A Discontinuous Galerkin Front Tracking Method for Two-Phase Flows with Surface Tension

Vinh-Tan Nguyen[a], Jaime Peraire[a,b], Boo Cheong Khoo[a,c], Per-Olof Persson[d]

[a] *Singapore-MIT Alliance, E4-04-10, 4 Engineering Drive 3, Singapore 117576*
[b] *Massachusetts Institute of Technology, 77 Massachusetts Avenue, MA 02139, USA*
[c] *National University of Singapore, 21 Lower Kent Ridge Road, Singapore 119077*
[d] *University of California, Berkeley, 970 Evans Hall #3840, Berkeley, CA 94720-3840, USA*

Abstract

A Discontinuous Galerkin method for solving hyperbolic systems of conservation laws involving interfaces is presented. The interfaces are represented by a collection of element boundaries and their position is updated using an arbitrary Lagrangian-Eulerian method. The motion of the interfaces and the numerical fluxes are obtained by solving a Riemann problem. As the interface is propagated, a simple and effective remeshing technique based on distance functions regenerates the grid to preserve its quality. Compared to other interface capturing techniques, the proposed approach avoids smearing of the jumps across the interface which leads to an improvement in accuracy. Numerical results are presented for several typical two-dimensional interface problems, including flows with surface tension.

Key words: front tracking, discontinuous Galerkin, material interface, mesh generation, surface tension

1. INTRODUCTION

Interfaces separating regions in space where sudden changes in material properties or flow conditions occur, are found in many engineering applications including compressible flows with shocks, multi-phase flow problems, and fluid-structure interactions. Consider, for instance, the problem of drop deformation under the presence of surface tension. In this case, the interface separates two different fluids and the effect of surface tension results in a jump of pressure across the interface. A successful numerical method for these problems has to resolve the discontinuities without any oscillations while keeping track of the interface propagation. Furthermore, it should be conservative and be accurate. These often conflicting requirements makes the design of numerical schemes for these flows particularly challenging.

Essentially, there are two major approaches for handling discontinuous solutions: the discontinuity capturing and the discontinuity tracking methods. In the discontinuity capturing methods, the discontinuities are not represented as sharp jumps but smeared over a certain length scale which depends on the grid resolution. The effect of representing the sharp jumps in a continuous manner over the mesh has the effect of reducing the accuracy of the solution to first order. These capturing approaches have been frequently applied and work well for nonlinear shock discontinuities, but they are less successful for problems involving contact discontinuities. For shock discontinuities, it is easy to maintain the width of the transition layer small as the integration progresses. This is because the nonlinearity in the solution drives the solution to become steeper as time progresses. The situation is very different for contact discontinuities. In such cases, the linear character of these discontinuities causes the width of the transition region to increase monotonically over time and, as a consequence, long time integrations can only be performed with very high order schemes.

Alternatively, in the front tracking method, the fronts are considered as internal boundaries and explicitly tracked within the mesh. This provides a much better resolution of the jumps across the interfaces but poses some serious meshing chellenges. The first implementation of a front tracking method was carried out by Glimm *et al* [11] for fluid discontinuities in two space dimensions, with extension to higher dimensions in [12]. In their approach, the sharp jump across the interface is handled by a Riemann solver which utilizes ghost cells where the unknowns are extrapolated across the interface. The use of extrapolation combined with ghost cells was further developed in the ghost fluid method (GFM) proposed by Fedkiw *et al* [5] and subsequently modified by Liu *et al* [22] for strong shock interactions. In the latter, the interface is represented by a level set function and a band of ghost cells is created at either side of the interface. The GFM has been shown to work well on a range of problems involving material interfaces and interaction with shock waves, and it is easily extended to problems in higher dimensions. However, the GFM method and its variants are not conservative and are only first order accurate due to their treatment of the discontinuities.

A number of conservative front tracking methods have been developed, for exampleby Glimm *et al* [9], Mao [25], and Gloth *et al* [13]. Glimm *et al.* [9] presented a scheme which tracks the discontinuities sharply while preserving the conserved quantities at a discrete level. It was further developed and modified in [10] with improved accuracy and various numerical experiments in one and two dimensions. This scheme is conservative with second order accuracy in the interior region and first order accuracy at the front. A general problem for all front tracking schemes is the handling of the topology of the front. In [10], the front is handled by a technique which is straight-forward in one space dimension but more complex in higher dimensions. Recently, Liu *et al* [23] proposes and extension of the method to consider system of nonlinear conservation laws in n dimensions. Another approach to handle the front using finite volumes on unstructured mesh methods is presented in Gloth *et al* [13]. Here, the location, geometry, and propagation of the fronts are described by the level set method.

In this article, we present a front tracking method for tracking discontinuities using the discontinuous Galerkin (DG) method. The interface is explicitly represented via internal boundaries in the DG mesh. Within each fluid domain an Arbitrary Eulerian-Lagrangian (ALE)method is used to account for the grid deformation. The motion of the interface between the different fluid region is either prescribed or obtained by solving a Riemann problems [40] for the moving velocity. As the interface is propagating, the computational mesh deforms and needs to be modified. This is done efficiently using a mesh generation technique [27] for implicit geometries described by signed distance functions. One of the main advantages of the proposed approach is the incorporation of the front tracking technique into the context of high order discontinuous Galerkin methods. The interface is sharply tracked while conservation errors are minimized. We present several numerical examples aimed at demonstrating the capabilities of the presented technique. In particular, we consider the problem of drop deformation under the effect of acoustic excitation.

2. THE DISCONTINUOUS GALERKIN FRONT TRACKING METHOD

2.1. The Discontinuous Galerkin ALE Formulation

Consider first a first order system of conservation laws

$$\boldsymbol{u}_t + \nabla \cdot \boldsymbol{F}(\boldsymbol{u}) = 0, \tag{1}$$

over the domain Ω with the appropriate boundary conditions applied on the domain boundary $\partial\Omega$, and the material interface ψ separating two regions containing fluid with different properties, as illustrated in figure

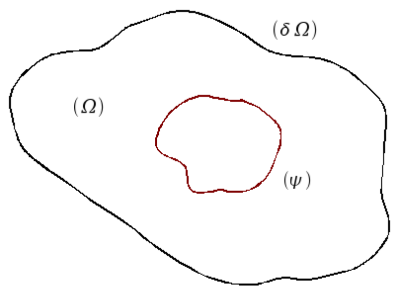


Figure 1: Computational domain Ω with interface ψ

1. Here, $\boldsymbol{u}(\boldsymbol{x},t) = \{u_i(\boldsymbol{x},t)\}_{i=1}^m$ is the conservative state vector with m components and $\boldsymbol{x} = (x_1,\ldots,x_d)$ is the position vector in d-dimensional space. The fluxes associated to the conserved variables are denoted by $\boldsymbol{F}(\boldsymbol{u}) = \{F_{ij}(\boldsymbol{u})\}_{i,j=1,1}^{m,d}$.

At any given time, we assume a triangulation $\mathcal{T}_h$ of the domain Ω into elements $\Omega = \bigcup_{\kappa\in\mathcal{T}_h}\kappa$, such that interface ψ can be represented as a collection of element edges. In addition, we consider the discontinuous finite element space associated with $\mathcal{T}_h$,

$$\mathcal{V}_h^p(\Omega) = \{\boldsymbol{v} \in L^2(\Omega)^m \mid \boldsymbol{v}|_\kappa \in [\mathcal{P}^p(\kappa)]^m, \kappa \in \mathcal{T}_h\}, \tag{2}$$

where $\mathcal{P}^p(\kappa)$ is the space of polynomials of degree p on the element κ. At a given instant, we consider an element κ with boundary $\partial\kappa$, deforming in time with according to a velocity field $\boldsymbol{\nu} = \boldsymbol{\nu}(\boldsymbol{x},t)$. To obtain a discontinuous Galerkin formulation, we consider the following variational statement derived from equation (1) over a time changing element $\kappa(t)$: find $\boldsymbol{u}_h \in \mathcal{V}_h^p(\Omega)$ such that for each $\kappa \in \mathcal{T}_h$,

$$\int\limits_{\kappa(t)} \frac{\partial \boldsymbol{u}_h}{\partial t}\cdot\boldsymbol{v}\ d\boldsymbol{x} + \int\limits_{\kappa(t)} (\nabla\cdot\boldsymbol{F}(\boldsymbol{u}_h))\cdot\boldsymbol{v}\ d\boldsymbol{x} = 0, \tag{3}$$

for all test functions $\boldsymbol{v} \in \mathcal{V}_h^p$.

