Towards High-Fidelity Multiphase Simulations: On the Use of Modern Data Structures on High Performance Computers

Fabian Föll, Timon Hitz, Jens Keim, and Claus-Dieter Munz

Abstract Compressible multi-phase simulations in the homogeneous equilibrium limit are generally based on real equations of state (EOS). The direct evaluation of such EOS is typically too expensive. Look-up tables, based on modern data-structures significantly, reduce the computation time while simultaneously increasing the memory requirements during the simulation. In the context of binary mixtures and large scale simulations this trade off is even more important due to the limited memory resources available on high performance computers. Therefore, in this work we propose an extension of our tabulation approach to shared memory trees based on MPI 3.0. A detailed analysis of benefits and drawbacks concerning the shared memory and the non-shared memory data-structure is described. Another research topic investigates the diffuse interface model of the isothermal Navier–Stokes–Korteweg equations. A parabolic relaxation model is implemented in the open-source code *FLEXI* and 3D simulations of binary head on collisions at various model parameters are shown.

1 Introduction

Typical technical applications, in which multiphase processes can be found, are fuel injection systems such as rocket combustion chambers. The problems inherently contain multiple scales. First, the liquid fuel is injected as a jet with a liquid core. Over time, the jet breaks up and ligaments and droplets form. At the surface of the

F. Föll (✉) · T. Hitz · J. Keim · C.-D. Munz
Institute of Aerodynamics and Gas Dynamics, University of Stuttgart, Pfaffenwaldring 21, 70569 Stuttgart, Germany
e-mail: foell@iag.uni-stuttgart.de

T. Hitz
e-mail: hitz@iag.uni-stuttgart.de

J. Keim
e-mail: keim@iag.uni-stuttgart.de

C.-D. Munz
e-mail: munz@iag.uni-stuttgart.de

W. E. Nagel et al. (eds.), *High Performance Computing in Science and Engineering '19*,
https://doi.org/10.1007/978-3-030-66792-4_25

liquid interface, phase change occurs and the gaseous environment is mixed with evaporated fuel. This mixture is then ignited.

In this project, we aim to understand the mixing processes leading up to the burning of the fuel/oxidizer mixture. Due to the multiscale character, we split the investigation into large scale jet simulations and more detailed simulations of single droplets. These processes face extreme ambient conditions that often exceed the critical state of the fuel. In these regimes, the liquid phase cannot be described incompressible any more and we have to consider the full coupling of hydrodynamics and thermodynamics, requiring the fully compressible flow equations.

The macroscopic modelling for jet simulations is based on the Homogeneous Equilibrium Model (HEM) [22], which considers a mixture of saturated liquid and saturated vapor under full thermodynamic equilibrium. An extension of the intrinsic assumption of vapor-liquid equilibrium in the HEM approach, towards binary mixtures, is the nested procedure of tangent plane distance (TPD) function [15] analysis and classical TPn-flash calculation [16]. These methods are restricted to modifications of the underlying equations of state (EOS), only. Especially with more than one species, the evaluation of the EOS becomes very costly. Therefore, we use look-up tables which shifts the evaluation costs into a pre-processing step while during runtime, only the look-up in an octree data structure is required [9]. For binary mixtures, the look-up tables become huge in storage size which causes problems if the size exceeds the memory of the CPU. Therefore, in this paper we propose a shared memory parallelization of look-up tables based on the MPI 3.0 standard. We provide performance results on benchmark test cases and show its practical use with the simulation of a binary mixing layer.

This project also investigates modelling strategies of phase interfaces, e.g. for droplets, such as sharp and diffuse interface models. As an example of the latter, we use a parabolic relaxation model of the isothermal Navier–Stokes–Korteweg (NSK) equations to simulate the collision of two droplets at varying model parameters. Numerical experiments were conducted using an extension of the open source code *FLEXI*.[1] It is based on a high order nodal discontinuous Galerkin spectral element method (DGSEM) [12].

The outline of the paper is as follows. In the next section, the governing equations are presented. This is followed by the description of the numerical methods, thermodynamic modelling and the look-up table approach. We then present the results on the performance of the look-up tables. Numerical experiments are shown of a two component mixing layer at super-critical conditions using a Peng-Robinson EOS as well as two colliding droplets using NSK diffuse interface model.

[1] http://www.flexi-project.org.

2 Governing Equations

2.1 The Compressible Navier–Stokes System for Multi-components

The compressible Navier–Stokes equations with multiple components are given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0\,, \tag{1}$$

$$\frac{\partial \rho Y_k}{\partial t} + \nabla \cdot (\rho Y_k \boldsymbol{u}) = \nabla \cdot (-\boldsymbol{J}_k)\ , \tag{2}$$

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot \left(\rho \boldsymbol{u} \otimes \boldsymbol{u} + p\underline{\boldsymbol{I}}\right) = \nabla \cdot \left(\underline{\boldsymbol{\tau}}\right)\ , \tag{3}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot [(E + p)\,\boldsymbol{u}] = \nabla \cdot \left(\underline{\boldsymbol{\tau}} \cdot \boldsymbol{u} - \boldsymbol{q}\right)\ , \tag{4}$$

with

$$\underline{\boldsymbol{\tau}} = \underbrace{2\mu\underline{\boldsymbol{S}} - 2/3\underline{\boldsymbol{I}} \cdot \nabla \boldsymbol{u}}_{\text{Stokes law}}, \quad \underline{\boldsymbol{S}} = 1/2\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^{\mathrm{T}}\right), \tag{5}$$

where ρ is the density, $\boldsymbol{u} = (u, v, w)^{\mathrm{T}}$ is the velocity vector, p is the static pressure, E is the total energy per unit volume, $\underline{\boldsymbol{I}}$ is the unit tensor. By considering N_k species, the system is extended by $N_k - 1$ concentration equations where $\boldsymbol{Y} = (Y_1, Y_2, \ \ldots\ , Y_{N_k-1})^{\mathrm{T}}$ with $Y_k = \frac{\rho_k}{\rho}$ is defined as the mass fraction of each species. For multi-component simulations, the heat flux is usually comprised of $\boldsymbol{q} = \boldsymbol{q}^{\mathrm{f}} + \boldsymbol{q}^{\mathrm{d}} + \boldsymbol{q}^{\mathrm{c}}$, where

$$\boldsymbol{q}^{\mathrm{f}} = -\lambda \nabla T \tag{6}$$

is the specific heat flux according to Fourier law with thermal conductivity λ and temperature T. The second term is the inter-species energy flux due to diffusion

$$\boldsymbol{q}^{\mathrm{d}} = \sum_k h_k \boldsymbol{J}_k. \tag{7}$$

Here, $\boldsymbol{q}^{\mathrm{c}}$ are cross-effects, like the Dufour effect, which are not considered in this paper. The viscous stress tensor $\underline{\boldsymbol{\tau}}$ with the strain rate tensor $\underline{\boldsymbol{S}}$ is defined for a Newtonian fluid. The concentration diffusion flux is usually comprised of $\boldsymbol{J}_k = \boldsymbol{J}_k^{\mathrm{f}} + \boldsymbol{J}_k^{\mathrm{c}} + \boldsymbol{J}_k^{\mathrm{b}}$, where

$$\boldsymbol{J}_k^{\mathrm{f}} = -\rho D_k \nabla Y_k\,, \quad k = 1, \ldots, N_k - 1, \tag{8}$$

is the concentration diffusion flux according to Fickian law and D_k is the species diffusion coefficient. Here $\boldsymbol{J}_k^c$ are cross-effects, like the Soret effect, which are also neglected in this paper. The third term,

$$\boldsymbol{J}_k^{\mathrm{b}} = -\rho Y_k \sum_{j=1}^{N_k} \left(D_j \nabla Y_j \right) \,, \quad k = 1, \ldots, N_k - 1, \tag{9}$$

is a correction for the mass balance and recovers $\sum_{k=1}^{N_k} \boldsymbol{J}_k = 0$ to guarantee conservation in cases where the species diffusion fluxes are significantly large [6]. Properties for the last species can be calculated via following relations

$$\sum_{k=1}^{N_k} Y_k = 1 \,, \qquad \sum_{k=1}^{N_k} \rho_k = \rho \,. \tag{10}$$

Since there are $5 + (N_k - 1)$ unknown variables, a closure relation is required between the variables pressure, density, specific internal energy per mass, ϵ, and the species composition,

$$E = \rho\epsilon + \frac{1}{2}\rho \boldsymbol{u} \cdot \boldsymbol{u} \,, \quad \epsilon = \epsilon(\rho, p, \boldsymbol{Y}) \,, \quad p = p(\rho, \epsilon, \boldsymbol{Y}) \,. \tag{11}$$

Such a functional relation is called an equation of state, more precise caloric EOS, and defines the thermodynamic relations between the state variables. For the temperature a thermal EOS (12)

$$T = T(\rho, p, \boldsymbol{Y}). \tag{12}$$

has also to be considered.

