

ExaFSA: Parallel Fluid-Structure-Acoustic Simulation



Florian Lindner, Amin Totounferoush, Miriam Mehl, Benjamin Uekermann, Neda Ebrahimi Pour, Verena Krupp, Sabine Roller, Thorsten Reimann, Dörte C. Sternel, Ryusuke Egawa, Hiroyuki Takizawa, and Frédéric Simonis

Abstract In this paper, we present results of the second phase of the project ExaFSA within the priority program SPP1648—Software for Exascale Computing. Our task was to establish a simulation environment consisting of specialized highly efficient and scalable solvers for the involved physical aspects with a particular focus on the computationally challenging simulation of turbulent flow and propagation of the induced acoustic perturbations. These solvers are then coupled in a modular, robust, numerically efficient and fully parallel way, via the open source coupling library preCICE. Whereas we made a first proof of concept for a three-field simulation (elastic structure, surrounding turbulent acoustic flow in the near-field, and pure acoustic wave propagation in the far-field) in the first phase, we removed several scalability limits in the second phase. In particular, we present new contributions to (a) the initialization of communication between processes of

F. Lindner · A. Totounferoush · M. Mehl (✉)

University of Stuttgart, Stuttgart, Germany

e-mail: florian.lindner@ipvs.uni-stuttgart.de; amin.totounferoush@ipvs.uni-stuttgart.de; miriam.mehl@ipvs.uni-stuttgart.de

B. Uekermann

Eindhoven University of Technology, Eindhoven, Netherlands

e-mail: b.w.uekermann@tue.nl

N. Ebrahimi Pour · V. Krupp · S. Roller

University of Siegen, Siegen, Germany

e-mail: neda.epour@uni-siegen.de; sabine.roller@uni-siegen.de

T. Reimann · D. C. Sternel

Technische Universität Darmstadt, Darmstadt, Germany

e-mail: thorsten.reimann@sc.tu-darmstadt.de; doerte.sternel@hpc-hessen.de

R. Egawa · H. Takizawa

Tohoku University, Sendai, Japan

e-mail: takizawa@tohoku.ac.jp

F. Simonis

Technical University of Munich, München, Germany

e-mail: simonis@in.tum.de

© The Author(s) 2020

H.-J. Bungartz et al. (eds.), *Software for Exascale Computing - SPPEXA*

2016–2019, Lecture Notes in Computational Science and Engineering 136,

https://doi.org/10.1007/978-3-030-47956-5_10

the involved independent solvers, (b) optimization of the parallelization of data mapping, (c) solver-specific white-box data mapping providing higher efficiency but less flexibility, (d) portability and scalability of the flow and acoustic solvers FASTEST and Ateles on vector architectures by means of code transformation, (e) physically correct information transfer between near-field acoustic flow and far-field acoustic propagation.

1 Introduction

The simulation of fluid-structure-acoustic interactions is a typical example for multi-physics simulations. Two fundamentally different physical sound sources can be distinguished: structural noise and flow-induced noise. As we are interested in accurate results for the resulting sound emissions induced from the turbulent flow, it is decisive to include not only the turbulent flow, but also the structure deformation and the interaction between both. High accuracy requires the use of highly resolved grids. As a consequence, the use of massively parallel supercomputers is inevitable. When we are interested in the sound effects far away from a flow induced fluttering structure, the simulation becomes too expensive, even for supercomputing architectures. Hence, we introduce an assumption, we call it the “far-field”. Far from the structure and, thus, the noise generation, we assume a homogeneous background flow and restrict the simulation in this part of the domain to the propagation of acoustic waves. This results in an overall setup with two coupling surfaces—between the elastic structure and the surrounding flow, and between the near-field and the far-field in the flow domain (see Fig. 1 for an illustrative example). Such a complex simulation environment implies several new challenges compared to

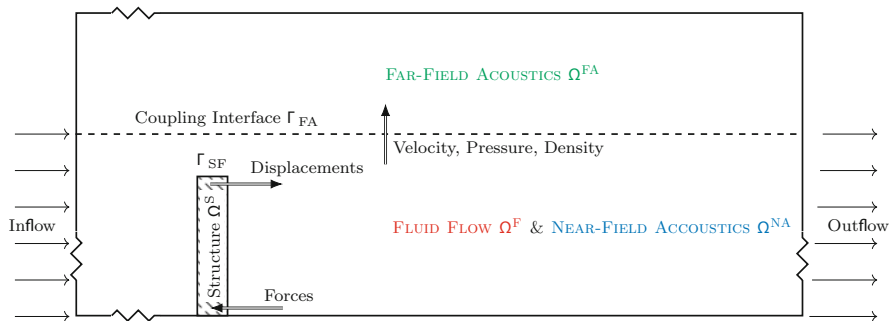


Fig. 1 Multiphysics fluid-structure-acoustic scenario as used in our simulations in Sect. 6. The domain is decomposed into a near-field ‘incompressible flow region’ $\Omega^F = \Omega^{\text{NA}}$, a far-field ‘acoustic only region’ Ω^{FA} , and an ‘elastic structure region’ Ω^S . Note that the geometry is not scaled correctly for better illustration