From the Reynolds transport theorem, we can write

$$\frac{d}{dt}\int\limits_{\kappa(t)} \boldsymbol{u}_h\cdot\boldsymbol{v}\ d\boldsymbol{x} = \int\limits_{\kappa(t)} \frac{\partial \boldsymbol{u}_h}{\partial t}\cdot\boldsymbol{v}\ d\boldsymbol{x} + \int\limits_{\kappa(t)} \boldsymbol{u}_h\cdot\frac{\partial \boldsymbol{v}}{\partial t}\ d\boldsymbol{x} + \oint\limits_{\partial\kappa(t)} \boldsymbol{u}_h\cdot\boldsymbol{v}\ \nu_n ds, \tag{4}$$

where $\nu_n = \boldsymbol{\nu}\cdot\boldsymbol{n}$ is the normal velocity of the element interface. Substituting (4) into (3) and integrating by parts, we obtain

$$\frac{d}{dt}\int_{\kappa(t)} \boldsymbol{u}_h \cdot \boldsymbol{v} \; d\boldsymbol{x} = \int_{\kappa(t)} \boldsymbol{u}_h \cdot \frac{\partial \boldsymbol{v}}{\partial t} \; d\boldsymbol{x} + \int_{\kappa(t)} \boldsymbol{F}(\boldsymbol{u}_h) : \nabla \boldsymbol{v} \; d\boldsymbol{x} - \oint_{\partial\kappa(t)} (\mathcal{F}_n(\boldsymbol{u}_h) - \boldsymbol{u}_h \nu_n) \cdot \boldsymbol{v} \; ds. \tag{5}$$

The discontinuous Galerkin formulation for a moving grid can now be expressed as follows: find $\boldsymbol{u}_h \in \mathcal{V}_h^p$ such that for each $\kappa \in \mathcal{T}_h$ and $\boldsymbol{v} \in \mathcal{V}_h^p$,

$$\frac{d}{dt}\int_{\kappa} \boldsymbol{u}_h \cdot \boldsymbol{v} \; d\boldsymbol{x} - \int_{\kappa} \boldsymbol{u}_h \cdot \frac{\partial \boldsymbol{v}}{\partial t} \; d\boldsymbol{x} - \int_{\kappa} \boldsymbol{F}(\boldsymbol{u}_h) : \nabla \boldsymbol{v} \; d\boldsymbol{x} + \oint_{\partial\kappa} \mathcal{F}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}, \nu_n) \cdot \boldsymbol{v} \; ds = 0, \tag{6}$$

where the numerical flux $\mathcal{F}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}, \nu_n)$ approximates $\boldsymbol{F}_n(\boldsymbol{u}) - \boldsymbol{u}\nu_n$ at interior element boundaries or domain boundaries with normal velocity ν_n. The $()^+$ and $()^-$ notion indicates the trace of the solution taken from the interior and exterior of the element, respectively, and $\boldsymbol{n}$ is the outward normal vector to the boundary of the element. Along the domain boundaries, the exterior state of the solution is constructed by weakly imposing the boundary conditions.

As the test functions $\boldsymbol{v}$ move with the grid velocity, their substantial derivatives vanish with the grid motion, i.e. $d\boldsymbol{v}/dt = \boldsymbol{0}$. Therefore, we have

$$\frac{\partial \boldsymbol{v}}{\partial t} = -\boldsymbol{\nu} \cdot \nabla \boldsymbol{v}. \tag{7}$$

Equation (6) can then be rewritten as: find $\boldsymbol{u}_h \in \mathcal{V}_h^p$ such that for each $\kappa \in \mathcal{T}_h$ and $\boldsymbol{v} \in \mathcal{V}_h^p$,

$$\frac{d}{dt}\int_{\kappa(t)} \boldsymbol{u}_h \cdot \boldsymbol{v} \; d\boldsymbol{x} = \int_{\kappa(t)} \big(\boldsymbol{F}(\boldsymbol{u}_h) - \boldsymbol{u}_h \boldsymbol{\nu}\big) : \nabla \boldsymbol{v} \; d\boldsymbol{x} - \oint_{\partial\kappa(t)} \mathcal{F}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}, \nu_n) \cdot \boldsymbol{v} \; ds. \tag{8}$$

We note that in the above expression, the original flux function is modified to reflect the movement of the grid. It can be seen that the DG front tracking formulation reduces to its standard DG form if the mesh is fixed ($\boldsymbol{\nu} = \boldsymbol{0}$).

The variational equation (8) is enforced separately in each element, and the coupling with the neighboring elements occurs via the numerical fluxes. The numerical fluxes and moving velocities along the tracked front are obtained by solving Riemann problems at the element interfaces.

2.2. Discontinuous Galerkin ALE Formulation the Compressible Navier-Stokes Equations

Here, we want to augment the original system of first order conservation laws (1) to include viscous effects. To this end, we write the Navier-Stokes equations

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \boldsymbol{F}^{\text{inv}}(\boldsymbol{u}) &= \nabla \cdot \boldsymbol{F}^{\text{vis}}(\boldsymbol{u}, \boldsymbol{q}) \\ \boldsymbol{q} - \nabla \boldsymbol{u} &= \boldsymbol{0} \end{aligned} \tag{9}$$

over the domain Ω with suitable boundary and initial conditions. Here, $\boldsymbol{u}$ is the conservative state vector which has density, momentum and total energy as components, $\boldsymbol{F}^{\text{inv}}(\boldsymbol{u})$ are the inviscid fluxes and $\boldsymbol{F}_i^{\text{vis}}(\boldsymbol{u}, \boldsymbol{q})$ denote the viscous fluxes. Note that as is customary in many DG formulations for elliptic problems (e.g. [3]), we have introduced the velocity gradient $\boldsymbol{q}$ as a new independent variable and thus cast the Navier-Stokes equations as a system involving only first order derivatives.

The discontinuous Galerkin formulation for the compressible Navier-Stokes equations (9) on a moving grid becomes: find $\boldsymbol{u}_h \in \mathcal{V}_h^p$ and $\boldsymbol{q}_h \in (\mathcal{V}_h^p)^d$ such that for each element $\kappa \in \mathcal{T}_h$

$$\begin{aligned} \frac{d}{dt} \int_{\kappa(t)} \boldsymbol{u}_h \cdot \boldsymbol{v} \; d\boldsymbol{x} &= \int_{\kappa(t)} \left(\boldsymbol{F}^{\text{inv}}(\boldsymbol{u}_h) - \boldsymbol{\nu} \boldsymbol{u}_h \right) : \nabla \boldsymbol{v} \; d\boldsymbol{x} - \oint_{\partial\kappa(t)} \mathcal{F}^{\text{inv}}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}, \nu_n) \cdot \boldsymbol{v} \; ds \\ &\quad - \int_{\kappa(t)} \boldsymbol{F}^{\text{vis}}(\boldsymbol{u}_h) : \nabla \boldsymbol{v} d\boldsymbol{x} + \oint_{\partial\kappa} \mathcal{F}^{\text{vis}}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{q}_h^+, \boldsymbol{q}_h^-, \boldsymbol{n}) \cdot \boldsymbol{v} \; ds, \\ \int_{\kappa(t)} \boldsymbol{q}_{\boldsymbol{h}} : \boldsymbol{p} \; d\boldsymbol{x} &= - \int_{\kappa(t)} \boldsymbol{u}_h \cdot (\nabla \cdot \boldsymbol{p}) \; d\boldsymbol{x} + \oint_{\partial\kappa} \mathcal{U}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}) : \boldsymbol{p} \; ds \end{aligned} \tag{10}$$

for all test functions $\boldsymbol{v} \in \mathcal{V}_h^p$ and $\boldsymbol{p} \in (\mathcal{V}_h^p)^d$. In the above equations, the inviscid numerical flux, $\mathcal{F}^{\text{inv}}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n}, \nu_n)$ is computed using the Roe or the Lax-Friedrich formula, except for the elements along the tracked front where the Riemann problem is solved to obtain the flux across the interface and the propagation speed of the front. The viscous numericalfluxes, $\mathcal{F}^{\text{vis}}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{q}_h^+, \boldsymbol{q}_h^-, \boldsymbol{n})$ and the numerical flux $\mathcal{U}(\boldsymbol{u}_h^+, \boldsymbol{u}_h^-, \boldsymbol{n})$ are defined according to the LDG scheme [3]. Numerical quadratures [36] are used to evaluate the volume and surface integrals. Finally, we note that, by proper choice of the numerical fluxes it is possible to eliminate $\boldsymbol{q}_h$ the discretized form of the above equations and hence cast the system as a set of coupled ODE's for the degrees of freedom associated to $\boldsymbol{u_h}$. These system of ODE's is then integrated using a Runge-Kutta method.