2.2 The Navier–Stokes–Korteweg Equations

The Navier–Stokes–Korteweg (NSK) equations are an extension of the Navier–Stokes equations where an interfacial stress is added that approximates capillary effects in phase interfaces of finite thickness. The NSK equations are given in the isothermal case for $T \equiv T_{\mathrm{ref}}$ by

$$\rho_t + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{13}$$

$$(\rho \mathbf{u})_t + \nabla \cdot \left(\rho \mathbf{u} \otimes \mathbf{u} + p \underline{\boldsymbol{I}} \right) = \nabla \cdot \underline{\boldsymbol{\tau}} + \nabla \cdot \underline{\boldsymbol{\tau}}_{\mathrm{K}}. \tag{14}$$

The NSK equations are non-dimensionalized such that the Stokes stress tensor, $\underline{\boldsymbol{\tau}} \in \mathbb{R}^{d \times d}$, and the Korteweg stress tensor, $\underline{\boldsymbol{\tau}}_{\mathrm{K}} \in \mathbb{R}^{d \times d}$, are given by

$$\underline{\boldsymbol{\tau}} = \frac{1}{\mathrm{Re}}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^{\mathrm{T}} - \frac{2}{3}\nabla \cdot \mathbf{u}\underline{\boldsymbol{I}}\right), \tag{15}$$

$$\underline{\boldsymbol{\tau}}_{\mathrm{K}} = \frac{1}{\mathrm{We}}\left(\rho \, \triangle \rho + \frac{1}{2}\left|\nabla \rho\right|^2\right)\underline{\boldsymbol{I}} - \frac{1}{\mathrm{We}}\nabla \rho \otimes \nabla \rho. \tag{16}$$

The Reynolds number, Re, and Weber number, We, are expressed in terms of the numbers $\epsilon_{\mathrm{K}} > 0$ and $\gamma_{\mathrm{K}} > 0$,

$$\frac{1}{\mathrm{Re}} = \epsilon_{\mathrm{K}}, \quad \frac{1}{\mathrm{We}} = \epsilon_{\mathrm{K}}^2 \gamma_{\mathrm{K}}. \tag{17}$$

Due to the capillary stress, Eq. (16), the momentum equation is a third order diffusion-dispersion equation. The system is closed by the pressure function of the Van-der-Waals law [24],

$$p = \frac{\rho R T_{\mathrm{ref}}}{1 - b\rho} - a\rho^2, \tag{18}$$

where a, b, R are material parameters. In reduced, non-dimensional, form, they are $a = 3$, $b = 1/3$, $R = 8/3$. For subcritical temperatures, Eq. (18) is non-convex and the eigenvalues of the hyperbolic flux Jacobian of the NSK equations may be imaginary numbers. The NSK system is therefore of hyperbolic-elliptic type and numerical methods that rely on the strict hyperbolicity of the conservation system cannot be used straight forward any more. To overcome these challenges, Corli et al. [7] proposed a parabolic relaxation scheme for diffusion-dispersion equations, which is extended to the isothermal NSK equations as

$$\rho_t^{\alpha} + \nabla \cdot (\rho^{\alpha}\mathbf{u}^{\alpha}) = 0, \tag{19}$$

$$(\rho^{\alpha}\mathbf{u}^{\alpha})_t + \nabla \cdot \left(\rho^{\alpha}\mathbf{u}^{\alpha} \otimes \mathbf{u}^{\alpha} + p^{\alpha}\underline{\boldsymbol{I}}\right) = \nabla \cdot \underline{\boldsymbol{\tau}}^{\alpha} + \alpha\rho^{\alpha}\nabla\left(c_{\mathrm{K}}^{\alpha} - \rho^{\alpha}\right), \tag{20}$$

$$\beta\left(c_{\mathrm{K}}^{\alpha}\right)_t - \epsilon_{\mathrm{K}}^2\gamma_{\mathrm{K}} \, \triangle c_{\mathrm{K}}^{\alpha} = \alpha\left(\rho^{\alpha} - c_{\mathrm{K}}^{\alpha}\right). \tag{21}$$

An additional unknown, the relaxation variable c_{K}, satisfies a linear parabolic evolution equation with constant relaxation parameters $\alpha, \beta > 0$. The system is of second order and of mixed parabolic-hyperbolic type. For $\alpha \to \infty$, the solution of the parabolic relaxation model approaches the solution of the classical NSK equations, i.e. $(\rho^{\alpha}, \mathbf{u}^{\alpha}) \to (\rho, \mathbf{u})$. The total energy of the relaxation system is given by

$$\mathcal{E}^{\alpha}[\rho] = \int_{\Omega}\left(\frac{1}{2}\rho\left|\mathbf{u}\right|^2 + W(\rho) + \frac{\alpha}{2}(\rho - c_{\mathrm{K}})^2 + \frac{1}{2}\epsilon_{\mathrm{K}}^2\gamma_{\mathrm{K}}\left|\nabla c_{\mathrm{K}}\right|^2\right)\mathrm{d}\,\mathbf{x}. \tag{22}$$

Admissible solutions to Eqs. (19)–(21) are minimizers of Eq. (22).

3 Numerical Methods

The multiphase solver is comprised of several building blocks. The bulk solver is based on a high order discontinuous Galerkin spectral element method (DGSEM). We use an efficient look up table to incorporate real gas equations of state. For the modelling of the phase interface we apply diffuse interface methods. In the Homogeneous Equilibrium Model (HEM), we rely on the EOS to describe phase transition. In the NSK model, capillarity effects are resolved in a phase interface of finite thickness.

3.1 Discontinuous Galerkin Method

The compressible Navier–Stokes equations and the parabolic relaxation model for the NSK equations are discretized by a discontinuous Galerkin spectral element method as described by [11, 12, 14]. The approach is suitable for general systems of conservation equations. In this paper we restrict ourself to the conservation equations of the form

$$\mathbf{U}_t + \nabla_x \cdot \underline{\boldsymbol{F}}(\mathbf{U}, \nabla_x \mathbf{U}) = \boldsymbol{Q}\,, \tag{23}$$

where $\mathbf{U}$ is the vector of the solution unknowns, $\underline{\boldsymbol{F}}$ is the corresponding flux containing the convective and the diffusive fluxes, and $\boldsymbol{Q}$ is the source term of the NSK relaxation model. The divergence operator in the physical space is defined as $\nabla_x = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)^T$.

