

# A discontinuous Galerkin-based sharp-interface method to simulate three-dimensional compressible two-phase flow

S. Fechter<sup>\*,†</sup> and C.-D. Munz

*Institute of Aerodynamics and Gas Dynamics, University of Stuttgart, Pfaffenwaldring 21, D-70569 Stuttgart, Germany*

## SUMMARY

A numerical method for the simulation of compressible two-phase flows is presented in this paper. The sharp-interface approach consists of several components: a discontinuous Galerkin solver for compressible fluid flow, a level-set tracking algorithm to follow the movement of the interface and a coupling of both by a ghost-fluid approach with use of a local Riemann solver at the interface. There are several novel techniques used: the discontinuous Galerkin scheme allows locally a subcell resolution to enhance the interface resolution and an interior finite volume Total Variation Diminishing (TVD) approximation at the interface. The level-set equation is solved by the same discontinuous Galerkin scheme. To obtain a very good approximation of the interface curvature, the accuracy of the level-set field is improved and smoothed by an additional  $P_N P_M$ -reconstruction. The capabilities of the method for the simulation of compressible two-phase flow are demonstrated for a droplet at equilibrium, an oscillating ellipsoidal droplet, and a shock-droplet interaction problem at Mach 3. Copyright © 2015 John Wiley & Sons, Ltd.

Received 25 August 2014; Revised 15 January 2015; Accepted 15 February 2015

**KEY WORDS:** DNS; discontinuous Galerkin; compressible multiphase flow; sharp-interface resolution; ghost-fluid method

## 1. INTRODUCTION

In many technical applications, multiphase flows meet conditions such as high-pressure environments and/or high velocities that prohibit the popular assumption of incompressibility. Important examples for such extreme ambient conditions include fuel injection systems of aeronautical, automotive, and rocket engines that operate at high-pressure and/or high-temperature conditions. In contrast to the incompressible treatment, which is typically found in two-phase numerical solvers in practical use today, all conservation equations are coupled via the equation of state (EOS) and have to be solved simultaneously. The decomposition and separate solution of hydrodynamics and thermodynamics as in the incompressible case is no longer possible. Hence, the numerical simulation of compressible multiphase flow introduces additional difficulties to the numerical approach mainly due to the necessary consistent approximation of thermodynamics and the large density and pressure jumps. The term multiphase denotes here the occurrence of two different phases at the same time in the computational domain, for example, liquid and vapor.

Two elements of the numerical algorithm for the simulation of two-phase flow are crucial: a method that allows to define the interface geometry and its temporal evolution, and a numerical strategy to treat the discontinuous nature of the interface together with the related physical processes such as surface tension or phase change. In here, the considered compressible case may include different equations of states on both sides of the phase interface that reveal the dissimilar behavior of

<sup>\*</sup>Correspondence to: Stefan Fechter, Institute of Aerodynamics and Gas Dynamics, University of Stuttgart, Pfaffenwaldring 21, D-70569 Stuttgart, Germany.

<sup>†</sup>E-mail: stefan.fechter@iag.uni-stuttgart.de

the liquid and gaseous phases. Usually the liquid phase is much stiffer and, hence, less compressible as the gaseous phase.

In literature, two main interface coupling methods are available. First, the diffuse interface approach, which is characterized by a smooth transition region between the bulk phases resulting in a finite interface thickness. Typical representatives are Baer–Nunziato and Navier–Stokes–Korteweg models (e. g., [1, 2]). The smoothing of the interface is provided by the numerical scheme or explicitly introduced to allow a stable resolution by the numerical approximation. The challenge here is to derive a thermodynamically consistent smoothening of the interface that does not violate the basic thermodynamic laws. The second coupling method is the sharp-interface representation of the interface that avoids any smoothening at the interface, which is a challenge for the coupling of the flow solver in the bulk phases and the interface treatment. The compressible flow solver resolves the macroscopic scales of the flow and necessarily allows discontinuous states at an accurate interface position. Appropriate jump conditions at the interface as well as the interface position are mandatory for the sharp-interface models. The coupling of the macroscopic flow solver and the interface tracking must establish the sharp-interface treatment. Beside the use of a moving grid (e. g., [3]) or the introduction of cutted cells [4], the so called ghost-fluid approach has been introduced. This formalism has been developed by Fedkiw *et al.* [5] for finite volume schemes. In this paper, fictitious fluid states are introduced near the interface, similar to the well-known concept of ghost cells to define boundary conditions.

In the following, we consider the latter sharp-interface approach based on the ideas of the ghost-fluid method. Information about the physics and the velocity of the interface is provided by an approximate local solution, which we call the micro-scale solver. In the present study, we rely here on Riemann-type solvers [6–8] and use the micro-scale model of Fechter *et al.* [8] at the interface. It is applicable to general EOS, even to nonanalytic EOS. Furthermore, it is straightforward to include the effects of surface tension or phase change into the algorithm as only the micro-scale solver at the phase interface has to be extended.

The macro-scale model is based on a discontinuous Galerkin spectral element method (DGSEM) [9] that allows for a high order of accuracy as well as efficient calculations. The discontinuous Galerkin (DG) formulation allows discontinuities at the element interfaces, which simplifies the implementation of the sharp-interface approach as the only numerical fluxes at the detected interface position that have to be adapted. The interface is tracked by a level-set method as proposed in [10]. Here, the high-order approximation of the level-set allows for an accurate curvature estimation at the phase boundary, an information that is needed for the inclusion of surface tension effects. The DG method is combined with a ghost-fluid approximation to preserve the discontinuous interface between the liquid and vapor phases. With the information of the interface location, the micro-solver determines the interface advection velocity for the level-set equation and the states from both sides of the interface, which are used in the flow solver as states in the ghost cells. The ghost cells then allow that the flow solver simulates either the gas or the liquid. Mixture states are circumvented in the sharp-interface approach.

The DG cells may be large due to the continuous in-cell resolution. At strong gradients or interfaces, this results to inaccuracies. In this paper, this is avoided by use of a subcell resolution, similar to ideas of Touil *et al.* [11] for spectral methods. The coarse DG cell with the interface is subdivided into multiple subcells, in which a finite volume approach is applied. This allows an improved resolution of the location of the interface as in the finite volume context discontinuities between each subcell is possible. Hence, this approach combines the high-order resolution of the DG scheme away from the interface with a refined finite volume resolution at the phase boundary. The interface tracking method, here the level-set approach, is always treated with the high-order DG method to take advantage of the low dispersion and dissipation properties. This is a favorable approach for the level-set advection as already described by Marchandise and Remacle [12].

This paper shows the first application of a high-order DG scheme for the simulation of compressible two-phase flow in three space dimensions that is based on a sharp-interface approximation via a ghost-fluid approach. High-order finite volume scheme has been proposed by Ferrari *et al.* [13] in one space dimension, by Hu *et al.* [7] and by Liu *et al.* [14] in multiple dimensions.

The paper is organized as follows: the physical model used for the simulation of compressible multiphase flow is introduced in Section 2, followed by a description of all building blocks in Section 3. Section 4 contains the overview of the information transfer between macro-scale and micro-scale models at the phase interface. In Section 5, numerical results of compressible droplet simulations are presented. The last section contains our conclusions.

## 2. MATHEMATICAL MODEL

We consider the compressible Euler equations as mathematical model for the simulation of compressible multiphase flows for both bulk phases. These equations can be written conveniently in the conservative form

$$\mathbf{U}_t + \nabla \cdot \mathbf{F}(\mathbf{U}) = \mathbf{0}, \quad (1)$$

with the three-dimensional (3D) Euler flux function  $\mathbf{F}(\mathbf{U})$

$$\mathbf{F}_1 = \begin{pmatrix} \rho u \\ \rho v^2 + p \\ \rho uv \\ \rho uw \\ u(\rho e + p) \end{pmatrix}, \quad \mathbf{F}_2 = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(\rho e + p) \end{pmatrix}, \quad \mathbf{F}_3 = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(\rho e + p) \end{pmatrix}, \quad (2)$$

and the vector of the conservative variable  $\mathbf{U} = (\rho, \rho \vec{v}, \rho e)^T$ , where  $\rho$  is the fluid density,  $\vec{v} = (u, v, w)^T$  is the velocity vector, and  $e$  is the specific total energy. In the following, we restrict ourselves to one species with two phases, for example, liquid and its vapor. The mixing of different components is not considered. This is out of the scope of this paper and has to be introduced by an additional equation for the mixture content.

For the sharp resolution of the phase interface of the liquid and its vapor, we use the ideas of the ghost-fluid method as developed by Fedkiw *et al.* [5] and apply jump conditions at the interface position. The jump conditions are provided by a Riemann-type solvers at the interface [6–8] and use the micro-scale model of Fechter *et al.* [8]. This model is applicable to general real equations of state being given even in a nonanalytic form. In our simulations, we use for simplicity the law of a perfect gas in the gaseous phase and the weakly compressible Tait EOS for the liquid phase. The inclusion of surface tension effects is carried out by introducing a surface force within the generalized Riemann problem at the interface. An alternative is the inclusion of the surface tension force as smeared volume source term that is denoted as continuous surface force model introduced by Hirt and Nichols [15].

## 3. THE NUMERICAL METHOD

In the following sections, we describe the basic building blocks of the proposed two-phase method. A short overview for orientation is given by

- Step 1: Computation of the curvature  $\kappa$  of the interface, whose location is given by a high-order numerical approximation of the level-set equation. To have a smooth and accurate approximation at the interface, the level-set variable is improved by an a posteriori reconstruction.
- Step 2: At the interface, a two-phase Riemann problem is solved, which provides information about the local normal interface velocity  $s_{PB}$  as well as the states at the interface from the right-hand to left-hand side. Such a Riemann problem is solved at each surface integration point of the phase interface and takes surface tension effects directly into account. Because of these source terms, it has to be considered as a generalized Riemann problem. The initial states at the interface are provided by extrapolation of the values from the liquid and gas bulk phases. In case phase transition occurs, additional information from the micro-scale has to be provided to ensure uniqueness of the solution.

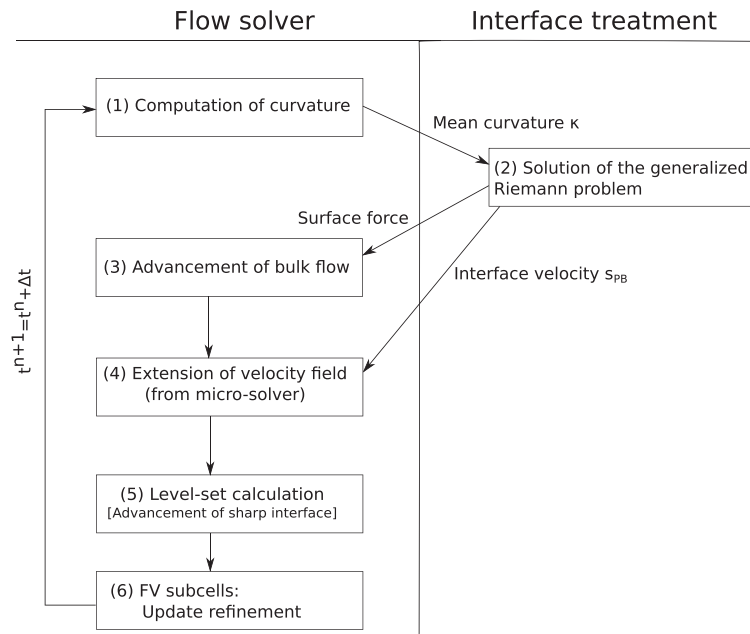


Figure 1. Numerical algorithm used for the simulation of compressible two-phase flows with surface tension and a local Riemann solver at the interface.

