Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 108C (2017) 1763–1772

Procedia Computer Science

International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland

Large Scale Simulation of Cloud Cavitation Collapse

U. Rasthofer, F. Wermelinger, P. Hadijdoukas, and P. Koumoutsakos

Computational Science and Engineering Laboratory, ETH Zurich, Switzerland
{urasthofer,fabianw,chatzidp,petros}@ethz.ch

Abstract

We present a high performance computing framework for large scale simulation of compressible multicomponent flows, applied to cloud cavitation collapse. The governing equations are discretized by a Godunov-type finite volume method on a uniform structured grid. The bubble interface is captured by a diffuse interface method and treated as a mixing region of the liquid and gas phases. The framework is based on our Cubism library which enables the efficient treatment of high-order compact stencil schemes that can harness the capabilities of massively parallel computer architectures and allows for processing up to 10^{13} computational cells. We present validations of our approach on several classical benchmark examples and study the collapse of a cloud of $\mathcal{O}(10^3)$ bubbles.

© 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the International Conference on Computational Science

Keywords: compressible multicomponent flow, high performance computing, diffuse interface method, bubble collapse, cloud cavitation

1 Introduction

Cavitation entails the rapid growth of vapor cavities in a liquid in regions of low pressure, followed by their violent collapse when transported to areas of higher external pressure. Cavitation causes material erosion on nearby surfaces, thus considerably reduces the expected lifespan of applications in marine propulsion, turbomachinery or fuel injection engines; see, e.g., [25]. Cavitating bubbles usually grow as a cloud which amplifies the destructive potential compared to the single bubble case. The relative location of the bubbles in a cloud and their proximity to other bubbles influences highly the bubble collapse process. Non-spherical bubble collapses oriented towards the center of the cloud evolve. Bubble collapses start at the periphery of the cloud and then progressively approach its center. Energy accumulates due to the earlier collapse of neighboring bubbles and is eventually released in the form of a strong pressure wave emerging from the center of the cloud. The violence of the process has challenged experimental works (see, e.g., [4, 6, 30]) as it is often associated with damage to the measurement devices. Computational studies (see, e.g., [1, 27]) of cloud cavitation collapse aim at complementing experiments and at providing detailed quantitative insight of the process.

We develop methods and software that enable simulations of cavitating bubble clouds at unprecedented scales. Such simulations can provide detailed information about the fundamental

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the International Conference on Computational Science
10.1016/j.procs.2017.05.158

process of bubble-bubble interactions and generate data that can assist the formulation of effective engineering models. In this context, our simulation goals involve clouds with thousands of resolved cavities, that is, two orders of magnitude larger than the reported state of the art, e.g., in [27].

Such simulations require an accurate treatment of the interface region as well as highly efficient flow solvers. Diffuse interface methods for compressible multicomponent flow (see, e.g., [2, 22, 24, 26]) introduce an artificial zone around the interface where the transition from one component to the other takes place in a smooth fashion. Therefore, these methods enable an adequate compromise between accuracy and computational efficiency. In the preceding study [21], we presented a compressible multicomponent flow solver, named Cubism-MPCF, with smoothed interface that is capable of processing grids with trillions of cells.

In this work, we investigate the performance of diffusive interface methods for compressible multicomponent flow and incorporate them into our open-source software Cubism-MPCF. We elaborate on the benefits in terms of the resolution requirements when properly treating the diffuse interface zone as a mixture of a liquid and a gas and not only as an artificial transition region from one component to the other. We find that this is of paramount importance for achieving the targeted bubble counts. Finally, we report on our findings from simulations of cloud cavitation collapse with $\mathcal{O}(10^3)$ cavities.

The paper is structured as follows: The governing equations and their discretization are introduced in Section 2. Section 3 provides an overview of our highly optimized flow solver Cubism-MPCF. Benchmark examples for validation as well as results from a collapsing cloud of 2500 bubbles are presented in Section 4. We conclude our work in Section 5.