“single-physics” simulations: (a) multi-scale properties in space and time (small-scale processes around the structure, multi-scale turbulent flow in the near-field, and large-scale processes in the acoustic far-field), (b) different optimal discretization and solver choices for the three fields, (c) highly ill-conditioned problem, if formulated and discretized as a single large system of equations, (d) challenging load-balancing due to different computational load per grid unit depending on the local physics.

Application examples for fluid-structure-acoustic simulations can be found in several technical systems: wind power plants, fans in air conditioning systems of buildings, cars or airplanes, car mirrors and other design details of a car frame, turbines, airfoil design, etc.

Fluid-structure interaction simulations as a sub-problem of our target system have been in the focus of research in computational engineering for many years, mainly aiming at capturing stresses in the structure more realistically than with a pure flow simulation. A main point of discussion in this field is the question whether monolithic approaches—treating the coupled problem as a single large system of equations—or partitioned methods—glueing together separate simulation modules for structures and fluid flow by means of suitable coupling numerics and tools—are more appropriate and efficient. Monolithic approaches require a new implementation of the simulation code as well as the development of specialized iterative solvers for the ill-conditioned overall system of equations, but can achieve very high efficiency and accuracy [3, 12, 19, 23, 38]. Partitioned approaches, on the other hand, offer large flexibility in choosing optimal solvers for each field, adding additional fields, or exchanging solvers. The difficulty here lies in both a stable, accurate, and efficient coupling between independent solvers applying different numerical methods and in establishing efficient communication and load balancing between the used parallel codes. For numerical coupling, numerous efficient data mapping methods [5, 26, 27, 32] have been published along with efficient iterative solvers [2, 7, 13, 20, 29, 35, 39, 41]. In [6], various monolithic and partitioned approaches have been proposed and evaluated in terms of a common benchmark problem. Three-field fluid-structure-acoustic interaction in the literature has so far been restricted to near-field simulations due to the intense computational load [28, 33].

To realize a three-field fluid-structure-acoustic interaction including the far-field, we use a partitioned approach and couple existing established “single-physics” solvers in a black-box fashion. We couple the finite volume solver FASTEST [18], the discontinuous Galerkin solver Ateles [42], and the finite element solver CalculiX [14] by means of the coupling library preCICE [8]. We compare this approach to a less flexible white-box coupling implemented in APESmate [15] as part of the APES framework and make use of the common data-structure within APES [31]. The assumption which is confirmed in this paper is, that the white-box approach is more efficient, but puts some strict requirements on the codes to be coupled, while the black-box approach is a bit less efficient, but much more flexible with respect to

the codes that can be used. Our contributions to the field of fluid-structure-acoustic interaction, which we summarize in this paper, include:

1. For the near-field flow, we introduce a volume coupling between background flow and acoustic perturbations in FASTEST accounting for the multi-scale properties in space and time by means of different spatial and time resolution.
2. For both near-field flow and far-field acoustics, we achieved portability and performance optimization of Ateles and FASTEST for vector machines by means of code transformation.
3. In terms of inter-field coupling, we
 - (a) increased the efficiency of inter-code communication by means of a new hierarchical implementation of communication initialization and a modified communicator concept,
 - (b) we improved the robustness and efficiency of radial basis function mapping,
 - (c) we identified correct interface conditions between near-field and far-field, optimized the position of the interface, and ensured correct boundary conditions by overlapping near-field and far-field,
 - (d) we developed and implemented implicit quasi-Newton coupling numerics that allow for a simultaneous execution of all involved solvers.
4. For a substantially improved inter-code load balancing, we use a regression-based performance model for all involved solvers and perform an optimization of assigned cores.
5. We present a comparison of our black-box and to the white-box approach for multi-physics coupling.