Step 3: The explicit DGSEM is used to advance the flow field to the next time level  $t^{n+1}$  using the compressible Euler equations. The flow solver is a single-phase solver for a general EOS and is applied either in the gas phase or in the liquid phase. The appropriate fluxes are calculated with the states given by the interface Riemann solver from step 2.

Step 4: The local normal interface speed  $s_{PB}$  from step 2 at the computational interface is extended along the level-set normals  $\mathbf{n}_{LS}$  to obtain a velocity field for the level-set advection equation.

Step 5: The new position of level-set zero is used to determine the updated position of the physical and computational interfaces.

An additional step will be described later. If the level-set approximation shows stronger jumps and an oscillation indicator of the density indicates oscillations, an inner cell refinement is applied introducing multiple finite volume cells inside the DG grid cell. A conservative interpolation/projection method is used to ensure consistency of the DG and finite volume methods at the interface.

In the following sections, the basic steps of the sharp-interface treatment are described in detail. The interaction between the different parts of the numerical algorithm is visualized in Figure 1.

### 3.1. The discontinuous Galerkin spectral element flow solver

In this section, we describe the flow solver as used in step 3 of the two-phase flow treatment. It is based on the density-based approach for the solution of the compressible Euler equations with a real EOS.

The description of the method is kept short, because it is an extension of a one-fluid flow solver as described in [16]. Hereby, the scheme is based on the DG spectral element approach as described in [9, 17]. The formulation of the DGSEM method is performed for the system of conservation equations in the form

$$\mathbf{U}_t + \nabla \cdot \mathbf{F}(\mathbf{U}) = \mathbf{0}, \quad (3)$$

where  $\mathbf{U}$  is the vector of the solution unknowns and  $\mathbf{F}$  is the corresponding physical flux. In the following, we just consider the Euler equations and neglect the viscous terms and heat conduction of the Navier–Stokes equations.

In a 3D domain, we subdivide the computational space into nonoverlapping hexahedral grid cells. Each grid cell is mapped onto the reference cube  $E := [-1, 1]^3$  by mapping  $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$ , where  $\boldsymbol{\xi} = (\xi, \eta, \zeta)$  is the coordinate vector of the reference cube. The mapping onto the reference element,  $E$ , transforms (3) to the system

$$\tilde{U}_t + \nabla_{\boldsymbol{\xi}} \cdot \tilde{\mathbf{F}}(\tilde{U}) = 0, \quad (4)$$

where

$$\tilde{U} = JU, \quad \tilde{\mathbf{F}}^i = J\mathbf{a}^i \cdot \mathbf{F} = \sum_{n=1}^3 (Ja_n^i) F_n \quad (5)$$

are the mapped quantities and  $J\mathbf{a}^i$  denote the metric terms of the transformation.

Kopriva [18] showed how to compute the metric terms to satisfy the free-stream preserving condition. The basis of the DG approximation is the weak form of the conservation law that is gained by multiplying (4) with a test function  $\phi$  and integration over the reference grid cell  $E$ . Applying an integration by parts leads to the weak formulation

$$\int_E \tilde{U}_t \phi d\boldsymbol{\xi} + \int_{\partial E} \tilde{\mathbf{F}} \cdot \hat{\mathbf{n}}_{\boldsymbol{\xi}} \phi dS - \int_E \tilde{\mathbf{F}} \cdot \nabla_{\boldsymbol{\xi}} \phi d\boldsymbol{\xi} = 0. \quad (6)$$

The approximate function is defined by

$$\tilde{U}^N(\boldsymbol{\xi}, t) \equiv \mathbb{I}^N \tilde{U} = \sum_{\lambda, \mu, \nu=0}^N \tilde{U}_{\lambda\mu\nu}(t) \psi_{\lambda\mu\nu}(\boldsymbol{\xi}), \quad (7)$$

where  $\mathbb{I}^N$  is the discrete interpolation operator of order  $N$ . The basis functions  $\psi$  are products of one-dimensional Lagrange polynomials

$$\psi(\boldsymbol{\xi}) = \psi_{ijk}(\boldsymbol{\xi}) = \ell_i(\xi)\ell_j(\eta)\ell_k(\zeta). \quad (8)$$

In  $\xi$  direction, the corresponding Lagrange polynomial  $\ell_j(\xi)$ ,  $j = 0, \dots, N$  of degree  $N$  is defined by

$$\ell_j(\xi) = \prod_{\substack{i=0 \\ i \neq j}}^N \frac{\xi - \xi_i}{\xi_j - \xi_i}, \quad j = 0, \dots, N, \quad (9)$$

where  $\xi_i$ ,  $i = 0, \dots, N$  are given points in  $[-1, 1]$ . To have good interpolation properties, favorable choices are the Gauss or Gauss–Lobatto points.

The integrals in (6) are approximated by numerical quadrature. Here, the same set of points is used as for the interpolation. Furthermore, we choose the test function equal to the basis functions ( $\phi = \psi$ ). To use the same set of points for interpolation and approximate integration strongly reduces the computational effort in this spectral element approach. The calculation of the 3D integral can be split into one-dimensional sweeps.

The approximation of the volume integral in (6) is given by

$$\int_E \tilde{\mathbf{F}}^N \nabla_{\boldsymbol{\xi}} \phi d\boldsymbol{\xi} = \int_E \mathbb{I}^N \tilde{\mathbf{F}} \nabla_{\boldsymbol{\xi}} \phi d\boldsymbol{\xi}, \quad (10)$$

which is the orthogonal projection with respect to the discrete inner product. In general,  $\tilde{\mathbf{F}} = \tilde{\mathbf{F}}(\boldsymbol{\xi}, \tilde{U})$  is a nonlinear function of the solution and depends on the transformation metric, as seen in (5). In collocation nodal spectral methods, one approximates each function as a polynomial and then projects the result onto the nodes. The contravariant flux  $\tilde{\mathbf{F}}$  is also represented as a polynomial of degree  $N$ .

To obtain the final nodal spectral DG approximation, the polynomial approximations for the solution and the fluxes are substituted into the weak form (6)

$$\int_E \tilde{U}_t^N \phi_{ijk} d\xi + \int_{\partial E} (\tilde{F}^N \cdot \hat{n}_\xi) \phi_{ijk} dS - \int_E \tilde{F}^N \nabla_\xi \phi_{ijk} d\xi = 0 \quad (11)$$

$$i, j, k = 0, 1, \dots, N,$$

where  $\hat{n}_\xi$  is the outwards directed normal on the surface  $\partial E$  of the reference cube. The approximate solution in adjacent elements is coupled to each other and to the physical boundaries through the surface integral. The numerical surface flux depends on the values from both sides of the surface and is determined as in finite volume schemes by a numerical flux, for example, based on Riemann solvers. Here, we use the numerical contravariant flux  $\tilde{F}$  for the bulk phases, calculated by an approximation of the flux of the local Riemann flux  $\mathbb{F}$  as

$$(\tilde{F}^d \cdot \hat{n}_\xi)^* = |J \mathbf{a}^d| \mathbb{F}(U^L, U^R; \hat{n}^d), \quad (12)$$

where  $U^L$  and  $U^R$  denote the states at the element faces,  $\hat{n}$  is the normal vector in the physical space, and  $d$  is the direction in the reference cube. For standard numerical fluxes and Riemann solvers, see, for example, the book of Toro [19].

The resulting discrete weak formulation of the DGSEM reads then as

$$\underbrace{\int_E \tilde{U}_t^N \phi_{ijk} d\xi}_{\text{Time derivative}} + \underbrace{\int_{\partial E} (\tilde{F}^N \cdot \hat{n}_\xi)^* \phi_{ijk} dS}_{\text{Surface integral}} - \underbrace{\int_E \tilde{F}^N \nabla_\xi \phi_{ijk} d\xi}_{\text{Volume integral}} = 0 \quad (13)$$

$$i, j, k = 0, 1, \dots, N,$$

and establishes the complete spatial approximation of the compressible flow solver for a real EOS, which is applied in the bulk phases.

For simplicity, we rely in this paper on a combination of two simple EOSs. In the gaseous phase, we use the EOS of a perfect gas, whereas in the liquid phase, the weakly compressible Tait EOS is applied. This allows to solve for the temperature and inner energy directly. The use of more complex EOS implies that the computational time for the EOS evaluation may increase strongly. Often the temperature and pressure cannot be written analytically in terms of the density and the inner energy or vice versa, which implies that this has to be determined iteratively. Alternatively, the primitive variables have to be precalculated and to be stored within a table, based on the ideas of Dumber *et al.* [20]. During the simulation, the primitive states are then interpolated from the table.

The numerical treatment of the two-phase interface and the resolution of the interfacial physics with coupling to the flow solver is described in Section 4.

### 3.2. Subcell resolution

The DG scheme is highly efficient in smooth parts of the flow, if it is applied on coarse grid cells with high order of accuracy. On the other hand, the sharp localization of an interface on a coarse grid is in this case a problem. To overcome this problem of the high-order DG scheme on big cells, we introduce inside any interface cell a set of equidistant subcells. Here, the DG polynomial of high degree in the macro-grid cell is projected to a piecewise constant function, constant in every small interior subgrid cell. This allows a sharper local resolution of the interface and applies a finite volume scheme within the DG grid cell. The approach is similar to the shock-capturing strategy described in [21, 22, 35].

This can be written in terms of a modified evaluation of the volume integral in (13). The volume integral is replaced by the sum of the finite volume sub-cell contributions. This reads as

$$\int_E \tilde{F}^N \nabla_\xi \phi_{ijk} d\xi = \sum_{i,j,k=1}^{N-1} \int_{\partial e} (\tilde{F} \cdot \hat{n}_\xi)^* dS, \quad (14)$$

where  $\partial e$  is the surface area of the finite volume sub-cell  $e$ . The sum of the right-hand side runs over all inner sub-cell side faces, evaluated as interior limit. This simply means that DG volume integral in (13) is expressed as the sum of surface contributions of the equidistant sub-cells. The internal point structure of the DG cell with Gaussian quadrature points is compared with a DG cell with finite volume subcells in Figure 2.

The calculation of the surface integral at the outer DG grid cell boundaries should be preserved, which faces the problem that different surface integration points are used: Gauss or Gauss–Lobatto distributed points in the DG cells and equidistant distributed points in the finite volume subcells. Here, a conservative flux projection and interpolation method is chosen according to

$$\int_E (\tilde{U}_{ijk} \phi_{ijk})_{\text{DG}} d\xi = \sum_{i,j,k=0}^N \int_e (\tilde{U}_{ijk} \cdot 1)_{\text{FV}} d\xi = \sum_{i,j,k=0}^N (\tilde{U}_{ijk})_{\text{FV}} \quad (15)$$

and a cell-local reconstruction as inverse. This guaranties that the overall calculation of the surface integral in (13) remains the same.

