

RESEARCH ARTICLE

Open Access



Compressible flow simulation with moving geometries using the Brinkman penalization in high-order Discontinuous Galerkin

Neda Ebrahimi Pour*, Nikhil Anand, Harald Klimach and Sabine Roller

*Correspondence:
neda.epour@uni-siegen.de
University of Siegen, 57076
Siegen, Germany

Abstract

In this work we investigate the Brinkman volume penalization technique in the context of a high-order Discontinuous Galerkin method to model moving wall boundaries for compressible fluid flow simulations. High-order approximations are especially of interest as they require few degrees of freedom to represent smooth solutions accurately. This reduced memory consumption is attractive on modern computing systems where the memory bandwidth is a limiting factor. Due to their low dissipation and dispersion they are also of particular interest for aeroacoustic problems. However, a major problem for the high-order discretization is the appropriate representation of wall geometries. In this work we look at the Brinkman penalization technique, which addresses this problem and allows the representation of geometries without modifying the computational mesh. The geometry is modelled as an artificial porous medium and embedded in the equations. As the mesh is independent of the geometry with this method, it is not only well suited for high-order discretizations but also for problems where the obstacles are moving. We look into the deployment of this strategy by briefly discussing the Brinkman penalization technique and its application in our solver and investigate its behavior in fundamental one-dimensional setups, such as shock reflection at a moving wall and the formation of a shock in front of a piston. This is followed by the application to setups with two and three dimensions, illustrating the method in the presence of curved surfaces.

Keywords: Moving geometries, Compressible flow, High-order Discontinuous Galerkin, Brinkman penalization

Introduction

In engineering applications we are often dealing with moving parts. When investigating the fluid motion in those scenarios, we therefore, need to consider complex, moving obstacles that influence or even drive the flow. In this work we look into a method to realize moving obstacles in simulations with a high-order discontinuous Galerkin scheme. High-order schemes need less number of degrees of freedom to represent smooth solutions. On

modern computing systems, where memory bandwidth is one of the main bottlenecks, this is a big advantage. The discontinuous Galerkin method combines a high-order local state representation within elements with a coupling of elements via fluxes on the surface as in the finite volume method. This relatively loose coupling via the element surfaces is a strong advantage on modern distributed parallel computing systems, as only comparatively little data needs to be communicated between elements.

Obstacles in such mesh-based schemes are usually represented by fitting the mesh to the geometry and prescribed via boundary conditions. An alternative option is the use of some immersed boundary method [22]. Immersed boundaries have the advantage that they do not require the mesh to be fitted to the geometries that are to be represented. This is especially attractive when considering moving obstacles as the movement does not necessitate a change of the mesh.

In high-order discretizations it also becomes difficult to achieve an accurate representation of the geometry by fitted meshes, due to the large elements that we want to employ. With moving geometries the problem of fitting the mesh of a high-order discretization to complex geometries gets magnified. Thus, immersed boundaries are a promising choice for moving obstacles in a high-order discontinuous Galerkin scheme. Comprehensive reviews of immersed boundary methods are available by Mittal and Iccarino [22], Sotiropoulos and Yang [27] and recently Maxey [21]. We use a penalization method as described by Liu and Vasilyev [20].

Volume penalization methods introduce penalty terms in the continuous equations, while imposing nothing in the discretization context [4]. Arquis and Caltagirone [5] introduced a type of volume penalization method for simulations of isothermal obstacles in incompressible flows that makes use of the Brinkman model for porous media. The basic idea is to model obstacles as a porous media, where the porous medium approaches a solid obstacle. A distinct advantage this method, in comparison to other penalization methods, is its error estimation, which can be rigorously predicted in terms of the penalization parameters [24]. Kevlahan and Ghidaglia [15] used a pseudo-spectral method and applied this Brinkman penalization method to model stationary, as well as moving geometries.

Liu and Vasilyev [20] used the Brinkmann penalization method for the compressible Navier-Stokes equations. They introduced new penalization term for the mass conservation and showed promising results for their acoustic setups. Several other successful investigations using different numerical methods speak for the effectiveness of the Brinkmann penalization method. Application in pseudo-spectral methods are found in [23] and [13]. In [24] the Brinkman penalization is used in the context of finite volumes and finite elements. To the authors knowledge, the Brinkman penalization method used for compressible flows, either involves non-moving obstacles or mostly considered for aero-acoustic problems e.g. in [18] or [20].

In [2] we applied the Brinkman penalization in our high-order discontinuous Galerkin method for compressible Navier-Stokes equations with non-moving obstacles. That analysis showed that the method can be successfully used to represent obstacles even when using high-order schemes and coarse elements. Thus, we now look into the application of this scheme to moving geometries. Our implementation is available in our open-source solver Ateles [26].

Numerical method

In the following we revisit the fundamental equations of compressible viscous flows. With the continuum assumption fluid motion is modeled in terms of the compressible Navier-Stokes equations. To model obstacles as immersed boundaries, we extend those equations by Brinkman penalization terms as described by Komatsu et al. [18]. With the introduction of the penalization terms we also discuss their treatment in the time integration scheme.

Compressible Navier-Stokes equations

Conservation of mass, momentum and energy applied to fluids leads to the equation system that we refer to as compressible Navier-Stokes equations. In conservative formulation with density ρ , momentum \mathbf{m} and energy E the system can be written as:

$$\partial_t \rho + \nabla \cdot \mathbf{m} = 0 \quad (1a)$$

$$\partial_t \mathbf{m} + \nabla \cdot (\mathbf{u} \otimes \mathbf{m} + p\mathbf{I}) - \nabla \tau = \rho \mathbf{f} \quad (1b)$$

$$\partial_t E + \nabla \cdot (\mathbf{u}((E + p))) - \nabla \cdot (\tau \mathbf{u} + \kappa \nabla T) = -\mathbf{m} \cdot \mathbf{f}. \quad (1c)$$

Where τ is the viscous stress tensor defined by (2).

$$\tau = \mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) - \lambda (\nabla \cdot \mathbf{u}) \mathbf{I} \quad (2)$$

Equation (1a) represents conservation of mass, (1b) conservation of momentum and (1c) conservation of energy. Velocity is denoted by $\mathbf{u} = \mathbf{m}/\rho$ and \mathbf{I} indicates the identity matrix. The parameter μ represents the dynamic viscosity of the fluid and λ the volume viscosity, assumed as $\lambda = 2\mu/3$ here. The temperature of the fluid is denoted by T and κ represents its thermal conductivity. Additional source terms, like gravity, are represented on the right hand side of (1b) and (1c) by \mathbf{f} . The equation system is closed by the equation of state for the ideal gas that provides a relation between pressure p , temperature and density:

$$p = \rho R T = (\gamma - 1) \left(\frac{E - \rho \mathbf{u}^2}{2} \right). \quad (3)$$

With $\gamma = c_p/c_v$ denoting the heat capacity ratio and R the ideal gas constant as given in (4).

$$R = c_p - c_v = (\gamma - 1)c_v \quad (4)$$

Brinkman penalization

The volume penalization methods introduce additional artificial terms to the equations to be solved in areas where obstacles are to be present. With the Brinkman penalization method these appear as source terms on the right hand sides of the equations. In [20] the Brinkman penalization method was introduced for compressible Navier-Stokes equations. While these included the possibility for moving obstacles, they were not Galilean invariant. This was corrected by Komatsu et al. [18], resulting in the formulation we use in this scheme.

The equation system with the penalization terms is given in (5).

$$\partial_t \rho + \nabla \cdot \mathbf{m} = - \left(\frac{1}{\phi} - 1 \right) \chi \frac{\partial(u_i - U_{oi})\rho}{\partial x_i} \quad (5a)$$

$$\begin{aligned} \partial_t m_i &= -\frac{\partial}{\partial x_j} (m_i u_j) - \frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} - \boxed{\frac{\chi}{\eta} (u_i - U_{oi})} \\ &\quad - \boxed{u_i \left(\frac{1}{\phi} - 1 \right) \chi \frac{\partial}{\partial x_j} [\rho (u_i - U_{oi})]} \end{aligned} \tag{5b}$$

$$\begin{aligned} \partial_t E &= -\frac{\partial}{\partial x_j} [(E + p)u_j] + \frac{\partial}{\partial x_i} (u_i \tau_{ij}) + \frac{\partial}{\partial x_j} \left(k \frac{\partial T}{\partial x_j} \right) - \boxed{\frac{\chi}{\eta_T} (T - T_o)} \\ &\quad - \boxed{\frac{\chi}{\eta} (\mathbf{u} - \mathbf{U}_o) \cdot \mathbf{u} - \frac{|\mathbf{u}|^2}{2} \left(\frac{1}{\phi} - 1 \right) \chi \frac{\partial}{\partial x_j} [\rho (u_i - U_{oi})]} \end{aligned} \tag{5c}$$

Where η and η_τ are the viscous and thermal permeability and ϕ the porosity of the Brinkman model. Velocity and temperature of the obstacle is given by U_o and T_o , respectively and its location is defined by the masking function χ . While the permeabilities only appear in source terms, the porosity affects the divergence operators and thereby changes the Eigenvalues of the convective system and necessitate smaller timesteps when considered in explicit time integration schemes. The geometrical modelling can therefore primarily be influenced by the permeability terms and the porosity. They allow to tune the modelling towards a solid obstacle. Hence, they need to be chosen appropriately.

