

## Direct numerical simulation of compressible multiphase flow using a Discontinuous Galerkin based multiscale approach

F. Jaegle\*, S. Fechter, M. Boger, C.-D. Munz  
Institute of Aerodynamics and Gas Dynamics, University of Stuttgart, Germany  
{Jaegle,Fechter,Boger,Munz}@iag.uni-stuttgart.de

### Abstract

In this study, a numerical method for the simulation of compressible two-phase flows is presented. The goal is the direct simulation (using a free surface technique) of droplet processes under extreme conditions as they are encountered, for example, when injecting liquid fuels in rocket or scramjet engines. We present several important components for the numerical approach, including the high-order discontinuous Galerkin spectral element method (DGSEM) and a sharp interface approach specifically tailored for this type of scheme. The method is capable of taking general equations of state (EOS) into account as well as including surface tension. The full method is presented for the three-dimensional case including basic reference examples. The capabilities of this approach are presented in a shock-droplet interaction.

---

### Introduction

In many technical applications, multiphase flows meet conditions such as high pressures and/or high velocities that prohibit the popular assumption of incompressibility. The most important examples are fuel injection systems of aeronautical, automotive and rocket engines. The numerical simulation of compressible multiphase flow presents additional difficulties to the numerical approach, compared to the incompressible treatment that is most often found in two-phase numerical solvers in practical use today.

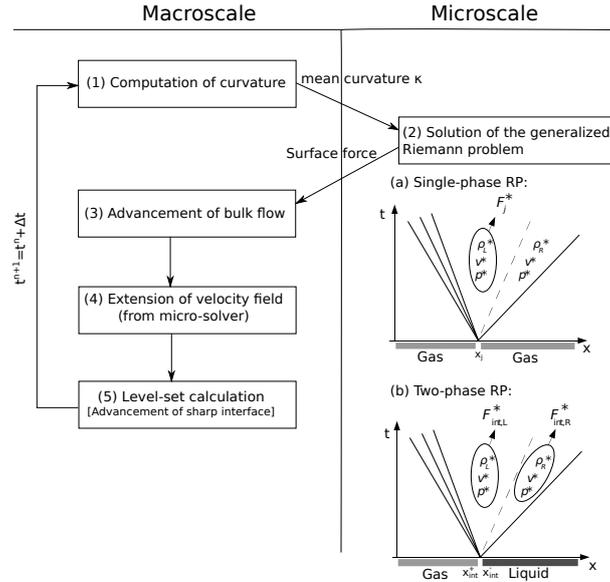
Two elements are crucial for the numerical simulation of two-phase flows: the first is a method that allows to define the geometry of the interface between the two phases as well as its movement depending on the flow field. The most widely used approaches are the volume-of-fluid method [1], where an additional variable corresponds to the volume fraction of one phase inside a given control volume, and the level-set method [2], where the interface is implicitly given as the zero-level of a variable that is initialized as a distance function.

The second element is a framework to establish the coupling between the two phases across the interface. This latter aspect is of particular importance for compressible flows as the coupling effects through the different equations of state (EOS) contain more complex physics than in the incompressible case. The widespread diffuse-interface approach allows the smearing out of the interface over several grid cells. In standard compressible formulations, this leads to intermediate states where the different pressure laws cannot simply be averaged, which would cause spurious pressure oscillations. In literature, several approaches are described that avoid this difficulty using special formulations for the intermediate EOS [3, 4]. The so-called sharp interface approach is an alternative coupling method that avoids this problem by devising special numerical procedures that allow to keep the discontinuities at the interface sharp and the EOS definitions separate. A well-known method of this type is the so-called ghost fluid formalism of Fedkiw et al. [5], developed for finite volume schemes, where fictitious fluid states are introduced near the interface, similar to the well-known concept of ghost-cells at boundaries. This method has been continuously refined, e.g. by Liu et al. [6], Hu et al. [7] and Ferrari et al. [8].

In this paper we present a sharp interface method based on a Discontinuous Galerkin spectral element method (DGSEM) that allows for a high order of accuracy as well as efficient calculations. The Discontinuous Galerkin (DG) formulation allows for discontinuities at the element interfaces, which simplifies the implementation of the sharp interface approach as only numerical fluxes have to be adapted. A further advantage is that it is straightforward to include the effects of surface tension into the algorithm. The high-order approximation of the level-set allows for a curvature calculation at the phase boundary. This information is needed to include surface tension in the algorithm. The current implementation is only capable to resolve the physical phase boundary at the nearest cell boundary. Due to the comparably large DG cells the numerical approximation of the phase boundary is very coarse. One approach to improve the phase boundary resolution is to use an Adaptive Mesh Refinement at the phase boundaries. This approach shows promising results for future investigations.