2 Computational Method

2.1 Governing Equations

The governing equation system, derived from the Baer-Nunziato model [3], reads as

$$\frac{\partial \alpha_1 \rho_1}{\partial t} + \nabla \cdot (\alpha_1 \rho_1 \mathbf{u}) = 0, \tag{1}$$

$$\frac{\partial \alpha_2 \rho_2}{\partial t} + \nabla \cdot (\alpha_2 \rho_2 \mathbf{u}) = 0, \tag{2}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + p \mathbf{I}) = \mathbf{0}, \tag{3}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot ((E + p)\, \mathbf{u}) = 0, \tag{4}$$

$$\frac{\partial \alpha_2}{\partial t} + \mathbf{u} \cdot \nabla \alpha_2 = K \nabla \cdot \mathbf{u}, \tag{5}$$

where

$$K = \frac{\alpha_1 \alpha_2 (\rho_1 c_1^2 - \rho_2 c_2^2)}{\alpha_1 \rho_2 c_2^2 + \alpha_2 \rho_1 c_1^2}; \tag{6}$$

see e.g. [17, 18] for derivation. This system of equations comprises two mass conservation equations, one for each component, conservation equations for momentum and total energy in mixture- (or single-)fluid formulation and a transport equation for the volume fraction of one of the two components with source/sink term on the right-hand side, referred to as "K-div term" in the following. In Equations (1)–(5), $\mathbf{u}$ denotes the velocity, p the pressure, $\mathbf{I}$ the identity

tensor, ρ the (mixture) density, E the (mixture) total energy $E = \rho e + 1/2\rho(\mathbf{u} \cdot \mathbf{u})$, where e is the (mixture) specific internal energy. Moreover, ρ_k, α_k and c_k with $k \in \{1, 2\}$ are density, volume fraction and speed of sound of the two components. It holds that $\alpha_1 + \alpha_2 = 1$ as well as $\rho = \alpha_1\rho_1 + \alpha_2\rho_2$ and $\rho e = \alpha_1\rho_1 e_1 + \alpha_2\rho_2 e_2$ for the mixture quantities.

The K-div term, originally derived in [14] for homogeneous mixtures, describes the reduction of the gas volume fraction in a mixture of gas and liquid when a compression wave travels across the mixing region. For an expansion wave, it recovers the increase of the gas volume fraction. The K-div term is non-zero only in the interface zone and frequently neglected (see, e.g., [2, 7, 18]). However, treating the interface zone indeed as mixture of gas and liquid and not only as an artificial blending region from one component to the other is important for properly capturing the dynamics of collapsing bubbles, as already indicated by the results shown in [26]. Furthermore, this term also allows for dynamically creating interfaces, corresponding to the generation of gas pockets in low pressure regions; see, e.g., [24].

The system of equations is closed by an appropriate equation of state for each of the phases. To capture liquids and gases, the stiffened equation of state (see, e.g., [16]), which enables a simple, analytic approximation to arbitrary fluids, is applied:

$$p = (\gamma_k - 1)\,\rho_k e_k - \gamma_k p_{\mathrm{c},k}, \tag{7}$$

where isobaric closure is assumed; see, e.g., [18]. Parameters γ_k and $p_{\mathrm{c},k}$ depend on the material. For $p_{\mathrm{c},k} = 0\,\mathrm{Pa}$ and γ_k being the ratio of specific heats, the equation of state for ideal gases is recovered. Unless otherwise specified, $\gamma_1 = 6.59$ and $p_{\mathrm{c},1} = 4.069 \cdot 10^8\,\mathrm{Pa}$ are used for water and $\gamma_2 = 1.4$ and $p_{\mathrm{c},2} = 0\,\mathrm{Pa}$ for air.

2.2 Numerical Method

System (1)–(5) is discretized using the method of lines. Our formulation builds on a finite volume method for uniform structured grids, where spatial operators are approximated using high-order stencil schemes. The approach yields a system of ordinary differential equations

$$\frac{d\mathbf{V}(t)}{dt} = \mathcal{L}(\mathbf{V}(t)), \tag{8}$$

where $\mathbf{V}$ is a vector of cell average values and $\mathcal{L}(\cdot)$ is a discrete operator that approximates the fluxes and the source term in the governing system. The temporal discretization of Equation (8) is obtained by an explicit third-order low-storage Runge-Kutta scheme proposed in [29]. The computation of the numerical fluxes is based on a Godunov-type scheme using the approximate HLLC Riemann solver introduced for single-phase flow in [28]. The Riemann initial states are determined by a shock capturing fifth-order WENO reconstruction (see [12]). Following [13], the reconstruction is carried out using primitive variables, and the HLLC Riemann solver is adapted to Equation (5) to prevent oscillations at the interface. The solution is advanced with a time-step size that satisfies the Courant-Friedrichs-Lewy condition. For the weighting coefficients of the Runge-Kutta stages, the values suggested in [9] are used, resulting in a total variation diminishing scheme.