2.3. The Geometric Conservation Law

In simulations of flow problems involving moving boundaries, it is important to assure that a numerical scheme exactly reproduces a constant solution. This preservation of constant solution is referred to as the Geometric Conservation Law [39], which simply states that a solution of a uniform flow under the numerical discretization scheme should be preserved exactly for an arbitrary mesh motion. Mathematically, it must be shown that the ALE formulation (8) and (10) satisfy the uniform flow exactly. However, since for a uniform flow the viscous fluxes vanish, we need to consider only the inviscid of equation (8). Inserting a constant solution, $\boldsymbol{u}(x,t) = \boldsymbol{u}_0$, into (8) and using the consistency property of the numerical fluxes,

$$\mathcal{F}(\boldsymbol{u}_0, \boldsymbol{u}_0, \boldsymbol{n}, \nu_n) = (\boldsymbol{F}(\boldsymbol{u}_0) - \boldsymbol{u}_0\boldsymbol{\nu}) \cdot \boldsymbol{n}, \tag{11}$$

we obtain the following expression after rearrangement

$$\begin{aligned} \boldsymbol{u_0} \cdot \frac{d}{dt} \int\limits_{\kappa(t)} \boldsymbol{v} d\boldsymbol{x} = \boldsymbol{F}(\boldsymbol{u}_0) : &\left(\int\limits_{\kappa(t)} \nabla \boldsymbol{v} \; d\boldsymbol{x} - \oint\limits_{\partial\kappa(t)} \boldsymbol{v}\boldsymbol{\nu} \cdot \boldsymbol{n} ds \right) \\ &+ \boldsymbol{u}_0 \cdot \left(\oint\limits_{\partial\kappa(t)} \boldsymbol{v}\boldsymbol{\nu} \cdot \boldsymbol{n} ds - \int\limits_{\kappa(t)} \nabla \boldsymbol{v} \cdot \boldsymbol{\nu} d\boldsymbol{x} \right). \end{aligned} \tag{12}$$

Applying the divergence theorem, the integrals associated with the flux function vanish:

$$\frac{d}{dt} \int\limits_{\kappa(t)} \boldsymbol{v} d\boldsymbol{x} = \oint\limits_{\partial\kappa(t)} \boldsymbol{v}\boldsymbol{\nu} \cdot \boldsymbol{n} ds - \int\limits_{\kappa(t)} \nabla \boldsymbol{v} \cdot \boldsymbol{\nu} d\boldsymbol{x}. \tag{13}$$

The time derivative of the integral on the left can be further expanded as

$$\frac{d}{dt} \int_{\kappa(t)} \boldsymbol{v} d\boldsymbol{x} = \int_{\kappa(t)} \frac{\partial \boldsymbol{v}}{\partial t} d\boldsymbol{x} + \oint_{\partial\kappa(t)} \boldsymbol{v}\boldsymbol{\nu} \cdot \boldsymbol{n} ds, \tag{14}$$

and substituting (14) into (13), we have

$$\int_{\kappa(t)} \left(\frac{\partial \boldsymbol{v}}{\partial t} + \nabla \boldsymbol{v} \cdot \boldsymbol{\nu} \right) d\boldsymbol{x} = 0. \tag{15}$$

As expected, this equation is always satisfied in the continuum case due to the fact that the basis functions move with the grid velocity as stated in (7). However, in the discrete case, some small errors can be introduced due to inexact integration. As shown in the numerical examples, these errors are very small. If necessary, it is actually possible to correct for this errors as described in [28], at the expense of introducing an additional equation.

2.4. The Interface Tracking Technique

The above expressions (8) and (10) define an algorithm to advance the numerical solution $\boldsymbol{u}_h$ provided the grid velocity $\boldsymbol{\nu}$ is known. In some situations however, we are interested in interfaces which deform according to the solution velocity field.

2.4.1. Interface Representation

The interface is approximated by a collection of element boundary edges. We use an isoparametric mapping with nodal shape functions to map the the reference triangle into the actual element [42] . Therefore, the shape of the actual elements and the interface is determined by the node positions. The use of higher order polynomials as shown in Figure 2 for $p = 3$ leads to curved approximations of the interface.

Once the interface velocity is known, the position of the nodes which define the interface can be obtained by solving an ordinary differential equation in time,

$$\frac{d\boldsymbol{X}_\psi^i}{dt} = \boldsymbol{\nu}^i \ , \qquad \text{for} \quad i = 1, \ldots, N_\psi \tag{16}$$

where $\boldsymbol{X}_\psi^i$ and $\boldsymbol{\nu}^i$ for $i = 1, \ldots, N_\psi$ are the positions and velocities, respectively, of the nodes on the interface and N_ψ is the number of mesh nodes on the interface. The above equation is integrated using the same Runge-Kutta time stepping employed for governing equations (10).

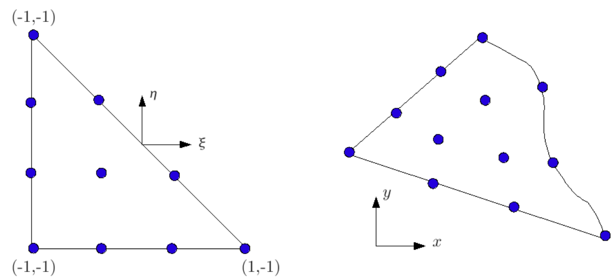


Figure 2: Isoparametric mapping from reference element

2.4.2. Interface Propagation Velocity

Except for the problems such as prescribed convection where the grid velocity is known beforehand, we determine the gird velocity at the interface by solving a Riemann problem in the normal direction to the interface.

The moving velocity must satisfy the Rankine-Hugoniot condition for the jump condition between the left (L) and right (R) states at a node which lies on the interface. The jump condition can be written in the following form

$$[\boldsymbol{F}(\boldsymbol{u}) \cdot \boldsymbol{n} - \nu_n \boldsymbol{u}]_L^R = 0, \tag{17}$$

where ν_n is the normal component of the interface velocity. This can be solved for $\boldsymbol{\nu}_n$ with the observation that the pressure and the normal velocity across the interface are constant [40]. Once $\boldsymbol{\nu}_n$ is determined, the interface velocity is set to be normal to the interface.

In the case of a nodal points located at the element vertices, the velocity at the node is double-valued. In this case, the interface velocity is simply constructed from the neighboring normal velocities ($\boldsymbol{\nu}_{n_1}$ and $\boldsymbol{\nu}_{n_2}$) so that its projections on the normal directions of the neighboring edges ($\boldsymbol{n}_1$ and $\boldsymbol{n}_2$) are preserved as depicted in Figure 3.That is,

$$\boldsymbol{\nu}_i \cdot \boldsymbol{n}_1 = \nu_{n_1} \tag{18}$$

$$\boldsymbol{\nu}_i \cdot \boldsymbol{n}_2 = \nu_{n_2}. \tag{19}$$

2.5. Automatic Mesh Regeneration

Once the velocity of the nodes at the interface is determined, we proceed to determining the velocity of the remaining nodes in the mesh with the objective of preserving a good mesh quality. At each timestep, we examine the

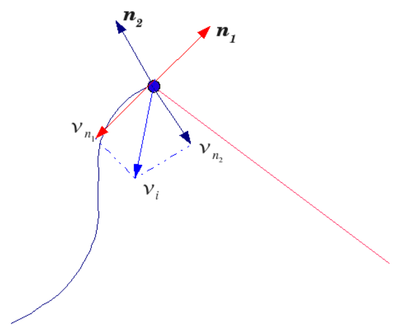


Figure 3: Velocity construction at element vertices where the interface velocity is projected from neigboring normal velocities.

mesh and perform the necessary mesh modifications such that the quality of the grid is preserved. A number of element shape parameters have been proposed for assessing the quality of a mesh [7]. For two-dimensional triangulations, a commonly used quantity which we have found to work well is the ratio of the inradius, r, to the circumradius, R, of the triangle,

$$q(\kappa) = \frac{2r}{R}. \tag{20}$$

This quantity has been shown to be a good measurement of the quality of element shapes.

In this work, the distance function mesh generation technique proposed in [27] is used for the mesh improvement. The inputs to the generator are the signed distance function $d(\boldsymbol{X})$ of the boundary and the mesh size function $h(\boldsymbol{X})$ giving the desired size of the elements. Once the motion of the nodes of the interface has been determined, the motion of the remaining nodes is determined by solving a force equilibrium system at the nodes. The force acting on an edge is proportional to the difference between the actual length l of the edge and its desired length l_0 which is set by the mesh size function $h(\boldsymbol{X})$ evaluated at the mid point of the edge. There are several alternatives for the force function $f(l, l_0)$ acting on each edge. In this work, a model of linear spring is used to describe the force function, acting as the repulsive forces. That is,

$$f(l, l_0) = \begin{cases} k(l_0 - l) & \text{if } l < l_0, \\ 0 & \text{if } l \geq l_0. \end{cases} \tag{21}$$

To solve for the force equilibrium, the forces at all the nodes are added to get $F(\boldsymbol{X}_{\text{nod}})$ and obtain a nonlinear system of equations $F(\boldsymbol{X}_{\text{nod}}) = \boldsymbol{0}$ for the node positions, $\boldsymbol{X}_{\text{nod}}$. A stationary solution of the system of ODEs

$$\frac{d\boldsymbol{X}_{\text{nod}}}{dt^*} = F(\boldsymbol{X}_{\text{nod}}), \quad t^* \geq 0 \tag{22}$$

is found using the forward Euler method. After each time step, any point that moved outside of the geometry is projected back to the boundary by a reaction force applied normal to the boundary.