The paper is organized as follows: in the first section, the numerical method is described followed by an outline of the information transfer between macro- and micro-scale model. In the next section results are presented and in the last section a short conclusion is shown.

\*Corresponding author: felix.jaegle@iag.uni-stuttgart.de



**Figure 1.** HMM algorithm used for the simulation of compressible multiphase flows.

### The numerical method

In this paper we present a novel multi-scale method for the calculation of compressible multiphase flows. To be able to include interface phenomena such as curvature into the algorithm, a heterogeneous multiscale method (HMM) is chosen. We consider the compressible Navier-Stokes equations as macro-scale model. The solution for the macro-scale model is provided by a Discontinuous Galerkin spectral element method (DGSEM) as developed in [9]. The micro-scale solver may be based on different concepts such as Riemann solvers or well-resolved 1D simulations. In the present study, we use a HLLC-type Riemann solver [7]. The effect of the Young-Laplace equation

$$(\rho(\mathbf{v} \cdot \mathbf{n} - s)\mathbf{v} + p(\rho)\mathbf{n}) = (d-1)\gamma\kappa\mathbf{n}, \quad (1)$$

describing the surface tension at the phase boundary enters the HMM as micro-scale model. With  $\kappa$  we denote the mean curvature of the level-set approximation associated with the orientation of the mean normal  $\mathbf{n}$ . The surface tension coefficient  $\gamma$  is assumed to be constant and positive. This problem can be handled in a generalized Riemann problem with non-homogeneous jump conditions. An accurate polynomial description of the phase interface is provided by the level-set dividing the cells such that a cell is either in the gaseous or in the liquid phase. The physical interface is described by the root of the level-set polynomial while the computational interface described by the surface dividing the cells that are predominately occupied by either the gas or the liquid phase.

The macro-scale solver is independent of the used EOS and different EOS can be applied for the different phases. In the present case we used different EOS for the different parts of the fluid. For the gaseous part we applied the ideal gas law whereas we used the weakly compressible Tait EOS to model the liquid phase. Note that the micro-scale solver is generally not independent of the EOS formulation. Multiphase Riemann solver formulations that satisfy this property for the case without phase change are available [10]. In the following we will describe the basic steps of the algorithm, that is visualized in figure 1.

Step 1: Computation of the discrete curvature at the faces of the computational interface.

Step 2: Solution of the generalized Riemann problem (micro-solver) at the surface integration points. At the numerical interface a generalized Riemann problem is solved that takes the local curvature into account. The initial conditions are provided by the approximate solutions for the liquid and gas phase. The solution of the micro-solver contains the phase boundary and is moving with the local normal interface speed  $s$ .

For the bulk phases a standard one-phase Riemann solver is used that takes the corresponding EOS into account.

Step 3: The explicit DGSEM scheme is used to advance the bulk flow to the next time level  $t^{n+1}$ . At the computational interface the respective flux contributions for the different phases from step 2 are used.

Step 4: The local normal interface speed  $s$  defined at the computational interface is extended to a global velocity field. This velocity field is used to advance the physical interface represented by the level-set variable.

Step 5: The new position of level-set zero is used to determine the new physical and computational interface. In case the interface has moved across a grid cell, the new state is extrapolated using adjacent grid cells.

### **The Macro-Scale Solver: a Discontinuous Galerkin spectral element method**

In this section we describe the the macro-scale solver used in Step 3 and Step 4 of the HMM Algorithm that is visualized in figure 1. In both cases we rely on a DGSEM. The description of the method is kept brief, for more details, we refer to [9, 11].

The key properties of the method are: the three-dimensional domain is divided into non-overlapping hexahedral elements, each mapped onto a reference cube element  $E := (-1, 1)^3$  by a mapping  $\mathbf{x}(\boldsymbol{\zeta})$ . The equations are solved in the reference element. The weak solution vector  $\mathbf{U}$  of a general system of conservation laws of the type

$$\mathbf{U}_t + \operatorname{div} \mathbf{F}(\mathbf{U}) = \mathbf{0} \quad (2)$$

with a flux  $\mathbf{F}(\mathbf{U}) = (\mathbf{F}^1(\mathbf{U}), \mathbf{F}^2(\mathbf{U}), \mathbf{F}^3(\mathbf{U}))^T$  is approximated by a tensor-product basis

$$\mathbf{U}(\boldsymbol{\zeta}) = \sum_{i,j,k=0}^N \hat{\mathbf{U}}_{ijk} \psi_{ijk}(\boldsymbol{\zeta}), \quad \psi_{ijk}(\boldsymbol{\zeta}) = l_i(\zeta^1) l_j(\zeta^2) l_k(\zeta^3), \quad (3)$$

where  $l_j(\zeta)$  are 1D Lagrange polynomials of degree  $N$  defined as:

$$l_j(\zeta) = \prod_{\substack{i=0 \\ i \neq j}}^N \frac{\zeta - \zeta_i}{\zeta_j - \zeta_i}, \quad j = 0, \dots, N. \quad (4)$$