3 Cubism-MPCF

The computational method presented in the previous section is implemented into Cubmis-MPCF. Cubism-MPCF was presented in its original form in [21] and later further developed

in [10, 11]. The most important features of Cubmis-MPCF are summarized in the following, additional details about design considerations can be found in the aforementioned references.

We define the computational kernels RHS, DT and UP for the processing of the full discretization. RHS stands for "right-hand side" and is responsible for the evaluation of the approximate Riemann problem. DT determines the admissible time-step size based on a global reduction. Finally, UP performs the Runge-Kutta update for a given set of weighting coefficients. The computational cost distribution of RHS, DT and UP is 90 %, 2 % and 8 %, respectively (neglecting I/O operations). The Cubism library provides a framework for the efficient treatment of high-order compact stencil schemes (e.g., RHS) as well as straightforward point-wise operations, such as DT and UP. It is a lightweight C++ template library that can be used for implementing any type of solver that relies on uniform structured grids. The software is open-source and can be downloaded from our lab repository[1]. Cubism-MPCF is a highly optimized compressible solver for multicomponent flows, based on the governing system described in Section 2.1. It is written in C++ and built on top of the Cubism library. The solver is open-source and available for download from our repository[2].

3.1 Hierarchical Data Structure based on Static Size Blocks

Cubism-MPCF uses a two-level hierarchical data structure utilizing uniformly spaced cubic blocks of static size. Data within the blocks is stored in an array of structures (AoS) format on a per cell basis. The computational domain is composed by a set of blocks arranged on a Cartesian topology, maintained by the Cubism library. The static size of the blocks is chosen such that cache utilization is maximized on multicore architectures. The hierarchical structure allows for increased locality of the data, which is a main concern for stencil operations. The vector $\mathbf{V}$ in Equation (8) inherits this hierarchical structure. A temporary buffer for the low-storage Runge-Kutta scheme additionally doubles the memory footprint for each block. Figure 1 shows the hierarchical data structure used in Cubism-MPCF. The circled block represents an entity that is accessed through the Cubism library, where the gray shaded region corresponds to the temporary storage required by the Runge-Kutta scheme. At the lowest level, each cell holds a vector of averaged conserved quantities.

3.2 Parallelization and Optimizations

The solver is parallelized with a hybrid paradigm using the MPI and OpenMP programming models. The software is split into three abstraction layers to separate specific optimizations, increase reusability of the code and allow for more efficient prototyping of new simulation cases. The realization of the domain decomposition and the inter-rank communication is accomplished on the cluster layer. The constant size subdomains are organized on a Cartesian processor topology. The cluster layer is responsible for the dispatch of the grid blocks to the node layer. The performance optimization techniques used for the block processing on the cluster layer are described in [10]. The thread level parallelism (TLP) is exploited on the node layer using the OpenMP standard. We employ dynamic work scheduling and a parallel granularity of one block per thread to hide potential load imbalances during the processing of the dispatched blocks. Each thread exclusively works on one block at a time. The block data and ghosts, required for the evaluation of the stencil at block boundaries, are loaded into a per-thread work buffer. The intra-rank ghosts are obtained from loading fractions of the surrounding blocks,

[1] `https://gitlab.ethz.ch/mavt-cse/Cubism`
[2] `https://gitlab.ethz.ch/mavt-cse/Cubism-MPCF`

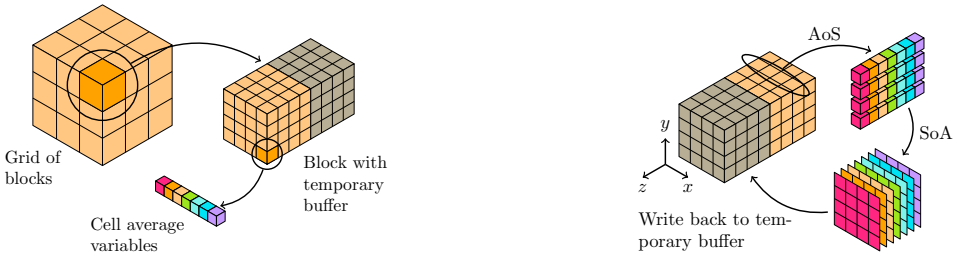

Figure 1: Hierarchical data structure used in Cubism-MPCF (left). Conversion from AoS to SoA format on the core layer (right). The figure is adapted from [21].