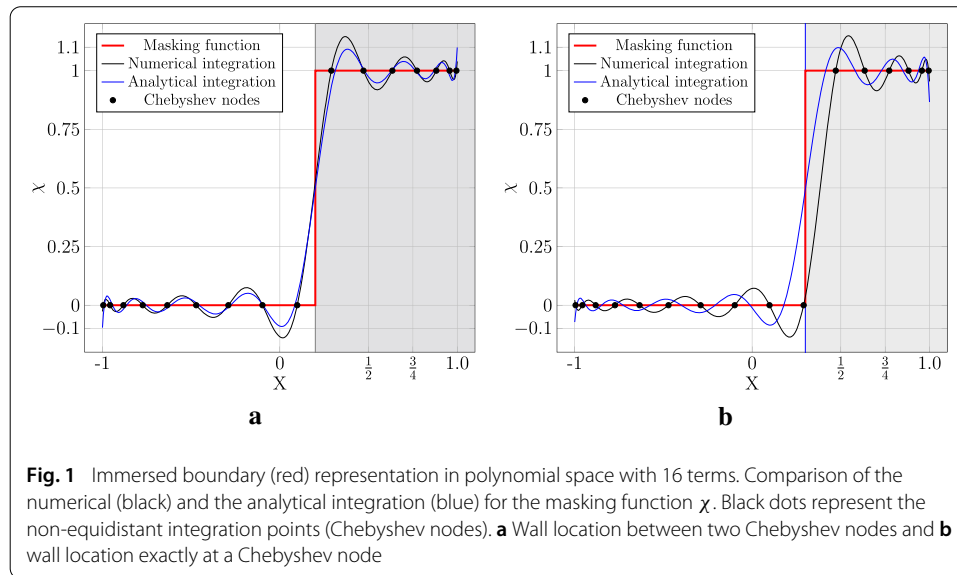
As we show in [2], it is feasible to ignore the porosity, introduced by Liu and Vasilyev [20] for compressible fluids, if the permeabilities are chosen sufficiently small. While the source terms get extremely stiff by tiny permeabilities, this allows us to avoid any complications by the porosity in the spatial divergence operators. The source terms are purely local and can be solved pointwise. Thus, a implicit mixed explicit time integration is deployed to deal with the source terms implicitly while the remaining parts are still solved explicitly. We ignore the porosity by choosing $\phi = 1$, which results in the following, simplified model:

$$\partial_t \rho = -\frac{\partial m_i}{\partial x_i} \tag{6a}$$

$$\partial_t m_i = -\frac{\partial}{\partial x_j} (m_i u_j) - \frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} - \boxed{\frac{\chi}{\eta} (u_i - U_{oi})} \tag{6b}$$

$$\begin{aligned} \partial_t E &= -\frac{\partial}{\partial x_j} [(E + p)u_j] + \frac{\partial}{\partial x_i} (u_i \tau_{ij}) + \frac{\partial}{\partial x_j} \left(k \frac{\partial T}{\partial x_j} \right) - \boxed{\frac{\chi}{\eta_T} (T - T_o)} \\ &\quad - \boxed{\frac{\chi}{\eta} (\mathbf{u} - \mathbf{U}_o) \cdot \mathbf{u}} \end{aligned} \tag{6c}$$

To solve the source terms introduced by this penalization implicitly, we use a diagonally implicit Runge-Kutta scheme [1]. In this work we make use of the values for the permeability proposed in our analysis of the method for non-moving obstacles, see also there for details on the time integration method [2]. We showed, that with a smaller value of β the modelling error can be decreased considerably, allowing to neglect the porosity in the conserved quantities, as the permeability enables to model solid obstacles. With the viscous permeability $\eta = \beta^2 \cdot \phi^2$ and the thermal viscosity $\eta_T = 0.4\beta \cdot \phi$, the expected modelling error as described in [20] has a magnitude of $\beta^{1/4}$ for a porosity of 1. For our simulations we consider the outcomes of Anand et al. and consider a scaling factor of $\beta \leq 10^{-6}$.



Representation of immersed boundaries

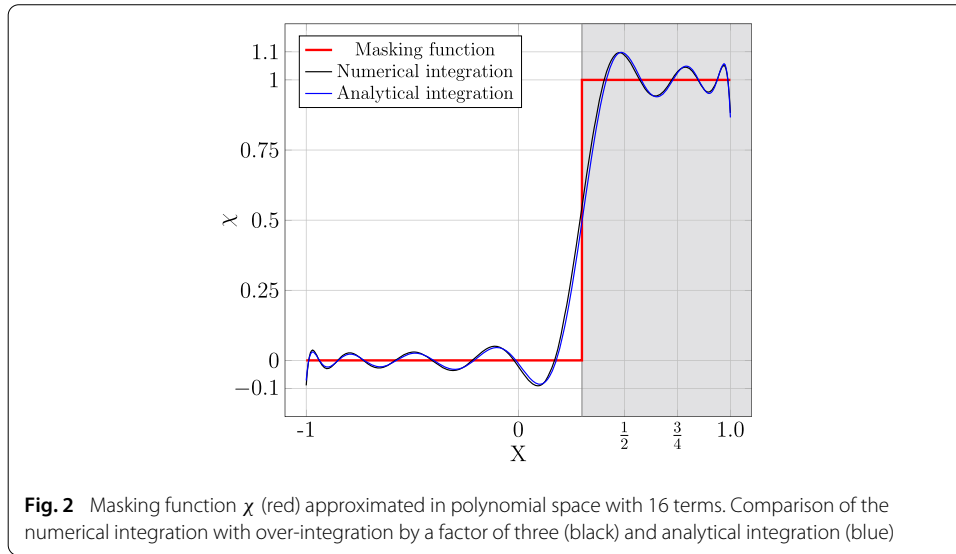
With the penalization method to represent obstacles, we can use the same discretization technique for the obstacles as for the solution in our scheme. Thus curvature of the geometries to represent poses no special problem as the discretization of the obstacles can be chosen from the same realm as the solution. In the penalization method, obstacles are given by the masking function χ . This function is one wherever an obstacle is to be modeled and zero everywhere else.

An algorithm to find this polynomial representation of the masking function for surfaces defined by polygons, for example in the form of STL files is proposed in [17].

To discuss the representation of χ in our high-order discontinuous Galerkin implementation on the basis of Legendre polynomials we consider a simple one dimensional setup with a single element. Figure 1a shows this single element and the masking function that models a wall within that element. To the left with $\chi = 0$ the fluid is to behave undisturbed, while to the right with $\chi = 1$ the flow is to be penalized. This step function is mapped to the polynomial space and uses a Legendre expansion with a maximal polynomial degree of 15. The blue line illustrates the projection with an analytic integration, while the black line indicates the resulting polynomial from a numerical integration with the shown 16 (Chebyshev) integration points.

With the numerical integration an error is introduced, as the jump of the masking function χ can only be determined up to the accuracy of the integration points. This is illustrated more clearly in Fig. 1b, which also clearly shows how this integration error depends on the location of the discontinuity in relation to the integration points. When considering a continuously moving wall, we will, therefore, see discrete jumps in the numerical representation of χ as the discontinuity moves across the integration point intervals. We further note that, as the integration points are not equidistantly spaced, this localization error is not constant, but depends on its location within the element. Closer to the element boundaries the localization is more accurate than in the middle of elements.

To decrease this error from numerical integration, we utilize over-integration and use more integration points for the approximation [16]. In Fig. 2 we consider an over-



integration factor of three. Comparing Fig. 1b without over-integration and the same case with this over-integration by a factor of three in Fig. 2, the improved representation of the wall location becomes apparent. As Fig. 2 shows, an over-integration by factor three already yields a polynomial approximation that is close to the analytical integration. Another approach to improve the geometrical representation is introduced by Engels et al. [7]. The authors assume the masking function to be smooth by introducing a thin smoothing layer. This avoids the strong discontinuity introduced by the jump of the masking function at the interface. However, this approach does not improve the convergence in the vicinity of the geometry surface.

Numerical results

Moving wall test case

To investigate the penalization scheme for a moving geometry in our discontinuous Galerkin implementation, we first analyze the fundamental behavior in a one-dimensional setup. We look into the reflection of an acoustic wave from a moving wall. Figure 3 sketches this setup with the initial wave and wall positions.

The traveling wave has the form of a Gaussian pulse given by Eq. (7). Its center is initially located at $x = 0.25$, while the wall is located at $x = 0.5$.

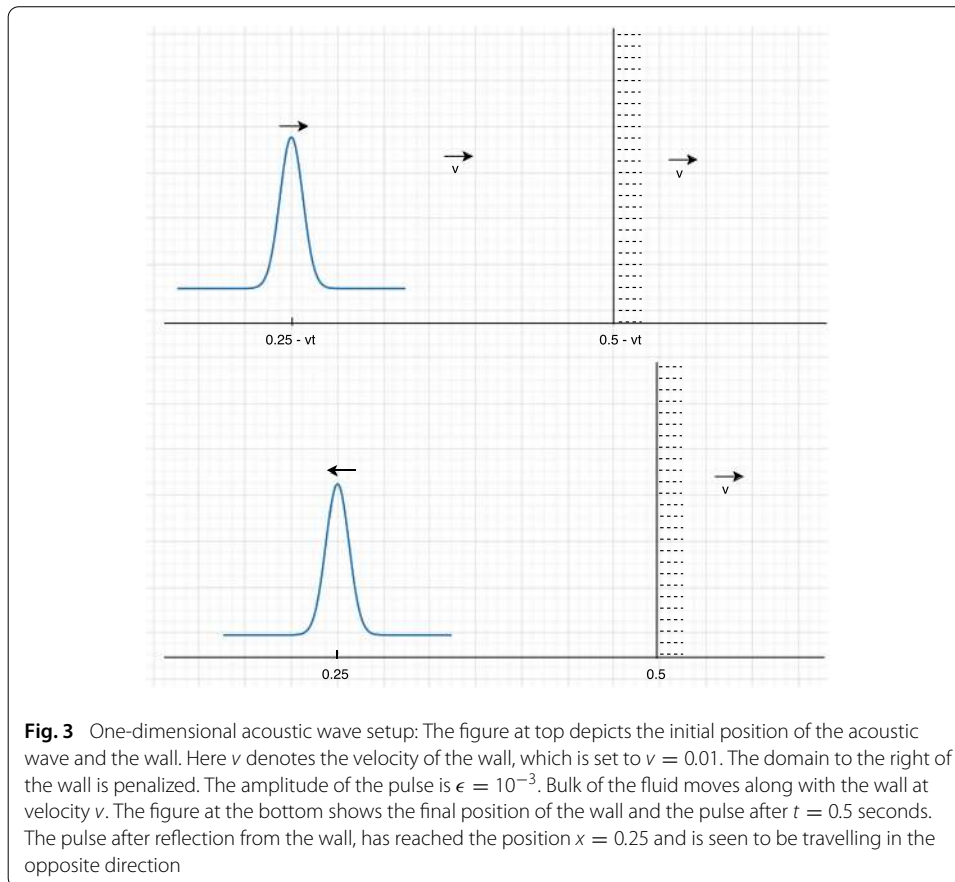
$$\rho' = u' = p' = \epsilon \exp \left[-\ln(2) \frac{(x - 0.25)^2}{0.004} \right] \tag{7}$$

The amplitude of the pulse is chosen to be $\epsilon = 10^{-3}$. The perturbations in density ρ' , velocity u' and pressure p' from (7) are applied to a constant, non-dimensionalized state with a speed of sound of 1. This results in the initial condition for the conservative variables density ρ , momentum m and total energy e as described in (8).

$$\rho = 1 + \rho', m = \rho(u' + v), e = \frac{1}{\gamma(\gamma - 1)} + \frac{p'}{\gamma - 1} + \frac{1}{2} \rho(u')^2, \tag{8}$$