Multiplying (2) with a test function  $\phi$  and integration by parts of the second term leads to three contributions: a volume integral of the time derivative term (a), a surface integral term (b) and a volume integral term (c), which now contains the gradient of the test function  $\phi$

$$\underbrace{\frac{\partial}{\partial t} \int_E \mathbf{J} \mathbf{U} \phi d\boldsymbol{\zeta}}_a + \underbrace{\int_{\partial E} (\mathbf{F}^* \cdot \mathbf{N}) \phi dS}_b - \underbrace{\int_E \mathbf{F}(\mathbf{U}) \cdot \operatorname{grad}(\phi) d\boldsymbol{\zeta}}_c = 0. \quad (5)$$

In this expression,  $\mathbf{J}$  is the Jacobian of the transformation to the reference cube element.

Volume as well as surface integrals are approximated by Gauss-Legendre or Gauss-Legendre-Lobatto quadrature. Quadrature points coincide with the interpolation points for the ansatz functions  $l_j$ . As no continuity constraint is enforced between the elements, the flux function  $\mathbf{F}(\mathbf{U})$  at the cell boundaries is replaced by a numerical flux function  $\mathbf{F}^*(\mathbf{U}^-, \mathbf{U}^+)$  depending on the left and right adjacent states  $\mathbf{U}^-$  and  $\mathbf{U}^+$ . The approximation (3) is inserted into the weak form (5) and the test functions are chosen as the ansatz functions  $\phi = \psi_{ijk}$ . Integrals are split in coordinate directions and approximated by Gauss-Legendre (-Lobatto) quadrature, which introduces the integration weights  $\omega_i$ ,  $i = 0, \dots, N$ . As the quadrature points are the same as the interpolation points of  $l_j$ , the Lagrange property  $l_j(\zeta_i) = \delta_{ij}$ ;  $i, j = 0, \dots, N$  can be exploited.

Plugging all the ansatz functions into the weak form, make use of the orthogonality of the ansatz functions and introduction of numerical integration yields the final semi-discrete form of the DGSEM scheme:

$$\begin{aligned} \left( \hat{\mathbf{U}}_{ijk} \right)_t = & -(\mathbf{J}_{ijk})^{-1} \left[ -\sum_{\lambda=0}^N \frac{\omega_\lambda}{\omega_i} D_{i\lambda} \mathbf{F}_{\lambda jk}^1 - \sum_{\mu=0}^N \frac{\omega_\mu}{\omega_j} D_{j\mu} \mathbf{F}_{i\mu k}^2 - \sum_{\nu=0}^N \frac{\omega_\nu}{\omega_k} D_{k\nu} \mathbf{F}_{ij\nu}^3 \right. \\ & + \left( [\mathbf{F}^* \hat{\mathbf{s}}]_{jk}^{+\zeta^1} \frac{l_i(1)}{\omega_i} - [\mathbf{F}^* \hat{\mathbf{s}}]_{jk}^{-\zeta^1} \frac{l_i(-1)}{\omega_i} \right) + \left( [\mathbf{F}^* \hat{\mathbf{s}}]_{ik}^{+\zeta^2} \frac{l_j(1)}{\omega_j} - [\mathbf{F}^* \hat{\mathbf{s}}]_{ik}^{-\zeta^2} \frac{l_j(-1)}{\omega_j} \right) \\ & \left. + \left( [\mathbf{F}^* \hat{\mathbf{s}}]_{ij}^{+\zeta^3} \frac{l_k(1)}{\omega_k} - [\mathbf{F}^* \hat{\mathbf{s}}]_{ij}^{-\zeta^3} \frac{l_k(-1)}{\omega_k} \right) \right]. \end{aligned}$$

The numerical fluxes  $\mathbf{F}^*$  are evaluated at the reference element faces in each coordinate direction. These terms are denoted by  $\square^{-\zeta^1}$  and  $\square^{+\zeta^1}$  for the left and right face in  $\zeta^1$ -direction and analogously for  $\zeta^2$  and  $\zeta^3$ . With  $D_{ij} = dl_j(\zeta)/d\zeta|_{\zeta=\zeta_i}$  a differentiation matrix is denoted, which is needed for the evaluation of the volume integral. The numerical integration weights for the Gaussian integration are symbolized with  $\omega$ . Time integration relies on an explicit third-order Runge-Kutta scheme.