In a three-dimensional domain we subdivide the computational space into non-overlapping hexahedral elements. Each element is mapped onto the reference cube element $E := [-1, 1]^3$ by a mapping $\boldsymbol{x}(\boldsymbol{\xi})$, where $\boldsymbol{\xi} = (\xi, \eta, \zeta)^{\mathrm{T}}$ is the coordinate vector of the reference element. The mapping onto the reference element E transforms Eq. (23) to the system

$$\boldsymbol{J}\mathbf{U}_t + \nabla_\xi \cdot \underline{\boldsymbol{\mathcal{F}}}\left(\mathbf{U}, \nabla_\xi \mathbf{U}\right) = \boldsymbol{J}\,\boldsymbol{Q}\,, \tag{24}$$

with the Jacobian $\boldsymbol{J}$ and the divergence operator in the reference space $\nabla_\xi = \left(\frac{\partial}{\partial \xi}, \frac{\partial}{\partial \eta}, \frac{\partial}{\partial \zeta}\right)^{\mathrm{T}}$. In each element, the solution and the fluxes are then approximated as polynomials

$$\mathbf{U}_h = \sum_{i,j,k=0}^{N} \hat{\boldsymbol{U}}_{ijk} \psi_{ijk}(\boldsymbol{\xi}) \quad \text{and} \quad \boldsymbol{\mathcal{F}}_h^m = \sum_{i,j,k=0}^{N} \hat{\boldsymbol{\mathcal{F}}}_{ijk}^m \psi_{ijk}(\boldsymbol{\xi})\,, \tag{25}$$

where the superscript $m = \{1, 2, 3\}$ denotes the flux in the direction of the Cartesian coordinates. The basis function $\psi_{ijk}(\boldsymbol{\xi}) = l_i(\xi) l_j(\eta) l_k(\zeta)$ is built by the tensor product of one-dimensional Lagrange polynomials l of degree N. As interpolation nodes we choose Gauss-Legendre points. Due to the nodal character of the Lagrange basis, the degrees of freedom $\hat{\boldsymbol{U}}_{ijk}$ and $\hat{\boldsymbol{\mathcal{F}}}^m_{ijk}$ are values of the approximations of the solution and the flux vectors at the interpolation nodes. To obtain the discontinuous Galerkin formulation, the approximations (25) are inserted into (24) which is then multiplied by a test function ϕ, identical to the basis function ψ, and then integrated in space. Integration by parts of the volume integral of the flux yields the weak formulation

$$\underbrace{\frac{\partial}{\partial t}\int_{\Omega}(\boldsymbol{J}\mathbf{U}_h\phi)\,\mathrm{d}\boldsymbol{\xi}}_{a}-\underbrace{\int_{\Omega}\left(\underline{\mathcal{F}}_h\cdot\nabla_{\xi}\phi\right)\mathrm{d}\boldsymbol{\xi}}_{b}+\underbrace{\int_{\partial\Omega}\left(\left[\underline{\mathcal{F}}_h\cdot\boldsymbol{n}\right]\phi\right)\mathrm{d}\boldsymbol{S}}_{c}=\int_{\Omega}\left(\boldsymbol{J}\,\boldsymbol{Q}_h\phi\right)\mathrm{d}\boldsymbol{\xi}\,. \tag{26}$$

We identify three contributing parts: the volume integral of the time derivative of the solution (a), a volume integral (b) and a surface integral of the fluxes (c). The integrals are evaluated by Gauss-Legendre quadratures. To obtain an approximation of the flux $\underline{\mathcal{F}}_h\cdot\boldsymbol{n}$ at the element surface, a numerical flux function $\boldsymbol{\mathcal{G}}=\boldsymbol{\mathcal{G}}(\mathbf{U}_L,\mathbf{U}_R)$ is introduced. It depends on the states left and right of the interface, $\mathbf{U}_L$ and $\mathbf{U}_R$, respectively. In case of the viscous and heat conduction fluxes, the gradients are needed in addition. For the numerical flux, we use standard approximative Riemann solvers of the HLL-type and Lax Friedrichs families [23]. The discrete formulation (26) is discretized in time using explicit third- or fourth-order Runge–Kutta schemes (RK) [13]. For the viscous fluxes, the approach of Bassi and Rebay [3, 4] is used.

The DG method with high order accuracy is favourable in smooth parts of the flow. At discontinuities or strong gradients we apply the shock capturing of Sonntag and Munz [20, 21]. We switch locally to a second order accurate finite volume (FV) scheme, where the interpolation nodes of the DG polynomials are reorganized as an equidistant sub-grid on which the solution is stored as integral mean values. A modal Persson indicator [19] is used to switch between DG and FV cells.

3.2 Equation of State and Thermodynamic Equilibrium

As thermodynamic coupling relation for the Navier–Stokes equations the cubic Peng-Robinson (PR) EOS [18] is used

$$p=\frac{R_m T}{\frac{M}{\rho}-b}-\frac{a}{\left(\frac{M}{\rho}+\delta_1 b\right)\left(\frac{M}{\rho}+\delta_2 b\right)}, \tag{27}$$

with the universal gas constant R_m and the molar weight of the mixture M. The parameter a takes intermolecular attraction forces into account, b is the co-volume and the PR EOS specific parameters $(\delta_1,\delta_2)=(1+\sqrt{2},1-\sqrt{2})$. The transformation of the pressure explicit thermal EOS to a caloric one is provided by a residual function ansatz [17]. In case of two-phase phenomena a thermodynamic modelling by use of the HEM approach is performed. The underlying assumption of thermodynamic equilibrium is defined by

$$T_v = T_l, \tag{28}$$

$$p_v = p_l, \tag{29}$$

$$\mu_v^k = \mu_l^k, \tag{30}$$

where the symbols v and l represent the vapor and liquid side respectively, μ is the chemical potential and equation (30) has to hold for all N_k species. In case of single species systems the vapor-liquid calculation is performed by use of the algorithm presented by [1], for mixtures a combined approach of TPD analysis and multi-species VLE calculation is used. The TPD function is defined in mole fraction space $\mathbf{z}$ and given by

$$TPD(\mathbf{z}^{trial}) = \sum_{i=k}^{N_k} z_k^{trial} \left[\mu_k(\mathbf{z}^{trial}, T, p) - \mu_k(\mathbf{z}^{test}, T, p)\right]. \tag{31}$$

The superscript $(\cdot)^{test}$ indicates for the feed composition, which is provided from the flow solver and $(\cdot)^{trial}$ for all other possible molar compositions, which fulfill the mass balance condition $\sum_k^{N_k} z_k^{trial} = 1$. The TPD analysis is based on the idea of direct evaluation of the Gibbs free energy surface [2] and checks for a global minimum in Gibbs free energy at the present feed composition. Hereby TPD values greater zero correspond to a stable state, smaller ones to an unstable one. For the analysis of the TPD function the local minimization method with multiple initial guesses presented by [15] is used. The thermodynamic consistent modeling of the states in the two-phase region is provided by the HEM approach with

$$\epsilon^{EQ} = x_v \epsilon_v + (1 - x_v)\epsilon, \tag{32}$$

where the specific inner energy per mass works as a dummy value for any caloric state variable and x_v is the vapor mass fraction defined by

$$x_v = \frac{1/\rho - 1/\rho_l}{1/\rho_v - 1/\rho_l}. \tag{33}$$

Due to the loss of hyperbolicity inside the spinodale region with real gas EOS, the sound speed in the two-phase region in the HEM approach is modeled with the relation presented by [25]

$$\frac{1}{\rho a^2} = \frac{\alpha_v}{\rho a_v^2} + \frac{1 - \alpha_v}{\rho \alpha_l^2}, \tag{34}$$

where a is the sound speed and α the volumetric vapor fraction given by

$$\alpha_v = \frac{x_v \rho}{\rho_v}. \tag{35}$$

3.3 Look up Tables and Extension to Shared Memory Trees

The current Cray machine Hazel Hen has about 185,088 cores in the current expansion stage. These are provided with 24 cores each at 7712 nodes. Each node is comprised of 128 GB memory. The next expansion stage, Hawk, which is planned for spring 2020, will be approximately 640,000 cores at 5000 nodes. The ratio of nodes to cores will accordingly increase more than quintupled from the present time of $N_{cores}/N_{nodes} = 24$ to $N_{cores}/N_{nodes} = 128$. It is important to consider that the available capacity of memory on a node is not increased and will therefore be 1 GB per core. Scalable and highly efficient CFD codes for high-performance (HPC) computers, which are perfectly adapted to old architectures, should keep pace with such new developments. To maintain efficiency, the algorithms have to be modified. Examples are memory-consuming algorithms, which can be found in multi-phase and multi-component simulations in combination with so-called look up tables approaches [9, 10]. Today, these tables are composed of modern data structures such as quadtree or octree data structures, see Figs. 1 and 2. Quadtrees and octrees make use of properties from so called space filling curves for fast data localization. Here the *Morton curve* is popular due to the inherent possibility to access the data via bit operations