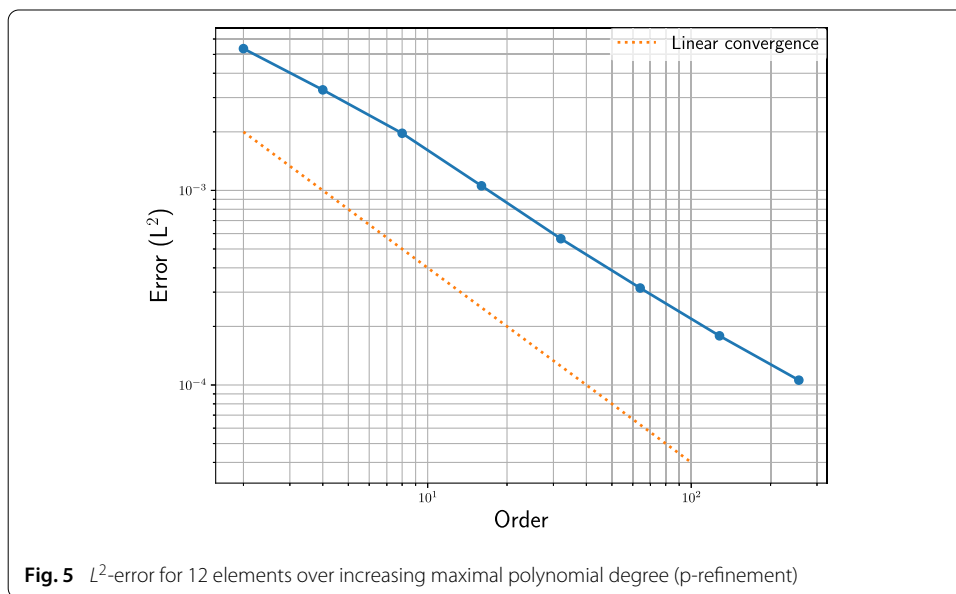
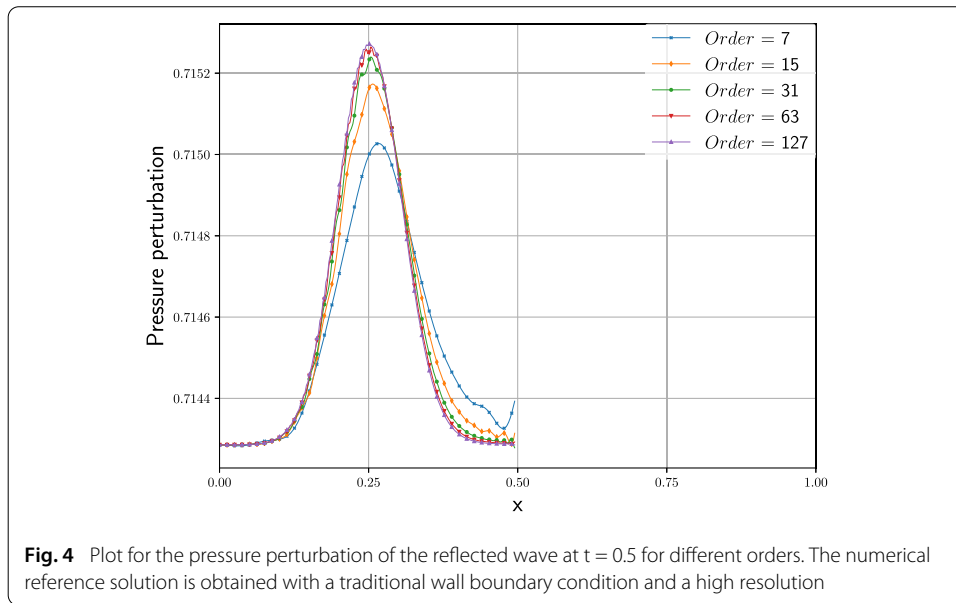
where v is the background velocity equal to the velocity of the moving wall.

The penalization with porous medium is applied in the right half of the domain ($x > 0.5$). In acoustic theory the reflection should be perfectly symmetric, and the reflected pulse



should have the same shape and size, only with opposite velocity. After the simulation time of $t = 0.5$, with linear acoustic wave transport and a speed of sound of 1, the position of the pulse will again be at a distance of $d = 0.25$ away from the position of the wall at that time. During this duration, the wall travels a distance of vt to the right, reaching the point $x = 0.5$. Therefore, the final position of the pulse has its center $x = 0.25$ after the time interval t .

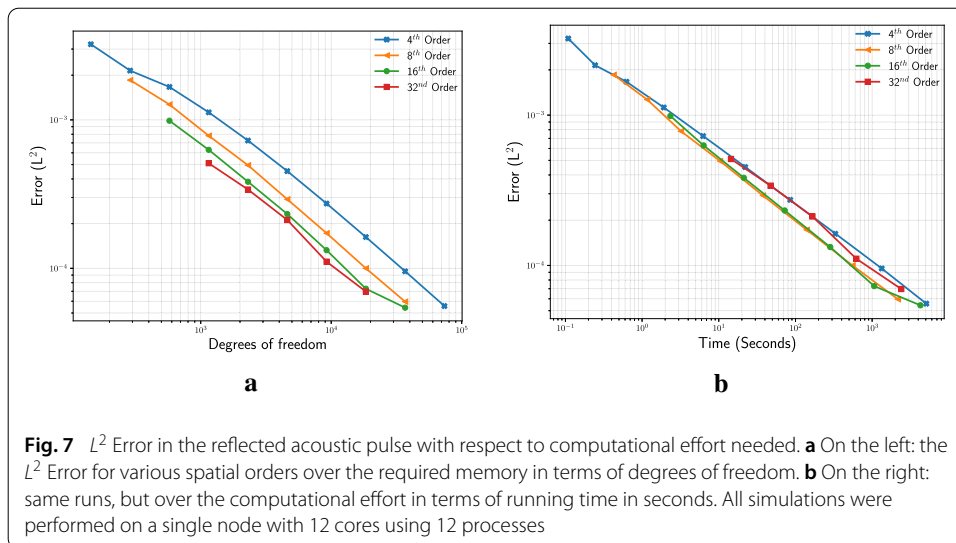
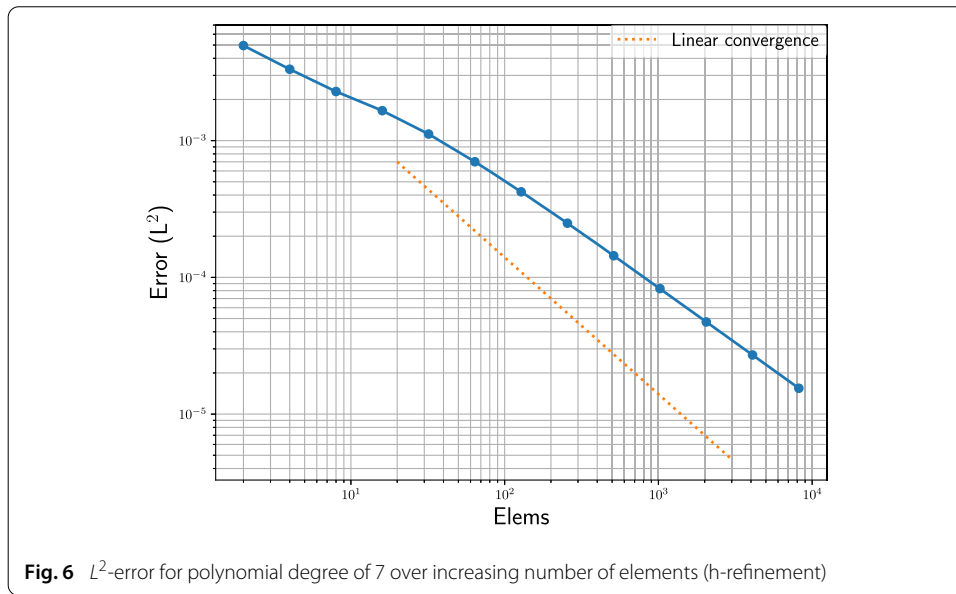
For the linear model we can easily compute the analytic solution and use this to judge the quality of the numerical computation with the penalization method. This simple setup allows us to analyze the dampening of the reflected wave amplitude as well as induced phase shifts. While the analytical result for linear wave transports provides a good reference in general for the acoustic wave with small perturbation, it sufficiently deviates from the nonlinear behavior to limit its suitability for convergence analysis in small error values. Therefore, we compare the simulation results with the penalization method and moving walls to numerical results with a traditional fixed wall boundary condition and a high resolution. This reference is computed with the same element length, but the domain ends at $x = 0.5$, which is the final position of the wall after $t = 0.5$ seconds, with a wall boundary condition and a maximal polynomial degree of 255 is used (256 degrees of freedom per element) to approximate the smooth solution. The pressure perturbation and the final position of the reflected wave (i.e. at time, $t = 0.5$), for different orders, is shown in Fig. 4. The domain for all the simulations shown in the figure comprises of 12 elements, i.e. $dx = 1/12$. Here, we can observe that both loss in wave amplitude as well as phase



shift of the reflected wave gets diminished with increasing orders. With sufficiently large degrees of freedom, the wave shows an excellent agreement with the reference solution. The error convergence is done in reference to this solution.

Figure 5 shows the error convergence in terms of the L^2 norm over the maximal polynomial degree in the discretization scheme (p-refinement) with a fixed number of 12 elements. We observe a linear convergence rate with increasing polynomial degree, which matches the worst case scenario from the analysis with a non-moving wall in [2]. As in the moving case the discontinuity of the masking function χ ranges through all locations of the elements, we can not expect a better convergence rate.

Figure 6 shows the L^2 norm of the error for the reflected pressure wave with a maximal polynomial degree of 7 over an increasing number of elements (h-refinement). Again a first order convergence can be observed in the overall domain. The moving discontinuity



in the masking function eliminates the fast error convergence rate of the high-order discretization close to the discontinuity. However, away from the discontinuity we maintain the fast convergence rate and still expect benefits.

In Fig. 7 the computational effort is shown for various spatial scheme orders. Figure 7a shows the error over the used number of degrees of freedom and, thus the required memory consumption. And Fig. 7b shows the respective running times of the simulations on a single computing node with 12 Intel Sandy-Bridge cores.