### **The Discontinuous Galerkin spectral element method for the 3D level-set transport**

The HMM approach used in this study relies on accurate geometrical information (position, curvature) of the phase boundary, which necessitates a suitable tracking formalism. To obtain this information we use a level-set formulation: an additional function  $\Phi$  is initialized as a signed distance function with respect to the interface. It is then advected by a velocity field  $\mathbf{s}$ . The advection equation for the level-set reads:

$$\frac{\partial \Phi}{\partial t} + \mathbf{s} \cdot \text{grad}(\Phi) = 0 \text{ in } \Omega \times (0, T). \quad (6)$$

Of course, this is only a conservation law as long as  $\text{div}(\mathbf{s}) = 0$ , which will not hold in our case. To discretize (6) also with the DGSEM scheme, equation (6) is re-cast as a conservation law with an additional right-hand-side containing the divergence of the level-set advection speed  $\mathbf{s}$ :

$$\frac{\partial \Phi}{\partial t} + \text{div}(\mathbf{s}\Phi) = \Phi \text{div}(\mathbf{s}). \quad (7)$$

Note that the divergence term can be readily obtained in the DGSEM by evaluating the necessary derivatives of the high-order ansatz polynomials.

The level-set advection speed  $\mathbf{s}$  may be chosen equal to the fluid velocity in the absence of phase transition. In a more general framework, an artificial smooth velocity field that locally coincides with the normal propagation speed of the interface at the level-set zero has to be generated [12].

### **Adaptive Mesh Refinement**

A difficulty arising with the used DG approach is that the numerical resolution of the phase interface is coarse and does not reflect the physical approximation of the phase boundary. This is a consequence of the comparatively large DG cells as the DG cells have an in-cell resolution. In the DG framework discontinuities are only allowed at the cell interfaces which implies that a phase boundary coupling is only possible at a cell boundary.

One approach to avoid this disadvantage of the DG scheme next to the phase boundary is to use an adaptive mesh refinement (AMR). In the surrounding of the phase boundary, that is detected using a level-set based indicator, the mesh is refined using a first-order finite volume method which has approximately the same number of degrees of freedom. This implies that one DG cell is refined into multiple finite-volume cells. At the resulting interface between the polynomial DG cells and the refined finite volume cells a conservative projection/reconstruction method is used to ensure consistency at the interface. This approach is used for the conservation variables as these quantities may be discontinuous across a phase interface. The level-set variable is always treated with the DG method as for the calculation of the curvature a high order solution is necessary. The input for the level-set, the advection velocity field, is interpolated onto the DG integration points.

This AMR approach combines both the high order resolution of a DG scheme as well as the fine resolution of discontinuities of a finite volume scheme.

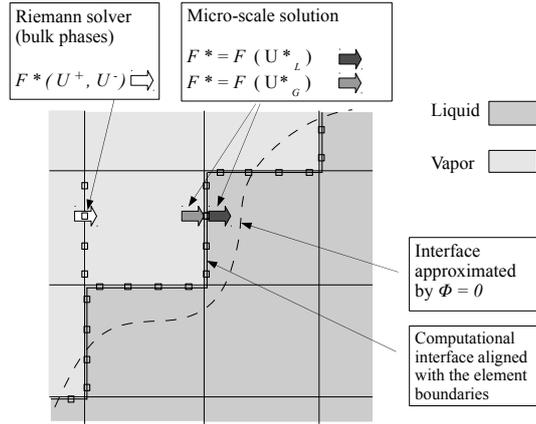
### **Micro-scale solver**

The micro-solver used here is an approximate HLLC Riemann solver [7] modified to take surface tension into account. It is employed at the computational interface. The density discontinuity across a phase interface is resolved using the wave pattern of the contact discontinuity. In order to include surface tension in this approximate Riemann solver, the pressure can be discontinuous across the contact discontinuity. The pressure drop magnitude is determined by the Young-Laplace law (1) using the mean curvature of the level-set that is determined in the macro-scale solver (step 1).