The subcell partition is carried out such that the number of the degrees of freedom remains constant, which implies that one DG grid cell is refined into  $(N + 1)^3$  interior subcells. This allows to switch from the spectral representation of DGSEM directly to a piecewise constant representation and vice versa. The approximation is advanced in time in the subcells by a standard second-order finite volume scheme with a TVD reconstruction. The same time step can be chosen because of the fact that the time restriction depends on the number of degrees of freedom. We note that the continuous level-set variable is always treated with the DG scheme as for the calculation of the curvature of a high-order approximation of the level-set function is advantageous.

The considered subcell approach significantly improves the localization of the phase interface. This enhancement is visualized in Figure 3 showing the density iso-contour of the liquid phase.

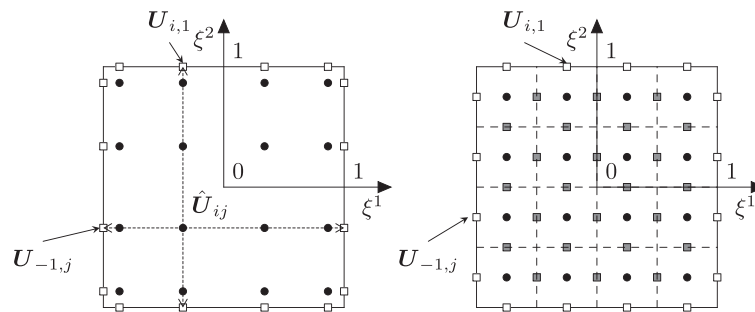


Figure 2. Comparison of the internal point structure of a discontinuous Galerkin cell with Gaussian quadrature points (left) and the finite volume subcell distribution (right). The gray rectangles indicate the additional flux evaluation points for the finite volume subcell approach.

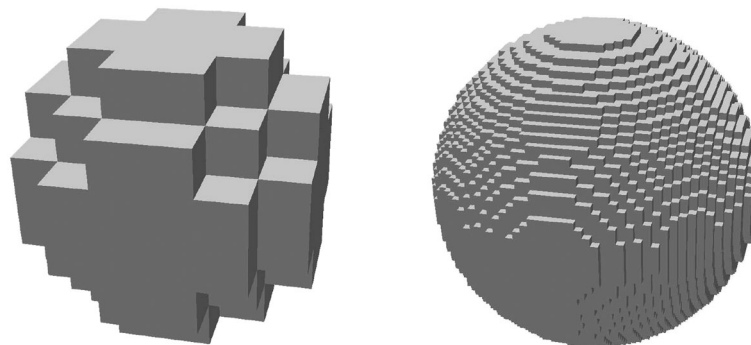


Figure 3. Comparison of the phase boundary approximation for a discontinuous Galerkin scheme of order six using the density iso-contour of the liquid phase. Left: approximation using a pure discontinuous Galerkin scheme. Right: approximation using the adaptive mesh refinement at the phase boundary.

For this study, a sixth-order DG scheme (polynomial degree  $N = 5$ ) is used. The subcell approach combines both, the high-order resolution of a DG scheme in smooth parts of the flow and the local resolution of discontinuities by a finite volume scheme on a fine grid. The finite volume scheme introduces larger dissipation errors on the fine grid and is less efficient with respect to the computational effort. Hence, it is applied only within a small region around the phase interface.

### 3.3. Interface tracking by the level-set approach

The sharp-interface approach used in this study relies on the treatment of the phase interface as a discontinuity. Hence, an accurate geometrical information with position, normals  $\mathbf{n}_{\text{LS}}$ , and curvature  $\kappa$  of the phase interface is needed. To obtain this information, we rely on a level-set approach

$$\frac{\partial \Phi}{\partial t} + s_{\text{PB}} \nabla \Phi = 0 \text{ in } \Omega \times (0, T) . \quad (16)$$

The variable  $\Phi$  is initialized as a signed distance function with respect to the interface and represents the temporal evolution of the phase interface. The zero iso-contour of the level-set function represents the physical position of the interface. A review of level-set methods is given in [23].

To apply the DGSEM scheme as described in Section 3.1, the level-set equation is recast to the nonhomogenous conservation equation

$$\frac{\partial \Phi}{\partial t} + \nabla \cdot (s_{\text{PB}} \Phi) = \Phi \nabla (s_{\text{PB}}) . \quad (17)$$

In the incompressible case, Grooss and Hesthaven [24] and Marchandise and Remacle [12] derived a DG formulation for the level-set advection equation and already showed the suitability of this approach. In the compressible case, the right-hand side term has to be considered in addition.

The level-set function is only important within a small region around the interface, where the geometry and the secondary interface quantities are needed. To facilitate the numerical treatment concerning boundary conditions, the distance function  $\tilde{\Phi}$  is smoothed in the outer region by

$$\Phi = \begin{cases} \tilde{\Phi} & \text{if } |\tilde{\Phi}| \leq \Phi_{\text{max}}, \\ p(\tilde{\Phi}) & \text{if } |\tilde{\Phi}| > \Phi_{\text{max}}, \end{cases} \quad (18)$$

with the continuous cropping function

$$p(\Phi) = \frac{1}{4} \Phi_{\text{max}} \tanh \left( \frac{4}{\Phi_{\text{max}}} (\Phi - \text{sign}(\Phi) \Phi_{\text{max}}) \right) + \text{sign}(\Phi) \Phi_{\text{max}} . \quad (19)$$

The level-set cropping level  $\Phi_{\text{max}}$  is a user-defined parameter that depends on the droplet size and the width of the narrow-bandwidth approach.

This has two advantages: first, the smoothened level-set function is easier to handle using a high-order method, which may be also advantageous calculating the derivatives at the interface. Second, the cropping facilitates the handling of the narrow-band approach and the implementation of the boundary conditions.

### 3.4. Geometry representation

The piecewise polynomial DG approximation with its discontinuities at the element boundaries may influence the accuracy to calculate the interface curvature  $\kappa$  from the level-set function  $\Phi$ , which is the second derivative. It is defined along with the level-set normals  $\mathbf{n}_{\text{LS}}$  as

$$\mathbf{n}_{\text{LS}} = \frac{\nabla \Phi}{|\nabla \Phi|}, \quad \kappa = \nabla \cdot \frac{\nabla \Phi}{|\nabla \Phi|} . \quad (20)$$

Formally, a two-order lower convergence order for the second derivative is obtained. This loss in order is counteracted through the use of a reconstructed level-set polynomial for the estimation of the curvature. We introduce a  $P_N P_M$ -reconstruction as proposed by Dumbser [25] up to the



order  $M = 3N_{\text{DG}} + 2$  to enhance the accuracy of the curvature estimation. We used a central reconstruction stencil on the grid to reconstruct the solution available at the polynomial degree  $N_{\text{DG}}$  to a higher polynomial degree  $M$ . The reconstructed level-set polynomial  $\bar{\Phi}$  has the property that it is much smoother at the element interfaces, and therefore, the gradients can be evaluated locally for each cell. Hence, no lifting of the gradients is necessary.

The  $P_N P_M$ -reconstruction operator for the level-set polynomial can be written as inverse of an  $\mathbb{L}_2$  projection. The reconstruction operator is a square matrix due to the choice of  $M = 3N_{\text{DG}} + 2$  and can be inverted easily. Note that the operator is not always exactly invertible, for choices of  $N \leq M < 3N_{\text{DG}} + 2$ , a least-square algorithm has to be used to compute the inverse. The reconstruction stencil consists of three grid cells in each axis direction with  $k = -1, \dots, 1$ . Exemplary reconstruction is written for the  $\xi$ -direction

$$\int_{-1}^1 \bar{\Phi}(\xi + 2k) \phi_i d\xi = \int_{-1}^1 \Phi^k(\xi) \phi_i(\xi) d\xi, \quad i = 0, \dots, N. \quad (21)$$

For the reconstruction of the level-set polynomial  $\Phi$ , we approximate the unknown function  $\bar{\Phi}$ , the polynomial data  $\Phi^k(\xi)$  are given. For the linear level-set function, the reconstruction operator can be written as constant reconstruction matrix-vector formulation

$$\mathbb{C}^k \hat{\Phi} = \hat{\Phi}^k \quad (22)$$

with

$$\mathbb{C}_{ij}^k = \int_{-1}^1 \bar{\phi}_j(\xi + 2k) \phi_i(\xi) d\xi. \quad (23)$$

Here,  $\phi_i(\xi)$  are the Lagrangean basis functions in  $[-1, 1]$ . The reconstruction of the degrees of freedom is then given by the inverse of the matrix  $\mathbb{C}$

$$\bar{\Phi} := \mathbb{C}^{-1} \Phi = \sum_{k=-1}^1 \mathbb{C}_k \Phi^k. \quad (24)$$

Note that due to the linear level-set function, the reconstruction matrix  $\mathbb{C}$  is computed once in a preprocessing step.

Because of the reconstruction, the accuracy of the gradient estimation is significantly enhanced. The continuous reconstruction facilitates the gradient estimation as the jumps at the element boundaries have less influence and a direct differentiation of the reconstructed polynomial is possible.

### 3.5. Reinitialization of the level-set variable

In order to keep the level-set function close to a signed distance function in the vicinity of the phase interface, reinitialization steps are needed after a number of time steps. This is carried out to limit the numerical diffusion because of too steep or flat level-set gradients. Small variations in the flow field may locally influence the level-set function and disturb the interface. Because of the fact that we even need second-order derivatives at the interface, even small disturbances in the level-set polynomial may lead to large inaccuracies in the curvature calculation. Typically, the reinitialization procedure is applied every 100 time steps.

We apply an iterative reinitialization procedure as introduced by Sussman *et al.* [10], modified to maintain the level-set restriction similar to the description in (18)

$$\Phi_t = S_\epsilon(\Phi_0)(l(\Phi) - |\nabla \Phi|), \quad (25)$$

with the  $l$ -function defined as

$$l(\Phi) = \begin{cases} 1 & \text{if } |\Phi| < \Phi_{\max}, \\ 0 & \text{if } |\Phi| > 1.25\Phi_{\max}, \\ p^*(\Phi) & \text{in between.} \end{cases} \quad (26)$$

The term  $S_\epsilon$  represents a sign function and  $\Phi_{\max}$  is the cropping limit of the level-set function. It has been observed that it is advantageous to use a smeared sign function instead of the sharp distance function (e. g., [10]). The sign function is defined as follows

$$S_\epsilon(\Phi_0) = \frac{\Phi_0}{\sqrt{\Phi_0^2 + \epsilon}} \quad (27)$$

and is only dependent of the initial conditions  $\Phi_0$  of the pseudo-iteration. The smoothening term  $\epsilon$  is chosen according to the minimal grid length in the computational domain. Equation (25) is iterated to steady-state, which is typically achieved after a few iterations.