The test was performed starting with 12 elements in each data series, which is the leftmost point in the plot. For subsequent points in the series, the number of elements are always increased by factor of 2. While we do not achieve a spectral convergence as in a purely smooth solution, we still find the higher order discretizations to be advantageous with respect to the required memory. In the computational effort a strong improvement from second to fourth order discretization is observed. However, for higher spatial scheme

orders than 16, the required computational running time increases again. Though this specific timing result is tied to the system, the simulations were run on, we generally expect the running time to be worse for higher orders in this metric on other computing systems as well.

There are mainly two contributions to this error that diminishes the convergence order. One cause is the Gibbs phenomenon that describes induced oscillations due to the discontinuity in the masking function and the other is the accuracy of the localization of the discontinuity from the numerical integration. Both effects are illustrated in Sect. [Representation of immersed boundaries](#). The inaccuracy from numerical integration can be limited by over-integration and using more points. The error from the Gibbs phenomenon can also be limited by employing a re-projection method [10]. Though, such a re-projection method potentially could restore the convergence up to the point of discontinuity, we do not consider it here.

Reflected shock wave

While in the previous setup we looked into a smooth acoustic wave, we now look into the reflection of a shock. We thus, not only have a discontinuity in the masking function χ but also in the solution itself. For the inviscid equations the solution to this problem can be computed analytically. Thus, we will neglect viscous terms in this setup and use the inviscid Euler equations. However, the same Brinkman penalization is used to model the moving wall. The reflection of a shock at a fixed wall with this method has been studied in [2]. Now, we move the wall similarly to the reflection of the acoustic pulse in the previous section. Of course the shock incurs more problems with a high-order discretization, nevertheless, this analysis shows that it is feasible to use it in this setting.

The state variables downstream (indicated by 1) are listed in Table 1. The upstream state is computed via the Rankine-Hugoniot conditions for shock Mach number M_s . The Rankine-Hugoniot conditions yield the following relations:

$$\frac{p_2}{p_1} = 1 + \frac{2\gamma(M_s - \frac{w_s}{u_s})^2 - 1}{\gamma + 1} \quad (9)$$

for the relation of the pressure p before and after the shock and

$$\frac{\rho_2}{\rho_1} = \frac{1 + \frac{\gamma+1}{\gamma-1} \cdot \frac{p_2}{p_1}}{\frac{\gamma+1}{\gamma-1} + \frac{p_2}{p_1}} \quad (10)$$

for the relation of the densities ρ . From these we can also find the pressure relation of before and after (p_3) the reflected shock, see [3]. Additionally, the wall velocity w_s needs to be taken into account here, leading to Eq. (11) for the pressure relation.

$$\frac{p_3}{p_2} = \frac{(M_s - \frac{w_s}{c_1})^2(3\gamma - 1) - 2(\gamma - 1)}{2 + (M_s - \frac{w_s}{c_1})^2(\gamma - 1)} \quad (11)$$

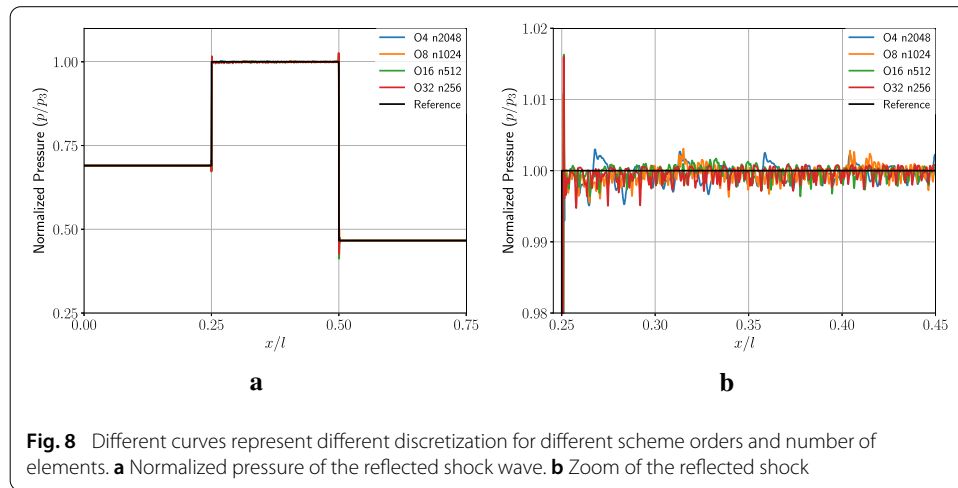
The speed of the reflected shock wave is than given by Eq. (12) [9].

$$u_r = \frac{1}{M_s} \left(1 + \frac{2(M_s^2 - 1)}{(\gamma + 1)/(\gamma - 1)} \right) c_1 \quad (12)$$

These relations provide us with the piecewise constant exact solution, that we use as a reference to compare the numerical results with.

Table 1 Initial condition of the shock state

| | | |
|---------------------------|----------|---------------|
| Downstream speed of sound | c_1 | 1.0 |
| Shock Mach number | M_s | 1.2 |
| Shock velocity | u_s | 1.2 |
| Wall velocity | w_s | 0.012 |
| Downstream density | ρ_1 | 1.0 |
| Downstream pressure | p_1 | γ^{-1} |
| Downstream velocity | u_1 | 0.012 |
| Isentropic coefficient | γ | 1.4 |



With a shock wave Mach number of $M_s = 1.2$ and a wall speed of $w_s = 0.012$ the exact solution for the pressure relation across the reflected shock wave is according to Eq. (11) 1.44911145. The simulation domain for this test case is the unit interval of $x \in [0, 1]$ with the wall initially located at $x = 0.49437415$. Hence the porous material covers at $t = 0$ more than half of the simulation domain ($x \in [0.49437415, 1]$). The initial shock position is $x = 0.24437415$ with a distance of 0.25 to the wall. The simulation is run until the wall reaches $x = 0.5$, the reflected shock is then at $x = 0.25$. Between wall and shock the fluid assumes the velocity of the wall $u_1 = w_s$.

For our investigation, we consider four different numerical discretization, 256, 512, 1024, and 2048 elements (n) in total ($\Delta x = 1/n$) and a scheme Order (O) of 32, 16, 8, and 4, respectively. This yields the same number of degrees of freedom (8196) across all runs. The penalization parameters are chosen to be $\phi = 1$ and $\beta = 10^{-6}$, the β value can be chosen that large as the numerical error is higher, when compared to the modelling error. Figure 8 presents the reflected shock wave for the different spatial resolutions. On the left in Fig. 8a the overall pressure distribution is shown along with the piecewise constant reference. As can be seen, all discretizations with the same number of degrees of freedom provide a good approximation of the exact solution (indicated by the black line). Thus, while the higher spatial scheme order does not offer a benefit in this scenario with piecewise constant solutions, it still is quite capable to represent the solution.

Figure 8b on the right shows a close-up of the solution close to the reflected shock and shows the oscillatory of the high-order approximation. These small oscillations were not observed in [2] with a non-moving wall. Thus, they are induced by the movement of the

Table 2 Comparison of simulation results with the exact solution

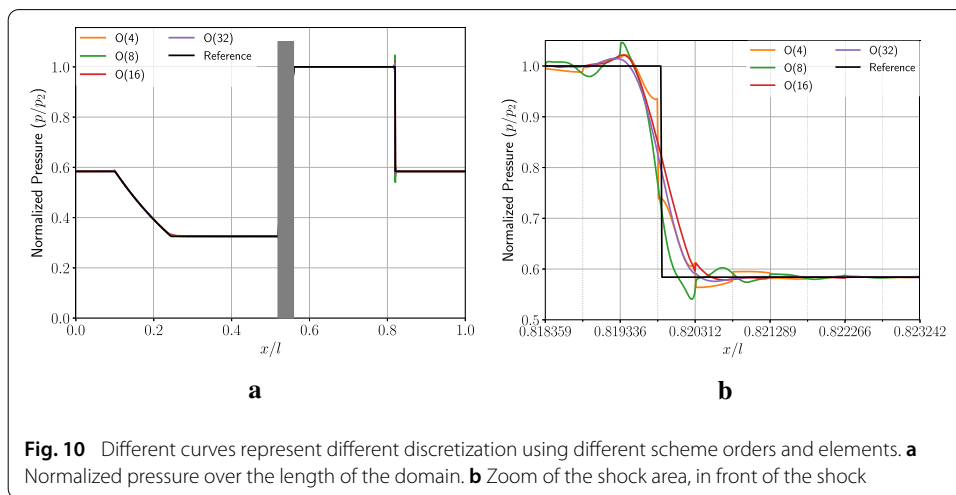
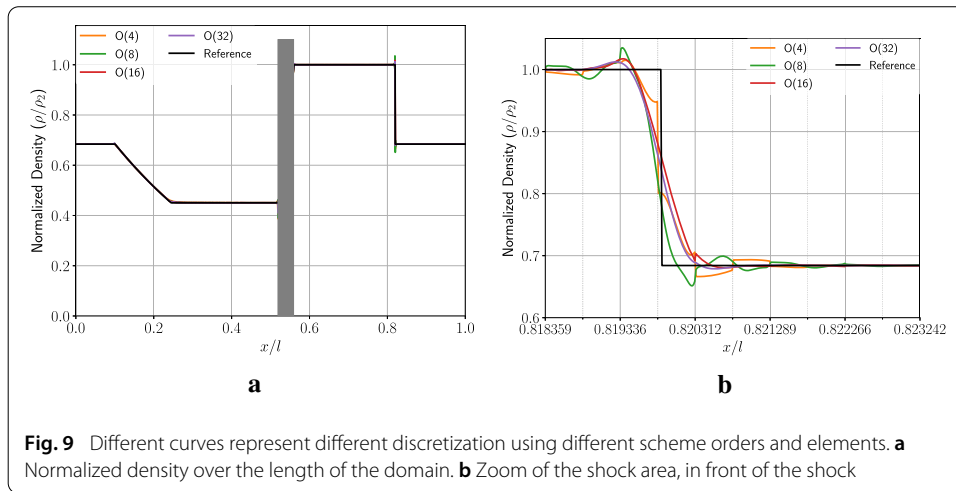
| Test case | p_3/p_2 | Error in p_3/p_2 in [%] | $\Delta x \cdot 10^{-4}$ |
|-------------|------------|---------------------------|--------------------------|
| O(4), n2048 | 1.44845372 | 0.04538827 | 6.17999817 |
| O(8), n1024 | 1.44847062 | 0.04422237 | 4.95925779 |
| O(16), n512 | 1.44858607 | 0.03625527 | 5.56962798 |
| O(32), n256 | 1.44871493 | 0.02736308 | 4.34888760 |

discontinuity in the masking function. As described in Sect. [Representation of immersed boundaries](#), the wall position is not uniformly well represented, which results in some oscillations as the discontinuity of the masking function moves through the stationary mesh. Yet, the error from this effect is relatively small and does not grow with higher scheme orders as shown in Table 2.