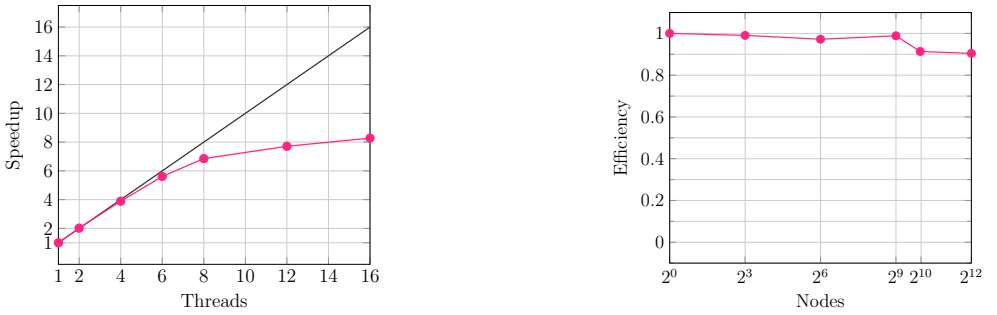

Figure 2: Strong scaling on a single node (left). Weak scaling across 4096 nodes (right).

whereas inter-rank ghosts are obtained from a global buffer. The computational kernels are executed on the core layer, which are called from the executing thread on the node. The core layer exploits data level parallelism (DLP) and instruction level parallelism (ILP) by relying on explicit vectorization and code fusion techniques reported in [10, 21]. In order to apply these techniques, the core layer converts the AoS layout into a structure of arrays (SoA) by converting the data into slices arranged on a ring buffer. Figure 1 illustrates this conversion. One ring buffer is required for each element in the vector of conserved variables. The core layer implements the RHS, DT and UP kernels introduced above and is the most crucial part in terms of performance. More details on software design regarding the parallelization strategy used in Cubism-MPCF can be found in [21].

3.3 Strong and Weak Scaling

We evaluate the performance of our solver on the Piz Daint supercomputer at Swiss National Supercomputing Centre (CSCS). The machine consists of 5272 Cray XC30 compute nodes, each equipped with an Intel Xeon E5-2670 8-core SandyBridge CPU and an Nvidia Tesla K20X GPU. The executable code has been generated with the GNU C++ compiler (v4.8). Figure 2 shows the detailed strong scaling on a single node with hyperthreading enabled. The size of the cubic blocks is set to 32 cells along each edge. Our code exhibits excellent strong scaling up to 8 cores. The additional gain due to hyperthreading is small because of the high per-thread cache utilization. The Craypat profiler reports a 95.3 % hit ratio for the LD1 cache and 99.5 % for the LD1 and LD2 caches combined. Figure 2 displays the weak efficiency across 4096 nodes. The work load on each node is composed of 4096 blocks, resulting in a memory footprint of

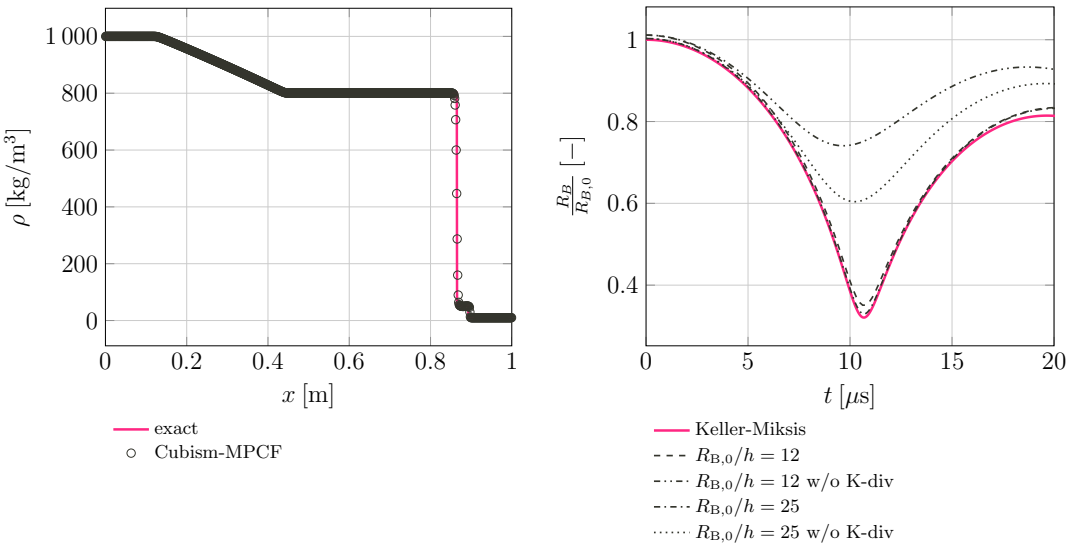

Figure 3: Density of liquid-gas shock tube using 800 cells (left). Bubble radius R_B of single-bubble collapse at $p_\infty/p_{B,0} = 10$ (right).