This function  $l(\Phi)$  is chosen such that it efficiently forces the level-set gradient in Equation (25) to 0, if  $|\Phi| > \frac{5}{4}\Phi_{\max}$  and to 1 if  $|\Phi| < \Phi_{\max}$  with a smooth transition in between. This smooth transition is provided by the  $p^*$ -function

$$p^*(\Phi) = -64\Phi^3 + 192\Phi^2 - 192\Phi + 65 \quad (28)$$

that ensures a two times differentiable transition.

The reinitialization equation is of Hamilton–Jacobi type and is discretized using a fifth-order Weighted Essentially Non-Oscillatory (WENO) scheme as developed by Jiang and Peng [26] together with a Godunov-type numerical Hamiltonian. For the application of the WENO scheme, we make use of the finite volume subcells (Section 3.2).

### 3.6. Extrapolation of the interface velocity field

The level-set velocity is only available locally at the phase interface as it is determined by the interface Riemann solver. This procedure is described in Section 4. But, for the level-set equation (Section 3.3), the advection velocity within a whole region in the volume is needed. This means that the velocity of the phase interface has to be extended to the volume around the droplet in an appropriate way.

A clever way to construct the advection velocity field was proposed by Adalsteinsson and Sethian [27]. The advection velocity is reconstructed such that

$$\mathbf{s}_{\text{ext}} \cdot \mathbf{n}_{\text{LS}} = 0, \quad (29)$$

starting with the initial advection velocity  $\mathbf{s}_{\text{PB}}$  determined by the micro-solver at the interface. This extrapolated velocity field  $\mathbf{s}_{\text{ext}}$  introduces minimal disturbances for the advection of the level-set function. Hence, the reinitialization procedure is called in larger time intervals. The underlying extrapolation procedure is discretized using a WENO scheme and a pseudo-time stepping approach separately for each velocity component of  $\mathbf{s}_{\text{ext}}$

$$\frac{\partial \mathbf{s}_{\text{ext}}}{\partial \tau} + \nabla \mathbf{s}_{\text{ext}} \nabla \frac{\Phi}{|\nabla \Phi|} = 0. \quad (30)$$

We note that the numerical procedure is similar to the approach for the reinitialization (Section 3.5).

## 4. THE COUPLING TO THE INTERFACE

In this section, we describe how we couple the physics at the interface to the flow solver in the bulk phases. This is established by a ghost-fluid approach based on the solution of a local Riemann problem at the interface. We first describe the solution of the Riemann problem and how to use it. Here, we restrict ourselves to the case without phase transition, which simplifies the Riemann solution as the traditional structure of the Riemann solution is retained. But, a general EOS on every side of the interface is allowed, and surface tension is taken into account. Such a Riemann solver and appropriate approximations have been developed by Fechter *et al.* [8]. The approximate solution is based on the linearized Lax curves, which has comparatively low computational costs

and needs solely an estimation of the sound speed in the bulk phases at the interface. It ensures a thermodynamic consistent approximation of the interface together with suitable jump conditions. Another example for a suitable interface Riemann solver is the Harten, Lax, van-Leer with contact restoration (HLLC) interface Riemann solver by Hu *et al.* [7]. Without phase transition and mass transfer, the Riemann solution has the usual structure and consists of four states separated by simple waves.

In the case of phase transition, this classical structure of the Riemann problem solution breaks down, and additional information is needed from the micro-scale that defines the mass transfer rate. The effects of phase transfer can be included by introducing into the Riemann problem solution an under-compressive shock as an additional wave, which is specified by a micro-scale, for example, by a kinetic relation. This construction is described by Rohde and Zeiler in [28] and in Chalons *et al.* [29] for the isothermal case and can be generalized to the general Euler equations. This case with phase transition is out of the scope of this paper and will be considered in future in detail.

Surface tension effects are taken into account at the interface, similar to the approach by Jaegle *et al.* [30]. The magnitude of the pressure jump is chosen according to the Young–Laplace law

$$(\rho(\mathbf{v} \cdot \mathbf{n}_{LS} - s_{PB} \cdot \mathbf{n}_{LS})\mathbf{v} + p(\rho, T)\mathbf{n}_{LS}) = 2\sigma\kappa\mathbf{n}_{LS}, \quad (31)$$

for which the mean curvature, evaluated at the interface position (as defined by  $\Phi = 0$ ), is needed as input parameter. Hence, the surface tension is resolved as a force at the interface and is not smeared out to a volume source term and approximated by a volume source as it is usually performed, for example, by continuum surface force method of Brackbill *et al.* [31]. More information about the surface tension modeling is found in Appendix A.

From the local solution of the Riemann problem to the normal direction of the interface, we obtain two types of information: the velocity of interface propagation  $s_{PB}$  and the states from both sides of the material interface ( $\mathbf{U}_L^*$  and  $\mathbf{U}_R^*$ ). The velocity  $s_{PB}$  is needed for the level-set equation. The states are then used to calculate the numerical fluxes in grid cells with an interface. According to the ghost-fluid approach, we have two sets of data in an interface grid cell. One set is used to calculate the numerical flux to fluid 1 and the other the numerical flux to fluid 2. The flow solver always operates in a single fluid. Looking at a typical wave structure of the interface Riemann problem as sketched in Figure 4, proper states for the ghost cells are the states left and right of the material interface  $\mathbf{U}_{int,L}^*$  and  $\mathbf{U}_{int,R}^*$ . In our simulations, we directly take the fluxes ( $\mathbf{F}_{int,L}^*$  and  $\mathbf{F}_{int,R}^*$ ).

This ghost-fluid approach shifts the waves, generated in the interface Riemann problem in the different fluids to the grid cell interface of the corresponding fluid. Defined in this way, the ghost data generate the proper waves in the bulk phases. Because of the different fluxes from the different sides at a grid cell with a material interface, the mixing is avoided, but the conservation can no longer be satisfied locally. A good quality of the determination of the interface velocity is important. The considered approach relies on the original ghost-fluid ideas of Fedkiw *et al.* [5] and its extension by Merkle and Rohde [6] and by Dressel and Rohde [32] for piecewise constant approximations in one space dimension and in the multidimensional case, respectively. In the bulk phases away from the

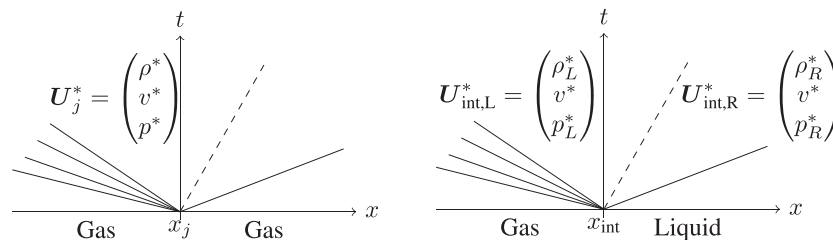


Figure 4. Comparison of the Riemann wave pattern. Left: traditional single-phase Riemann problem with single intercell flux  $\mathbf{F}^* = \mathbf{F}^*(\mathbf{U})$ . Right: Riemann problem at the phase interface without phase transfer. The fluxes  $\mathbf{F}_{int,L}^* = \mathbf{F}^*(\mathbf{U}_{int,L})$  and  $\mathbf{F}_{int,R}^* = \mathbf{F}^*(\mathbf{U}_{int,R})$  are the intercell fluxes.

interface, the numerical flux of the DGSEM scheme is calculated by a standard one. The transfer of information from the local interface Riemann problem to the flow solver is illustrated in Figure 5 in a one-dimensional case and in Figure 6 for the two-dimensional (2D) case.

In Figure 6,  $U_L := U^-$  and  $U_G := U^+$  denote the respective states in the two different phases in some quadrature point at the numerical interface. These states are the input for the interface Riemann solver together with the mean interface curvature, if surface tension is considered. The multimaterial Riemann solver delivers for each quadrature point a liquid state  $U_L^*$  and a gaseous state  $U_G^*$ . The DGSEM now uses the flux evaluations  $F^* = \tilde{F}(\tilde{U}_L^*)$  for the computation on the liquid element and  $F^* = \tilde{F}(\tilde{U}_G^*)$  on the gaseous element. Inside the finite volume subcells, the numerical fluxes are used on each face of the subcells.

In case the interface position, as defined by the zero iso-contour of the level-set function, has moved across one grid cell, the state in this cell has to be redefined according to its new fluid state. As no information about the state in the cell is available, this state has to be extrapolated using the adjacent grid cells. This is due to the approximation of the interface in the sharp-interface context. We approximate the state in this cell by the mean value of the surrounding cells that are in the

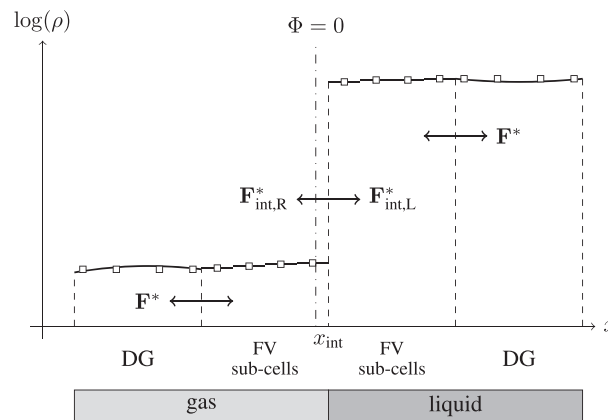


Figure 5. One-dimensional treatment of the fluxes at the phase interface. Within the bulk phases, the standard single-phase Riemann flux  $F^*$  is used. The interface position is marked with  $\Phi = 0$ . At the interface position, the fluxes  $F_{\text{int,L}}^*$  and  $F_{\text{int,R}}^*$  are applied to the left and right phases. DG, discontinuous Galerkin; FV, finite volume.

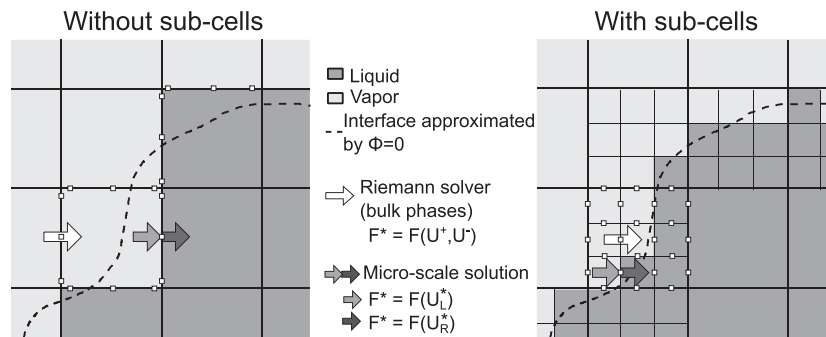


Figure 6. Schematic diagram of a typical setting in the two-phase approach for the sharp-interface approach: the liquid–vapor interface is approximated by the zero level-set  $\Phi = 0$ . According to the ghost-fluid approach, the numerical fluxes at the grid cell boundaries are determined either by a standard Riemann solver (bulk phase) or the interface Riemann solver and the ghost-cell data. The white dots visualize the integration points at the element boundaries.

same phase

$$\mathbf{U}_{\text{new}} = \frac{1}{\sum_{i=1}^{\text{nCell}} \delta_{i,\text{new}}} \sum_{i=1}^{\text{nCell}} \delta_{i,\text{new}} \mathbf{U}_i \quad (32)$$

with

$$\delta_{i,\text{new}} = \begin{cases} 1 & \text{if fluid}(i) = \text{fluid}(\text{new}), \\ 0 & \text{else.} \end{cases} \quad (33)$$

This state averaging is carried out for all surrounding nCell cells of the same fluid phase (in 3D: nCell  $\in [1, 6]$ ). To avoid pressure discontinuities due to the state extrapolation, the average pressure is calculated instead of the average total energy.