The contact discontinuity is the standard choice to resolve the phase interface as across the contact discontinuity no mass transfer is occurring. Furthermore, this discontinuity represents well the occurrence in multiphase flows as typically only the densities of the different fluids vary in some order of magnitudes. The velocity of contact discontinuity is taken as a model for the propagation speed of the interface  $\mathbf{s}$ .

### **Transfer of information between macro-scale and micro-scale solver**

The transfer of information from the micro- to the macro-solver relies on the numerical fluxes at the computational interface as illustrated in Fig. 2. In the bulk phases away from the computational interface, the DGSEM incorporates a standard HLLC Riemann solver to obtain the numerical flux  $F^*$ . For the element faces that form the computational interface, the micro-scale solver is used instead. Let  $U_L := U^-$  and  $U_G := U^+$  be the respective states in the two different phases in some quadrature point. These states are the input for the micro-scale solver (together with the curvature). The micro-scale solver delivers for each quadrature point a liquid state  $U_L^*$  and a



**Figure 2.** Schematic of a typical setting in the HMM approach, involving the liquid-vapor interface, approximated by the zero level-set  $\Phi = 0$ , the computational interface and the different ways to apply the numerical fluxes provided by either a standard Riemann solver (bulk phase) or the micro-scale solver at the computational interface.

gaseous state  $U_G^*$ . The DGSEM now uses the flux evaluations  $F^* = F(U_L^*)$  for the computation on the liquid element and  $F^* = F(U_G^*)$  for the computation on the gaseous element. This method of calculating two different fluxes for each phase results in a solution where variables such as the density remain discontinuous across the (computational) interface and the typical numerical smearing of discontinuities in numerical schemes is avoided. For piecewise constant approximations the method is similar to the ghostfluid method [5, 13], where two different fluxes are obtained via additional ghost states near the interface.

In this study, we include the effect of surface tension, which necessitates two elements: the first is the evaluation of the local curvature of the interface, the second is the application of the resulting surface force on the flow. In the literature, a common way to do the latter is to replace the surface force by a distribution of a volume source term, e.g. the continuum surface force method of Brackbill and Kothe [14]. In the HMM approach of the present study, the effect of surface tension is entirely handled in the micro-scale solver. This way, the surface tension force is exerted on the macro-scale through the fluxes at the computational interface. Inversely, it is necessary to provide the micro-scale with information on the mean curvature  $\kappa$ , which is obtained from the divergence of the level-set normal:

$$\kappa = \operatorname{div} \left( \frac{\operatorname{grad}(\Phi)}{|\operatorname{grad}(\Phi)|} \right). \quad (8)$$

This quantity is only meaningful at the zero level-set, which in our approach does not coincide with the computational interface, where the micro-solvers are located. Therefore, for each micro-solver, the curvature value to be used as an initial condition is obtained through a 1D search in a direction normal to the cell boundary until the nearest level-set zero is found.

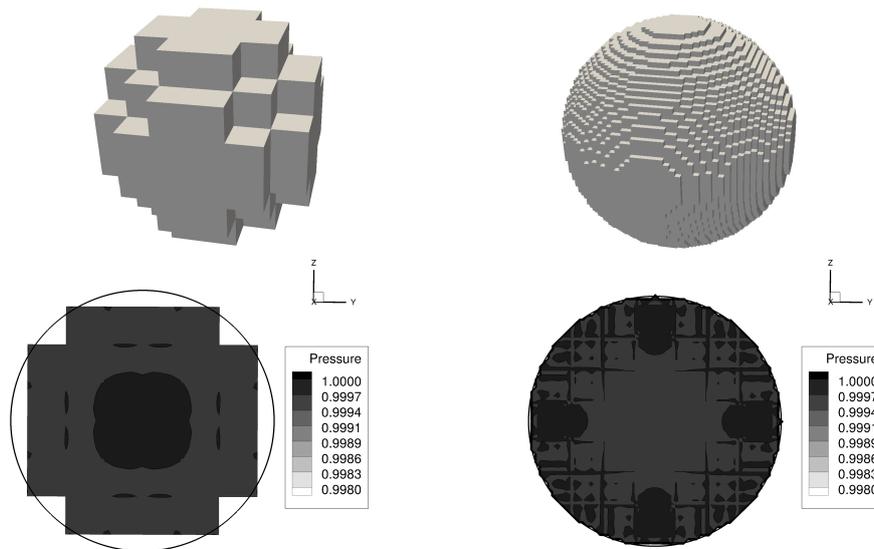
## Results

To illustrate the performance of the method, three test cases with increasing complexity are presented. All cases consider three-dimensional, spherical droplets.

