.W. 1967 *Difference Methods for Initial-Value Problems*, Inter-science, New york.
- [15] MORETTI, G., GROSSMAN, B. & MARCONI, F. 1972 *A complete numerical technique for the calculation of three dimensional inviscid supersonic flow*, Rep. No. 72-192, American Institute for Aerodynamics and Astronautics, New York.

- [16] GLIMM, J., ISAACSON, E., MARCHESIN, D. & MCBRYAN, O. 1981 Front tracking for hyperbolic systems, *Adv.Appl.Math.* 2, **91-119**.
- [17] GLIMM, J. & MCBRYAN, O. 1985 A computational model for interfaces, *Adv.Appl.Math.* 6, **422-435**.
- [18] GLIMM, J., GROVE, J.W., LI, X.L., SHYUE, K.-M., ZHANG, Q. & ZENG, Y. Three-dimensional front tracking, *SIAM J.Sci.Comput.* 19, **201-225**
- [19] TRYGGVASON, G. & UNVERDI, S.O. Computations of three-dimensional Rayleigh-Taylor instability, *Phys.Fluids A.* 5, **656-659**.
- [20] UNVERDI, S.O. & TRYGGVASON, G. 1992 Computations of multi-fluid flows, *Phys. Fluids D* 60, **70-83**.
- [21] UNVERDI, S.O. & TRYGGVASON, G. A front-tracking method for viscous, incompressible multi-fluid flows, *J.Comput.Phys.* 100, **25-37**
- [22] LEVEQUE, R.J. & SHYUE, K.-M. 1996 2-dimensional front tracking based on high resolution wave propagation methods, *J.Comp.Phys.* 123, **354-368**.
- [23] HYMAN, J.M. 1984 Numerical methods for tracking interfaces, *Phys.D.* 12, **396-407**.
- [24] RUDMAN, J.M. 1997 Volume-tracking methods for interfacial flow calculations, *Int. J. Numer. Heat Fluid Flow* 24, **671-691**.
- [25] DE ZEEUW, D., POWELL, K.G. 1993 An adaptively refined Cartesian mesh solver for the Euler equations, *J.Comp.Phys.* 104(1), **56-68**.
- [26] PEMBER, R.B., BELL, J.B., COLLELA, P., CRUTCHFIELD, W.Y. & WELCOME, M.L. 1994 An adaptive Cartesian mesh method for unsteady compressible flow in irregular regions, *J.Comp.Phys.* 120, **278-304**.
- [27] QUIRK, J. 1994 An alternative to unstructured meshes for computing gas dynamic flows around arbitrarily complex two-dimensional bodies, *Comput.Fluids* 23, **125-142**.
- [28] COIRIER, W.J. & POWELL, K.G. 1995 An accuracy assessment of Cartesian-mesh approaches for Euler equations, *J.Comp.Phys.* 117, 121-131.
- [29] YANG, G., CAUSON, D.M., INGRAM, D.M., SAUNDERS, R. & BATTEN, P. 1997 A Cartesian cut element method for compressible flows. Part A: Static body problems, *Aeronaut. J.* 2, **47-56**.
- [30] YANG, G., CAUSON, D.M., INGRAM, D.M., SAUNDERS, R. & BATTEN, P. 1997 A Cartesian cut element method for compressible flows. Part B: Moving body problems, *Aeronaut. J.* 2, **57-65**.
- [31] YANG, G., CAUSON, D.M. & INGRAM 2000 Calculation of compressible flows about complex moving geometries using a three-dimensional Cartesian cut element method, *Int.J.Numer.Meth.Fluids* 33, **1121-1151**.
- [32] ALMGREN, A.S., BELL, J.B., COLLELA, P. & MARTHALER, T. 1997 A Cartesian mesh projection method for the incompressible Euler equations in complex geometries, *SIAM J.Sci.Comp.* 18(5), **1289-1309**.
- [33] JOHANSEN, H. & COLELLA, P. 1998 A Cartesian mesh embedded boundary method for Poisson's equation on irregular domains,

- J.Comp.Phys.* 147(1), **60–85**.
- [34] CAUSON, D.M., INGRAM, D.M., MINGHAM, C.G., YANG, G. & PEARSON, R.V. 2000 Calculation of shallow water flows using a Cartesian cut element approach, *Adv.Water Resour.* 23, **545–562**.
- [35] UDAYKUMAR, H.S., KAN, H.C., SHYY, W., TRAN SON TAY, R. 1997 Multiphase dynamics in arbitrary geometries on fixed Cartesian meshes, *J.Comput.Phys.* 137(2), **366–405**.