## 5. NUMERICAL RESULTS

To illustrate the capabilities of the method, three test cases of 3D droplet dynamics with increasing complexity are presented after a one-dimensional validation problem. The first 3D test case validates the inclusion of the surface tension effects in the interface Riemann solver. We simulate a droplet at rest as a generic test case to assess the amount of parasitic currents induced by the surface tension modeling and to validate the subcell mesh refinement at the interface. The second test case is an initially ellipsoidal droplet that starts to move and to oscillate with decreasing amplitude because of surface tension. The last case is as the most demanding one. A shock-droplet interaction is considered, in which a planar shock wave impinging a droplet at rest.

In all the test cases under consideration, the Euler equations are solved with equations of states that are different in both bulk phases. In the gaseous phase, the perfect gas law is applied, whereas in the liquid phase, the Tait EOS is used. This simplified EOS model does not describe the real fluid behavior very accurately but includes all features as the jump at the interface and surface tension at lower computational effort. An extension to more accurate EOS in a tabulated representation is straightforward and is work in progress.

### 5.1. One-dimensional shock-tube problem

In this first one-dimensional test case, the numerical interface coupling approach is validated versus an exact solution. As in the single-phase context, Riemann problems may serve as good test cases. The case is chosen in preparation for the shock-droplet interaction problem presented later in the 3D situation. Here, a plane Mach 3 shock wave in the gas phase is impinging onto a plane liquid phase. Additionally, strong surface tension forces are present at the interface in this test case. The surface tension effect is modeled by a pressure jump of  $\Delta p_\sigma = 1$  at the interface position in the one-dimensional framework. For the gas phase, the EOS of a perfect gas is taken, while for the liquid phase, the Tait EOS is used.

The initial conditions are Riemann problem data given by

$$(\rho, v, p, \gamma) = \begin{cases} (1.32575, 0.246215, 1.48783, 1.4) & \text{if } 0 \leq x \leq 0.5, \\ (1000, 0.0, 1.0, 7.15) & \text{else,} \end{cases} \quad (34)$$

$$\Delta p_\sigma = 1.$$

The nondimensional parameters of the Tait EOS are chosen to  $B = 3310$ ,  $A = 1$  and  $\rho_0 = 1000$ . The final simulation time is  $t = 0.07$ .

The exact solution is calculated by solving this generalized Riemann problem taking into account the surface tension. In Figure 7, the results for density, pressure, and velocity at the final simulation time are shown as a solid line. The exact solution exists of four constant states separated by three waves: the shock wave to the right within the liquid can be clearly seen in the plot of the pressure. The density jump as well as the change of velocity in the liquid due to the pressure change is

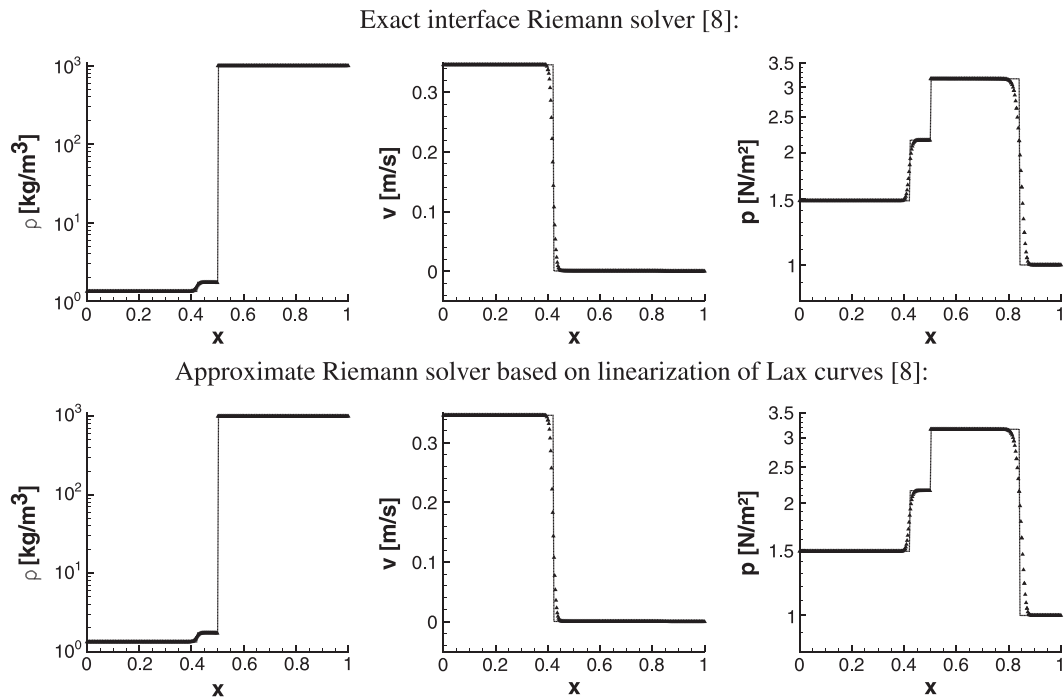


Figure 7. Results of the one-dimensional shock-droplet interaction test case for the linearized and exact two-phase Riemann solvers. Left: density, middle: velocity, and right: pressure. The solid line represents the exact solution and the triangles the numerical approximation.

very small and not visible. At the liquid gas interface, the pressure jumps because of the surface tension. The left going wave is a shock wave generated by reflection of the incident shock. This exact solution is compared with numerical results obtained on a grid with 100 grid points. Here, we performed two simulations with a different interface treatment. In the upper row of Figure 7, the numerical results compared with the exact solution are shown using the exact generalized Riemann solver at the interface of the numerical simulation. It is clearly seen that the phase interface is kept sharp because of the use of the ghost cell and the Riemann solver coupling approach. In the lower row, the exact Riemann solver at the phase interface is replaced by the linearized Riemann solver as proposed in [8]. Both simulations produce almost the same results with a good approximation of the exact solution of the Riemann problem including surface tension and related pressure jump at the interface. The solutions are oscillation-free despite the large pressure jump at the interface.

## 5.2. Spherical droplet at rest

The second test is a simulation of a droplet at rest including the described surface tension approach. As described in Section 4, the linearized Riemann problem solver is applied at the phase boundary taking into account surface tension effects as surface force. Therefore, the mean interface curvature is needed. The droplet is initialized at a slightly higher pressure than the surrounding gas, which is balanced by surface tension. The parameters given in detail in Figure 8 are chosen such that the problem is steady. If the calculation of the curvature and the resulting local solution at the interface are accurate enough, the approximation should preserve the steadiness. Hence, this test case allows for an assessment of the induced parasitic currents because of the surface tension modeling.

The simulation has been conducted over a dimensionless time of  $t = 0.5$  representing about 1040 iterations. The Cartesian grid consists of  $40 \times 40 \times 40$  equidistant cells, and the polynomial degree of  $N = 3$  has been chosen to provide an accurate estimate for the interface curvature. The computational domain is a box measuring  $[0.8, 0.8]$  in all axis directions. This resolution resolves one droplet diameter with 80 points.

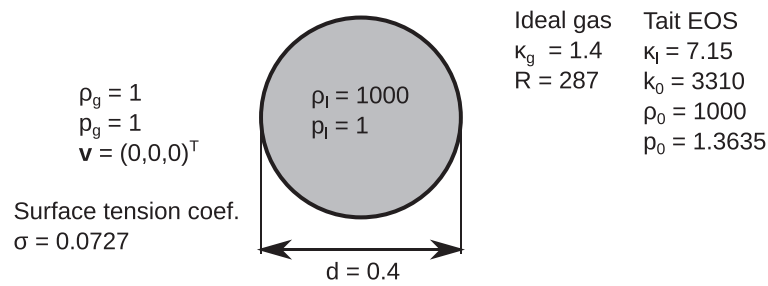


Figure 8. Initial conditions of the steady droplet with surface tension. Values chosen represent a water droplet in air. EOS, equation of state.

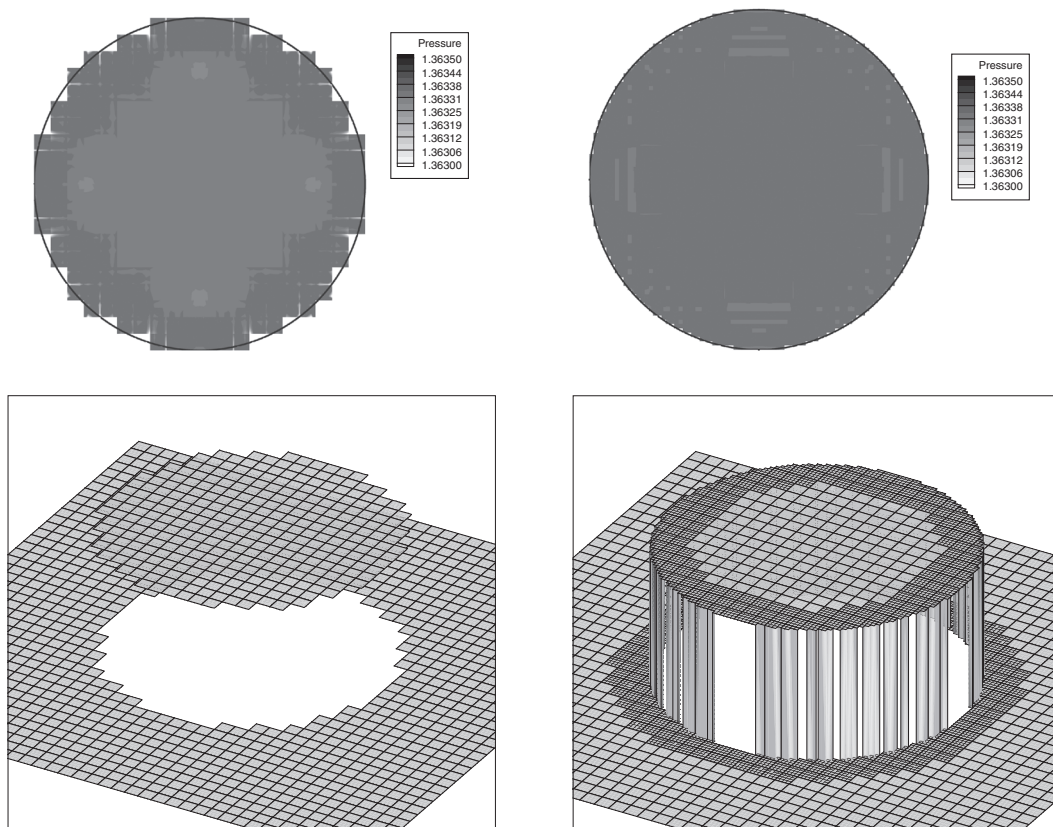


Figure 9. Result of the droplet at rest test case with surface tension at  $t = 0.5$ . The black line visualizes the physical phase boundary (zero level-set iso-contour). Left: pressure contours on the droplet median plane obtained for the case without subcell refinement. Right: pressure contours on the droplet median plane with subcell refinement at the phase boundary. Note that the color scale in the top row does not cover the total pressure jump. The range is chosen to highlight the minimal oscillations in the liquid phase.