### *Steady droplet with surface tension*

The first test is a simulation of a steady-state droplet with surface tension. The droplet is initialized at a slightly higher pressure than the surrounding gas (see table 1 for a summary of the initial conditions). The resulting pressure difference matches the pressure jump resulting from surface tension at the given droplet diameter, therefore leading to a steady-state if the calculation of the curvature and the resulting micro-scale solution are accurate.

The simulation has been conducted over a dimensionless time of  $t = 0.3$ . A Weber number based on this time period and the diameter of  $d = 0.3 \text{ m}$  is of the order of  $10^6$ . The cartesian grid is very coarse, consisting only of  $12 \times 12 \times 12$  cells, the polynomial degree of the DG scheme has been chosen as  $N = 6$ . Figure 3 shows pressure information extracted on the mid-plane of the droplet at  $t = 0.15$ . The jump in the pressure is visible at the cell boundaries near the level-set zero. The constant states on the gaseous side has been perfectly conserved while



**Figure 3.** Result of the steady droplet test case with surface tension. On top the density iso-contours for the case without AMR (left) and AMR (right) is shown. Below the corresponding pressure solution is shown, while the black line visualizes the physical phase boundary.

Droplet radius	Inner state	Outer state
$r_{\text{ini}} = 0.4$	$\rho = 1000 \quad  \mathbf{v}  = 0.0 \quad p = 1.0$	$\rho = 1.0 \quad  \mathbf{v}  = 0.0 \quad p = 0.996365$
Surface tension coeff.	Tait EOS parameters inner state	Ideal gas EOS parameters outer state
$\gamma = 0.000727$	$\kappa_L = 7.15, k_0 = 3310, \rho_0 = 1000, p_0 = 1.0$	$\kappa = 1.4$

**Table 1.** Initial conditions of the steady droplet with surface tension.

minor fluctuations have developed on the liquid side. The corresponding result is shown for the case without the use of AMR at the phase boundary and with AMR. The resolution of the physical interface is much better using AMR especially for a high order of accuracy. In the AMR case minor fluctuations due to the stiff Tait EOS occur next to the interface that might be related to the postprocessing of the results.

#### ***Droplet interacting with concentric spherical waves***

The interaction of a liquid-gaseous interface with waves is one of the more interesting phenomena for compressible two-phase flow. In order to provide a quantitative assessment of such an interaction, a spherically symmetric test case is considered first. This way, the solution of the full 3D solver can be compared to a well-resolved result of a 1D code in spherically symmetric formulation. The cartesian grid consists of  $50 \times 50 \times 50$  cells, the polynomial degree is  $N = 2$ . The 1D reference solution is obtained on a grid with 200 cells and  $N = 2$ . The droplet is initialized with a zero velocity field. A pressure difference between the inner and outer regions of the droplet is applied such that the droplet is out of equilibrium (table 2). This leads to a rarefaction propagating into the droplet while a weak shock moves outwards from the interface as shown in Fig. 4. The same figure also shows the shape of the zero level-set, which remains perfectly spherical. For a quantitative analysis, the comparison with the well-resolved 1D code is shown in Fig. 5. The 3D result matches the reference data well, the main visible discrepancy is due to the much lower resolution that leads to smeared-out discontinuities. Another minor difference is the different position of the density jump (i.e. the computational interface). As this position must coincide with the nearest grid cell boundary, it is slightly shifted for the coarse grid of the 3D simulation. The pressure jump due to surface tension matches very well, even though curvature information in the 3D case comes from the derivation of the DG polynomials as opposed to a direct evaluation from the droplet diameter in the 1D result.