$$\text{data position} = f(\text{bit number}), \tag{36}$$

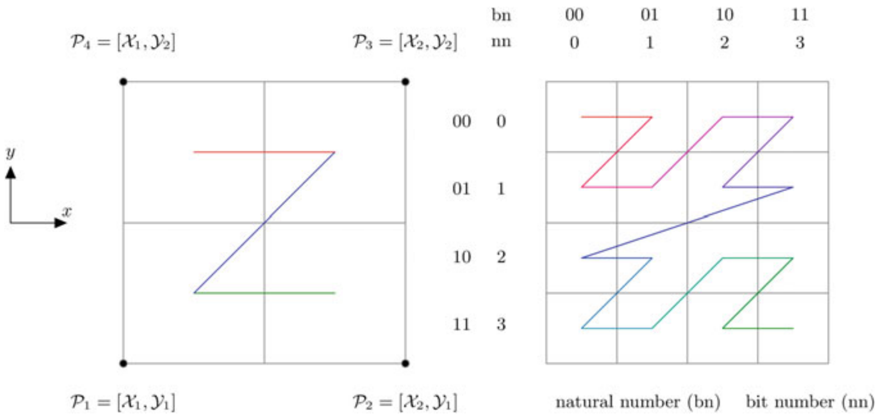

bn
00
01
10
11
$\mathcal{P}_4 = [\mathcal{X}_1, \mathcal{Y}_2]$
$\mathcal{P}_3 = [\mathcal{X}_2, \mathcal{Y}_2]$
nn
0
1
2
3
y
x
00 0
01 1
10 2
11 3
$\mathcal{P}_1 = [\mathcal{X}_1, \mathcal{Y}_1]$
$\mathcal{P}_2 = [\mathcal{X}_2, \mathcal{Y}_1]$
natural number (bn)
bit number (nn)

Fig. 1 Quadtree table: *left* first stage, *right* second stage, bit numbers (bn) are used for fast localization of quadtree elements in the definiton area given by $\mathcal{P}_i$

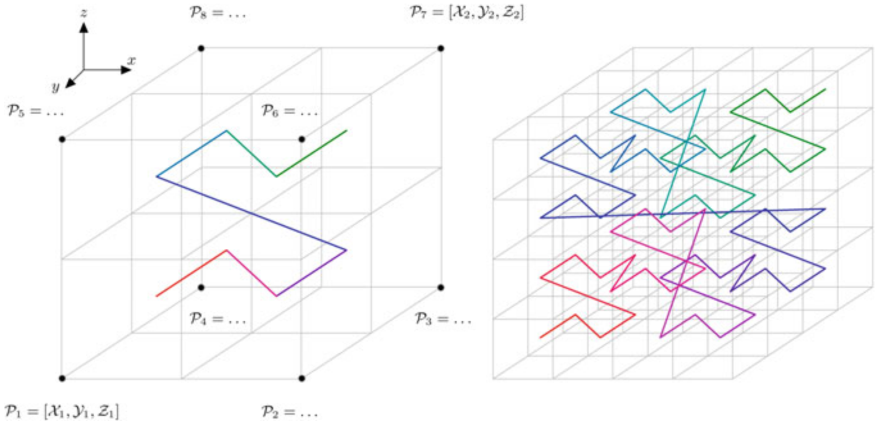


Fig. 2 Octree table—left: one stage, right: two stages

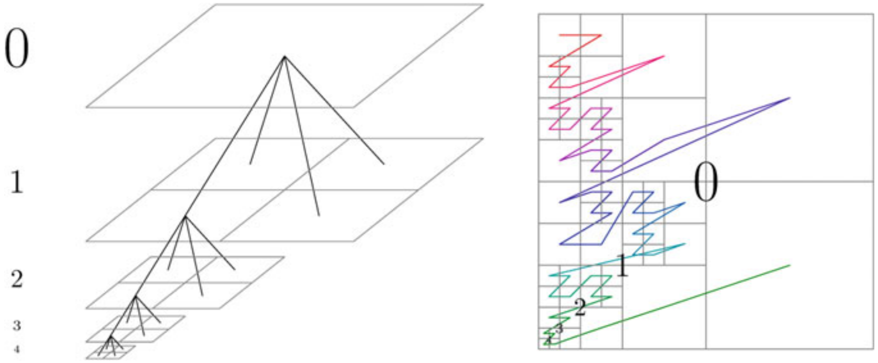


Fig. 3 Left: tree data structure, right: refined quad tree

see Fig. 1. Despite of the usage of such modern data structures, today's CFD simulations may reach the memory limits fast, if large scale high fidelity simulations are performed. In this context we want to discuss in this paper the implementation and application of tree structures on high-performance computers associated with MPI 3.0 and shared memory (Fig. 3).

In the last period, we have extended our tabulation framework, in order to use the look up tables as efficiently as possible on future high performance computers. Initially we will give some information about the parallelization strategy of the CFD solver FLEXI [8]. FLEXI is based on the so-called domain decomposition, which divides the computational grid into heterogeneously distributed MPI processes depending on the number of cores used, see Fig. 4. For the domain decomposition, again a space filling curve, more precisely the so-called *Hilbert curve*, is used. The curve has the special property to optimally distribute the different MPI regions with respect to the volume/surface ratio, even on unstructured grids. Figure 4 shows such

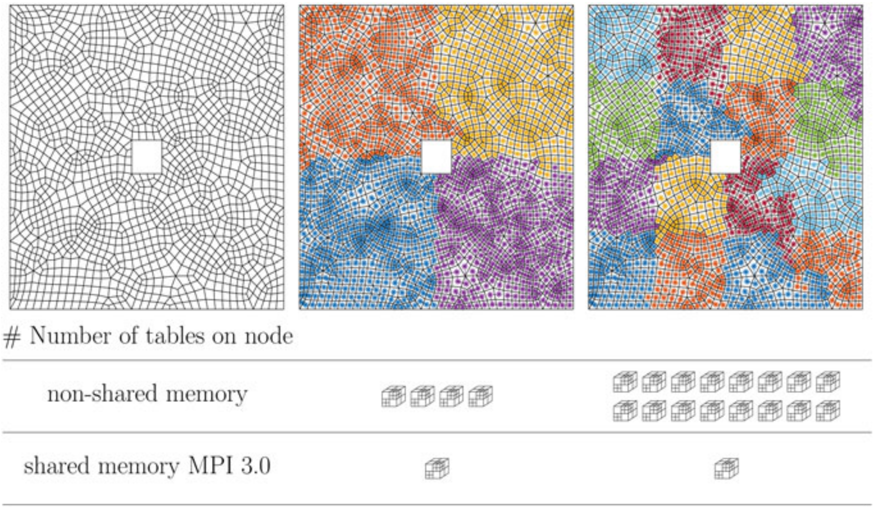


Fig. 4 FLEXI [8] parallelization strategy with domain decomposition via space filling curve: *left* computational quad mesh, *middle* decomposition with 4 MPI processes, *right* decomposition with 16 MPI processes

a division. To ensure that each MPI process can access the data in the table, each MPI process has to initialize and allocate its own table when using standard MPI features. By considering MPI 3.0 features, like shared memory windows, the number of tables for each node can be reduced to one table for each node.