8 GB per node. The time to solution for this configuration is 18.3 s (one full time step). The code achieves excellent weak efficiency on the (almost) full machine. The loss in efficiency is due to the collective MPI all-reduce operation required to synchronize the admissible time-step size after the DT kernel finished. The average time required by the block processing algorithm for each stage of the third-order Runge-Kutta scheme is 6.05 s and 6.13 s for 1 and 4096 nodes, respectively. The timings correspond to one evaluation of the RHS kernel on the whole domain. We achieve this almost perfect compute/transfer overlap by employing the methods explained in [10].

4 Numerical Examples

4.1 Liquid-Gas Shock Tube

A 1D two-component shock tube, as considered, e.g., in [8], is examined first. The initial conditions in the domain $\Omega = [0, 1]$ m read as

$$(\rho, u, p, \alpha_2)_0 = \begin{cases} (1000.0\,\mathrm{kg/m^3}, 0\,\mathrm{m/s}, 1.0 \cdot 10^9\,\mathrm{Pa}, 0) & \text{for } 0\,\mathrm{m} \leq x \leq 0.75\,\mathrm{m}, \\ (10.0\,\mathrm{kg/m^3}, 0\,\mathrm{m/s}, 1.0 \cdot 10^5\,\mathrm{Pa}, 1) & \text{for } 0.75\,\mathrm{m} < x \leq 1.0\,\mathrm{m}. \end{cases}$$

The parameters of the stiffened-gas equation of state are set to $\gamma_1 = 4.4$ and $p_{c,1} = 6.0 \cdot 10^8$ Pa for the liquid in the left part and to $\gamma_2 = 1.4$ and $p_{c,2} = 0$ Pa for the gas in the right part. Results are evaluated at time $t = 0.234$ ms, comparing them to the curves obtained from an exact two-component Riemann solver. Figure 3 exemplarily depicts the solution for the density obtained without K-div term, as it vanishes for this case due to $\nabla \cdot \mathbf{u} = 0$ at the interface. All results are in good agreement with the exact solution.

4.2 Single-Bubble Collapse

An isolated air bubble collapsing in water due to a prescribed jump between the initial pressure in the interior of the bubble, $p_{\mathrm{B},0}$, and the far-field, p_∞, is investigated next. A pressure ratio of $p_\infty/p_{\mathrm{B},0} = 10$ is considered. Initially, a zero-velocity field is assumed. The density of water amounts to $\rho_{1,0} = 1000\,\mathrm{kg/m^3}$ and of air to $\rho_{2,0} = 1\,\mathrm{kg/m^3}$. The initial radius $R_{\mathrm{B},0}$ of the bubble is set to 1 mm. The initial pressure and gas fraction field are given by

$$\alpha_{2,0} = \frac{1}{2}\left[1 - \tanh\left(\frac{r - R_{\mathrm{B},0}}{1.5h}\right)\right] \text{ and } p_0 = \begin{cases} p_{\mathrm{B},0} & \text{if } 0 \leq r \leq R_{\mathrm{B},0}, \\ p_\infty + \frac{R_{\mathrm{B},0}}{r}\left(p_{\mathrm{B},0} - p_\infty\right) & \text{otherwise,} \end{cases}$$

where h denotes the cell length and r the distance between a point $\mathbf{x}$ of the domain and the center of the bubble; see [26]. The solution of the Keller-Miksis equation [15], which constitutes an ordinary differential equation for $R_{\mathrm{B}}(t)$ in the weakly compressible case, serves as a reference for our investigations. The bubble radius from our 3D simulations is obtained based on the gas volume fraction α_2 via $R_{\mathrm{B}} = \sqrt[3]{3/(4\pi)\int \alpha_2 \mathrm{d}\Omega}$.