During the mesh deformation iteration, we examine the mesh and if necessary perform some topological changes. In the original mesh generator [27] this was done using Delaunay triangulations, but this can be rather complicated and inefficient for moving interfaces. Instead local operations consisting of edge flipping, node addition and deletion are implemented to improve the mesh quality. The mesh modification procedures continue until all the elements satisfy a preset threshold for the mesh quality. We note that for most timesteps, the mesh modification process is very inexpensive since no topological changes take place and only a few relaxation iterations ($\sim 1-2$) are required to solve for the interior node positions (22).

2.5.1. Edge Flipping

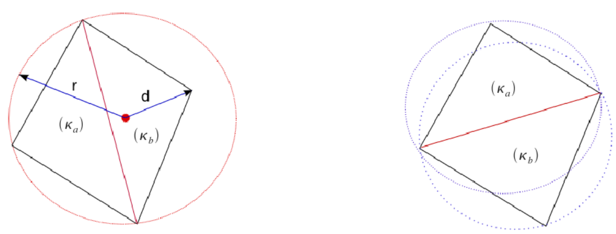


Figure 4: Edge flipping. Left: initial triangles κ_a and κ_b with the circumradius r of the triangle κ_a. Right: after flipping

The criterion used for edge flipping is that the circumcircle of any triangle should not contain any other triangles in the mesh, and if it does, the shared edge between two triangles is flipped and the velocity field is updated correspondingly. An example is shown in Figure (4), where the third node of triangle κ_b is inside the circumcircle of triangle κ_a. This is handled by flipping the shared edge between the two triangles and updating the velocity field correspondingly. In the rare situations where the edge to be flipped is an interface edge, then this operation is not performed.

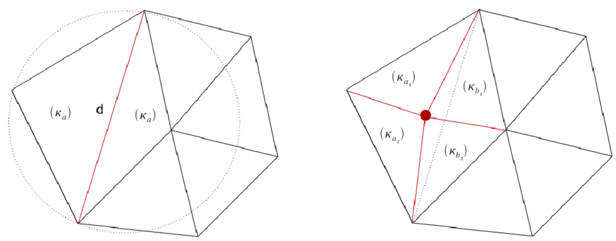


Figure 5: Node addition. Left: initial grid with a large edge between κ_a and κ_b. Right: After splitting, the long edge is split and the associated new elements (κ_{a1}, κ_{a2}, κ_{b1} and κ_{b2}) are formed.

2.5.2. Node addition

It is sometimes necessary to add points to the mesh. If any edge is too long compared to the desired value based on the distance function evaluated at the midpoint, then the midpoint is inserted as a new mesh point and the element is split into two elements as shown in Figure 5.

2.5.3. Node deletion

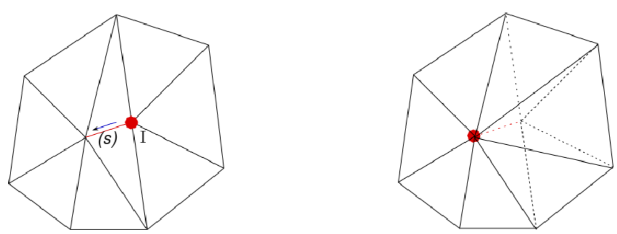


Figure 6: Node deletion. Left: Initial mesh with short edge (s). Right: Deleting of intermediate node (I) and reconnecting associated edges.

Conversely, if an edge is too short compared to its expected value, a node is removed from the grid and the edge is collapsed as shown in Figure 6.

2.6. Solution Updating

When the mesh connectivities are changed due to the mesh improvement operations described above, the solution has to be reconstructed on the modified mesh such that the conservation of the solution is maintained, that is

$$\sum_{\kappa'_i \in \mathcal{T}'_h} \int_{\kappa'_i} \boldsymbol{u}'_h \, d\boldsymbol{x} = \sum_{\kappa_i \in \mathcal{T}_h} \int_{\kappa_i} \boldsymbol{u}_h \, d\boldsymbol{x}, \tag{23}$$

where $\mathcal{T}_h$ is the original triangulation and $\mathcal{T'}_h$ is the triangulation after the mesh modifications. Similarly, $\boldsymbol{u}_h$ and $\boldsymbol{u}'_h$, represent the solutions on $\mathcal{T}_h$ and $\mathcal{T}'_h$, respectively. We obtain an approximation to (23) by interpolating the values of $\boldsymbol{u}_h$ on the nodes of the elements of $\mathcal{T}_h^{'}$ and performing the following least squares projection: find $\boldsymbol{u}'_h \in \mathcal{V}'^p_h$ such that for all $\boldsymbol{v}' \in \mathcal{V}'^p_h$

$$\int\limits_{\kappa'} \boldsymbol{u}'_h \boldsymbol{v}' \; d\boldsymbol{x} = \int\limits_{\kappa'} \boldsymbol{u}_h \boldsymbol{v}' \; d\boldsymbol{x}, \qquad \text{for all} \qquad \kappa' \in \mathcal{T}'_h. \tag{24}$$

Here, $\boldsymbol{v}' \in \mathcal{V}'^p_h$ is the DG space associated with the modified triangulation $\mathcal{T}'_h$. This interpolation procedure followed by the projection is very efficient but may introduce some conservation errors. Our numerical experiments indicate that these errors can be made very small when the solution is sufficiently resolved.

3. NUMERICAL RESULTS

3.1. Front tracking for scalar problems in two dimensions

Example 3.1. *Linear Convection*

This example considers the scalar convection problem in two space dimensions proposed by Zalesak in [41] and is a standard test for front tracking and capturing schemes. Consider the convection equation

$$\frac{\partial \Phi}{\partial t} + \boldsymbol{U} \nabla \Phi = 0 \tag{25}$$

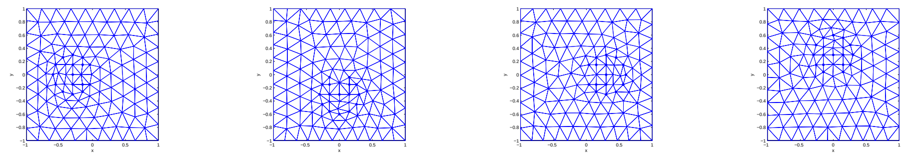
in the domain $(x, y) \in [-1, 1] \times [-1, 1]$. A slotted circle C of radius $r = 0.3$ is centered at $(0.0, 0.0)$. The width of the slot is 0.15 and the height of the slot is 0.15. The velocity field is $\boldsymbol{U} = (u, v)^T$ is given by

$$\begin{aligned} u &= \frac{\pi}{3.14}(-y) \\ v &= \frac{\pi}{3.14}x \end{aligned}$$

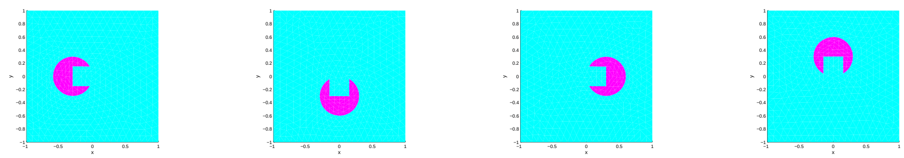
and the initial condition is given by the indicator function

$$\Phi_0(x, y) = \begin{cases} 1.0 & \text{if } (x, y) \in C \\ 0.0 & \text{otherwise.} \end{cases} \tag{26}$$

The slotted circle will rotate about $(0.0, 0.0)$ with velocity $\boldsymbol{U}$. In this case we want to refine the mesh near the interface and thus the mesh size



(a) Grid at $t = \pi/2$ (b) $t = \pi$ (c) $t = 3\pi/2$ (d) $t = 2\pi$



(e) Solution at $t = \pi/2$ (f) $t = \pi$ (g) $t = 3\pi/2$ (h) $t = 2\pi$

Figure 7: Example 3.1: Zalesak problem. Grids and solutions over one period using the cubic interpolations and a fourth order Runke-Kutta method.

function $h(\boldsymbol{X})$ is specified as $h(\boldsymbol{X}) = \min(1 + 1.5|\psi(\boldsymbol{X})|, 1.5)$, where $\psi(\boldsymbol{X})$ is the distance function of the interface. The grids and the solutions are shown over one period of evolution of the slotted circle using the cubic polynomials to represent the spatial variation of the solution and a fourth order Runge-Kutta scheme to perform the time integration. The difficulty of this problem is to accurately follow the interface as it rotates around the center. It can be seen from the presented result that the interface is tracked very accurately. Compared to results using the level set method, e.g. in [37], our high order method performs much better in terms of tracking the interface and maintaining the conservation of the solution.