These contributions have been achieved as a result of the project ExaFSA—a cooperation between the Technische Universität Darmstadt, the University of Siegen, the University of Stuttgart, and the Tohoku University (Japan) in the Priority Program SPP 1648—Software for Exascale Computing of the German Research Foundation (DFG) in close collaboration with the Technical University of Munich. In the first funding phase (2013–2016), we showed that efficient yet robust coupled simulations are feasible and can be enhanced with an in-situ visualization component as an additional software part, but we still reached limits in terms of scalability and load balancing [4, 9]. This paper focuses on results of the second funding phase (2016–2019) and demonstrates significant improvements in scalability and accuracy as well as robustness based on the above-mentioned contributions.

In the following, we introduce the underlying model equations of our target scenarios in Sect. 2 and present our solvers and their optimization in Sect. 3 as well as the black-box coupling approach and new contributions in terms of coupling in Sect. 4. In Sect. 5, we compare black-box coupling to an alternative, efficient, but solver-specific and, thus, less flexible white-box coupling for uni-directional flow-acoustic coupling. Finally, results for a turbulent flow over a fence scenario are presented in Sect. 6.

2 Model

In this section, we shortly introduce the underlying flow, acoustic and structure models of our target application. We use the Einstein summation convention throughout this section.

2.1 Governing Equations

The multi-physics scenario we investigate describes an elastic structure embedded in a turbulent flow field. The latter is artificially decomposed into a near-field and a far-field. See Fig. 1 for an example.

Near-Field Flow In the near-field region $\Omega^F = \Omega^{\text{NA}}$, the compressible fluid flow is modeled by means of the density $\bar{\rho}$, the velocity \bar{u}_i and the pressure \bar{p} . As we focus on a low Mach number regime, we can split these variables into an incompressible part ρ, u_i, p , and acoustic perturbations ρ', u'_i, p' :

$$\bar{\rho} = \rho + \rho', \quad \bar{u}_i = u_i + u'_i, \quad \bar{p} = p + p'. \quad (1)$$

The incompressible flow is described by the Navier-Stokes equations¹

$$\frac{\partial u_i}{\partial x_i} = 0, \quad (2a)$$

$$\frac{\partial}{\partial t} (\rho u_i) + \frac{\partial}{\partial x_j} (\rho u_i u_j - \tau_{ij}) = \rho f_i, \quad (2b)$$

where ρ is the density of the fluid, and f_i summarizes external force density terms. The incompressible stress tensor τ_{ij} for a Newtonian fluid is described by

$$\tau_{ij} = -p\delta_{ij} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \quad (3)$$

with μ representing the dynamic viscosity and δ_{ij} the Kronecker-Delta.

¹To capture the moving structure within the near-field, we actually formulate all near-field equations in an arbitrary Lagrangian-Eulerian perspective. For the relative mesh velocity, we use a block-wise elliptic mesh movement as described in [30]. As we do not show fluid-structure interaction in this contribution, however, we formulate all near-field equations in a pure Eulerian perspective for the sake of simplicity.

Acoustic Wave Propagation The propagation of acoustic perturbations in both the near-field and the far-field is modeled by the linearized Euler equations, where in the far-field a constant background state is assumed (which implies $\frac{\partial p}{\partial t} = 0$).

$$\frac{\partial \rho'}{\partial t} + \rho \frac{\partial u'_i}{\partial x_i} + u_i \frac{\partial \rho'}{\partial x_i} = 0 \quad (4a)$$

$$\rho \frac{\partial u'_i}{\partial t} + \rho u_j \frac{\partial u'_i}{\partial x_j} + \frac{\partial p'}{\partial x_i} = 0 \quad (4b)$$

$$\frac{\partial p'}{\partial t} + \rho c^2 \frac{\partial u'_i}{\partial x_i} + u_i \frac{\partial p'}{\partial x_i} = -\frac{\partial p}{\partial t} \quad (4c)$$

Here c is the speed of sound. In the near-field, the background flow quantities u_i and p are calculated from (2), whereas they are assumed to be constant in the acoustic far-field. The respective constant value is read from the coupling interface with the near-field, which implies that the interface has to be chosen such that the background flow values are (almost) constant at the coupling interface. In both cases, the coupling between background-flow and acoustic perturbations is unidirectional from the background flow to the acoustic equations (4) by means of p and u_i .

Elastic Structure The structural subdomain Ω^S is governed by the equations of motion, here in Lagrangian description:

$$\rho^S \frac{\partial^2 \vartheta_i}{\partial t^2} = \frac{\partial S_{jk} F_{ik}}{\partial X_j^S} + \rho^S f_i^S. \quad (5)$$

With $x_i^S = X_i^S + \vartheta_i$ being the position of a particle in the current configuration, X_i^S is the position of a particle in the reference configuration, and ϑ_i the displacement. F_{ij} is the deformation gradient. S_{ij} is the second Piola-Kirchhoff tensor, and ρ^S describes the structural density. The Cauchy stress tensor τ_{ij}^S relates to S_{ij} via

$$\tau_{ij}^S = \frac{1}{\det(F_{ij})} F_{ik} S_{kl} F_{jl}. \quad (6)$$

We assume linear elasticity to describe the stress-strain relation.