Exploiting the symmetry of the problem, the computation domain is a cube of size $\Omega = [0, 15R_{\mathrm{B},0}]^3$, which contains one eighth of the bubble centered at the origin. Symmetry boundary conditions are imposed at the three faces of the cube intersected by the interface, while zeroth-order absorbing boundary conditions are assumed for the remaining faces. A grid refinement study is provided in Figure 3. The initial bubble radius is resolved by approximately 12 cells for the coarser grid and by 25 cells for the finer one. When using the finer grid, differences between the 3D simulation and the solution of the Keller-Miksis equation are only marginal. Already with the coarser grid, the curve from the 3D simulation is close to the reference data. In addition, results obtained with a formulation neglecting the K-div term, which is also frequently used in literature for compressible multiphase flow (see, e.g., [7, 18, 21]), are included. A pronounced positive impact of the K-div term is observed for both resolutions. For the rather coarse resolutions used here, the minimum radius is reached at an earlier time and deviates significantly from the solution of the Keller-Miksis equation when the K-div term is omitted. Although convergence towards the reference curve can be stated, significantly higher resolutions would be required if the K-div term is omitted. Closer investigations, not shown here, have revealed a thickening of the interface, when the K-div term is not considered, leading to the time shift in the collapse process. Moreover, it appears to "stiffen" the system such that a further compression of the bubble is prevented, and large deviations from the expected minimum radius are observed; see Figure 3. The stiffened behavior may be traced back to the simple advection of α_2. However, to ensure a correct thermodynamic behavior (see [23]), the gas content α_2 has to adapt to pressure changes in the mixture. Exactly this behavior is recovered by the K-div term. Therefore, the K-div term has to be considered for the present applications.

4.3 Cloud Cavitation Collapse

In the following, results from a collapsing cloud comprising 2500 gas bubbles are presented. For the setup of the cloud, bubbles are randomly positioned within a sphere of radius $R_{\mathrm{C}} = 3.5\,\mathrm{mm}$. The radius of the bubbles is also chosen at random from the interval $[0.10; 0.15]$ mm such that the average bubble radius amounts to 0.11 mm. While the positions of the bubbles are obtained from a uniform distribution, a log-normal distribution is used for their radii. The considered cloud contains a gas content of 8.4%, resulting in a cloud interaction parameter of $\beta = 85$ (see, e.g., [5] for a definition). The computational domain $\Omega = [0; 10]^3\,\mathrm{mm^3}$ is discretized by 1024^3

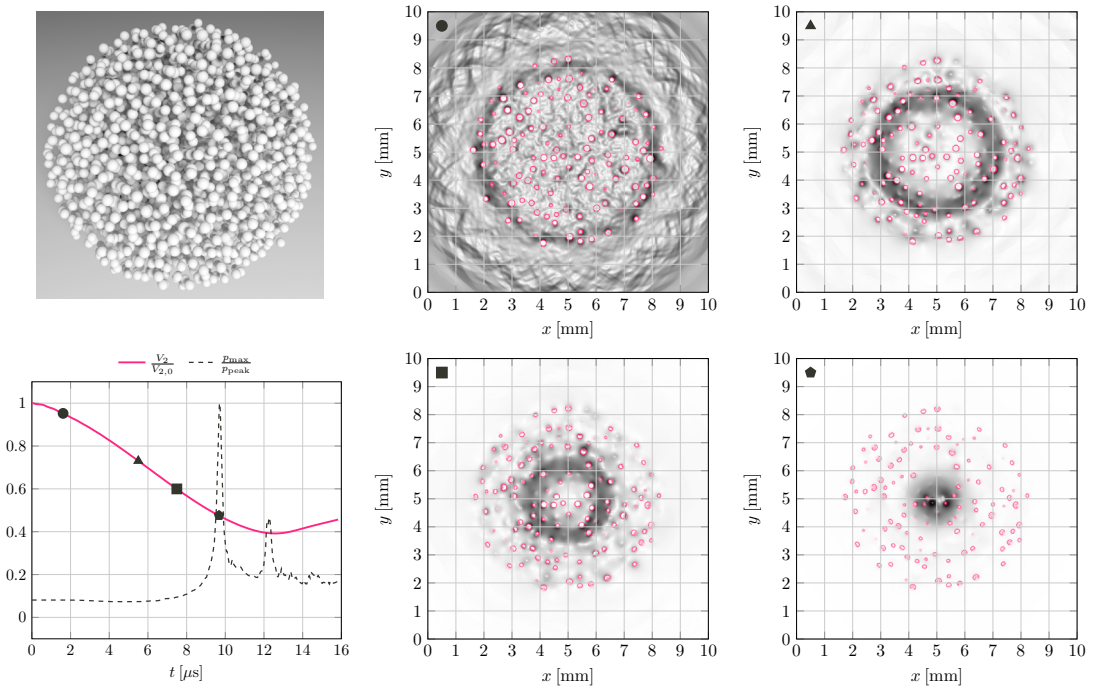

Figure 4: Initial configuration of 2500 bubble cloud (top left). Normalized gas volume of cloud and maximum pressure versus time (bottom left). The remaining four figures illustrate (from left to right and from top to bottom) schlieren visualizations of the pressure gradient (monochrome color) and the bubble interfaces (red color) at the indicated time instants.