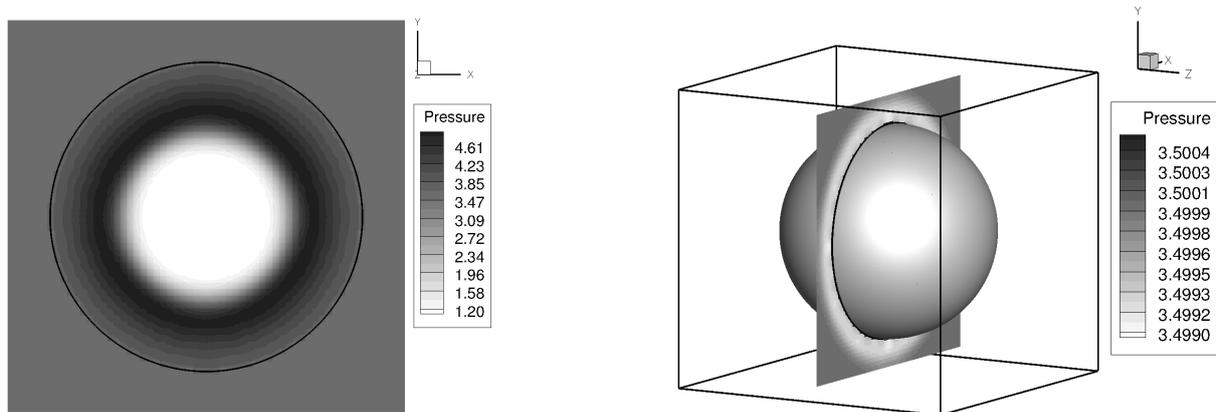
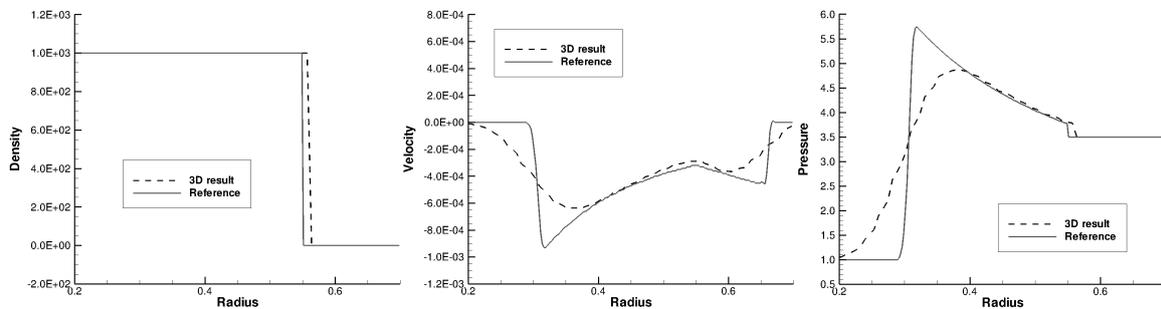
#### ***Droplet interacting with a planar shock wave***

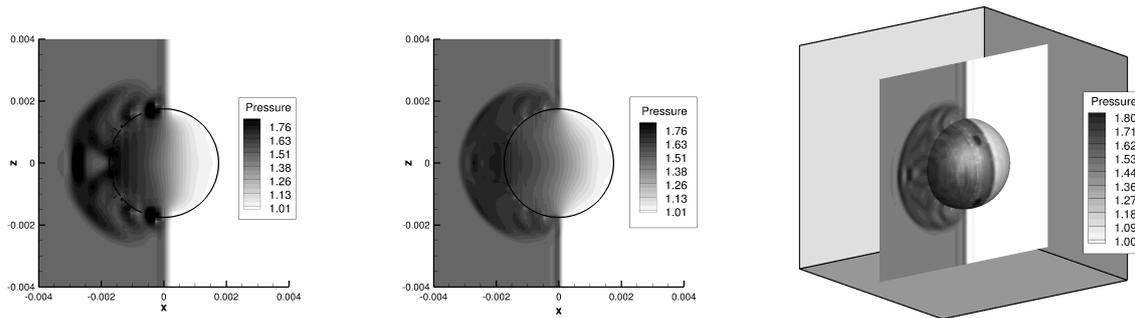
The third test considers a planar shock wave impinging on a spherical droplet. The initial conditions are given in table 3. The cartesian grid consists of  $52 \times 52 \times 52$  cells, the polynomial degree is  $N = 2$ . Surface tension is not considered in this simulation. The result is shown in Fig. 6, showing pressure contours inside the droplet

Droplet radius	Inner state	Outer state
$r_{\text{ini}} = 0.4$	$\rho = 1000$ $ \mathbf{v}  = 0.0$ $p = 1.0$	$\rho = 1.0$ $ \mathbf{v}  = 0.0$ $p = 3.5$
Surface tension coeff.	Tait EOS parameters inner state	Ideal gas EOS parameters outer state
$\gamma = 0.0727$	$\kappa_L = 7.15$ , $k_0 = 3310$ , $\rho_0 = 1000$ , $p_0 = 1.0$	$\kappa = 1.4$

**Table 2.** Initial conditions for the droplet interacting with concentric spherical waves.

Droplet radius	Inner state	Outer state (pre/post-shock)
$r_{\text{ini}} = 0.00175$	$\rho = 1000$ $ \mathbf{v}  = 0.0$ $p = 1.0$	$\rho = 1.0$ $ \mathbf{v}  = 0.0$ $p = 1.0$
		$\rho = 1.32575$ $\mathbf{v} = (0.346215, 0, 0)$ $p = 1.48783$
	EOS parameters inner state	EOS parameters outer state
	$\kappa_L = 7.15$ , $k_0 = 3310$ , $\rho_0 = 1000$ , $p_0 = 1.0$	$\kappa = 1.4$