Table 2 shows the pressure ratio (p_3/p_2) and the distance between the numerical shock and the exact shock location. The pressure ratio in the numerical solution is obtained by averaging the solution left and right of the shock. Apparently the pressure relation is captured quite well by all scheme orders with this fixed number of 8196 degrees of freedom and we observe an error of less than 0.05% with a decreasing tendency for higher spatial scheme orders. The shock location in the numerical result is assumed to be at the middle point of the jump, and again all simulations provide a good approximation of the reference shock location. Again the highest scheme order yields the smallest error, but the trend is not as clear as in the pressure relation. This can be explained by the approximately same spatial resolution of the wall position in the numerical integration. With the fixed number of degrees of freedom we also have a fixed average density of integration points, which limits the accuracy of the wall position. However, in comparison to the distance between wall and reflected shock (0.25) at this point, the error is again negligibly small and in the largest case of the fourth order approximation less than 0.25% of the distance to the wall.

Formation of shock: moving Piston

In the previous section we investigated the reflection of a shock at a moving wall. Now we want to look into a shock that is formed by a moving wall itself. A piston moves in a tube of fluid at rest. Ahead of it a shock will form and travel away from the moving wall. Behind the piston a suction area will build up with a rarefaction traveling away from the moving wall. This one-dimensional setup is well studied in literature [19] and the exact solution for the inviscid equations is known. For this simulation we use a rigid rectangular piston, modeled by the Brinkman penalization and moving with velocity $v_p = 150$ in a one-dimensional domain. The domain length L is set to 1.0 and the width of the piston is 0.04. Initially, the piston is positioned at 0.4. The fluid is initially at rest, with density $\rho = 1.0 \text{ kg/m}^3$ and pressure $p = 10^5 \text{ Pa}$. A time interval of 0.008s is computed. This setup is particularly significant, as it indicates, whether conservation is maintained by the wall model. A shock will not form, if conservation is not maintained or will have a wrong speed, as a loss of conservation properties occur [19]. We have a closer look into the density and pressure. Important here is the correct capturing of the shock position as well as the pressure and density relation behind and ahead of the formed shock. The exact position of the shock after $t = 0.008\text{s}$ is expected at $x = 0.819870$, whereas the piston interface is located at $x = 0.56$ at the end of the simulation. During post-processing we



apply a smoothing filter that addresses Gibbs oscillations close to the shock [28]. As in the previous Sect. **Reflected shock wave** with the reflected shock, we consider discretizations with different scheme orders (O) but with the same overall number of degrees of freedom. We use spatial schemes of order $O(4)$, $O(8)$, $O(16)$ and $O(32)$ with element counts (n) of $n2048$, $n1024$, $n512$ and $n256$, respectively. Thus, the domain is resolved by 8192 degrees of freedom in each case.

The state of the simulation result after $t = 0.008s$ is shown in Fig. 9 for the density and in Fig. 10 for the pressure. Once for the overall domain in Figs. 9a and 10a and once with close-ups of the shock in Figs. 9b and 10b. The overall image shows that the state is well captured by all scheme orders for this number of degrees of freedom. In the close-ups the shown grid indicates the mesh for the fourth order discretization and discontinuities at the element surfaces can be observed. The discretizations with higher spatial scheme orders span accordingly multiple grid cells in Figs. 9b and 10b. In the run with 32nd order a single element spans 8 of the shown 10 grid cells, all but the outer left and outer right.

As with the shock reflection in the previous section we find that the high order can properly handle both the moving geometry and the discontinuity in the solution. Thus, it

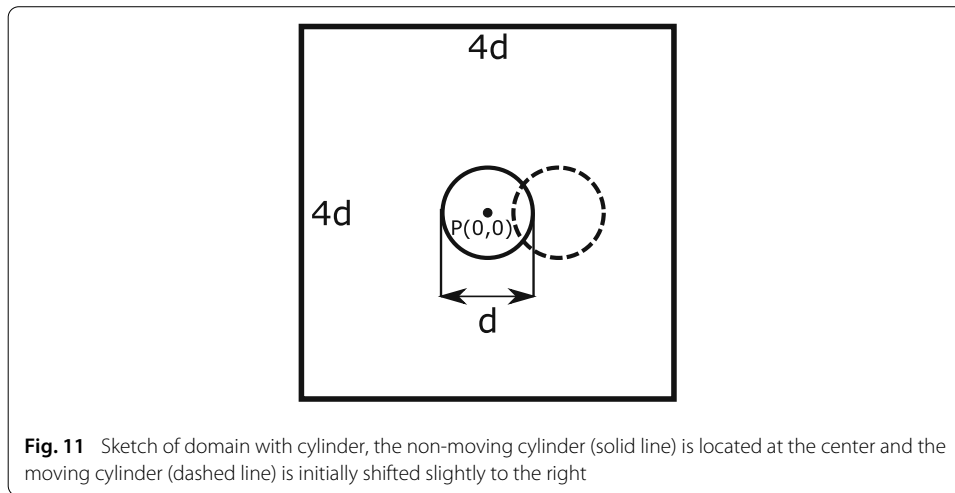


Fig. 11 Sketch of domain with cylinder, the non-moving cylinder (solid line) is located at the center and the moving cylinder (dashed line) is initially shifted slightly to the right

is viable to use a high spatial scheme order in the complete domain, across discontinuities in state and masking function without special measures.

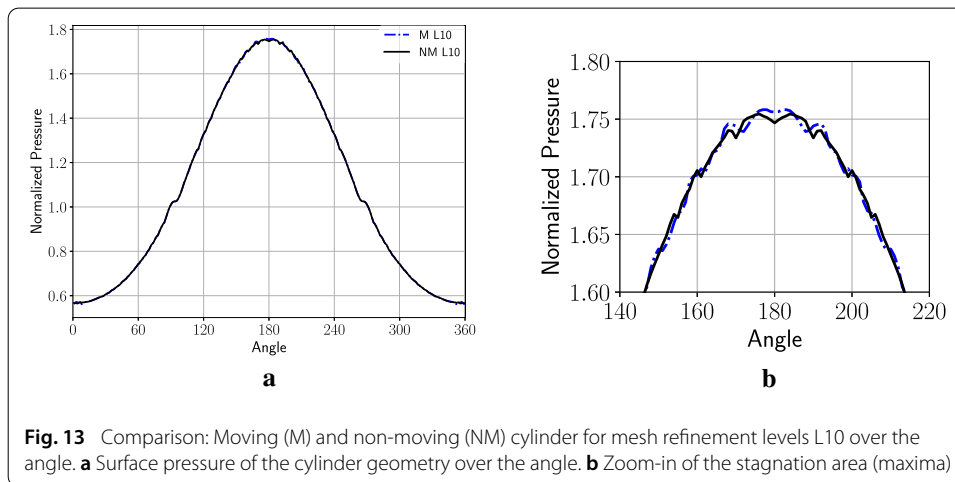
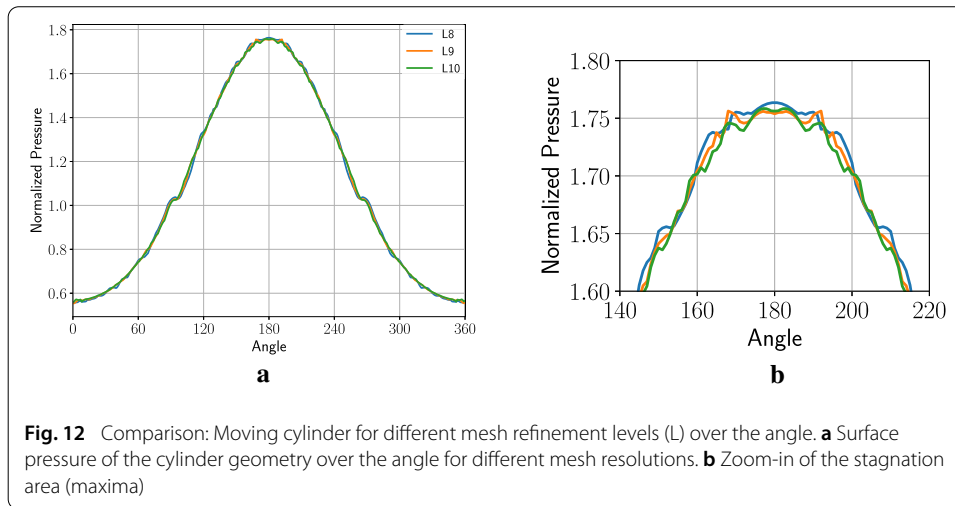
Subsonic cylinder movement

We now move on to a two-dimensional test case, where we consider the movement of a cylinder through a fluid with a low velocity. As a reference for this simulation, we use the simulation of a cylinder at rest within a fluid moving with a higher velocity. In this setup we neglect again the viscosity and only solve the penalized compressible Euler equations. We define the simulation domain with an extent of $[4d \times 4d]$ where d is the diameter of the cylinder. In the reference solution the fixed cylinder is put in the center of the domain. Accordingly, the moving cylinder is put initially shifted from a center such that it reaches the center of the domain after a simulation time of $0.04s$. The relative motion between cylinder and fluid has a Mach number of $Ma = 0.5$. In the reference computation this defines the velocity of the fluid around the cylinder in rest. For the moving cylinder we split the velocity parts and have the cylinder move with half the velocity difference $Ma = 0.25$ upstream through the fluid that has a velocity of $Ma = 0.25$.

Figure 11 sketches the simulation domain. In the case of the moving cylinder, the cylinder is initially slightly shifted towards the right (dashed line cylinder) and moves to the left during the simulation. The fluid moves from left to right in both computations.

In the simulation we use a spatial scheme order of $O(16)$ with three different meshes. The coarsest mesh (L8) uses $64 \times 64 = 4096$, the intermediate (L9) $128 \times 128 = 16384$ and the finest (L10) $256 \times 256 = 65536$ elements. We compare the solutions by measuring the pressure along the surface of the cylinder with a resolution of one degree. In Fig. 12 all three mesh resolutions (L) for the moving cylinder are shown. We can observe how the pressure, normalized by the background pressure converges with finer mesh resolution. For the mesh refinement L9 and L10 the normalized pressure at the stagnation point is almost the same. Additionally, in all three cases we can clearly observe the decrease of the pressure value at the stagnation point, as the resolution becomes better. Thus, also the representation of the cylinder.