The coupling between fluid and structure is bi-directional by means of dynamic and kinematic conditions, i.e., equality of interface displacements/velocities and stresses, i.e.,

$$u_i^{\Gamma^F} = \frac{\partial \vartheta_i^{\Gamma^S}}{\partial t}, \quad \tau_{ij}^{\Gamma^F} = \tau_{ij}^{\Gamma^S} \quad (7)$$

at $\Gamma^I = \Gamma^S \cap \Gamma^F$ with $\Gamma^F = \partial\Omega^F$ and $\Gamma^S = \partial\Omega^S$.

3 Solvers and Their Optimization

Following a partitioned approach, the respective subdomains of the multi-physics model as described in Sect. 2 (elastic structure domain, near-field, and far-field) are treated by different solvers. We employ the flow solver FASTEST presented in Sect. 3.1 to solve for the incompressible flow equations, Eq. (2), and near-field acoustics equations, Eq. (4), the Ateles solver described in Sect. 3.2 for the far-field acoustics equations, Eq. (4), and finally the structural solver CalculiX introduced in Sect. 3.3 for the deformation of the obstacle, Eq. (5). For performance optimization of FASTEST and Ateles, we make use of the Xevolver framework, which has been developed to separate system-specific performance concerns from application codes. We report on the optimization of both solvers further below.

3.1 FASTEST

FASTEST is used to solve both the incompressible Navier-Stokes equations (2) and the linearized Euler equations (4) in the near-field.

Capabilities and Numerical Methods The flow solver FASTEST [24] solves the three-dimensional incompressible Navier-Stokes equations. The equations are discretized utilizing a second-order finite-volume approach with implicit time-stepping, which is also second order accurate. Field data are evaluated on a non-staggered, body-fitted, and block-structured grid. The equations are solved according to the SIMPLE scheme [11], and the resulting linear equation system is solved by ILU factorization [36]. Geometrical multi-grid is employed for convergence acceleration. The code generally follows a hybrid parallelization strategy employing MPI and OpenMP. FASTEST can account for different flow phenomena, and has the capability to model turbulent flow with different approaches. In our test case example, we employ a detached-eddy simulation (DES) based on the $\zeta - f$ turbulence model [30].

In addition, FASTEST contains a module to solve the linearized Euler equations to describe low Mach number aeroacoustic scenarios, which are solved by a second order Lax-Wendroff scheme with various limiters.

Since all equation sets are discretized on the same numerical grid, advantage can be taken from the multi-grid capabilities to account for the scale discrepancies of the fluid flow and the acoustics. Since the spatial scales of the acoustics are considerably larger than those of the flow, a coarser grid level can be used for them. In return, the finer temporal scales can be considered by sub-cycling a CFD time step with various CAA time steps. This way a very efficient implementation of the viscous/acoustic splitting approach can be realized.

Performance Optimization Concerning performance optimization, one interesting point of FASTEST is that some of its kernels were once optimized for old vector machines, and thus important kernels have their vector versions in addition to the default ones. The main difference between the two versions is that nested loops in the default version are collapsed into one loop in the vector version. Since the loops skip accessing halo regions, the compiler is not able to automatically collapse the loops, resulting in short vector lengths even if the compiler can vectorize them. To efficiently run the solver on a vector system, performance engineers usually need to manually change the loop structures. In this project, Xevolver is used to express the differences between the vector and default versions as code transformation rules. In other words, vectorization-aware loop optimizations are expressed as code transformations. As a result, the default version can be transformed to its vector version, and the vector version does not need to be maintained any longer to achieve high performance on vector systems. That is, the FASTEST code can be simplified without reducing the vector performance by using the Xevolver approach. Ten rules are defined to transform the default kernels in FASTEST to their vector kernels. Those code transformations plus some system-independent minor code modifications for removing vectorization obstacles can reduce the execution time on the NEC SX-ACE vector system by about 85%, when executing a simple test case that models a three-dimensional Poiseuille flow through a channel based on the Navier-Stokes equations, in which the mesh contains two blocks with 426,000 cells each. The code execution on the SX-ACE vector processor works about 2.7 times faster than on the Xeon E5-2695v2 processor, since the kernel is memory-intensive and the memory bandwidth of SX-ACE is $4\times$ higher than that of Xeon. Therefore, it is clearly demonstrated that the Xevolver approach is effective to achieve both high performance portability and high code maintainability for FASTEST.