**Table 3.** Initial conditions for the droplet interacting with a planar shock wave.**Figure 4.** Result for a droplet with concentric spherical waves. **Left:** pressure contours on the mid-plane of the droplet, range adapted to highlight the rarefaction inside the droplet. **Right:** iso-surface of the level-set zero with pressure contours on the mid-plane of the droplet. Range adapted to highlight the shock outside the droplet.**Figure 5.** Result for a droplet with concentric spherical waves. Comparison between a well-resolved 1D result (3rd order accurate, 200 grid cells) under the assumption of spherical symmetry and data from the 3D simulation extracted along a line in radial direction.



**Figure 6.** Result of a droplet interacting with a planar shock. **Left:** pressure contours on the droplet median plane on the coarse grid ( $30 \times 30 \times 30$  cells). **Middle:** pressure contours on the droplet median plane on the fine grid ( $50 \times 50 \times 50$  cells). **Right:** pressure contours on the droplet median plane as well as on the zero-level set isosurface.

on the median plane as well as outside in the gaseous region and on the zero-level set isosurface. The shock is partly reflected on the droplet, the other part enters into the liquid, where it travels at much higher speed and has therefore been reflected at and distorted by the curved surface of the droplet. The main features of the simulations appear as physical and resemble the (higher resolved but 2D) results shown by Hu et al. [7]. Unlike in the previous test, the staircase pattern of the computational interface appears as a visible perturbation in the result. While the reflected shock front quickly regains a smooth shape after clearing the angular computational interface, the airflow following the shock leads to unphysical fluctuations when passing over the jagged regions, revealing the limits of the method without the use of AMR. For coarser grid resolutions the staircase pattern becomes a more severe disturbance in the flow leading to an unphysical flow distribution around the droplet. For finer resolutions the flow approaches the solution of Hu et al. in the limit.

### Summary and Conclusions

A numerical method for compressible two-phase flows using a sharp interface method is described and applied in 3D. 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. 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 correctly using the solution of two-phase Riemann problems.

The presented 3D test cases show the capabilities of the used solution algorithm for compressible fluids. A simplified one-dimensional comparison of the solution shows a good agreement. It is shown that the solution can be improved using an adaptive mesh refinement together with a high-order DG scheme. The whole methodology is able to cope with break-up of liquids but is not tested yet. Further investigations on this topic have to be done.

### Acknowledgements

The authors gratefully acknowledge the support by the German Research Foundation (DFG) through SFB TRR 75 “Tropfdynamische Prozesse unter extremen Umgebungsbedingungen” as well as the Carl Zeiss Foundation.

### References

- [1] C. W. Hirt, B. D. Nichols, *J. Comput. Phys.* 39 (1) (1981) 201–225.
- [2] S. Osher, J. A. Sethian, *J. Comput. Phys.* 79 (1) (1988) 12–49.
- [3] R. Saurel, R. Abgrall, *J. Sci. Comput.* 21 (3) (2000) 1115–1145.
- [4] E. H. van Brummelen, B. Koren, *J. Comput. Phys.* 185 (1) (2003) 289–308.
- [5] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, *J. Comput. Phys.* 152 (2) (1999) 457–492.
- [6] T. G. Liu, B. C. Khoo, K. S. Yeo, *J. Comput. Phys.* 190 (2) (2003) 651–681.
- [7] X. Hu, N. Adams, G. Iaccarino, *J. Comput. Phys.* 228 (17) (2009) 6572–6589.
- [8] A. Ferrari, C.-D. Munz, B. Weigand, *Commun. Comput. Phys.* 9 (2010) 205–230.
- [9] D. A. Kopriva, G. Gassner, *J. Sci. Comput.* 44 (2010) 136–155.
- [10] F. Jaegle, V. Schleper, *Preprint 2012*.
- [11] D. Kopriva, *Implementing Spectral Methods for Partial Differential Equations* (2009) 293–354.
- [12] F. Jaegle, C. Zeiler, C. Rohde, *Preprint 2012*.
- [13] C. Merkle, C. Rohde, *M2AN Math. Model. Numer. Anal.* 41 (6) (2007) 1089–1123.
- [14] J. Brackbill, D. Kothe, C. Zemach, *Journal of computational physics* 100 (2) (1992) 335–354.