However, modern data structures generally consist of chained pointer lists, which are not directly applicable with the MPI 3.0 shared memory feature. This is due to the fact, that each MPI process is linked with its own virtual memory space, see Fig. 5. This has consequences for the way in which the tree structure has to be read in and accessed during the simulation on HPC systems. In Fig. 6, the standard approach to store and access the tree data is depicted. Here, each branch of the tree stores a small portion of the whole data. Furthermore, each MPI process reads and allocates the data during IO. In Fig. 7 the alternative approach to store and access the tree data with MPI 3.0 shared memory window is depicted. Here, unlike before, each branch of the tree only stores two integer IDs depicting a range in the global shared memory array. An important aspect is the fact, that during IO only one MPI process on the node is allowed to read and allocate the data. Nevertheless, each MPI process has to read and store the empty tree. This is necessary because each MPI process has still to know the relative path to the unique IDs in the last branch. With this approach it is possible to maintain the efficient tree data structure while simultaneously be capable to store and access several magnitudes of data.

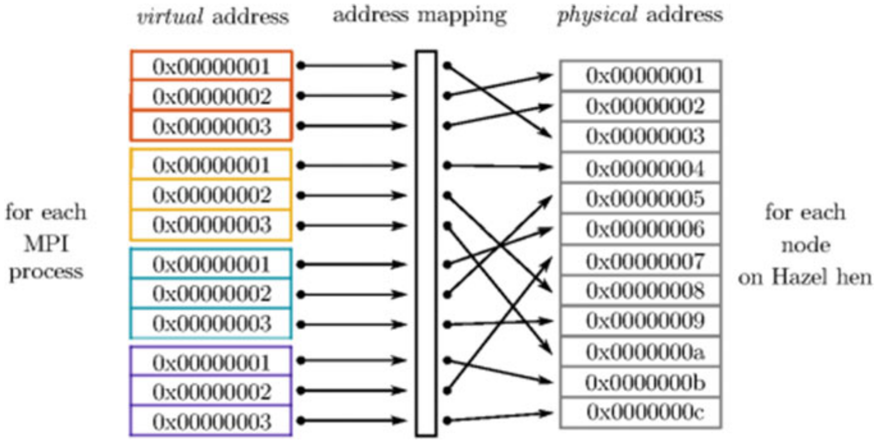


Fig. 5 Mapping from virtual address space for each MPI process to physical address space, here 4 MPI processes (the colors are chosen consistent to the domain decomposition in Fig. 4)

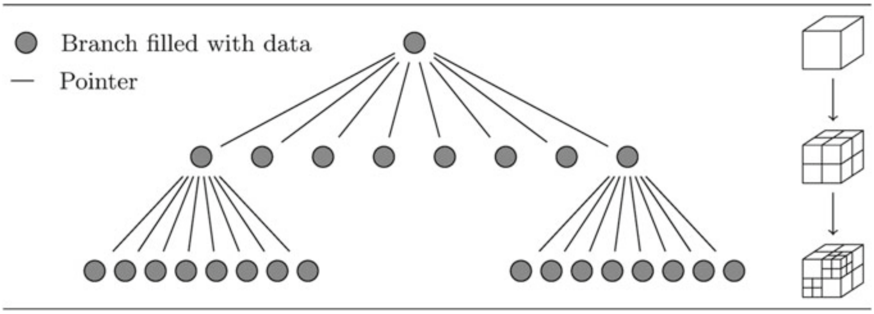


Fig. 6 Tree data structure without MPI 3.0 shared memory: each branch contains the data

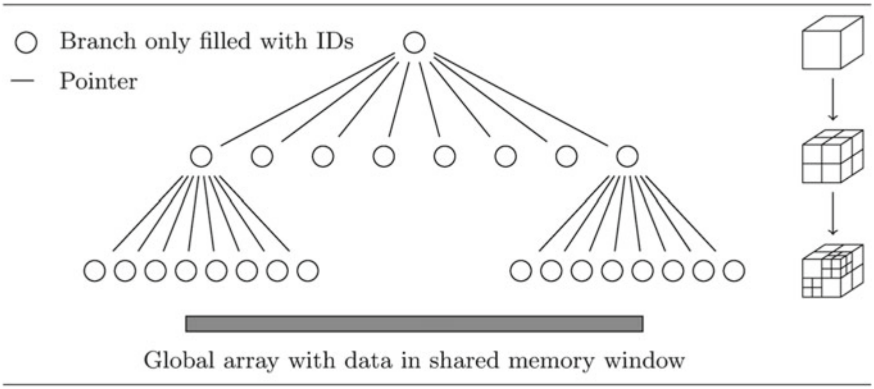

○ Branch only filled with IDs
— Pointer
Global array with data in shared memory window

Fig. 7 Tree data structure with MPI 3.0 shared memory: each branch only contains an integer id for the global data array, the global data array is allocated in a shared memory window

4 Results

4.1 Performance Comparison of Tree Data Structures with and Without MPI 3.0 Shared Memory

In this section we investigate the different data structures in terms of performance and memory usage.

For the comparison we use the performance index

$$\text{PID} = \frac{\text{wall-clock-time} \cdot \#\text{processors}}{\#\text{DOF} \cdot \#\text{time steps} \cdot \#\text{RK-stages}}. \tag{37}$$

The results are obtained with the open source code FLEXI in combination with octree tables. Note that FLEXI is based on the HDF5 standard. To ensure a fair comparison, we have chosen a simple test case, the standard lid driven cavity in two dimensions, see [5]. We choose a binary mixture with two different ideal gases, instead of performing a one-component simulation as it is typically done in the literature. First, we look at the performance of both data structures that we defined in Sect. 3.3. We perform each simulation six times and average the measurement to cancel out hardware influences. The comparison was done on 8 nodes with 192 processes. The tree data was refined up to 7 levels resulting in about ≈ 2.8 GB memory size. Each octant represents the data in a three dimensional polynomial basis of degree 4. In the first two lines of Table 2, we have listed the results for the performance test. We notice a slightly higher PID for the MPI 3.0 implementation, which is most likely due to additional index mapping used to get the position in the global shared memory array. In the third line we compare the time which was used to read and allocate the data before the simulation starts. We note that the IO of the MPI 3.0 implementation is different in the way that we do not read in the whole tree from the HDF5 at once. By using the shared memory option, we read, allocate and deallocate each octant successively from the HDF5 file, to save as much memory as possible. Here, we notice a non negligible longer IO time for the MPI 3.0 implementation. The factor between the standard and the shared memory approach is about 6 (Table 1).

The next two Tables 3 and 4 contain memory comparisons.

Table 1 Numerical setup of the Lid driven cavity test problem

Species	Y_k	κ	N	# elements	Re
Helium	0.5	1.60	5	1024	100
Air	0.5	1.40			

Table 2 Performance comparison for the different data structures

	Standard (non-shared memory)	MPI 3.0 (shared memory)
PID (s)	2.70×10^{-5}	3.05×10^{-5}
Simulation time (s)	370.45	410.03
IO time (s)	116.05	742.34

Table 3 Memory usage depending on tree level, here we tabulated a binary mixture of Helium/Air

Tabulation type	Tree level	Memory usage in (GB)
Octree with N=4 in each direction	7	2.8
	8	10.0
	9	32.2

Table 4 Theoretical memory usage by using non-shared memory and shared memory data structures on different architectures

Tabulation of octrees	Standard (non-shared memory)	MPI 3.0 (shared memory)
Max memory on Hazel Hen (GB) 24 cores on each node	5.33	128.00
Max memory on Hawk (GB) 128 cores on each node	1.00	128.00

Here, we notice the huge improvement with the MPI 3.0 shared memory implementation. For the planned architecture Hawk we will (theoretically) be able to store and access about 128 times more memory than with the old algorithm.