3.2 *Ateles*

In our project, Ateles is used for the simulation of the acoustic far-field. Since acoustics scales need to be transported over a large distance, Ateles' high-order DG scheme can show its particular advantages of low dissipation and dispersion error in this test case.

Capabilities and Numerical Methods The solver Ateles is integrated in the simulation framework APES [31]. Ateles is based on the Discontinuous Galerkin (DG) discretization method, which can be seen as a hybrid method, combining the finite-volume and finite-element methods. DG is well suited for parallelization and the simulation of aero-acoustic problems, due to its inherent dissipation and dispersion properties. This method has several outstanding advantages, that are among others the high-order accuracy, the faster convergence of the solution with

increasing scheme order and fewer elements compared to a low order scheme with a higher number of elements, the local h-p refinement as well as orthogonal hierarchical bases. The DG solver Ateles includes different equation systems, among others the compressible Navier-Stokes equations, the compressible inviscid Euler equations and the linearized Euler equations (used in this work for the acoustics far-field). For the time discretization, Ateles makes use of the explicit Runge-Kutta time stepping scheme, which can be either second or fourth order.

Performance Optimization Analyzing the performance of Ateles, originally developed assuming x86 systems, we found out that four kinds of code optimization techniques are needed for a total of 18 locations of the code in order to migrate the code to the SX-ACE system. Those techniques are mostly for collapsing the kernel loop and also for directing the NEC compiler to vectorize the loop. In this project, all the techniques are expressed as one common code transformation rule. The rule can take the option to change its transformation behaviors appropriately for each code location. This means that, to achieve performance portability between SX-ACE and x86 systems, only one rule needs to be maintained in addition to the Ateles application code. We executed a small testcase solving Maxwell equations with an 8th order DG scheme on 64 grid cells. The code transformation leads to $7.5\times$ higher performance. The significant performance improvement is attributed to loop collapse and insertion of appropriate compiler directives, which increases the vectorization length by a factor of 2 and the vectorization ratio from 71.35% to 96.72%. Finally, in terms of the execution time, the SX-ACE performance is 19% the performance of Xeon E5-2695v2. The code optimizations for SX-ACE reduce the performances of Xeon and Power8 by 14% and 6%, respectively. In this way, code optimizations for a specific system are often harmful to other systems. However, by using Xevolver, such a system-specific code optimization is expressed separately from the application code. Therefore, the Xevolver approach is obviously useful for achieving high performance portability across various systems without complicating the application code.

3.3 *CalculiX*

As structure solver, we use the well-established finite element solver CalculiX[14], developed by Guido Dhont und Klaus Wittig.² While CalculiX also supports static and thermal analysis, we only use it for dynamic non-linear structural mechanics. As our main research focus is not the structural computation per se, but the coupling within a fluid-structure-acoustic framework, we merely regard CalculiX as a black box. The preCICE adapter of CalculiX has been developed in [40].

²www.calculix.de.

4 A Black-Box Partitioned Coupling Approach Using preCICE

Our first and general coupling approach for the three-field simulation comprising (a) the elastic structure, (b) the near-field flow with acoustic equations, and (c) the far-field acoustic propagation follows a black-box idea, i.e., we only use input and output data of dedicated solvers at the interfaces between the respective domains for numerical coupling. Such a black-box coupling requires three main functional coupling components: intercode-communication, data-mapping between non-matching grids of independent solvers, and iterative coupling in cases with strong bi-directional coupling. preCICE is an open source library³ that provides software modules for all three components. In the first phase of the ExaFSA project, we ported preCICE from a server-based to a fully peer-to-peer communication architecture [9, 39], increasing the scalability of the software from moderately to massively parallel. To this end, all coupling numerics needed to be parallelized on distributed data. During the second phase of the ExaFSA project, we focused on several costly initialization steps and further necessary algorithmic optimizations. In the following, we shortly sketch all components of preCICE with a particular focus on innovations introduced in the second phase of the ExaFSA project and on the actual realization of the fluid-acoustic coupling between near-field and far-field and the fluid-structure coupling.