cubic grid cells. Hence, the bubble radii are resolved by 10 to 15 cells. At the boundaries, non-reflecting, characteristic-based boundary conditions are applied; see, e.g., [19]. Initially, the bubble pressure is set to $p_{\mathrm{B},0} = 45\,\mathrm{bar}$. The ambient pressure is set to $p_\infty = 100\,\mathrm{bar}$, and the initial pressure field is determined as explained in [27]. Pressurized air with $\rho_{2,0} = 11.352\,\mathrm{kg/m^3}$ is assumed for the gas in the bubble and water with $\rho_{1,0} = 1000\,\mathrm{kg/m^3}$ for the surrounding liquid.

Figure 4 displays the initial cloud, the evolution of the gas volume V_2 and the maximum pressure $p_{\max}$ as well as numerical schlieren images (see [20]) from the center plane of the cloud. Consistent with the experimental findings in [30], the peak of the maximum pressure appears several micro-seconds before the minimum volume is reached. The schlieren images show the formation of a spherical pressure wave, in accordance with the symmetry of the configuration. While the pressure wave propagates through the cloud, it increases in strength such that the highest pressures are observed in the core of the cloud. The pressure wave is accompanied by a subsequent strong deformation of the bubbles. The deformation of the bubbles is caused by inward-pointing micro-jets due to bubble-bubble interactions. As the imposed pressure ratio is rather low, the strength of the individual bubble collapses is relatively weak. Together with the relatively high β, we encounter a regime of weak collapse strength, but strong bubble interactions. As a result, most of the energy of the pressure jump is transferred into kinetic energy via the micro-jets. Owing to the strong micro-jets, all bubbles get pierced during the collapse process.

5 Conclusions

We have presented a solver capable of preforming the largest ever simulations of cloud cavitation collapse. We use a diffuse interface method to capture the interface on a uniform structured, fixed grid. The computational approach is based on a Godunov-type finite volume method and has been implemented into our open-source software Cubism-MPCF. The distinct feature of this code is the Cubism library which provides a framework for the efficient treatment of high-order compact stencil schemes as well as point-wise operations. We have validated our approach for two classical benchmark examples, a liquid-gas shock tube and a single-bubble collapse. Using the latter test case, we have demonstrated the importance of recovering the compression and expansion of the gas in the interface zone when applying smoothed interfaces to compressible flow problems. Eventually, we have investigated a cloud of 2500 collapsing bubbles, which has highlighted the high potential of the proposed approach and solver.

Ongoing and future research includes the investigation of collapsing clouds of $\mathcal{O}(10^4)$ gas bubbles, a comparative study with respect to approaches based on bubble-particle models, studies of uncertainty quantification as well as developments towards software for hybrid compute architectures.

Acknowledgements. An award of computer time was provided by the Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program under the project CloudPredict. This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357. This work was also supported by a grant from the Swiss National Supercomputing Centre (CSCS) under project ID s500. Provided computational resources are gratefully acknowledged.

References

[1] N. A. Adams and S. J. Schmidt. Shocks in cavitating flows. In *Bubble Dynamics and Shock Waves*, pages 235–256. Springer Nature, 2013.

[2] G. Allaire, S. Clerc, and S. Kokh. A five-equation model for the simulation of interfaces between compressible fluids. *Journal of Computational Physics*, 181:577–616, 2002.

[3] M. Baer and J. Nunziato. A two-phase mixture theory for the deflagration-to-detonation transition (DDT) in reactive granular materials. *International journal of multiphase flow*, 12:861–889, 1986.

[4] N. Bremond, M. Arora, C.-D. Ohl, and D. Lohse. Controlled multibubble surface cavitation. *Physical Review Letters*, 96, 2006.

[5] C. Brennen, G. Reisman, and Y.-C. Wang. Shock waves in cloud cavitation. Twenty-First Symposium on Naval Hydrodynamics, 1997.

[6] E. A. Brujan, T. Ikeda, and Y. Matsumoto. Shock wave emission from a cloud of bubbles. *Soft Matter*, 8:5777, 2012.

[7] V. Coralic and T. Colonius. Finite-volume WENO scheme for viscous compressible multicomponent flows. *Journal of Computational Physics*, 274:95–121, 2014.

[8] N. Favrie, S. Gavrilyuk, B. Nkonga, and R. Saurel. Sharpening diffuse interfaces with compressible flow solvers. *Open Journal of Fluid Dynamics*, 4:44–68, 2014.