4.2 Navier–Stokes Multi-component Simulations

The multi-component Navier–Stokes model was used for comparison simulations conducted with direct use of the EOS and tables with different refinement levels. As test case a two-dimensional shear layer of nitrogen and n-dodecane of the dimension $[0, 0.2] \times [-0.15, 0.15]\ \mathrm{m}^2$ was investigated. The initial states of the pure species are summarized in Table 5. As initial condition a base flow in x-direction superposed by a y-velocity disturbance was used, which are given by

Table 5 Specified initial conditions of the base flow for the pure species of the mixing layer test case

Species	Sum formula	u_0 (m/s)	p (MPa)	T (K)	ρ (kg/m^3)
Mitrogen	N_2	205.46	6	1000	19.88
n-heptane	C_7H_{16}	−85.59	6	600	255.63

$$u_{N_2} = 2M_{c,0}a_{N_2}\left[1+\left(\frac{a_{N_2}}{a_{C_7H_{16}}}\right)\sqrt{\frac{\rho_{N_2}Z_{N_2}}{\rho_{C_7H_{16}}Z_{C_7H_{16}}}}\right]^{-1}, \tag{38}$$

$$u_{C_7H_{16}} = -\sqrt{\frac{\rho_{N_2}Z_{N_2}}{\rho_{C_7H_{16}}}}u_{N_2}, \tag{39}$$

$$u(x,t=0) = u_0\left|erf\left(\frac{\sqrt{\pi}y}{\delta_{\omega,0}}\right)\right|, \tag{40}$$

$$Y_{C_7H_{14}}(x,t=0) = 1 - y_{N_2}, \tag{41}$$

$$Y_{N_2}(x,t=0) = 0.5 + 0.5\ erf\left(\frac{\sqrt{\pi}y}{\delta_{\omega,0}}\right), \tag{42}$$

$$v(x,y,t=0) = 0.1\ max\ (u_0)\sin\left(\frac{8\pi x}{\delta_{\omega,0}}\right)\exp\left\{-\left(\frac{y}{\delta_{\omega,0}}\right)^2\right\} \tag{43}$$

and

$$\rho = \rho(T,p,\mathbf{Y}). \tag{44}$$

Here Z is the compressibility factor, $M_{c,0}$ is the Mach number which was chosen to 0.4 and $\delta_{\omega,0}$ is the initial blending thickness between the two species with $\delta_{\omega,0} = 6.859 \cdot 10^{-3}$ m.

The achieved results are visualized in Figs. 8 and 9. In both snapshots we can observe some differences in between the three computations. This is due to the fact, that the chosen Kelvin Helmholtz test problem is a highly sensitive initial value problem. The different thermodynamic approximations quickly lead to different results. In summary we can show that the tabulation approach is suitable for multi-component simulations in the super-critical regime, nevertheless future investigations are necessary.

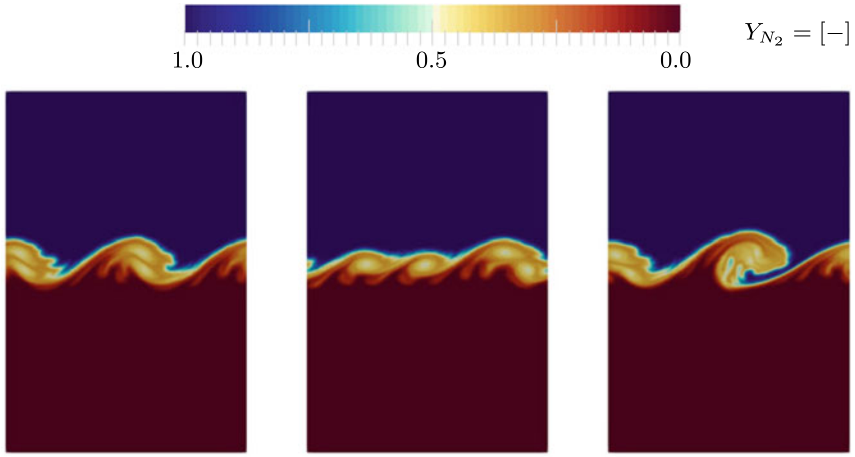


Fig. 8 Temporal snapshot of nitrogen mass fraction at $t = 4$ ms for simulations conducted by direct use of the EOS (left), coarse tables (middle) and refined tables (right)

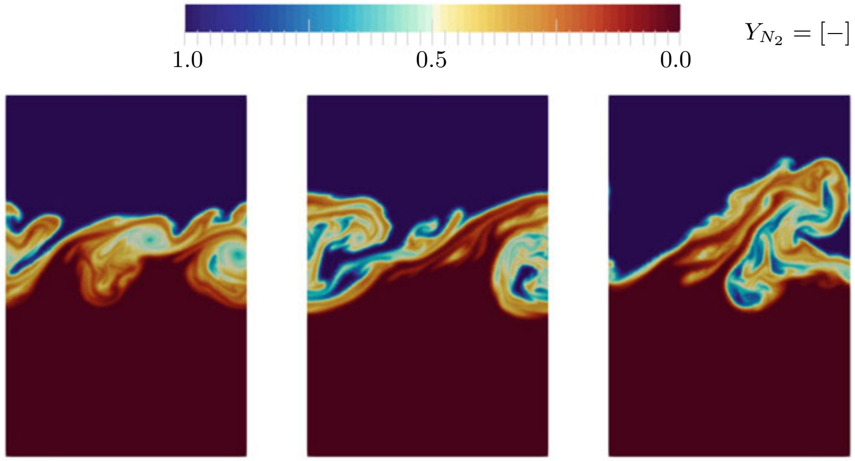


Fig. 9 Temporal snapshot of nitrogen mass fraction at $t = 6$ ms for simulations conducted by direct use of the EOS (left), coarse tables (middle) and refined tables (right)

4.3 Navier–Stokes–Korteweg

The parabolic relaxation model for the NSK equations was used to investigate head on collisions of two droplets.

4.3.1 Simulation Setup

The initial conditions were

$$\rho(\mathbf{x}, t = 0) = \rho_{\text{vap}} + \frac{\rho_{\text{vap}} - \rho_{\text{liq}}}{2} \sum_{i=1}^{2} \left(\tanh \left(\frac{d_i - r_i}{2\sqrt{\gamma_{\text{K}} \epsilon_{\text{K}}^2}} \right) \right) \tag{45}$$

$$u(\mathbf{x}, t = 0) = \begin{cases} \frac{v_{\text{ini}}}{2} + \left(1 - \tanh \left(\frac{d_1 - r_{\text{d}}}{2\sqrt{\gamma_{\text{K}} \epsilon_{\text{K}}^2}} \right) \right) & \text{if} \quad x < 0.5, \\ \frac{-v_{\text{ini}}}{2} + \left(1 - \tanh \left(\frac{d_2 - r_{\text{d}}}{2\sqrt{\gamma_{\text{K}} \epsilon_{\text{K}}^2}} \right) \right) & \text{if} \quad x \geq 0.5, \end{cases} \tag{46}$$

$$v(\mathbf{x}, t = 0) = 0, \tag{47}$$

$$w(\mathbf{x}, t = 0) = 0, \tag{48}$$

where $\rho_{\text{vap}} = 0.3197$, $\rho_{\text{liq}} = 1.8071$ are the Maxwellian densities at $T_{\text{ref}} = 0.85$. The droplet radii were $r_1 = r_2 = 0.5$ and the distance was given by

$$d_i = \parallel \mathbf{x} - \mathbf{x}_{0,i} \parallel, \tag{49}$$

where $\mathbf{x}_{0,1} = (0.3, 0.5, 0.5)^\top$ and $\mathbf{x}_{0,2} = (0.7, 0.5, 0.5)^\top$ are the initial positions of the droplets. Four cases were investigated where the droplet number, position, and size remained the same and the model parameters and initial velocities were changed. The parameters are summarized in Table 6. The computation domain was $\Omega = [0, 1]^3$ and it was discretized by 64 elements in each direction. The polynomial degree was $N = 3$ which yielded 256^3 degrees of freedom (DOF). Time integration was done implicit with CFL $= 100$ using a fourth order ESDIRK scheme with six stages. The simulations were performed on the Hazel Hen supercomputer at HLRS using 200 nodes.