4.1 (Iterative) Coupling

To simulate fluid-structure-acoustic interactions such as in the scenario shown in Fig. 1, two coupling interfaces have to be considered with different numerical and physical properties: (a) the coupling between fluid flow and the elastic structure requires an implicit bi-directional coupling, i.e., we exchange data in both directions and iterate in each time step until convergence; (b) the coupling between fluid flow and the acoustic far-field is uni-directional (neglecting reflections back into the near-field domain), i.e., results of the near-field fluid flow simulation are propagated to the far-field solver as boundary values once per time step. In order to fulfil the coupling conditions at the fluid-structure interface as given in Sect. 2, we iteratively solve the fixed-point equation

$$\begin{pmatrix} S(f) \\ F(u) \end{pmatrix} = \begin{pmatrix} u \\ f \end{pmatrix}, \quad (8)$$

³www.precice.org.

where f represents the stresses, u the velocities at the interface Γ_{FS} , S the effects of the structure solver on the interface (with stresses as an input and velocities as an output), F the effects of the fluid solver on the interface (with interface velocities as an input and stresses as an output). preCICE provides a choice of iterative methods accelerating the plain fixed-point iteration on Eq. (8). The most efficient and robust schemes are our quasi-Newton methods that are provided in a linear complexity (in terms of interface degrees of freedom) and fully parallel optimized versions [35]. As most of our achievements concerning iterative methods fall within the first phase of the ExaFSA project, we omit a more detailed description and refer to previous reports instead [9].

For the uni-directional coupling between the fluid flow in the near-field and the acoustic far-field, we transfer perturbation in density, pressure, and velocity from the flow domain to the far-field as boundary conditions at the interface. We do this once per acoustic time step, which is chosen to be the same for near-field and far-field acoustics, but which is much smaller than the fluid time step size (and the fluid-structure coupling), as described in Sect. 3.1.

Both domains are time-dependent and subject to mutual influence.

In an aeroacoustic setting, the near-field subdomain Ω^{NA} and far-field subdomain Ω^{FA} , with boundaries $\Gamma^{\text{NA}} = \partial\Omega^{\text{NA}}$ and $\Gamma^{\text{FA}} = \partial\Omega^{\text{FA}}$ are fixed, which means, all background information in the far-field are fixed to a certain value. Therefore there is only influence of Ω^{NA} onto Ω^{FA} , as backward propagation can be neglected. Then the continuity of shared state variables on the interface boundary $\Gamma^{\text{IA}} = \Gamma^{\text{NA}} \cap \Gamma^{\text{FA}}$ is

$$\rho_i^{\Gamma^{\text{FA}}} = \rho_i^{\Gamma^{\text{NA}}}, u_i^{\Gamma^{\text{FA}}} = u_i^{\Gamma^{\text{NA}}}, p_i^{\Gamma^{\text{FA}}} = p_i^{\Gamma^{\text{NA}}}. \quad (9)$$

4.2 Data Mapping

Our three solvers use different meshes adapted to their specific problem domain. To map data between the meshes, preCICE offers three different interpolation algorithms: (a) Nearest-neighbor interpolation is based on finding the geometrically nearest neighbor, i.e. the vertex with the shortest distance from the target or source vertex. It excels in its ease of implementation, perfect parallelizability, and low memory consumption. (b) Nearest-projection mapping can be regarded as an extension to the nearest-neighbor interpolation, working on nearest mesh elements (such as edges, triangles or quads) instead of merely vertices and interpolating values to the projection points. The method requires a suitable triangulation to be provided by the solver. (c) Interpolation by radial-basis functions is provided. This method works purely on vertex data and is a flexible choice for arbitrary mesh combinations with overlaps and gaps alike.

In the second phase of the ExaFSA project, we improved the performance of the data mapping schemes in various ways. All three interpolation algorithms contain a lookup-phase which searches for vertices or mesh elements near a given set of

positions. As there is no guarantee regarding ordering of vertices, this resulted in $O(n \cdot m)$ lookup operations, $n, m \in \mathbb{N}$ being the size of the respective meshes. In the second phase, we introduced a tree-based data structure to facilitate efficient spatial queries. The implementation utilizes the library `Boost Geometry`⁴ and uses an `r-tree` in conjunction with the `r-star` insertion algorithm. The integration of the tree is designed to fit seamlessly into `preCICE` and avoids expensive copy operations for vertices and mesh elements of higher dimensionality. Consequently, the complexity of the lookup-phase was reduced to $O(\log_a n) \cdot m$ with a being a parameter of the tree, set to ≈ 5 . The tree index is used by nearest-neighbor, nearest-projection, and RBF interpolation as well as other parts in `preCICE` and provides a tremendous speedup in the initialization phase of the simulation.