3.2. The Front Tracking Method for Flows with Surface Tension

In some problems involving fluids of different properties densities and viscosities, the damping effect of surface tension becomes more important than that of the viscosity. The presence of surface tension results in an unbalanced force acting on the interface. The inadequate treatment of these forces can affect the accuracy of the tracking scheme and cause spurious currents around the interface region.

For the tracking of interfaces in the presence of surface tension, Riemann problems at the interface have to be solved to compute the numerical flux and the interface motion taking into account the added forces due to surface tension [40]. Some examples of interface tracking with surface tension are presented below, including drop deformations and bubble oscillations under acoustic waves.

3.2.1. Surface Tension and Curvature

The introduction of surface tension requires the calculation of the surface tension force which is a function of the interface curvature. The surface tension force is considered as a distributed external force applied at the interface. The jump in pressure due to the surface tension must be satisfied across the interface, by incorporating it into the Riemann solver together with the Rankine-Hugoniot condition in order to solve for the flux across the interface and the moving velocity of the interface. Since the interface is defined by piecewise polynomial segments which are only continuous and typically have small discontinuities in the derivatives, we have found it necessary to interpolate a smooth function across the interface nodes in order to obtain accurate surface tension approximations. We use a B-spline interpolation [29] through the interface points. The curvature at a particular point on the interface is computed by projecting that point to the B-spline curve and then directly evaluating the curvature at that point on the B-spline.

3.2.2. Applications

Example 3.2. *Flow under surface tension*

We consider the flow in the square domain of $[-1, 1] \times [-1, 1]$. A circle-shaped membrane of zero thickness with a radius of $R = 0.3$ centered at the origin is embedded in the flow, separating the bubble from the surrounding fluid. The flow is governed by compressible Navier-Stokes equations charecteried by the non-dimensional Reynolds number $Re = \frac{\bar{u}R}{\mu}$ where $\bar{u}$ is the characteristic velocity and mu is the fluid viscosity. In this case, the fluids inside and outside the bubble are of the same type. The effect of the membrane is modeled by the surface tension σ giving rise to the surface tension force acting on the flow,

$$\boldsymbol{f}(\boldsymbol{X}) = \kappa\sigma\boldsymbol{n}\delta(\boldsymbol{X} - \boldsymbol{X}_\psi), \tag{27}$$

where κ is the surface curvature, $\boldsymbol{n}$ is the interface normal vector and δ is two dimensional Kronecker delta function. The flow is initially at rest with unit pressure and density ($P_0 = \rho_0 = 1.0$). At steady state, the pressure jump ΔP across the interface is given as

$$\Delta P = \sigma\kappa = \sigma/R. \tag{28}$$

In this example, we want to verify that the above expression is satisfied across the interface by comparing the numerical result with the analytical expression.

Table 1: Pressure jump and spurious currents around the circular bubble, $Re = 100$, using cubic interpolations

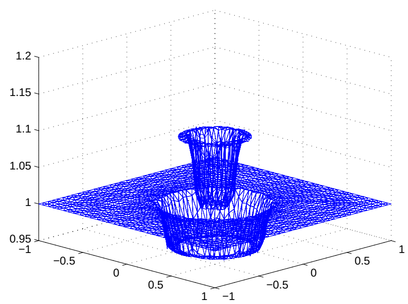
Grid size	No elements	ΔP	$\epsilon_{\Delta P}$
0.1	365	0.1531	0.0057
0.075	645	0.1492	0.0018
0.05	1413	0.1485	0.0011
0.035	2816	0.1478	0.0004

(a) Pressure jump error for $k = 1/Ca = 10$

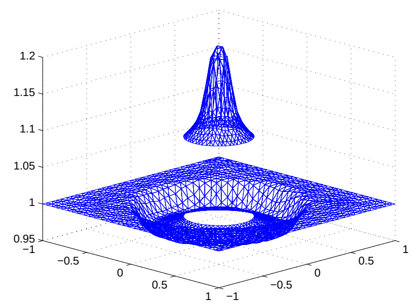
Grid size	$\lvert\boldsymbol{u}\rvert_{max}$	Ca	$\lvert\boldsymbol{u}\rvert_{max}$
0.1	2.8200E-04	1/50	7.7872E-004
0.075	2.5386E-04	1/20	6.3902E-004
0.05	2.3492E-04	1/10	2.5386E-004
0.035	2.1325E-04	1/5	1.3204E-004

(b) Spurious current

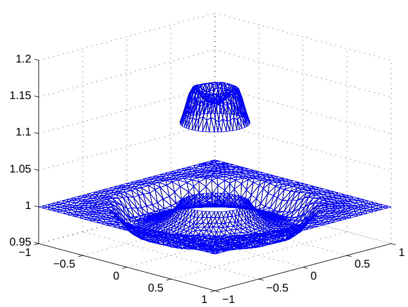
In Figure 8, the solutions for the pressure at different times are shown for $Re = 100$ on a grid with 365 elements using cubic polynomials to represent the solution inside each element at the capillary number of the flow $Ca = 1/10$ computed with respect to the speed of sound, c, as $Ca = c\mu/\sigma$. It can be observed that there is a sharp jump in pressure across the interface which can be tracked explicitly, and that the interface and the jump are sharply captured. In Table 2(a), the pressure jumps at steady state are computed on different grids and compared with the analytical results. The results show that the jump across the interface converges to the analytical value as the grid is refined. The convergence rate of the pressure jump, and the error, $\epsilon_{\Delta P}$, between the numerical result and the analytical value obtained from (28) is found to be more than second order with respect to the grid size (the exponent is about 2.35).



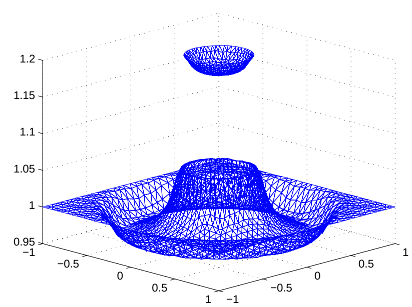
(a) Pressure, t=0.1



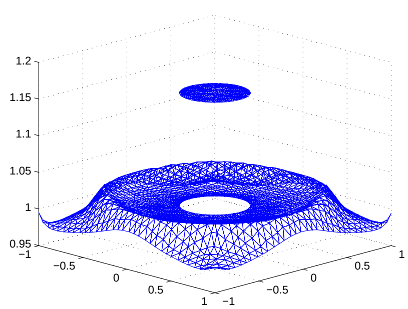
(b) Pressure, t=0.3



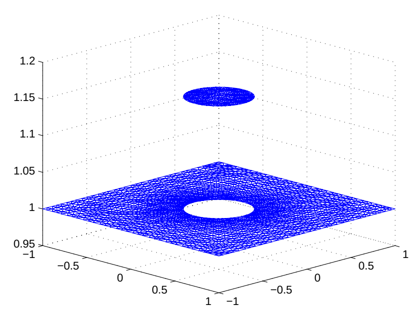
(c) Pressure, t=0.8



(d) Pressure, t=0.5



(e) Pressure, t=1.0



(f) Pressure, t=4.0

Figure 8: Example 3.2: Circular bubble under surface tension. Pressure field at different time steps with $Re = 100$, $k = 10$, using cubic polynomials on a grid of 347 elements

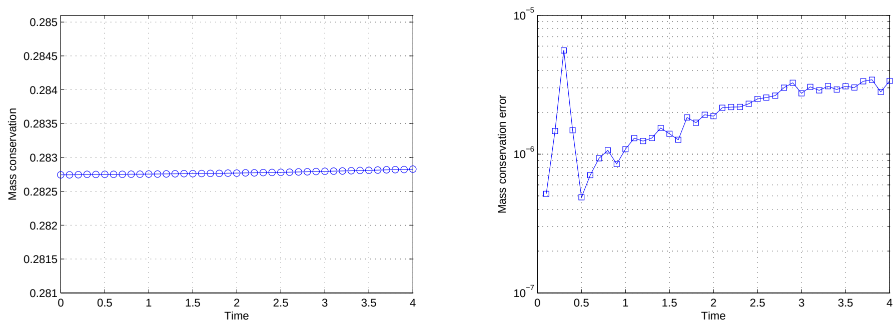
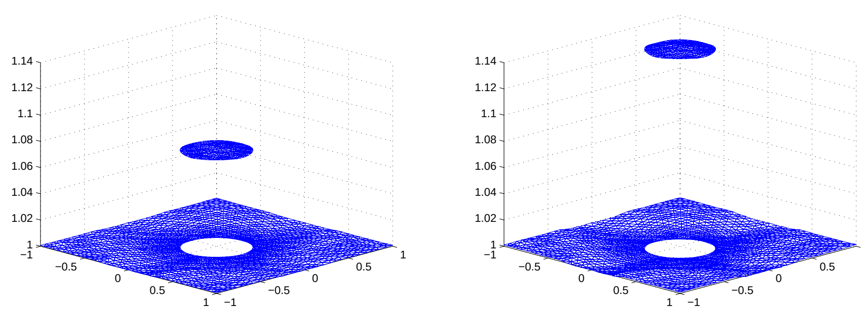


Figure 9: Example 3.2: Mass conservation inside the bubble under the effect of surface tension. Note at the small mass conservation error.