- [36] UDAYKUMAR, H.S., MITTAL, R., RAMPUNGGON, P. & KHANNA, A. 2001 A sharp interface Cartesian mesh method for simulating flows with complex moving boundaries, *J.Comput.Phys.* 174(1), **345–380**.
- [37] UDAYKUMAR, H.S., MITTAL, R. & RAMPUNGGON, P. 2002 Interface tracking finite volume method for complex solid-fluid interactions on fixed meshes, *Comm.Nummer.Meth.Eng.* 18(2), **89–97**.
- [38] TUCKER, P.G. & PAN, Z. 2000 A Cartesian cut element method for incompressible viscous flow, *Appl.Math.Modell.* 24, **591–606**.
- [39] YE, T., MITTAL, R., UDAYKUMAR, H.S. & SHYY, W. 1999 An accurate Cartesian mesh method for viscous incompressible flows with complex immersed boundaries, *J.Comp.Phys.* 156(2), **209–240**.
- [40] BOKHOVE, O., WOODS, A.W., & DE BOER, A. 2005 Magma Flow through Elastic-Walled Dikes. *Theor. Comput. Fluid Dyn.* 19, **261–286**.
- [41] RYSKIN, G. & LEAL, L.G. 1984 Numerical solution of free-boundary problems in fluid mechanics, Part 2: Buoyancy-driven motion of a gas bubble through a quiescent liquid, *em J.Fluid Mech.* 148, **19–36**.
- [42] DANDY, D.S. & LEAL, L.G. 1989 Buoyancy-driven motion of a deformable drop through a quiescent liquid at intermediate Reynolds numbers, *J.Fluid.Mech.* 339, **161–192**.
- [43] CUENOT, B., MAGNAUDET, J. & SPENNATO, B. 1997 The effects of slightly soluble surfactants on the flow around a spherical bubble, *J.Fluid.Mech.* 339, **25–53**.
- [44] FRITTS, M.J., COWLEY, W., TREASE, H.E, EDS. 1985 *The Free Lagrange Method*, Lect. Notes Phys. Vol. 238, Springer-Verlag, New York.
- [45] FYFE, D.E., ORAN, E.S. & FRITTS, M.J. 1988 Surface tension and viscosity with Lagrangian hydrodynamics on a triangular mesh, *J.Comput.Phys.* 76, **349–384**.
- [46] FUKAI, J., SHIIBA, Y., YAMAMOTO, T., MIYATAKE, O. POULIKAKOS, D. ET.AL. 1993 Wetting effects on the spreading of a liquid droplets colliding with a flat surface: experiment and modeling, *Phys.Fluids A.* 5, **2588–2599**.
- [47] MAGNAUDET, J., RIVERO, M. & FABRE, J. 1995 Accelerated flows around a rigid sphere or a spherical bubble. Part I: Steady straining flow, *J.Fluid.Mech.* 284, **97–136**.
- [48] VAN DER VEGT, J.J.W. & VAN DER VEN, H. 1998 Discontinuous Galerkin finite element method with anisotropic local grid refinement for inviscid compressible flows, *J. Comput. Phys.* 141, **46–77**.
- [49] VAN DER VEGT, J.J.W. & VAN DER VEN, H. 2002 Space-time discontinuous Galerkin finite element method with dynamic mesh motion for

- Inviscid Compressible Flows, *J. Comput. Phys.* 182, **546–585**.
- [50] KLAIJ, C.M., VAN DER VEGT, J.J.W. & VAN DER VEN, H. 2006 Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations, *J. Comput. Phys.* 217, **589–611**.
- [51] AMBATI, V.R. & BOKHOVE, O. 2007 Space-time discontinuous Galerkin finite element method for shallow water flows, In press & online *J. Comp. Appl. Math.*
- [52] SUDIRHAM, J.J., VAN DER VEGT, J.J.W. & VAN DAMME, R.M.J. 2006 Space-time discontinuous Galerkin method for advection-diffusion problems on time-dependent domains, *Appl. Num. Math.* 56, **1491–1518**
- [53] VAN DER VEGT, J.J.W. & XU, Y. 2006 Space-time discontinuous Galerkin method for nonlinear water waves, accepted by Journal of Computational Physics, November 2006.
- [54] REED, W.H. & HILL, T.R. 1973 *Triangular mesh methods for the neutron transport equation*, Los Alamos Scientific Laboratory report LA-UR-73-479, Los Alamos, NM.
- [55] COCKBURN, B. & SHU, C.W. 1989 TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws II: General framework, *Math. Comp.* 52, **411–435**.
- [56] COCKBURN, B., LIN, S.Y. & SHU, C.W. 1989 TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws III: One dimensional systems, *J. Comp. Phys.* 84, **90–113**.