Figure 9 shows pressure information extracted on the mid-plane of the droplet at  $t = 0.5$ . The pressure jump associated with the surface tension is visible at the cell boundaries near the level-set zero. The constant states on the gaseous side have been perfectly conserved, while minor fluctuations have been developed on the liquid side. This is related to the relatively stiff EOS for the liquid phase. Minor density fluctuations emerge as visual pressure disturbances. These fluctuations originate from the nonexact numerical curvature calculation that leads to some numerical noise. The corresponding result is shown for the case without and with the use of subcell refinement at the phase boundary. The resolution of the physical interface is much better using the subcell refinement especially for the here considered high-order scheme.

Table I.  $\mathbb{L}_2$  error of the curvature approximation for two different curvature approximation methods ( $P_N P_M$  and DG lifting) and the corresponding approximation order for the droplet at rest test case.

Grid spacing	$P_N P_M$ [25]		DG lifting [33]	
	$\mathbb{L}_2$ curvature	Order	$\mathbb{L}_2$ curvature	Order
1/10	5.828 E−01	—	7.745 E−01	—
1/20	1.152 E−02	5.66	1.647 E−01	2.23
1/40	1.292 E−04	6.47	5.043 E−02	1.71
1/80	1.604 E−05	3.01	1.719 E−02	1.55

DG, discontinuous Galerkin.

Note that the curvature calculation is the same for the case with and without refinement at the interface.

Table II.  $\mathbb{L}_2$  error norms of the momentum as a measure of the induced parasitic currents at  $t = 0.5$  with and without refinement at the interface.

Grid spacing	DOF per diameter	$\max(\mathbb{L}_2)$ DG	$\max(\mathbb{L}_2)$ DG subcells
1/10	20	7.485 E−04	6.838 E−05
1/20	40	1.921 E−04	2.683 E−05
1/40	80	7.549 E−05	1.642 E−05

DOF, degrees of freedom; DG, discontinuous Galerkin.

Here, the  $P_N P_M$  method is applied to improve the curvature calculation.

We compare the accuracy of the curvature calculation and the corresponding order of convergence for a spherical droplet geometry as shown in Table I. Two different approaches to compute the curvature are used: first is a DG lifting procedure based on the Bassi-Rebay 1 (BR1) method of Bassi and Rebay [33]. The lifting is necessary because of the discontinuous ansatz functions. The second approach is the  $P_N P_M$ -reconstruction as introduced in Section 3.4 with an additional reconstruction step prior to the level-set lifting. The curvature estimation method based on the  $P_N P_M$ -reconstruction has a higher convergence order compared with the DG lifting procedure because of the reconstruction step. This higher convergence order is reflected as well in lower discretization errors on the same mesh but implies larger computational costs.

The amount of induced parasitic currents is measured by the maximum  $\mathbb{L}_2$ -norm for the momentum terms shown in Table II. Because of numerical and modeling inaccuracies, some spurious velocities or currents are visible but are very small and do not falsify the steady state, even over long times.

### 5.3. Wobbling droplet

The third test case is an example of a surface tension driven flow. Here, an initially ellipsoidal-shaped droplet is considered that is not in the equilibrium state. Because of the surface tension force, the droplet starts to oscillate until it reaches its final spherical shape. The final state is a spherical droplet with a pressure drop fulfilling the Young–Laplace equation in (31).

The simulation parameters are chosen similar to the second test case, only the interface geometry is changed to an ellipsoid. The initial conditions are summarized in Figure 10. An initial ellipsoidal-shaped interface geometry is assumed with the semi-axis 0.72 in  $x$ -direction and 0.42 in both other directions. A box measuring  $[-1.4 \times 1.4]^3$  is used as computational domain; at the boundaries of the computational domain wall boundary, conditions are assumed. A mesh with  $40 \times 40 \times 40$  equidistant grid cells and a polynomial degree of  $N = 3$  is used.

The shape of the droplet during one oscillation period is visualized in Figure 11. Because of numerical dissipation (physical dissipation is not present as the Euler equations are considered), the oscillation amplitude diminishes with time and reaches finally a spherical state. Because of the low numerical diffusion of the used DG scheme, this takes a long time.



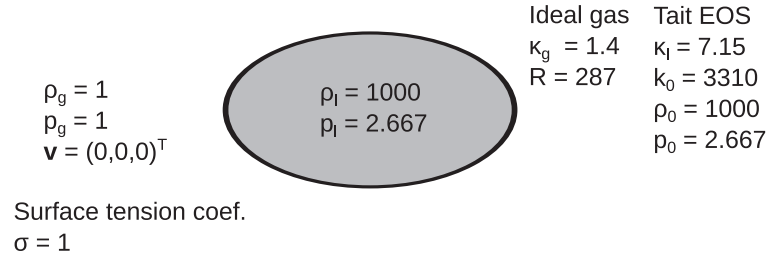


Figure 10. Initial conditions of the steady droplet with surface tension. The ellipsoidal semi-axis are chosen to 1.2 in  $x$ -direction and 0.7 in  $y$ -direction and  $z$ -direction. EOS, equation of state.

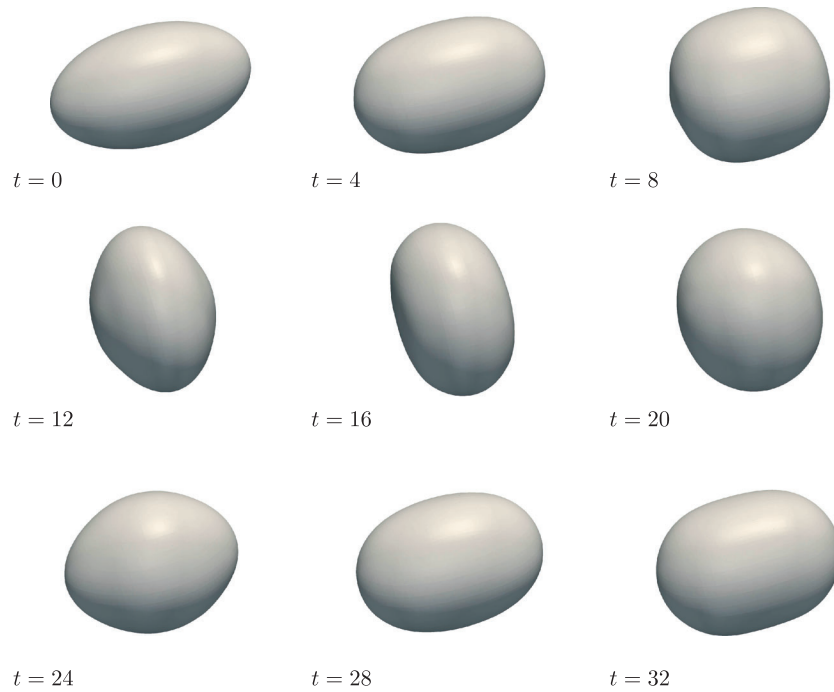


Figure 11. Three-dimensional surface contour (physical interface) for the oscillating droplet test case during one oscillation period. The numerically obtained oscillation frequency is  $f_2^{\text{num}} \approx 29$  s.

To be able to assess the quality of the oscillatory motion of the water droplet, the oscillation frequency is compared with the analytical formula derived by Lamb [34]. He found for the resonance mode frequency  $f_l$  of the  $l$ th oscillation mode

$$f_l^{\text{ana}} = \sqrt{\frac{\sigma l(l-1)(l+1)}{3\pi\rho_0 V}} \quad (35)$$

assuming a droplet with small oscillation amplitudes in vacuum or air.  $\rho_0$  denotes the density of the droplet medium and  $V$  the initial volume of the droplet. The amplified motion of the droplet corresponds to the mode  $l = 2$  (Rayleigh mode) and has an oscillation frequency of  $f_2^{\text{ana}} = 29$  s. The numerically obtained oscillation amplitude is plotted in Figure 12 in terms of the maximum amplitude in  $x$ -direction and  $y$ -direction (the  $z$ -direction is similar to the  $y$ -direction). It can be seen that sinusoidal oscillation is reached quickly after the initial deflection. The present high-order oscillation modes (higher harmonics) diminish quickly, and the Rayleigh frequency dominates the oscillation. The found oscillation frequency  $f_2^{\text{num}} \approx 29$  s is in good agreement to the analytical estimation of Lamb.

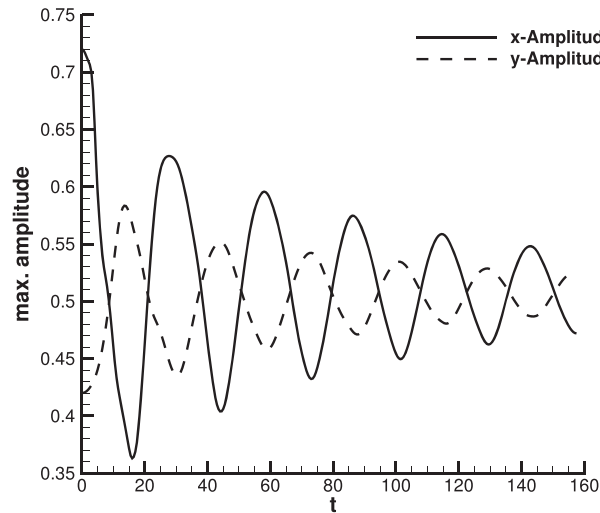


Figure 12. Plot of the maximum oscillation amplitude over time for an initially ellipsoidal droplet including surface tension effects. Note that the  $y$ -amplitudes and  $z$ -amplitudes are identical.

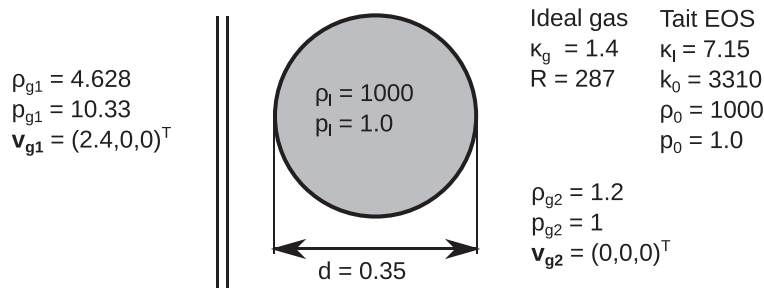


Figure 13. Initial conditions of the shock-droplet interaction test case. EOS, equation of state.