(a) $Re = 1000$, $k = 50$ (b) $Re = 1000$, $k = 100$

Figure 10: Example 3.2: Circular bubble under surface tension. Steady solution of pressure field with different values of capillary number, cubic elements on the grid of 347 elements

It is also interesting to study the velocity field around the bubble, which is expected to vanish at steady state. However, as the bubble is relaxed to a circular stable shape there is still a small amplitude velocity field around the interface, called a spurious current, due to the imbalance between the stresses at the interface. It was shown in [32] that this spurious current scales with the surface tension and the viscosity as $|\boldsymbol{u}|_{max} = C\sigma/\mu$ where C is a constant. This is equivalent to having a constant value of $|\boldsymbol{u}|_{max}\mu/\sigma$. The spurious current was computed as the norm of the velocity field at steady state and is shown in Table 2(b) for various grid sizes and different capillary numbers. From Table 2(b), it is observed that the spurious current is approximately constant on different grid sizes at a given viscosity and surface tension corresponding to the Reynolds number and the capillary number of $Re = 100$ and $k = 10$. The experiment was then repeated on a fixed grid of 645 elements and $Re = 100$ but with different values of capillary numbers. The spurious current is then found to scale with the inverse of the capillary number as predicted in [32].

The conservation of mass inside the circular bubble is also measured and shown in Figure 9 and observed to be very small. The steady state solution for pressure obtained for two values of the capillary number and shown in Figure 10.

Example 3.3. ***Oscillation of a drop***

In the next example, we study the oscillation of a drop under surface tension. This problem has been studied extensively before, and Rayleigh [31] derived the formulation for the oscillation of cylindrical jets under capillary force. Under a small perturbation in the plane perpendicular to the axis of the cylindrical droplet, the frequency ω_n of the oscillation at a particular mode l depends on the surface tension σ, the density ρ, and the unperturbed radius of the drop R_0 as follows,

$$\omega_n^2 = (l^3 - l)\frac{\sigma}{\rho R_0^3}, \tag{29}$$

where the surface of the drop is given in polar coordinates by

$$r = R_0 + \epsilon_R \cos(l\theta). \tag{30}$$

The oscillation period is then computed as $T = 2\pi/\omega$. In the first mode ($l = 1$) the drop is moving rigidly and there is no deformation in the drop shape. For the second mode ($l = 2$), the drop has the shape of an ellipsoid in which the major axis alternates between the horizontal and the vertical axis. For $l = 3$ and $l = 4$, the drop has triangular and square shapes with rounded corners. Fritts *et al* [8] extended the Rayleigh theory to apply to the oscillation of a drop in an external fluid, giving a frequency of

$$\omega_n^2 = (l^3 - l)\frac{\sigma}{(\rho_d + \rho_o)R_0^3}, \tag{31}$$

where ρ_d and ρ_o are the density inside and outside of the drop, respectively.

We consider a deformable bubble which has an initial ellipsoidal shape with major axis $a = 0.45$ and minor axis $b = 0.3$. The computational domain is $[-1, 1] \times [-1, 1]$. The ellipsoidal bubble of density $\rho_d = 1.0$ and pressure $P_d = 1.0$ is surrounded by the fluid of the same density $\rho_0 = 1.0$ and pressure $P_0 = 1.0$. The flow is initially at rest with $u_0 = v_0 = 0.0$. Under the effect of surface tension and viscosity the bubble oscillates and finally converge to a circular shape.

The profile of the pressure is presented in Figure 11 at various times on a grid with 323 elements. It can be observed that the sharp jump due to surface tension is well resolved. In Figure 12b, the shape of the bubble is shown at different times for Reynolds number $Re = 50$ and capillary number $k = 5$. The interface is initially at rest in the ellipsoidal shape with zero kinetic energy. Under the effect of surface tension the bubble oscillates until the equilibrium is reached. The damping effect from the viscosity results in a decay of the oscillation amplitude.

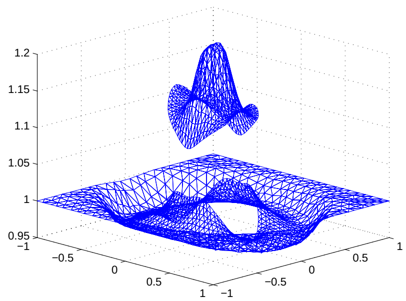
Figure 12a shows the evolution of the radius of the drop in the x and the y directions. Under the above described flow conditions the Rayleigh frequency and oscillation period can be calculated from (31) as

$$\omega_n = 1.5457, \qquad T_n = 2\pi/\omega_n = 4.0648. \tag{32}$$

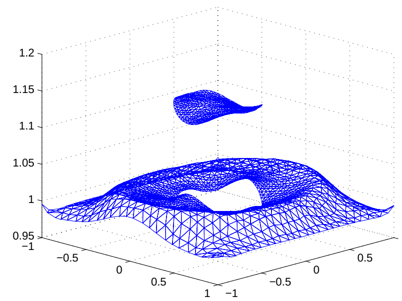
From the numerical result, it is found that the oscillation frequency is $\omega = 1.4710$ and the time period is $T = 4.2713$. This frequency has been obtained by fitting the response of a damped linear to the motion shown in Figure 12 therefore taking into account the effect of damping. The error in oscillation frequency is attributed to the effect of viscosity, which resulted in a damping coefficient of $\xi = 0.0325$ and to the finite amplitude of the oscillations. We note that Rayleigh theory is only applicable for small perturbation, namely linear behavior.

Example 3.4. ***Drop deformation and oscillation under acoustic wave***

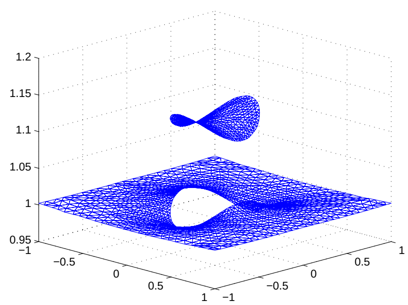
The deformation of immiscible drops in fluid flow is studied in this example, which has been previously explored both experimentally and numer-



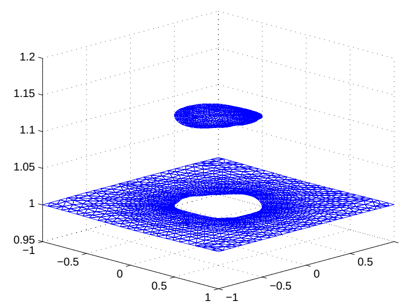
(a) Pressure, t=0.5



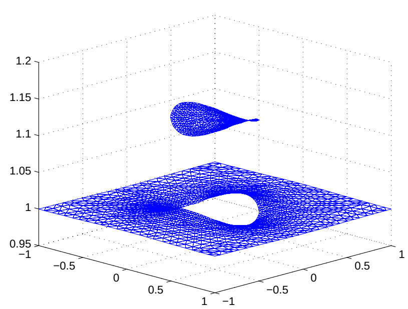
(b) Pressure, t=1.0



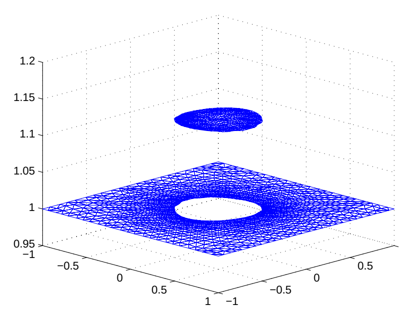
(c) Pressure, t=2.0



(d) Pressure, t=3.7

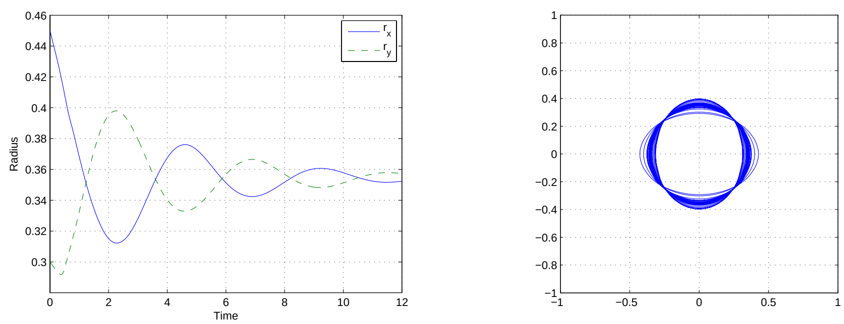


(e) Pressure, t=4.7



(f) Pressure, t=12.0

Figure 11: Example 3.3: Oscillation of ellipsoidal bubble under surface tension. Pressure field at different time steps with $Re = 50$, $k = 5$, using cubic polynomials on a grid of 323 elements.