In the course of integrating the index, the RBF interpolation profited from a second performance improvement. In contrast to the nearest-neighbor and nearest-projection schemes it creates an explicit interpolation matrix. Setting values one by one results in a large number of small memory allocations with a relatively large per-call overhead. To remedy this, a preallocation pattern is computed with the help of the tree index. This results in a single memory allocation, speeding up the process of filling the matrix. A comparison of the accuracy and runtime of the latter two interpolation methods is provided in Sect. 5.

4.3 Communication

Smart and efficient communication is paramount in a partitioned multi-physics scenario. As `preCICE` is targeted at HPC systems, a central communication instance would constitute a bottleneck and has to be avoided. At the end of phase one, we implemented a distributed application architecture. The main objective in its design is not a classical speed-up (as it is for parallelism) but not to deteriorate the scalability of the solvers and rendering a central instance unnecessary. Still, a so-called *master* process exists, which has a special purpose mainly during the initialization phase.

At initialization time, each solver gives its local portion of the interface mesh to `preCICE`. By a process called re-partitioning, the mesh is transferred to the coupling partner and partitioned there, i.e., the coupling partner's processes select interface data portions that are relevant for their own calculations. The partitioning pattern is determined by the requirements of the selected mapping scheme. The outcome of this process is a sparse communication graph, where only links between participants exist that share a common portion of the interface. While this process was basically in place at the end of phase one, it was refined in several ways.

MPI connections are managed by means of a communicator which represents an n -to- m connection including an arbitrary number of participants. The first imple-

⁴www.boost.org.

mentation used only one communication partner per communicator, essentially creating only 1-to-1 connections. To establish the connections, every connected pair of ranks had to exchange a connection token generated by the accepting side. This exchange is performed using the network file system, as the only a-priori existing communication space common to both participants. However, network file systems tend to perform badly with many files written to a single directory. To reduce the load on the file system, a hash-based scheme was introduced as part of the optimizations in phase two. With that, writing of the files is distributed among several directories, as presented in [26]. This scheme features a uniform distribution of files over different directories and, thus, minimizes the files per directory.

However, this obviously resulted in a large number of communicators to be created. As a consequence, large runs hit system limits regarding the number of communicators. Therefore, a new MPI communication scheme was created as an alternative. It uses only one communicator for an all-to-all communication, resulting in significant performance improvements for the generation of the connections. This approach also solves the problem of the high number of connection tokens to be published, though only for MPI. As MPI is not always available or the implementation is lacking, the hash-based scheme of publishing connection tokens is still required for TCP based connections.

4.4 Load Balancing

In a partitioned coupled simulation solvers need to exchange boundary data at the beginning of each iteration, which implies a synchronization point. If computational cores are not distributed in an optimal way among solvers, one solver will have to wait for the other one to finish its time step. Thus, the load imbalance reduces the computational performance. In addition, in a one way coupling scenario, if the data receiving solver is much slower than the other one, the sending partner has to wait until the other one is ready to receive (in synchronized communication) or store the data in a buffer (in asynchronous communication). In the first phase, the distribution of cores over solvers was adjusted manually and only synchronized communication was implemented, resulting in idle times.

Regression Based Load Balancing We use the load balancing approach proposed in [37] to find the optimal core distribution among solvers: we first model the solver performance against the number of cores for each domain and then optimize the core distribution to minimize the waiting time. Since mathematical modeling of the solvers' performance can be very complicated, we use an empirical approach as proposed in [37], first introduced in [10], to find an appropriate model.

Assuming we have a given set of m data points, consisting of pairs (p, f_p) mapping the number of ranks p to the run-time f_p , we want to find a function $f(p)$ which predicts the run-time against p . Therefore, we use the Performance Model

Normal Form (PMNF) [10] as a basis for our prediction model:

$$f^i(p) = \sum_{k=1}^n c_k p^{i_k} \log_2^{j_k}(p), \quad (10)$$

where the superscript i denotes the respective solver, n is a a-priori chosen number of terms, $i_k, j_k \in \mathbb{N}_0$ and c_k is the coefficient for the k th regression term. The next step is to optimize the core distribution such that we achieve minimal overall run time which can be expressed by the following optimization problem:

$$\begin{aligned} & \underset{p_1, \dots, p_l}{\text{minimize}} && F(p_1, \dots, p_l) \quad \text{with } F(p_1, \dots, p_l) = \max_i(f^i(p_i)) \\ & \text{subject to} && \sum_{i=1}^l p_i \leq P. \end{aligned}$$

This optimization problem is a nonlinear, possibly non-convex integer program. It can be solved by the use of branch and bound techniques. But, if we assume that the f^i are all monotonically decreasing, i.e., assigning more cores to a solver never increases the run-time, we can simplify the constraints to $P = \sum_{i=0}^l p_i$ and solve the problem by brute-forcing all possible choices for p_i . That is, we iterate over all possible combinations of core numbers and choose the pair that minimizes the total run-time. For more details, please refer to [37].