#### 5.4. Shock-droplet interaction

The fourth test considers a planar shock wave impinging on a spherical droplet. First, a 2D test case is considered as the published reference results consider a shock impinging onto a water column. In a second step, the test case is generalized to a fully 3D shock-droplet interaction.

The initial conditions of the pre-shock and post-shock states as well as the initial state of the droplet are given in Figure 13. At the domain boundaries in  $y$ -direction and  $z$ -direction, a wall boundary condition is assumed. The droplets initial position is  $(0.55, 0, 0)$ , the initial position of the planar  $M = 3$  shock is set to  $x = 0.35$ . The nondimensional parameters as described by Hu *et al.* [7] are used in this test case.

A Persson indicator [35] based on the density is used for shock-capturing purposes, which reliably detects the shock position. Depending on the indicator value, the time update is calculated using the more accurate DG scheme in smooth region or the TVD stable finite-volume scheme. The latter has the ability to cope with strong discontinuities and shocks and is therefore used in nonsmooth regions as found at the interface and at the shock front. A high-order DG scheme would not be able to resolve the strong shock involved in this case.

**5.4.1. Two-dimensional simulation.** The Cartesian mesh in the 2D case consists of  $90 \times 100$  cells, and the computational domain extends to  $[0, 1.2]$  in  $x$ -direction and  $[-0.7, 0, 7]$  in  $y$ -direction. In the third direction, one grid cell with periodic boundary conditions is used. The polynomial degree of the DG approximation is chosen to  $N = 3$ , and surface tension effects are not considered. The pressure jump of the impinging shock wave that dominates the pressure distribution and surface tension

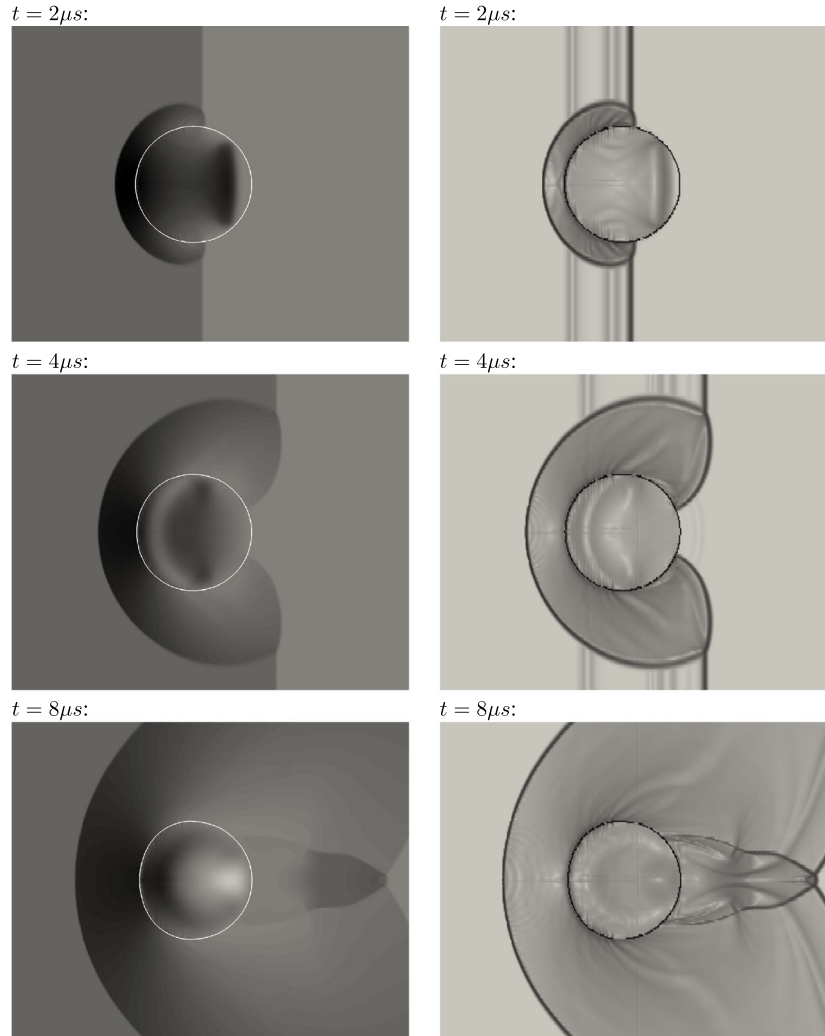


Figure 14. Result of a two-dimensional water column interacting with a planar shock at various time instances. Left: pressure contours in the range of  $-20$  to  $40$  atm. The solid white line indicates the interface position. Right: Schlieren-type image of the logarithmic density gradient  $\log(\nabla\rho + 1)$ .

effects have only a minor impact. The result is visualized in Figure 14 at various time instances, showing pressure contours and a visualization of the logarithmic density gradient (Schlieren-type visualization).

The impinging planar shock wave is partly reflected at the surface of the water column and partly enters the liquid phase. Inside the liquid, the shock travels at much higher speed because of the larger liquid sound velocity. At the curved surface, the planar shock wave gets reflected and distorted. With advancing simulation time, the interface geometry gets deformed because of the impacting  $M = 3$  flow. Within the liquid phase, negative pressure values are encountered due to the here considered simple EOS that does not account for physical effects like cavitation that would take place at low pressures. In reality, the water would vaporize, and a gas region is created.

The main features of the simulations appear physical and resemble the higher resolved results shown by Hu *et al.* [7] and Chang and Liou [36]. The deformation of the water column is in accordance with an experimental shadow-graph picture shown by Theofanous *et al.* [37] at the final simulation time of  $24 \mu\text{s}$  as well as a reference simulation of Hu *et al.* [7] (see Figure 15).

Because of the staircase-like approximation of the phase boundary, some disturbances directly next to the interface position are visible that diminish quickly away from the interface. The shock

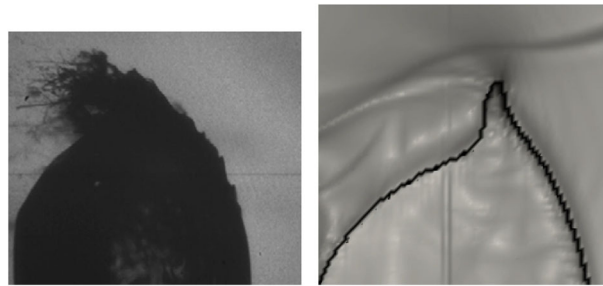


Figure 15. Schlieren photos for shock-droplet interaction problem. Left: Tributyl Phosphate droplet subjected to  $M = 3.0$  flow,  $We = 2500$  taken from Theofanous *et al.* [37]. Right: numerical result by water droplet with  $M = 3.0$  and  $We = \infty$ .

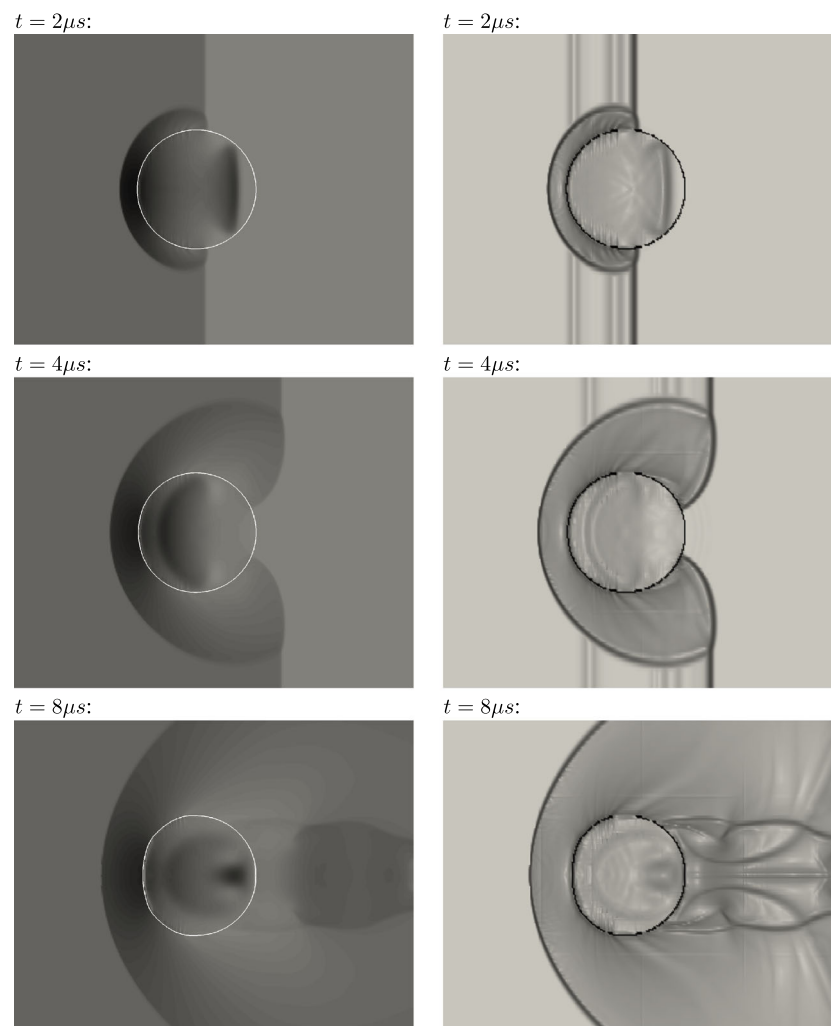


Figure 16. Result of a three-dimensional water droplet interacting with a planar shock at various time instances. Left: pressure contours in the range of  $-20$  to  $40$  atm. The solid white line indicates the interface position. Right: Schlieren-type image of the logarithmic density gradient  $\log(\nabla \rho + 1)$ .

front recovers quickly after passing the interface. The high-pressure region next to the stagnation point is clearly visible and not distorted by the staircase pattern approximation of the computational interface.

**5.4.2. Three-dimensional simulation.** The Cartesian mesh is bit coarser compared with the 2D case and consists of  $80 \times 90 \times 90$  cells. The computational domain extends to  $[0, 1.2]$  in  $x$ -direction and  $[-0.7, 0, 7]$  in  $y$ -direction and  $z$ -direction. Similarly to the boundary condition in  $y$ -direction, a wall boundary condition is assumed in  $z$ -direction.

The results, plotted on the droplet median plane, are shown in Figure 16 using a pressure and density gradient visualization, similar to the 2D simulation. The 3D results resemble the findings of the previous 2D investigation. Because of the spherical droplet geometry, the transmitted shock wave is slightly weaker behind the droplet. The impinging shock wave is reflected at the spherical interface resulting in a 3D shock structure in front and in the wake of the droplet.

Because of the reduced numerical resolution, the approximation of the droplet interface is a bit coarser and, hence, the induced numerical disturbances at the interface larger. The overall flow features in both calculations look alike because of the symmetric flow field (computational domain represents a shock tube as used in the experiments by Theofanous *et al.* [37]). This also influences the pressure distribution inside the droplet as more disturbances close to the interface are visible.