(a) Radius in the x and the y directions

(b) Ellipse shapes

Figure 12: Example 3.3: Oscillation of an ellipsoidal bubble under surface tension. Deformation of the ellipse membrane under surface tension, $Re = 50$, $k = 5$, DGMP3 on the grid of 323 elements.

ically [35, 21]. A suspended drop containing a fluid of different density and possibly different viscosity is subjected to an acoustic wave. Surface tension is applied at the interface separating the drop from the outside fluid. When the frequency of the acoustic wave matches the natural frequency of the system (31), which is dependent on the tension parameter, the drop enters into resonance.

We considering a drop of undeformed radius R_0 placed at the center of the domain. The drop, having the values of density ρ_d, viscosity μ_d, and the surface tension σ, is suspended in the flow of density ρ_o and viscosity μ_o. The surface tension force acting on the fluid is given by (27). The acoustic wave travels from the left to the right boundaries. Rigid boundary conditions are applied at the top and the bottom boundaries. Outflow boundary conditions are specified at the right boundary. On the left boundary, the acoustic wave conditions of pressure, density, and velocity are specified as follows,

$$\begin{gathered} P(t) = P_0(1 + \bar{P}cos(\omega t)) \\ \rho(t) = \rho_0\left(P(t)/P_0\right)^{1/\gamma} \\ u(t) = u_0 + \tfrac{2}{\gamma-1}(c(t) - c_0) \\ v(t) = 0 \end{gathered} \tag{33}$$

where $\bar{P}$ is the amplitude of the wave and ω is the wave frequency. P_0, ρ_0 and c_0 are the pressure, density, and sound speed of the flow outside the drop.

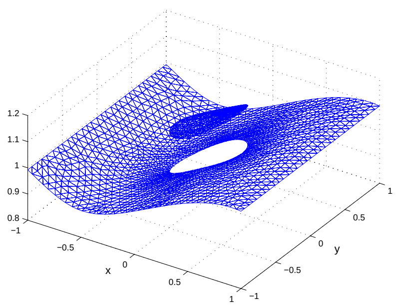
The acoustic impact is characterized by the Strouhal number ($St = \frac{\omega R}{\bar{u}}$). In this example, we set $P_0 = \rho_0 = 1.0$ and the fluid inside and outside of the drop are taken to be the same with $\mu_0 = \mu_d$.

Figure 13 shows a series of pressure distributions at different times with $Re = 100$, $k = 10$ and $St = 0.25\pi$ using cubic elements. The flow is initially at rest ($u_0 = v_0 = 0.0$). The simulation is done on a grid of 347 elements. It can be seen that the pressure jump across the interface is very well resolved while the front is explicitly tracked. The deformed drop shape is quantified and measured by the radii in the x and the y directions, r_x and r_y. In Figure 14 the radius of the drop is plotted with time at the different values of Strouhal number $St = 0.15\pi$ and 0.5π. It is known that the pulsation of a deformed drop (ρ_d) surrounded by another fluid (ρ_0) is given by the Rayleigh formula (31). In this case the second mode of oscillation ($l = 2$) is again considered. By varying the frequency of the acoustic wave, as measured by the Strouhal number, the drop oscillates at different frequencies. There is a phase shift in the oscillation between r_x and r_y as shown in Figure 14b which results from the combination of different oscillation modes. The maximum and minimum values of the radii are shown in Figure 15 together with the amplitude of the oscillation. The Rayleigh frequency computed from (31) is $\omega_n = 2.1091$ under the current test case condition. In our simulation, the maximum amplitude of oscillation is obtained at $St = 0.16\pi$ corresponding to the acoustic wave frequency $\omega = 1.9825$. The computational result is close to the analytical solution, with a discrepancy of about 6% for the resonance frequency, which once again is attributed to the fact that our system is both damped and the drop oscillations have finite amplitude.

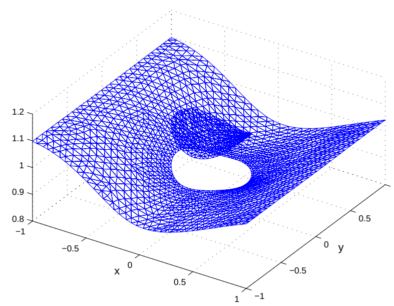
4. CONCLUSIONS

A discontinuous Galerkin front tracking scheme has been presented. The material interface is tracked explicitly. Although not strictly conservative, the method is found to be accurate and the mass conservation errors are found to be very small. The front propagation speed is determined by solving Riemann problems at the element interface. The interface is represented by a collection of edges which are element boundaries and therefore is approximated by high order polynomials. To maintain the quality of the grid during the propagation of the interface, the grid is optimized at every time step.

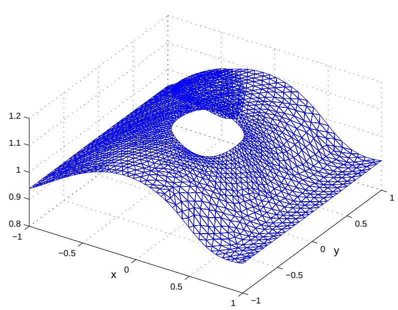
To handle interfaces in flows with surface tension, the jump in the solution due to the surface tension is incorporated into the Riemann solver. In order to compute the interface curvature required to evaluate the surface



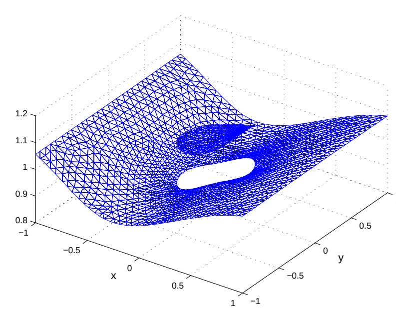
(a) Pressure, t=2.0



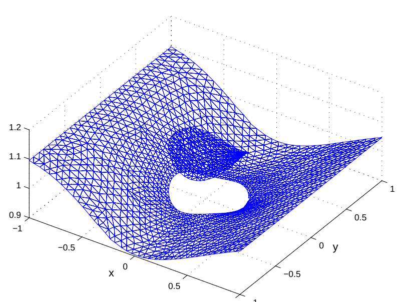
(b) Pressure, t=2.5



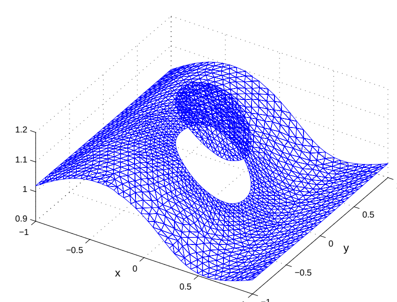
(c) Pressure, t=3.25



(d) Pressure, t=4.25

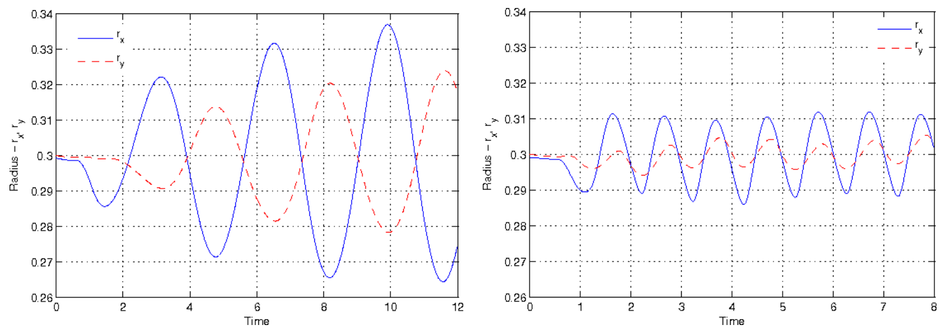


(e) Pressure, t=4.5



(f) Pressure, t=5.0

Figure 13: Example 3.4: Drop deformation and oscillation under acoustic wave. Pressure field at different time steps with $Re = 100$, $k = 10$, $St = 0.25\pi$ using cubic interpolations on the grid of 347 elements.