Table 6 Parameters for head on droplet collision simulations

Parameter	ϵ_{K} [−]	γ_{K} [−]	α [−]	β [−]	v_{ini} [−]
Case A	1×10^{-3}	100	100	1×10^{-2}	3.0
Case B	1×10^{-2}	1.00	100	1×10^{-2}	3.0
Case C	1×10^{-2}	0.05	100	1×10^{-2}	3.0
Case D	1×10^{-2}	0.05	100	1×10^{-2}	4.0

4.3.2 Simulation Results

The isocontour of the mean density, $\rho_{\text{mean}} = 1.0634$, of the solution of case A is shown in Fig. 10 for different time instances. Two droplets were pushed towards each other and coalesce. A flat disc formed for $t > 0.12$ which broke up into a ring and a small droplet in the center at $t \approx 0.24$. Both the ring and the centered droplet evaporated and for $t \to \infty$ only vapour remained, since the average density was in the stable vapour phase.

In Case B ϵ_{K} was increased and γ_{K} was decreased such that different phenomena were observed. The isocontour of the mean density is shown in Fig. 11. Again, the two droplets merged and a disc formed. The disc flattened and break up occurred at its centre, however no droplet was formed and only a ring remained. Eventually, the ring evaporated and the domain was filled by a stable vapour phase.

Case C reduced γ_{K} further, which led to a thinner phase interface. The isocontour is shown in Fig. 12. After coalescence, the disc formed again but no break up occurred and the disc remained at that form until it evaporated completely.

Case D used the same parameters as Case C but increased the initial velocity of the droplets. The isocontour is shown in Fig. 13. The momentum of the droplets was increased and the impact was stronger such that the disc quickly broke up and a ring and centered droplet remained.

The total energy, Eq. (22), was calculated in each time step. As seen in Fig. 14, the total energy decreased monotonously until a minimum was reached. Hence, the solutions produced by the relaxation model were admissible.

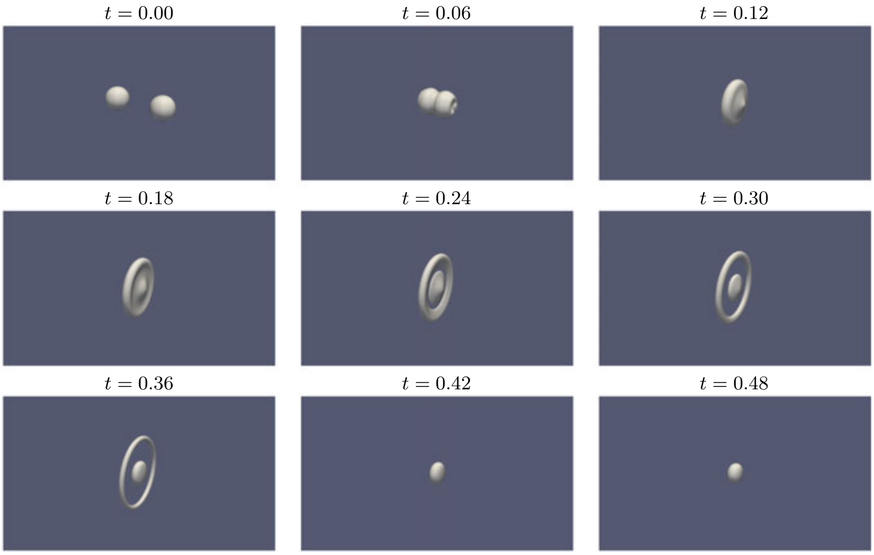


Fig. 10 Results for Case A with parameters $\epsilon_{\text{K}} = 1\,\exp{-3}$, $\gamma_{\text{K}} = 100$, $v_{\text{ini}} = 3.0$

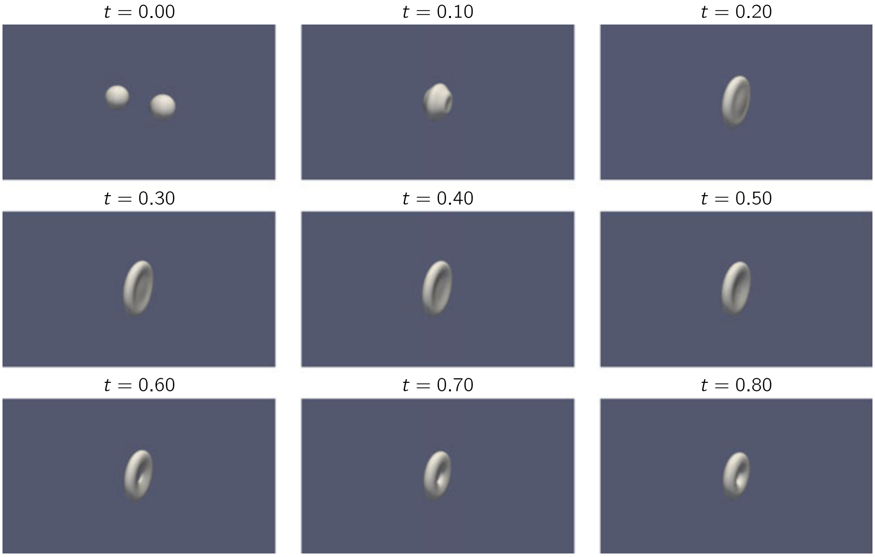


Fig. 11 Results for Case B with parameters $\epsilon_{\mathrm{K}} = 1\exp -2$, $\gamma_{\mathrm{K}} = 1.00$, $v_{\mathrm{ini}} = 3.0$

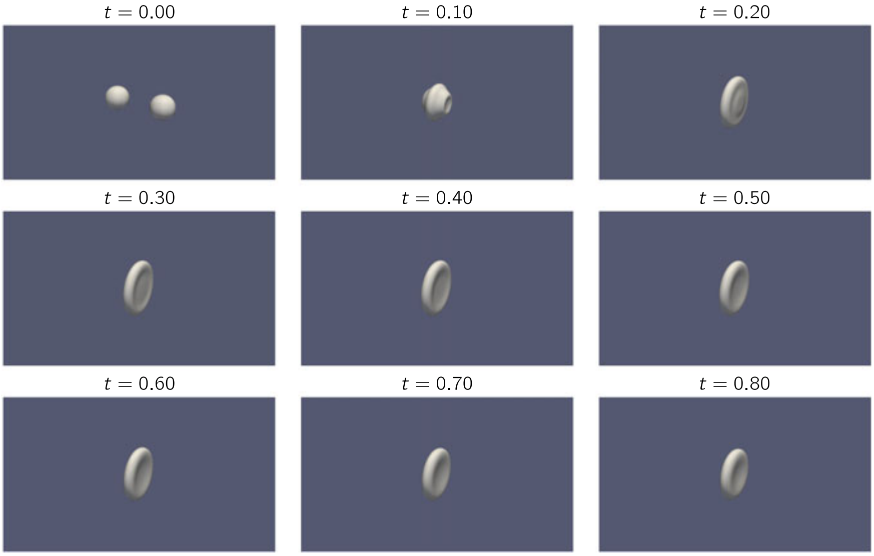


Fig. 12 Results for Case C with parameters $\epsilon_{\mathrm{K}} = 1\exp -2$, $\gamma_{\mathrm{K}} = 0.05$, $v_{\mathrm{ini}} = 3.0$