## 6. CONCLUSION

A numerical method for compressible two-phase flows using a sharp-interface method is described and applied in three dimensions. The present numerical approach allows for a high order of accuracy as well as efficient calculations. The high order is of advantage for the resolution of the interface as well as its curvature. These estimates are needed to include surface tension effects in the algorithm for the simulation of compressible two-phase flow.

Basic ideas are taken from the popular ghost-fluid approach and are adapted for the DG framework. The numerical fluxes at the computational interface are obtained consistently using the solution of two-phase Riemann problems. It is shown that the resolution of the numerical interface can be considerably improved through the use of a finite volume subcell refinement near the phase boundary. Extensions to realistic EOSs are possible without problems, the main drawback is the high additional computational effort.

The presented 3D test cases show the capabilities of the used solution algorithm for compressible fluids, and the results show good agreement to reference solutions, for example, by Hu *et al.* [7], using a 2D approach with a larger number of degrees of freedom.

## APPENDIX A: SURFACE TENSION MODELING WITH $P_N P_M$ RECONSTRUCTION

Surface tension acts as a pressure force on the interface between liquid and vapor phases. Its magnitude can be determined by the Young–Laplace law

$$(\rho(\mathbf{v} \cdot \mathbf{n}_{LS} - \mathbf{s}_{PB} \cdot \mathbf{n}_{LS})\mathbf{v} + p(\rho, T)\mathbf{n}_{LS}) = 2\sigma\kappa\mathbf{n}_{LS}. \quad (\text{A.1})$$

Input for the estimation of the surface tension pressure jump is the interface curvature (Section 3.4) and the surface tension coefficient  $\sigma$  that is considered to be constant and positive. Often the surface tension force is approximatively included by a reformulation to a volume source. Hence, the surface pressure term is smeared out over a small region next to the interface. For incompressible flow, such an approach is the often used continuous surface force model of Hirt and Nichols [15], which introduces source terms in the momentum equations. For compressible flow, such an approach is much more subtle, because the thermodynamic variables and the hydrodynamical variables are closely coupled, and a reformulation has to be found, which is consistent with the thermodynamics and introduces source terms in momentum and energy equation.

Our sharp-interface tracking based on the local solution at the interface allows in a quite natural way to take the surface tension into account. It is part of the generalized Riemann problem at the

phase interface. The Riemann problem takes the pressure jump due to surface tension forces into account, and the final fluxes automatically contain the corresponding pressure jump. The generalized jump conditions at the phase interface can be written as

$$\begin{aligned}\rho_L^* v_L^* - \rho_R^* v_R^* - S_M (\rho_L^* - \rho_R^*) &= 0, \\ \rho_L^* (v_L^*)^2 + p_L^* - \rho_R^* (v_R^*)^2 - p_R^* - S_M (\rho_L^* v_L^* - \rho_R^* v_R^*) &= (d-1)\sigma\kappa, \\ v_L^* (\rho_L^* e_L^* + p_L^*) - v_R^* (\rho_R^* e_R^* + p_R^*) - S_M (\rho_L^* e_L^* - \rho_R^* e_R^*) &= (d-1)\sigma\kappa S_M,\end{aligned}\tag{A.2}$$

with  $S_M$  being the velocity and  $U_L^*$  and  $U_R^*$  the states left and right of the phase interface. In case no surface tension forces are considered, the Rankine–Hugoniot equations are recovered.

The inclusion of surface tension effects heavily depends on an accurate estimation of the interface curvature  $\kappa$ . As described in Section 3.4, the accuracy of the level-set method is enlarged by a posteriori  $P_N P_M$ -reconstruction. This reconstructed polynomial is in a second step used to estimate the interface curvature  $\kappa$ , the second derivative of the level-set polynomial. Because of the reconstruction step, the derivatives can be obtained by direct differentiation. As a final step, the curvature is projected back onto a polynomial representation of order  $N + 1$  and then onto the finite volume subcells. Because of the locality of the operator, the reconstruction and projection are carried out individually for each DG grid cell. This defines a global curvature field in the computational domain. An improvement is to only estimate the curvature in a narrow band around the interface.

As the interface curvature has only a meaningful value at the interface position, the polynomial curvature distribution is evaluated at the position of the phase interface ( $\Phi = 0$ ). As the level-set function is assumed to be linear within the narrow band around the interface (Equation (18)), the curvature value at the interface can be estimated using linear interpolation at the numerical interface. This procedure allows for an accurate estimation of the interface curvature. The gain in using a polynomial evaluation of the curvature at the interface  $\Phi = 0$  is not that high as it would justify the expensive numerical operation.

#### ACKNOWLEDGEMENTS

The authors gratefully acknowledge a number of discussions with Christian Rohde, Christoph Zeiler, and Veronika Schleper. This research was supported by the German Research Foundation (DFG) through SFB TRR 75 ‘Droplet dynamics under extreme ambient conditions’.

#### REFERENCES

1. Saurel R, Petitpas F, Abgrall R. Modelling phase transition in metastable liquids: application to cavitating and flashing flows. *Journal of Fluid Mechanics* 2008; **607**(1):313–350.
2. Anderson D, McFadden GB, Wheeler A. Diffuse-interface methods in fluid mechanics. *Annual Review of Fluid Mechanics* 1998; **30**(1):139–165.
3. Nguyen VT, Peraire J, Khoo BC, Persson PO. A discontinuous Galerkin front tracking method for two-phase flows with surface tension. *Computers & Fluids* 2010; **39**(1):1–14.
4. Hu XY, Khoo B, Adams N, Huang F. A conservative interface method for compressible flows. *Journal of Computational Physics* 2006; **219**(2):553–578.
5. Fedkiw RP, Aslam T, Merriman B, Osher S. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Computational Physics* 1999; **152**(2):457–492.
6. Merkle C, Rohde C. The sharp-interface approach for fluids with phase change: Riemann problems and ghost fluid techniques. *ESAIM: Mathematical Modelling and Numerical Analysis* 2007; **41**(6):1089–1123.
7. Hu XY, Adams NA, Iaccarino G. On the HLLC Riemann solver for interface interaction in compressible multi-fluid flow. *Journal of Computational Physics* 2009; **228**(17):6572–6589.
8. Fechter S, Jaegle F, Schleper V. Exact and approximate Riemann solvers at phase boundaries. *Computers & Fluids* 2013; **75**:112–126.
9. Kopriva DA. *Implementing Spectral Methods for Partial Differential Equations*. Springer Media B. V., Houten: Houten, 2009.
10. Sussman M, Smereka P, Osher S. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics* 1994; **114**(1):146–159.
11. Toulil H, Hussaini M, Sussman M. Tracking discontinuities in hyperbolic conservation laws with spectral accuracy. *Journal of Computational Physics* 2007; **225**(2):1810–1826.

12. Marchandise E, Remacle JF. A stabilized finite element method using a discontinuous level set approach for solving two phase incompressible flows. *Journal of Computational Physics* 2006; **219**(2):780–800.
13. Ferrari A, Munz C-D, Weigand B. A high order sharp-interface method with local time stepping for compressible multiphase flows. *Communications in Computational Physics* 2010; **9**:205–230.
14. Liu TG, Khoo BC, Yeo KS. Ghost fluid method for strong shock impacting on material interface. *Journal of Computational Physics* 2003; **190**(2):651–681.
15. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**(1):201–225.
16. Hindenlang F, Gassner GJ, Altmann C, Beck A, Staudenmaier M, Munz C-D. Explicit discontinuous Galerkin methods for unsteady problems. *Computers & Fluids* 2012; **61**:86–93.
17. Kopriva DA, Gassner GJ. On the quadrature and weak form choices in collocation type discontinuous Galerkin spectral element methods. *Journal of Scientific Computing* 2010; **44**:136–155.
18. Kopriva DA. Metric identities and the discontinuous spectral element method on curvilinear meshes. *Journal of Scientific Computing* 2006; **26**(3):301–327.
19. Toro EF. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer-Verlag: Berlin, 1999.
20. Dumbser M, Iben U, Munz C-D. Efficient implementation of high order unstructured WENO schemes for cavitating flows. *Computers & Fluids* 2013; **86**:141–168.
21. Sonntag M, Munz C-D. Shock capturing for discontinuous Galerkin methods using finite volume subcells. In *Finite Volumes for Complex Applications VII-Elliptic, Parabolic and Hyperbolic Problems*. Springer: Berlin, 2014; 945–953.
22. Dumbser M, Zanotti O, Loubère R, Diot S. A Posteriori Subcell Limiting of the Discontinuous Galerkin Finite Element Method for Hyperbolic Conservation Laws. *Journal of Computational Physics* 2014; **278**:47–75.
23. Sethian J, Smereka P. Level set methods for fluid interfaces. *Annual Review of Fluid Mechanics* 2003; **35**(1):341–372.
24. Grooss J, Hesthaven J. A level set discontinuous Galerkin method for free surface flows. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**(25–28):3406–3429.
25. Dumbser M. Arbitrary high order  $P_N P_M$  schemes on unstructured meshes for the compressible Navier–Stokes equations. *Computers & Fluids* 2010; **39**(1):60–76.
26. Jiang GS, Peng D. Weighted ENO schemes for Hamilton–Jacobi equations. *SIAM Journal on Scientific Computing* 2000; **21**(6):2126–2143.
27. Adalsteinsson D, Sethian J. The fast construction of extension velocities in level set methods. *Journal of Computational Physics* 1999; **148**(1):2–22.
28. Rohde C, Zeiler C. A relaxation Riemann solver for compressible two-phase flow with phase transition and surface tension. *Applied Numerical Mathematics* 2014. DOI: 10.1016/j.apnum.2014.05.001.
29. Chalons C, Coquel F, Engel P, Rohde C. Fast relaxation solvers for hyperbolic-elliptic phase transition problems. *SIAM Journal on Scientific Computing* 2012; **34**(3):A1753–A1776.
30. Jaegle F, Rohde C, Zeiler C. A multiscale method for compressible liquid-vapor flow with surface tension. *ESAIM: Proceedings*, Vol. 38, Marseille, 2012; 387–408.
31. Brackbill JU, Kothe DB, Zemach C. A continuum method for modeling surface tension. *Journal of Computational Physics* 1992; **100**(2):335–354.
32. Dressel A, Rohde C. A finite-volume approach to liquid-vapour fluids with phase transition. In *Finite Volumes for Complex Applications V*. Wiley: Aussois (France), 2008; 53–68.
33. Bassi F, Rebay S. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *Journal of Computational Physics* 1997; **131**:267–279.
34. Lamb H. *Hydrodynamics*. Cambridge University Press: Cambridge, 1932.
35. Persson PO, Peraire J. Sub-cell shock capturing for discontinuous Galerkin methods. *Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, Reno, Nevada, 2006.
36. Chang CH, Liou MS. A robust and accurate approach to computing compressible multiphase flow: stratified flow model and AUSM<sup>+</sup>-up scheme. *Journal of Computational Physics* 2007; **225**(1):840–873.
37. Theofanous T, Li G, Dinh TN, Chang C. Aerobreakup in disturbed subsonic and supersonic flow fields. *Journal of Fluid Mechanics* 2007; **593**:131–170.