(a) $Re = 100,\ k = 10,\ St = 0.15\pi$ (b) $Re = 100,\ k = 10,\ St = 0.5\pi$

Figure 14: Example 3.4: Drop deformation and oscillation under acoustic wave. Evolution of the drop shape with different Strouhal number, $Re = 100$, $k = 10$ and cubic elements

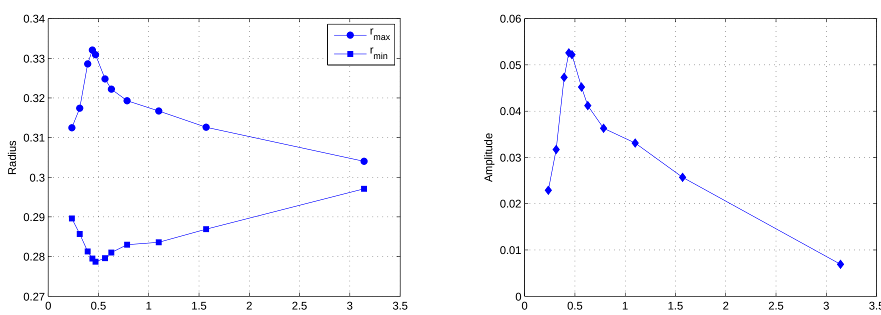


Figure 15: Example 3.4: Drop deformation and oscillation under acoustic wave. Radius and amplitude of the drop at different Strouhal number, $Re = 100$, $k = 10$, and cubic elements

tension, a smooth representation of the interface is obtained using B-splines. Results of various compressible Navier-Stokes flows with surface tension have been shown, including the oscillation of a drop with and without the presence of an externally imposed acoustic wave. The numerical results show a rather close agreement with the analytical results based on the inviscid linearized theory of Rayleigh. Overall, the proposed discontinuous Galerkin front tracking method is deemed robust and able to deal with material interfaces involving surface tension and general geometries.

5. Acknowledgement

We would like to thank the Singapore-MIT Alliance for the support of this work.

References

[1] Arnold DN, Brezzi F, Cockburn B and Marini LD. Unified analysis of discontinuous Galerkin methods for elliptic problems. SIAM Journal on Numerical Analysis 2002;39:1749-1779.

[2] Cockburn B and Shu C-W. Runge-Kutta Discontinuous Galerkin Methods for Covective-Dominated Problems. Review Article, J. Sci. Comp. 2001;16:173-261.

[3] Cockburn B and Shu C-W. The local discontinuous Galerkin method for time dependent convection-diffusion systems. SIAM J. Numer. Anal. 1998;35:2440-2463.

[4] Dubiner M. Spectral methods on triangles and other domains. J. Sci. Comput. 1991;6:345-357.

[5] Fedkiw R, Aslam T, Merriman B, and Osher S. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). J. Comput. Phys. 1999;152:457-492.

[6] Taylor MA, Wingate BA, and Vincent RE. An algorithm for computing Fekete points in the triangle. SIAM J. Numer. Anal. 2000;38:1707-1720.

[7] Field DA. Qualitative measures for initial meshes. Int. J. Numer. Meth. Engng. 2000;47:887-906.

[8] Fyfe DE, Oran ES, and Fritts MJ. Surface tension and viscosity with Lagrangian hydrodynamics on a triangular mesh. J. Comput. Phys. 1998;76:349-384.

[9] Glimm J, Li XL, Liu Y, and Zhao N. Conservative front tracking and level set algorithms. Proceedings of the National Academy of Sciences (PNAS) 2001;98:14198-14201.

[10] Glimm J, Li XL, Liu Y, Xu Z, and Zhao N. Conservative front tracking with improved accuracy. SIAM J. Numer. Anal 2003;41(5):1926-1947.

[11] Glimm J, Marchesin D, and McBryan O. Subgrid resolution of fluid discontinuities II. J. Comput. Phys. 1980;37:336-354.

[12] Glimm J, Grove JW, Li X-L, Shyue K-M, Zeng Y, and Zhang Q. Three-dimensional front tracking. SIAM J. Sci. Comput. 1998;19:703-727.

[13] Gloth O, Hanel D, Tran L,Vilsmeier R. A front tracking method on unstructured grids, Computers and Fluids 2003;32:547-570.

[14] Gottlieb S, and Shu C-W. Total variation diminishing Runge-Kutta schemes. Math. Comp. 1998;67:73-85.

[15] Gottlied S, Shu C-W, and Tadmor E. Strong stablity preserving high order time discretization methods, SIAM review 2001;43(1):89-112.

[16] Goodman JB and LeVeque RJ. On the Accuracy of Stable Schemes for 2D Scalar Conservation Laws. Mathematics of Computation 1985; 45(171):15-21.

[17] Godunov SK. A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics. Mat. Sb. 1959;47:357-393.

[18] Harten A and Hyman JM. Self-adjusting grid methods for one-dimensional hyperbolic conservation laws. J. Comp. Phys. 1983;50:235-269.

[19] Hesthaven JS. From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex. SIAM J. Numer. Anal. 1998;35(2):655-676.

[20] Koornwinder TH. Askey-Wilson polynomials for root systems of type BC. Comptemp. Math. 1992;138:204-237.

[21] Li X and Sarkar K. Drop dynamics in an oscillating external flow at finite Reynolds numbers. Phys. Fluids 2005;17:27-103.

[22] Liu TG, Khoo BC and Yeo KS. Ghost fluid method for strong shock impacting on material interface. J. Comput. Phys. 2003;190:651-681.

[23] Liu J, Lim H-K, Glimm J and Li X. A conservatve front tracking method in N-dimensions. J. Sci. Comput. 2007;31:213-236

[24] Liu Y and Vinokur M, Exact integrations of polynomials and symmetric quadratures formulas over arbitrary polyhedra grids. J. Comput. Phys. 1998;140:122-147.

[25] Mao D-K. Toward front tracking based on conservation in two dimensional space. SIAM J. Sci. Comput. 2000;22(1):113-151.

[26] Perigaud S and Saurel R. A compressible flow model with capillary effects. J. Comput. Phys. 2005;209:139-178.

[27] Persson P-O, Strang G. A Simple Mesh Generator in MATLAB. SIAM Review 2004;46(2):329-345.

[28] Persson P-O, Bonet J and Peraire J. Discontinuous Galerkin Solution of the Navier-Stokes Equations on Deformable Domains. To appear in Comp. Meth. in Appl. Engngr.

[29] Piegl L and Tiller W. The NURBS Book, 2nd edition, Springer-Verlag 1997.

[30] Popinet S and Zaleski S. A front tracking algorithm for accurate representation of surface tension. Int. J. Numer. Meth. Fluids 1999;30:775-793.

[31] Rayleigh L. On capillary phenomina of jets. Proc. Royal Society London 1897;29:71-97.

[32] Lafaurie B, Nardone C, Scardovelli R, Zaleski S and Zanetti G. Modelling merging and fragmentation in multiphase flows with SURFER. J. Comput. Phys. 1994;113:134-147.

[33] Reed WH and Hill TR. Triangular mesh methods for the neutron transport equation. Tech. Report LA-UR-73-479, Los Alamos Scientific Laboratory 1973.

[34] Roe PL. Approximate Riemann solvers, parametter vectors and difference schemes. J. Comput. Phys. 1981;43:357-372.

[35] Sarkar K and Schowalter WR. Deformation of a two dimensional drop at non-zero Reynolds number in time-periodic extensional flow: numerical simulation. J. Fluid Mech. 1991;436:177-206.

[36] Solin P, Segeth K, and DoleZel I. Higher-Order Finite Element Methods. Chapman and Hall 2003.

[37] Sussman M, Fatemi E, Smereka P, and Osher S. An Improved Level Set Method for Incompressible Two-phase Flows. Computers and Fluids 1998;27(5-6):663-680.

[38] Tang HZ and Tang T. Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws. SIAM J. Numer. Anal 2003;41(2):487-515.

[39] Thomas PD and Lombard CK. Geometric conservation law and its application to flow computations on moving grids. AIAA J. 1979;17:1030-1037.

[40] Toro EF. Riemann Solvers and Numerical Methods for Fluid Dynamics. Springer-Verlag. Second Edition 1999, 624 pages.

[41] Zalesak ST. Fully multidimensional flux-corrected transport algorithms for fluids. J. Comput. Phys. 1979;31(3):335-362.

[42] Zienkiewicz OC and Taylor RL. The Finite Element Method: Vol. 3, Fluid Dynamics. Butterworth-Heinemann 2000, 352 pages.