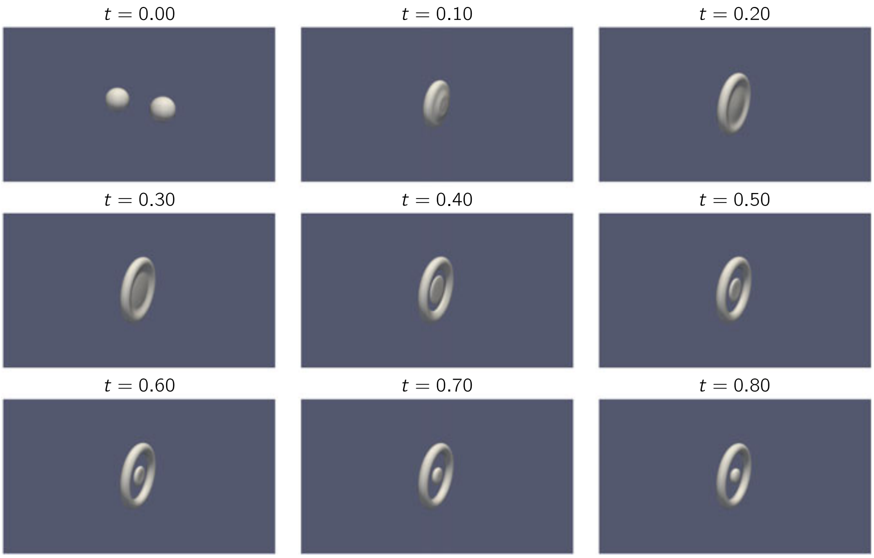
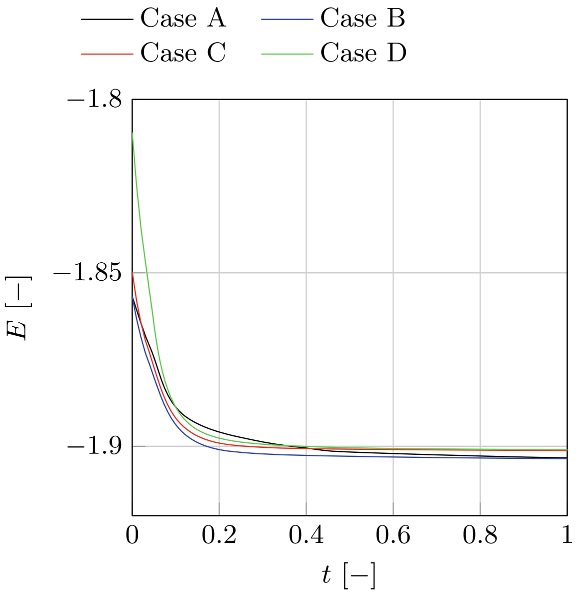


Fig. 13 Results for Case D with parameters $\epsilon_{\mathrm{K}} = 1\exp -2$, $\gamma_{\mathrm{K}} = 0.05$, $v_{\mathrm{ini}} = 4.0$

Fig. 14 Decay of total energy for the head on droplet collisions



5 Summary and Conclusions

In this work we carried out investigations on the use of modern data structures on high performance computers. In this context, a new implementation strategy for shared memory look up tables for binary mixtures was introduced. We were able to show that a change in hardware architecture on high performance computers, e.g. from Hazel Hen to Hawk, has a great impact on the old algorithms. With the new implementation we are able to store and access about 128 times more memory than with the old algorithm. The simulation and comparison of a multi-component real gas shear layer with exact EOS and tabulation approach led to reasonable results, however further investigations are necessary.

In addition, 3D simulations of colliding droplets were carried out using a parabolic relaxation model of the Navier–Stokes–Korteweg diffuse interface model. A variation of model parameters produced a variation in the coalescence behaviour. Future research aims at validation with experimental results.

Acknowledgements We gratefully acknowledge the Deutsche Forschungsgemeinschaft (DFG) through SFB-TRR 40 "Fundamental Technologies for the Development of Future Space-Transport-System Components under High Thermal and Mechanical Loads" and SFB-TRR 75 "Droplet dynamics under extreme ambient conditions" Computational resources have been provided by the Bundes-Höchstleistungsrechenzentrum Stuttgart (HLRS).

References

1. R. Akasaka, A reliable and useful method to determine the saturation state from Helmholtz energy equations of state. J. Therm. Sci. Technol. **3**, 442–451 (2008)
2. L.E. Baker, A.C. Pierce, K.D. Luks, Gibbs energy analysis of phase equilibria. SPE J. **22**, 731–742 (1982)
3. F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. J. Comput. Phys. **131**(2), 267–279 (1997). https://doi.org/10.1006/jcph.1996.5572
4. F. Bassi, S. Rebay, Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier-Stokes equations. Int. J. Numer. Methods Fluids **40**(1–2), 197–207 (2002). https://doi.org/10.1002/fld.338
5. O. Botella, R. Peyret, Benchmark spectral results on the lid-driven cavity flow. Comput. Fluids **27**(4), 421–433 (1998). https://doi.org/10.1016/S0045-7930(98)00002-4
6. T. Coffee, J. Heimerl, Transport algorithms for premixed, laminar steady-state flames. Combust. Flame **43**(Supplement C), 273–289 (1981). https://doi.org/10.1016/0010-2180(81)90027-4
7. A. Corli, C. Rohde, V. Schleper, Parabolic approximations of diffusive-dispersive equations. J. Math. Anal. Appl. **414**(2), 773–798 (2014). https://doi.org/10.1016/j.jmaa.2014.01.049
8. FLEXI, Description and source code (2018), https://www.flexi-project.org/. Accessed 02 Oct 2018
9. F. Föll, T. Hitz, C. Müller, C.D. Munz, M. Dumbser, On the use of tabulated equations of state for multi-phase simulations in the homogeneous equilibrium limit. Shock. Waves **1** (2019). https://doi.org/10.1007/s00193-019-00896-1
10. F. Föll, S. Pandey, X. Chu, C.D. Munz, E. Laurien, B. Weigand, High-fidelity direct numerical simulation of supercritical channel flow using discontinuous Galerkin spectral element method,

in *High Performance Computing in Science and Engineering ' 18*, ed. by W.E. Nagel, D.H. Kröner, M.M. Resch (Springer International Publishing, 2019), pp. 275–289

11. J. Hesthaven, T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, 1st edn. (Springer Publishing Company, Incorporated, 2008). https://doi.org/10.1007/978-0-387-72067-8
12. F. Hindenlang, G. Gassner, C. Altmann, A. Beck, M. Staudenmaier, C. Munz, Explicit discontinuous Galerkin methods for unsteady problems. Comput. Fluids **61**, 86–93 (2012). https://doi.org/10.1016/j.compfluid.2012.03.006
13. C.A. Kennedy, M.H. Carpenter, R. Lewis, Low-storage, explicit Runge–Kutta schemes for the compressible Navier–Stokes equations. Appl. Numer. Math. **35**(3), 177–219 (2000). https://doi.org/10.1016/S0168-9274(99)00141-5
14. D. Kopriva, *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*, 1st edn. (Springer Publishing Company, Incorporated, 2009)
15. M.L. Michelsen, The isothermal flash problem. Part 1. Stability. Fluid Phase Equilib. **9**, 1–19 (1982a)
16. M.L. Michelsen, The isothermal flash problem. Part 2. Phase-split calculation. Fluid Phase Equilib. **9**, 21–40 (1982b)
17. M.L. Michelsen, J.M. Mollerup, *Thermodynamic Models: Fundamentals & Computational Aspects*, 2nd edn. (Tie-Line Publications, Holte, 2007)
18. D.Y. Peng, D.B. Robinson, A new two-constant equation of state. Ind. Eng. Chem. Fundam. (1976)
19. P. Persson, J. Peraire, Sub-cell shock capturing for discontinuous galerkin methods, in *44th AIAA Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings, American Institute of Aeronautics and Astronautics* (2006), https://doi.org/10.2514/6.2006-112
20. M. Sonntag, C.D. Munz, Shock capturing for discontinuous Galerkin methods using finite volume subcells, in *Finite Volumes for Complex Applications VII-Elliptic, Parabolic and Hyperbolic Problems*, ed. by J. Fuhrmann, M. Ohlberger, C. Rohde (Springer International Publishing, 2014), pp. 945–953
21. M. Sonntag, C.D. Munz, Efficient parallelization of a shock capturing for discontinuous Galerkin methods using finite volume sub-cells. J. Sci. Comput. **70**(3), 1262–1289 (2017). https://doi.org/10.1007/s10915-016-0287-5
22. H.B. Stewart, B. Wendroff, Two-phase flow: models and methods. J. Comput. Phys. **56**(3), 363–409 (1984). https://doi.org/10.1016/0021-9991(84)90103-7
23. E. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction* (Springer, Berlin, 2009). https://doi.org/10.1007/b79761
24. J. Van der Waals, Over de Continuiteit van den Gas-en Vloeistoftoestand. Ph.D. thesis, University of Leiden (1873)
25. A.B. Wood, *A Textbook of Sound*, 1st edn. (G. Bell and Sons, 1941)