In Fig. 13 we compare the solution in normalized pressure for the mesh refinement level L10 for the moving (M) and non-moving (NM) cylinder (cf. Fig. 13a). The pressure



behavior for both cases is very close and in agreement with each other. We can observe a slight shift in the stagnation point between both solutions (cf. Fig. 13b), which is again due to the movement of the cylinder and the different representation in each time step. However, apart from that, both curves show the same behavior in pressure and are in very good agreement.

Supersonic movement

In this section we increase the velocity of the moving cylinder to reach the supersonic regime, where we again see shocks in the flow, but now in a two-dimensional simulation. The simulation domain is rectangular of the form $[L \times H]$ with the length $L = 4 \cdot H$ and H is the height of the domain. The cylinder is $1/4$ of the height of the domain and has a diameter of $d = 0.25$ m. Initially the cylinder is located close to the right boundary (outlet) and moves with Mach 1.5 towards the left boundary (inlet). The fluid has a velocity of Mach 1.5 towards the right, resulting in a total relative velocity between fluid and obstacle of Mach 3. Upper and lower boundaries are set to slip walls. This setup is also described in a report by P. Hu et al. [12]. The mesh has elements of the length $1/128$ resulting in 65536 elements in total. We solve the inviscid Euler equations with a spatial scheme order of six.

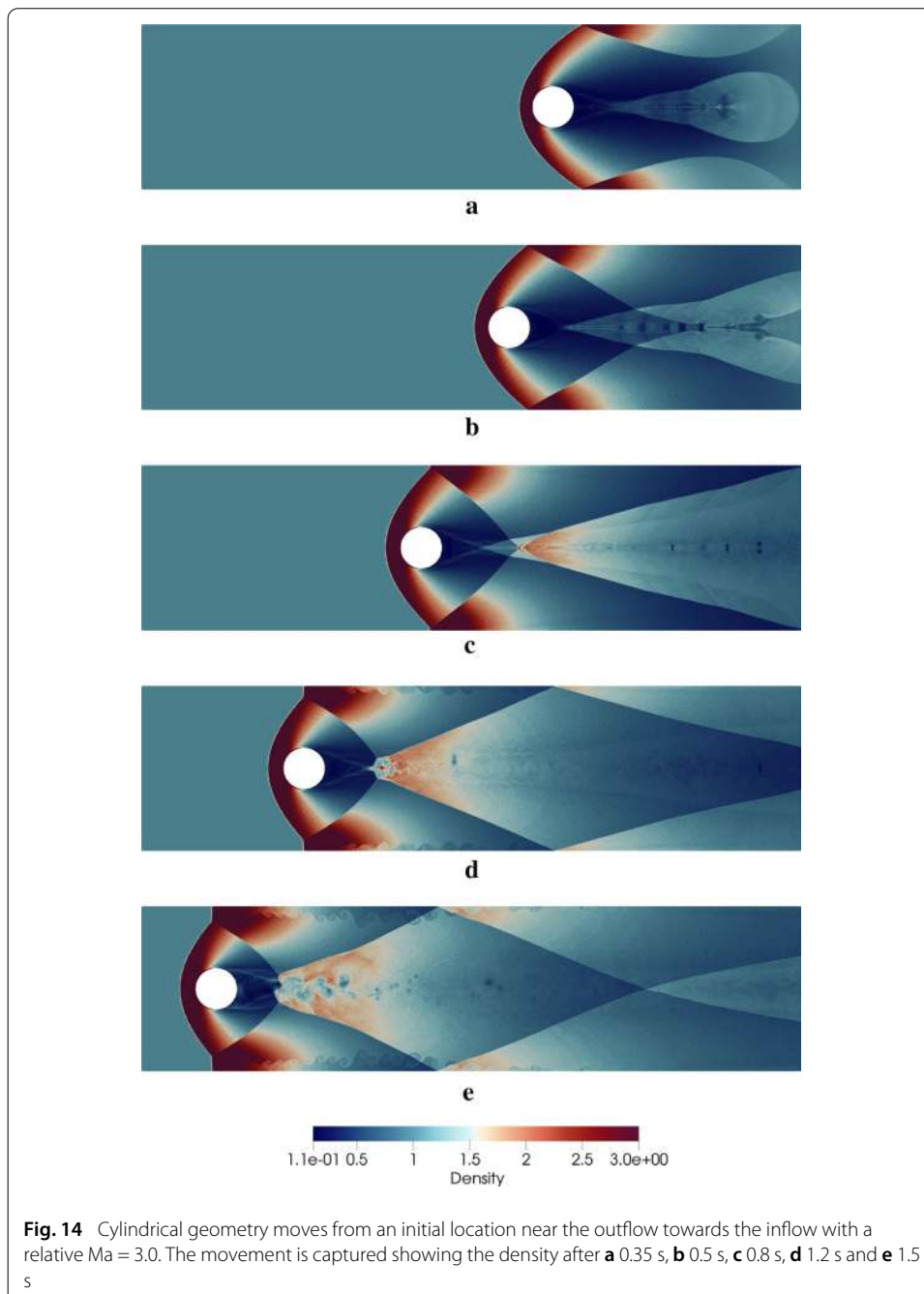
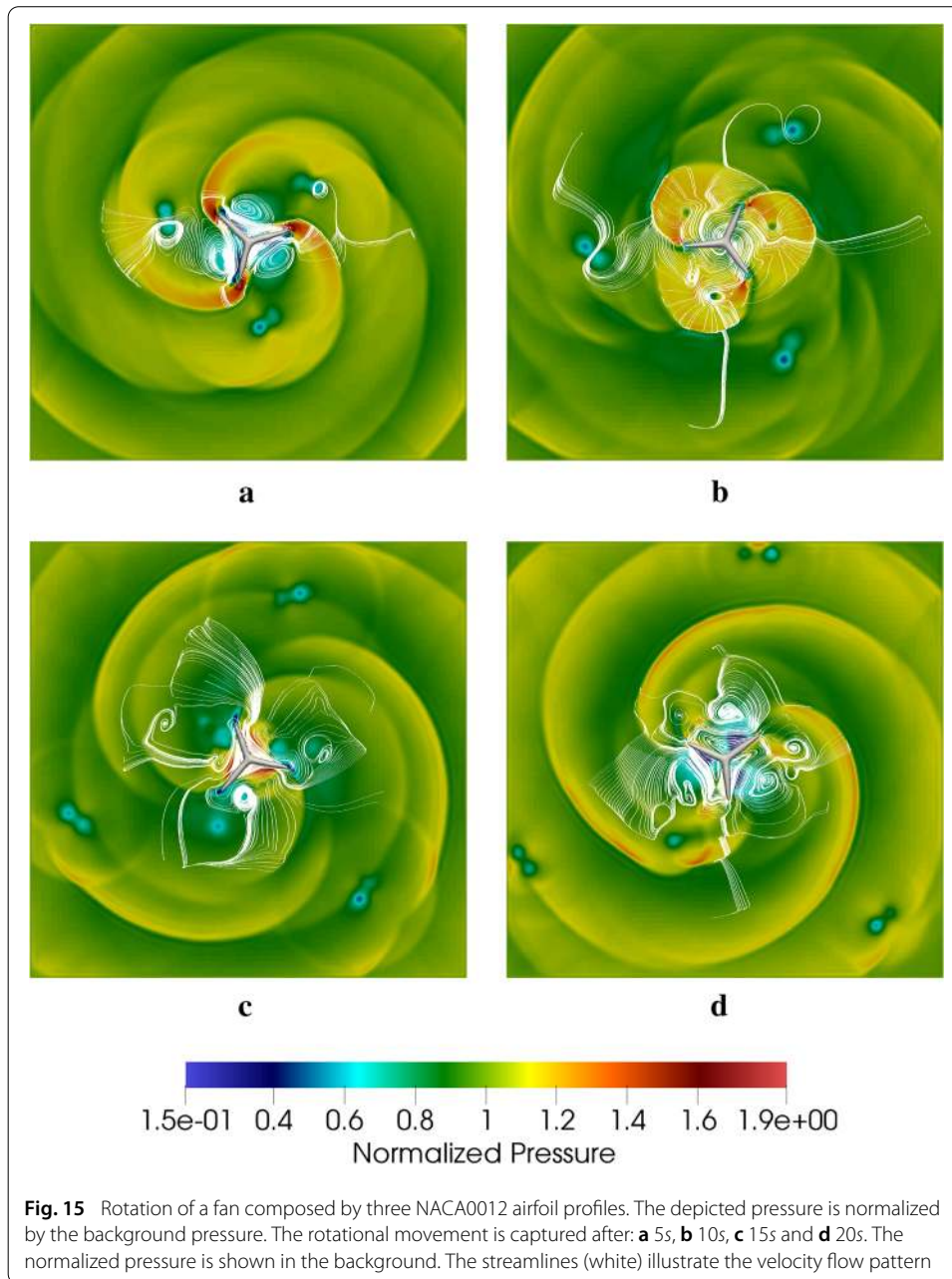


Figure 14 shows the density for various positions of the cylinder at consecutive points in time. Ahead of the cylinder the formation of a bow shock can be observed (cf. Fig. 14a–e). The shock is sharply resolved and drastic change in state can be observed in this supersonic setting. In Fig. 14a we can observe the interaction of the bow shock with the walls. The shock is reflected at the upper and lower wall. Besides the interaction of the shock with the wall, we can observe the interaction of the reflected shock with vortices in the wake of the obstacle (cf. Fig. 14d, e). Further, we can nicely observe the structure of a Richtmyer-Meshkov instability behind the shock along the walls. All figures show a good resolution



of the flow features, as small structures are clearly visible. The cylindrical geometry is well modeled and its movement causes the expected physical phenomena. Our results are in good agreement with the report of Hu et al. [12]. However, we can observe that small scale structures are more clearly resolved in the presented high-order simulation here. This highlights the advantage of high-order methods even with discontinuities present in the solution. It also shows, that the Brinkman penalization can successfully be used in the supersonic regime.