- [57] COCKBURN, B., HOU, S. & SHU, C.W. 1990 TVB Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws IV: The multidimensional case, *Math. Comp.* 54, **545–581**.
- [58] COCKBURN, B. & SHU, C.W. 1998 The Runge-Kutta discontinuous Galerkin finite element method for conservation laws V: Multidimensional systems, *J. Comput. Phys.* 141, **199–224**.
- [59] COCKBURN, B. & SHU, C.W. 1998 The local discontinuous Galerkin method for time-dependent convection-diffusion systems, *SIAM J. Numer. Anal.* 35(6), **2440–2463**.
- [60] ATKINS, H.L. & SHU, C.-W. 1998 Quadrature-free implementation of discontinuous Galerkin methods for hyperbolic equations, *AIAA J.* 36, **775–782**.
- [61] BASSI, F. & REBAY, S. 1997 A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations, *J. Comput. Phys.* 131, **267–279**.
- [62] BISWAS, R., DEVINE, K.D. & FLAHERTY, J. 1994 Parallel, adaptive finite element methods for conservation laws, *Appl. Numer. Math.* 14, **255–283**.
- [63] ODEN, J.T., BABUSKA, I. & BAUMANN, C.E. 1998 A discontinuous hp finite element method for diffusion problems, *J. Comput. Phys.* 146, **495–519**.
- [Tassi et.al.] TASSI, P., BOKHOVE, O. & AND VIONNET, C. 2007 Space discontinuous Galerkin method for shallow water flows. In press *Advances*

- in water resources 2006.
- [64] COCKBURN, B., KARNIADAKIS, G.E. & SHU, C.W. 1999 *Discontinuous Galerkin Methods: Theory, Computation and Applications*, Springer.
- [65] UNGOR, A. & SHEFFER, A. 2002 Pitching tents in space-time: mesh generation for discontinuous Galerkin method, *Int.J.Found.Comp.Sci.* 13, **201–221**.
- [66] JAFFRE, J., JOHNSON, C. & SZEPESSY, A. 1995 Convergence of the discontinuous Galerkin finite element method for hyperbolic conservation laws *Math.Models Meth.Appl.Sci.* 5, **367**.
- [67] KRÖNER 1996 *Numerical schemes for conservation laws.*, Wiley und Teubner.
- [68] KLAIJ, C.M., VAN DER VEGT, J.J.W. & VAN DER VEN, H. 2006 Pseudo-time stepping methods for space-time discontinuous Galerkin discretizations of the compressible Navier-Stokes equations, *J.Comp.Phys.* 219, **622–643**
- [Pesch et.al.] PESCH, L., BELL, A., SOLLIE, W.E.H., AMBATI, V.R., BOKHOVE, O. & VAN DER VEGT, J.J.W. 2006 hpGEM- A software framework for Discontinuous Galerkin finite element methods, *ACM Transactions on Software.* under revision per Nov., submitted May 31st 2006.
- [69] TORO, E. F. 1997 *Riemann solvers and numerical methods for fluid dynamics.* Springer Verlag, Berlin.
- [70] WOODS, A.W., SPARKS, S., BOKHOVE, O., LEJEUNE, A.M., CONNOR, C., HILL, B. 2002 Modelling magma-drift interaction at the proposed high-level radioactive waste repository at Yucca Mountain, Nevada, USA, *Geophys. Res. Lett.* 29 10.1029.
- [71] BOKHOVE, O. 2002 Decompressie van magma in opslagtunnels, *Nederlands Tijdschrift voor Natuurkunde* 68, **232–235**, English version: <http://eprints.eemcs.utwente.nl/>.
- [72] BOKHOVE, O. 2001 Numerical modeling of magma-repository interactions, University of Twente, 97 pp, <http://eprints.eemcs.utwente.nl/>.
- [73] EDWARDS, D.A., BRENNER, H. & WASAN, D.T. 1991 *Interfacial processes and rheology.* Butterworth-Heinemann, Stoneham, Reed publishing.
- [74] SCRIVEN, L.E. 1960 Dynamics of a fluid interface, *Chem. Eng. Sci.* 12, **98–108**.
- [75] STROUD, A.H. 1971 *Approximate calculation of multiple integrals.* Prentice-Hall, New Jersey.
- [76] ABRAMOWITZ, M., STEGUN, I.A. 1964 *Handbook of mathematical functions.* Dover publications, New York.
- [77] AMBATI, V.R. 2006 Flooding and drying in discontinuous Galerkin discretizations of shallow water equations, ECCOMAS Egmond aan zee, European Conference on Computational Fluid Dynamics. <http://proceedings.fyper.com/eccomas CFD2006/>.