[9] S. Gottlieb and C.-W. Shu. Total variation diminishing Runge-Kutta schemes. *Mathematics of Computation of the American Mathematical Society*, 67:73–85, 1998.

[10] P. E. Hadjidoukas, D. Rossinelli, B. Hejazialhosseini, and P. Koumoutsakos. From 11 to 14.4 PFLOPs: performance optimization for finite volume flow solver. In *Proceedings of the 3rd International Conference on Exascale Applications and Software*, 2015.

[11] P. E. Hadjidoukas, D. Rossinelli, F. Wermelinger, J. Sukys, U. Rasthofer, C. Conti, B. Hejazialhosseini, and P. Koumoutsakos. High throughput simulations of two-phase flows on Blue Gene/Q. In *Parallel Computing: on the Road to Exascale, Proceedings of the International Conference on Parallel Computing, ParCo 2015, 1-4 September 2015, Edinburgh, Scotland, UK*, pages 767–776, 2015.

[12] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *Journal of computational physics*, 126:202–228, 1996.

[13] E. Johnsen and T. Colonius. Implementation of WENO schemes in compressible multicomponent flow problems. *Journal of Computational Physics*, 219:715 – 732, 2006.

[14] A. K. Kapila, R. Menikoff, J. B. Bdzil, S. F. Son, and D. S. Stewart. Two-phase modeling of deflagration-to-detonation transition in granular materials: reduced equations. *Physics of Fluids*, 13:3002–3024, 2001.

[15] J. B. Keller and M. Miksis. Bubble oscillations of large amplitude. *The Journal of the Acoustical Society of America*, 68:628–633, 1980.

[16] R. Menikoff and B. J. Plohr. The Riemann problem for fluid flow of real materials. *Reviews of Modern Physics*, 61:75–130, 1989.

[17] A. Murrone and H. Guillard. A five equation reduced model for compressible two phase flow problems. *Journal of Computational Physics*, 202:664–698, 2005.

[18] G. Perigaud and R. Saurel. A compressible flow model with capillary effects. *Journal of Computational Physics*, 209:139–178, 2005.

[19] T. J. Poinsot and S. K. Lele. Boundary conditions for direct simulations of compressible viscous flows. *Journal of Computational Physics*, 101:104–129, 1992.

[20] J. J. Quirk and S. Karni. On the dynamics of a shock-bubble interaction. *Journal of Fluid Mechanics*, 318:129–163, 1996.

[21] D. Rossinelli, B. Hejazialhosseini, P. Hadjidoukas, C. Bekas, A. Curioni, A. Bertsch, S. Futral, S. J. Schmidt, N. A. Adams, and P. Koumoutsakos. 11 PFLOP/s simulations of cloud cavitation collapse. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '13, pages 3:1–3:13, New York, NY, USA, 2013. ACM.

[22] R. Saurel and R. Abgrall. A simple method for compressible multifluid flows. *SIAM Journal on Scientific Computing*, 21:1115–1145, 1999.

[23] R. Saurel, O. Le Métayer, J. Massoni, and S. Gavrilyuk. Shock jump relations for multiphase mixtures with stiff mechanical relaxation. *Shock Waves*, 16:209–232, 2007.

[24] R. Saurel, F. Petitpas, and R. A. Berry. Simple and efficient relaxation methods for interfaces separating compressible fluids, cavitating flows and shocks in multiphase mixtures. *Journal of Computational Physics*, 228:1678–1712, 2009.

[25] D. P. Schmidt and M. L. Corradini. The internal flow of diesel fuel injector nozzles: a review. *International Journal of Engine Research*, 2:1–22, 2001.

[26] A. Tiwari, J. B. Freund, and C. Pantano. A diffuse interface model with immiscibility preservation. *Journal of Computational Physics*, 252:290–309, 2013.

[27] A. Tiwari, C. Pantano, and J. Freund. Growth-and-collapse dynamics of small bubble clusters near a wall. *Journal of Fluid Mechanics*, 775:1–23, 2015.

[28] E. F. Toro, M. Spruce, and W. Speares. Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves*, 4:25–34, 1994.

[29] J. Williamson. Low-storage Runge-Kutta schemes. *Journal of Computational Physics*, 35:48–56, 1980.

[30] K. Yamamoto. Investigation of bubble clouds in a cavitating jet. In *Mathematical Fluid Dynamics, Present and Future*, pages 349–373. Springer Nature, 2016.