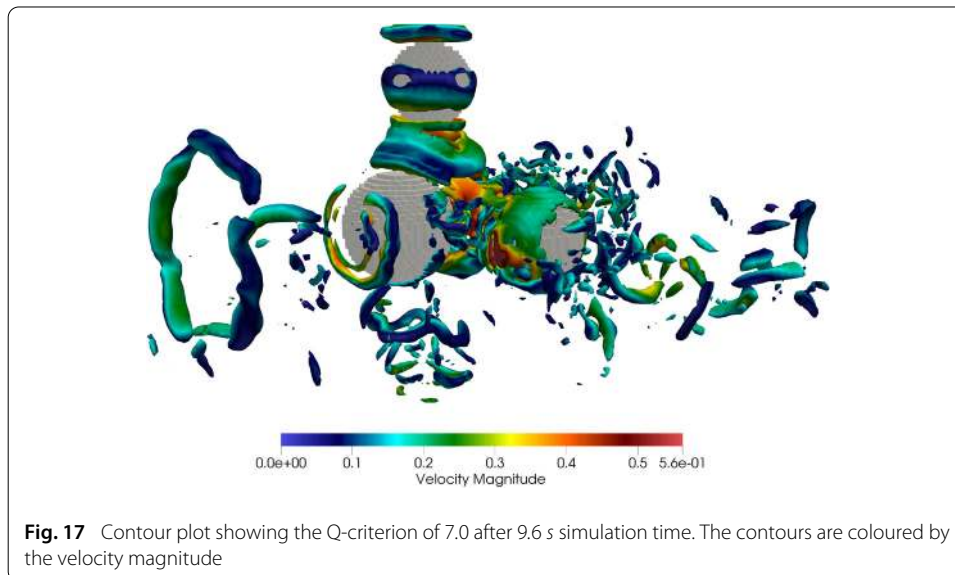
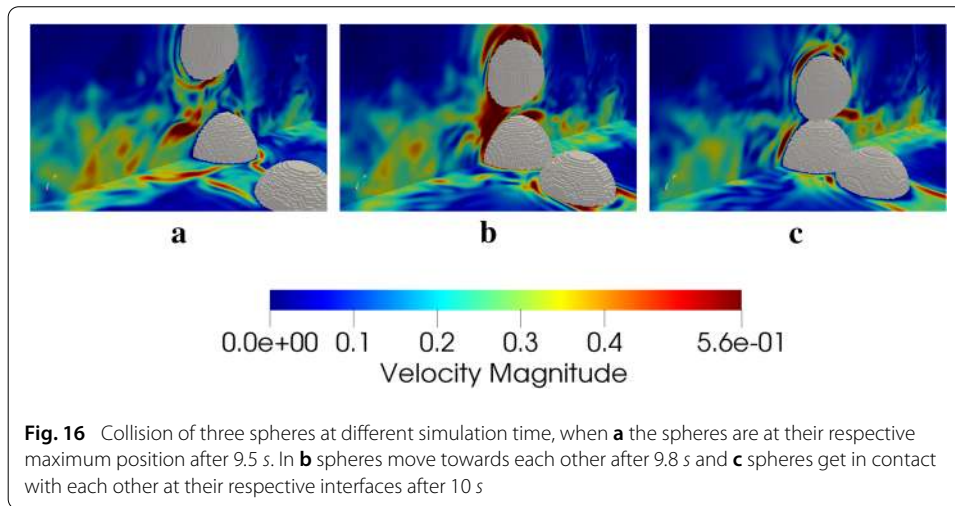
Rotation of a fan

This three dimensional test case is solved with the full compressible viscous Navier-Stokes equations. The setup features a rotating fan, that is composed by three NACA0012 airfoils

with a chord length of $1m$. The first blade is initially positioned at an angle of 90° , the second one at 210° and the third at 330° . The pressure is set to $1bar$, the density to $1kg/m^3$ and the fluid is initially at rest. The fan rotates with a frequency of $6.283Hz$. The Reynolds number with respect to the chord length is $67, 114$. The kinematic viscosity is defined to be $1.49 \cdot 10^{-5}m^2/s$ and the thermal conductivity is set to $0.024W/(m \cdot K)$. All lengths are normalized by the cord length. The computational domain is of size $[10 \times 10 \times 2]$ unit length and the mesh consists of a total of 102400 cubical elements with an edge length of $1/8m$. A spatial scheme order of $O(6)$ is used on this mesh. The fan rotates counter-clockwise around its center point at $P(0, 0, 1)$ of the domain. In Fig. 15 the time evolution of the rotating fan is shown after 5s, 10s, 15s and 20s from Fig. 15a to d respectively. The pressure, normalized by the background pressure is shown as a color field, while white streamlines illustrate the velocity pattern in the vicinity of the rotating obstacle. Three strong vortices appear, that are due to the sudden movement of the fan in the initially not moving fluid. Those are still visible in 15d, but are far from the obstacle at that point and the influence from the initial condition vanishes. Clearly the meeting point of high and low pressures at the tip of the blades can be observed. Resulting pressure waves are well resolved and propagated through the domain forming the depicted spiral pattern. The three vortices from the start-up travel towards the outer boundaries, disturbing the outgoing pressure waves (cf. Fig. 15b to d). Between the fan blades the streamlines illustrate how the flow is captured in circulation areas confined by the blades. The geometry is well represented and the observed behavior is in agreement with the expectations. The simulation was conducted on Germans national supercomputer JUWELS located in Jülich Supercomputing Center using 15k core-h. Each compute node on the system is equipped with 48 cores [14].

Collision of three spheres

The three-dimensional test case is solved with the compressible Euler equations. The simulation domain has a length of $L = 4m$, a height of $H = 4m$ and a width of $W = 1.5m$. Three spheres are located in the domain, that are at the beginning of the simulation in contact with each other at their respective interfaces. The first sphere is located at $S_1(-0.1, 0.0, 0.0)$ and has a radius of $0.2m$, the second and third sphere have a radius of $0.15m$ and are located at $S_2(0.0, 0.45, 0.0)$ and $S_3(0.0, 0.0, -0.45)$ respectively. All spheres move with a predefined cosine function, away from each other and towards each other, that is defined as $A \cdot \cos 2\pi t$ with A being the amplitude, which is 0.1 . The Euler equations are solved with a scheme order of $O(8)$. The fluid is in rest, the pressure is defined to be $1bar$ and the density is $1kg/m^3$. The computational mesh has 98304 elements, with an edge length of $1/16m$. The simulation is carried out for $10s$. In z -direction the computational domain is periodic. In Figure 16 different positions of the spheres at different points in time are shown. Figure 16a illustrates the spheres at their maximum position, with the largest distance to each other after $9.5s$. The spheres move from that position again back to their original location, that is shown in Fig. 16b (after $9.8s$). As expected, the fluid is compressed between the spheres, resulting in high velocity value on the right and left of the spheres, as the fluid is squeezed out from the region between them. Finally, the spheres reach their respective position in Fig. 16, where their interfaces get in contact with each. It needs to be highlighted, that the numerical scheme is able to handle the



collision of objects, without further stability restrictions. Furthermore, the presence of multiple moving geometries can be handled accordingly by the solver and the method used to model the geometries.

In Fig. 17 the Q-criterion of 7.0 is shown for the time 9.6 s, when the spheres move again back to their origin. Again, high velocity values can be observed in the region between the spheres, where the fluid is compressed. Small structures appear close and away from the spheres, where the fluid is disturbed by the movement of the three geometries. Please note that simulation results provided by the solver are represented by polynomial series. For visualization purposes this representation is voxelized during post-processing. Therefore, the spheres in Fig. 16a appear relatively rough, as the voxelization was done in a coarse resolution for the geometry here, while the sliced data of the flow state is resolved to a finer level.

Conclusion

We applied the Brinkman penalization method in our modal discontinuous Galerkin implementation Ateles and the presented analysis shows that it offers an effective method to model moving obstacles. While the convergence order breaks down at the discontinuity of the masking function for the penalization, the accuracy is still high in the remaining domain and overall not worse than with a lower order discretization. Thus, the complete domain can be computed with the same discretization without the need to adapt to the moving obstacles. The Brinkman penalization scheme worked in our implementation for smooth flow states like a reflected acoustic wave as well as for shocks. It is working in the subsonic and supersonic regimes and independent of the location of obstacles relative to the mesh. Small and closing gaps between moving parts can effortlessly modelled up to the contact of obstacles.

We observe small scale oscillations in the solution that are not present in simulations with fixed walls, but they remain small and decrease with better resolutions. An over-integration in the representation of the masking function is necessary and we found that an over-integration by a factor of three is sufficient to mostly recover an accurate geometrical approximation. The presented method enables us to model moving geometries of arbitrary shape within a high-order discontinuous Galerkin scheme and provides a large flexibility in its application.

Further improvements could be offered by incorporating re-projection methods that promise the reconstruction of error convergence even in the vicinity of discontinuities e.g. in [8].

Acknowledgements

Simulation results were obtained on the Noctua system at Paderborn Center for Parallel Computing (PC2) and the Supermuc-NG supercomputer at Leibniz Rechenzentrum (LRZ) of Bayerischen Akademie der Wissenschaften. The authors wish to thank for the computing time and the technical support.

Authors' contributions

Conceptualization by NEP and HK; NEP wrote the original draft preparation and NA contributed the pulse reflection setup; all authors were involved in the review and editing process; NA, NEP and HK worked on the methodology; NEP and HK worked on the used software; investigation and validation was carried out by NA and NEP, they also did the visualization to produce the graphs and images. Supervision and funding acquisition by SR. All authors read and approved the final manuscript.

Funding

Open Access funding enabled and organized by Projekt DEAL. N. Ebrahimi Pour was financially supported by the priority program 1648-Software for Exascale Computing 214 (www.sppexa.de) of the Deutsche Forschungsgemeinschaft.

Availability of data

All configuration files for the shown test cases are available in [6] and can be run with our open-source simulation framework APES [25] for pre- and post-processing.

Declarations

Competing interests

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix

The discontinuous Galerkin discretization

This section is devoted to a brief review, of the semi-discrete form of the discontinuous Galerkin Finite Element method (DG) for compressible flows. For brevity we restrict this

discussion to the inviscid part. The viscous terms are discretized using the DG method with symmetric interior penalty [11]. We refer to the equations obtained by neglecting diffusive terms from the compressible Navier-Stokes equations as the Euler equations. They provide a model for the conservation of mass, momentum and energy of the fluid and are defined in vectorial notation by

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

with respective initial and boundary conditions. \mathbf{u} is a vector, that includes all variables to be conserved. Further the flux function $\mathbf{F}(\mathbf{u}) = (\mathbf{f}(\mathbf{u}), \mathbf{g}(\mathbf{u}))^T$ is defined for the spatial dimensions and is in 2D given by

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \quad \mathbf{f}(\mathbf{u}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho E + p)u \end{bmatrix}, \quad \mathbf{g}(\mathbf{u}) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho E + p)v \end{bmatrix},$$

with density ρ , velocity $\mathbf{v} = (u, v)^T$, specific total energy E , and pressure p . The system is closed by the equation of state. Which for an ideal gas provides $p = (\gamma - 1)\rho (e - \frac{1}{2}(u^2 + v^2))$. Here $\gamma = \frac{c_p}{c_v}$ is the ratio of specific heat capacities and e is the total internal energy per unit mass.

The discontinuous Galerkin formulation of the mentioned equation is obtained, by multiplication with a test function ψ and integration over the domain Ω . Afterward, we apply integration by parts and obtain the weak formulation

$$\int_{\Omega} \psi \frac{\partial \mathbf{u}}{\partial t} d\Omega + \oint_{\partial\Omega} \psi \mathbf{F}(\mathbf{u}) \cdot \mathbf{n} ds - \int_{\Omega} \nabla \psi \cdot \mathbf{F}(\mathbf{u}) d\Omega = 0, \quad \forall \psi. \tag{14}$$

Where ds denotes a segment on the surface integral. The overall domain Ω is split into smaller, n non-overlapping elements $\Omega_i | i = 1, 2, \dots, n$, such that $\Omega = \cup_{i=1}^n \Omega_i$ and $\Omega_i \cap \Omega_j = \emptyset \forall i \neq j$. Within in each of these elements a polynomial series P^m with a maximal degree of $m \geq 0$ is used to approximate the solution \mathbf{u} . We refer to this approximate solution as $\mathbf{u}_h(\mathbf{x}, t)$, in each element i .

$$\mathbf{u}_h(\mathbf{x}, t) = \sum_{i=0}^m \hat{\mathbf{u}}_i \phi_i \tag{15}$$

The expansion coefficients $\hat{\mathbf{u}}_i$ are the degrees of freedom of the numerical solution. It is important to keep in mind, that no global continuity requirement for \mathbf{u}_h in the previous definition is needed. Dividing the integrals in Eq. (14) into a sum of integrals over elements Ω_i , we can obtain the space-discrete variational formulation with test functions ψ_j in each element

$$\frac{\partial}{\partial t} \int_{\Omega_i} \psi_j \mathbf{u}_h d\Omega + \oint_{\partial\Omega_i} \psi_j \mathbf{F}(\mathbf{u}_h) \cdot \mathbf{n} ds - \int_{\Omega_i} \nabla \psi_j \cdot \mathbf{F}(\mathbf{u}_h) d\Omega = 0, \quad \forall \psi_j. \tag{16}$$

By using $m + 1$ linearly independent test functions ψ_0, \dots, ψ_m we arrive at a fully determined linear system of $m + 1$ equations for the $m + 1$ degrees of freedom. The appearing inner products of ϕ_i and ψ_j , can be precomputed and we can represent the system in the form of matrices that are multiplied with the degrees of freedom $\hat{\mathbf{u}}$ representing the state. Since the numerical representation is only supported locally, the flux term is not uniquely defined at the element interfaces. Thus, the flux function has to be replaced by an approximative numerical flux function $\mathbf{F}^*(\mathbf{u}_h^-, \mathbf{u}_h^+, \mathbf{n})$, where \mathbf{u}_h^- and \mathbf{u}_h^+ are the interior and exterior traces at the element face in the normal direction \mathbf{n} to the interface.

To simplify the above equation, we can re-write it in a matrix vector notation as

$$\frac{\partial}{\partial t} \hat{\mathbf{u}} = M^{-1} \left(S \cdot \mathbf{F}(\hat{\mathbf{u}}) - M^F \cdot \mathbf{F}(\hat{\mathbf{u}}) \right) =: rhs(\hat{\mathbf{u}}). \quad (17)$$

Hereby M, S are the mass and the stiffness matrices and M^F are the so called face mass lumping matrices. The computational cost to perform these operations obviously depends on the choice of basis functions. Using the orthogonality of Legendre polynomials the mass matrix gets diagonalized and trivially invertible, such that the multiplication with M^{-1} can be performed in $\mathcal{O}(m + 1)$ operations. The stiffness matrix is not as trivially applied, however due to the recursive properties of the Legendre polynomials it can still be computed in $\mathcal{O}(m + 1)$ operations by exploiting this recursiveness.

The obtained ordinary differential equations (17) can be solved in time using any standard time-stepping method, e.g. a Runge-Kutta method [2]. In our DG discretization, we usually employ the HLL flux as numerical flux.

Received: 9 October 2020 Accepted: 1 April 2021

Published online: 27 April 2021

References

- Alexander R. Diagonally implicit runge-kutta methods for stiff O.D.E.'s. *SIAM J Numer Anal.* 1977;14(6):1006–21. <https://doi.org/10.1137/0714068>.
- Anand N, Ebrahimi Pour N, Klimach H, Roller S. Utilization of the brinkman penalization to represent geometries in a high-order discontinuous galerkin scheme on octree meshes. *Symmetry.* 2019;11(9):1126. <https://doi.org/10.3390/sym11091126>.
- Ben-Dor G, Igra O, Elperin T, editors. *Handbook of shock waves*, vol. 3. 1st ed. San Diego: Academic Press; 2000.
- Brown-Dymkoski E, Kasimov N, Vasilyev OV. A characteristic based volume penalization method for general evolution problems applied to compressible viscous flows. *J Comput Phys.* 2014;262:344–57. <https://doi.org/10.1016/j.jcp.2013.12.060>.
- E. Arquis JC. Sur les conditions hydrodynamiques au voisinage d'une interface milieu fluide-milieu poreux: application a' la convection naturelle. *C.R. Acad. Sci. Paris II* 299; 1984, p. 1–4.
- Ebrahimi Pour N, Anand N. Configuration files for the simulation; 2020. <https://geb.sts.nt.uni-siegen.de/EbrahimiPour-moving-config.zip>. Accessed 08 Oct 2020.
- Engels T, Kolomenskiy D, Schneider K, Sesterhenn J. Flusi: a novel parallel simulation tool for flapping insect flight using a fourier method with volume penalization; 2015.
- Gelb A, Tanner J. Robust reprojection methods for the resolution of the gibbs phenomenon. *Appl Comput Harmonic Anal.* 2006;20(1):3–25. <https://doi.org/10.1016/j.jacha.2004.12.007>. <http://www.sciencedirect.com/science/article/pii/S1063520305000588>. *Computational Harmonic Analysis—Part 2*.
- Glazer E, Sadot O, Hadjadj A, Chaudhuri A. Velocity scaling of a shock wave reflected off a circular cylinder. *Phys Rev E.* 2011;83(6):066317. <https://doi.org/10.1103/PhysRevE.83.066317>.
- Gottlieb D, Shu C. On the Gibbs phenomenon and its resolution. *SIAM Rev.* 1997;39(4):644–68. <https://doi.org/10.1137/S0036144596301390>.
- Hartmann R, Houston P. Symmetric interior penalty dg methods for the compressible navier-stokes equations ii: goal-oriented a posteriori error estimation. *Int J Numer Anal Model* 2006;3(2):141–62. http://global-sci.org/intro/article_detail/ijnam/894.html.
- Hu PC, Flaherty JE, Shephard MS. Solving fluid / rigid body interaction problem by a discontinuous galerkin level set method; 2005.
- Jause-Labert C, Godefert F, Favier B. Numerical validation of the volume penalization method in three-dimensional pseudo-spectral simulations. *Comput Fluids.* 2012;67:41–56. <https://doi.org/10.1016/j.compfluid.2012.06.026>.
- Jülich Supercomputing Centre: JUWELS: Modular Tier-0/1 Supercomputer at the Jülich Supercomputing Centre. *Journal of large-scale research facilities* 2019;5(A135). <https://doi.org/10.17815/jlsrf-5-171>.
- Kevlahan NKR, Ghidaglia JM. Computation of turbulent flow past an array of cylinders using a spectral method with brinkman penalization. *Eur J Mech B/Fluids.* 2001;20(3):333–50. [https://doi.org/10.1016/s0997-7546\(00\)01121-3](https://doi.org/10.1016/s0997-7546(00)01121-3).
- Kirby RM, Karniadakis GE. De-aliasing on non-uniform grids: algorithms and applications. *J Comput Phys.* 2003;191(1):249–64. [https://doi.org/10.1016/s0021-9991\(03\)00314-0](https://doi.org/10.1016/s0021-9991(03)00314-0).
- Klimach HG, Zudrop J, Roller SP. Generation of high order geometry representations in octree meshes. *Peer J Comput Sci.* 2015;1:e35. <https://doi.org/10.7717/peerj-cs.35>.
- Komatsu R, Iwakami W, Hattori Y. Direct numerical simulation of aeroacoustic sound by volume penalization method. *Comput Fluids.* 2016;130:24–36. <https://doi.org/10.1016/j.compfluid.2016.02.016>.
- Liepmann H, Roshko A. *Elements of gasdynamics galcit aeronautical series*. Hoboken: Wiley; 1957.
- Liu Q, Vasilyev OV. A brinkman penalization method for compressible flows in complex geometries. *J Comput Phys.* 2007;227(2):946–66. <https://doi.org/10.1016/j.jcp.2007.07.037>.
- Maxey M. Simulation methods for particulate flows and concentrated suspensions. *Annu Rev Fluid Mech.* 2017;49(1):171–93. <https://doi.org/10.1146/annurev-fluid-122414-034408>.
- Mittal R, Iaccarino G. Immersed boundary methods. *Annu Rev Fluid Mech.* 2005;37(1):239–61. <https://doi.org/10.1146/annurev.fluid.37.061903.175743>.

23. Pasquetti R, Bwemba R, Cousin L. A pseudo-penalization method for high reynolds number unsteady flows. *Appl Numer Math.* 2008;58(7):946–54. <https://doi.org/10.1016/j.apnum.2007.04.011>.
24. Ramière I, Angot P, Belliard M. A fictitious domain approach with spread interface for elliptic problems with general boundary conditions. *Comput Methods Appl Mech Eng.* 2007;196(4–6):766–81. <https://doi.org/10.1016/j.cma.2006.05.012>.
25. Simulationstechnik und Wissenschaftliches Rechnen Uni Siegen: Apes simulation framework; 2019. <http://apes.osdn.io>. Accessed 08 Oct 2020.
26. Simulationstechnik und Wissenschaftliches Rechnen Uni Siegen: Ateles source code; 2019. <https://osdn.net/projects/apes/scm/hg/ateles/>. Accessed 26 Aug 2019.
27. Sotiropoulos F, Yang X. Immersed boundary methods for simulating fluid-structure interaction. *Progress Aerospace Sci.* 2014;65:1–21. <https://doi.org/10.1016/j.paerosci.2013.09.003>. <http://www.sciencedirect.com/science/article/pii/S0376042113000870>.
28. Zudrop J. Efficient numerical methods for fluid- and electrodynamics on massively parallel systems. Ph.D. thesis, RWTH Aachen University; 2015.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