Asynchronous Communication and Buffering For our fluid-structure-acoustic scenario shown in Fig. 1, we perform an implicitly coupled simulation of the elastic structure interacting with the incompressible flow over a given discrete time step (marked simply as ‘Fluid’ in Fig. 2). This is followed by many small time steps for the acoustic wave propagation in the near-field, which are coupled in a loose, uni-directional way to the far-field acoustic solver (executing the same small time steps). To avoid waiting times of the far-field solver while we compute the fluid-structure interactions in the near-field, we would like to ‘stretch’ the far-field calculations such that they consume the same time as the sum of fluid-structure time steps and acoustic steps in the near-field (see Fig. 2). To achieve this, we introduced a fully asynchronous buffer layer, by which the sending participant was decoupled from the receiving participant, as shown in Fig. 2. Special challenges to tackle were the preservation of the correct ordering of messages, especially for TCP communication which does not implement such guarantees in the protocol.

4.5 Isolated Performance of preCICE

In this section, we show numerical results for preCICE only. This isolated approach is used to show the efficiency of the communication initialization. In addition,

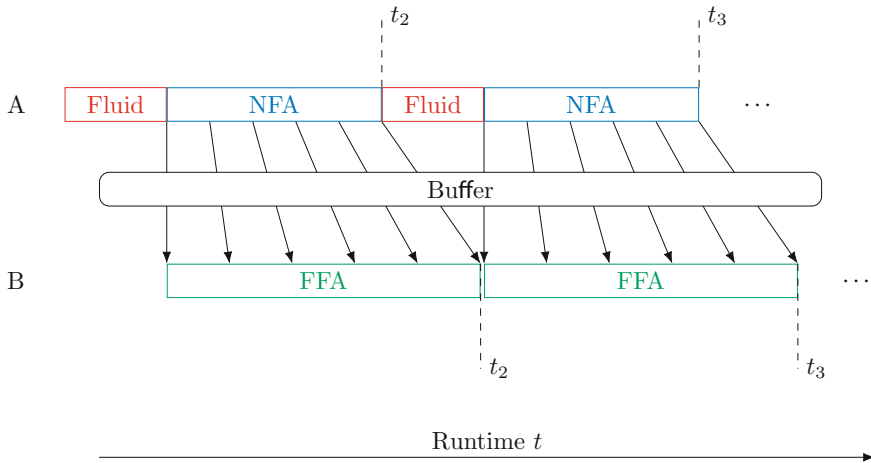


Fig. 2 Coupling scenario between participant A (performing a time step for the incompressible fluid (or fluid-structure interaction) followed by many time steps of the near-field acoustic simulation (NFA)) and participant B (performing the same small acoustic steps for the far-field (FFA) after receiving acoustic data from the near-field solver). Without buffering, inevitable idle times for participant B are created. NFA is linked to FFA through send operations. Therefore, the runtimes of NFA and FFA are matched through careful load-balancing. Shown here: A send buffer decouples NFA and FFA solver for send operations, prevents idle times, and allows for a more flexible processor assignment

we show stand-alone upscaling results. Other aspects are considered elsewhere: (a) the mapping accuracy is analyzed in Sect. 5, (b) the effectiveness of our load balancing approach as well as the buffering for uni-directional coupling are covered in Sect. 6. If not denoted otherwise, the following measurements are performed on the supercomputing systems SuperMUC⁵ and HazelHen.⁶

Mapping Initialization: Preallocation and Matrix Filling As described previously, one of the key components of mapping initialization is the spatial tree which allows for performance improvements by accelerating the interpolation matrix construction. Figure 3 compares different approaches to matrix filling and preallocation: (a) no preallocation: using no preallocation at all, i.e., allocating each entry separately, (b) explicitly computed: calculate matrix sparsity pattern in a first mesh traversal, allocate entries afterwards, and finally fill the matrix in a second mesh traversal, (c) computed and saved: additionally cache mesh element/data point relations from the first mesh traversal and use them in the second traversal to fill the matrix with less computation, (d) spatial tree: use the spatial tree instead of brute-force pairwise comparisons to determine mesh components relevant for the

⁵28× Intel-Xeon-E5-2697 cores, 64 GB memory per node.

⁶24× Intel Xeon-E5-2680 cores, 128 GB memory per node.

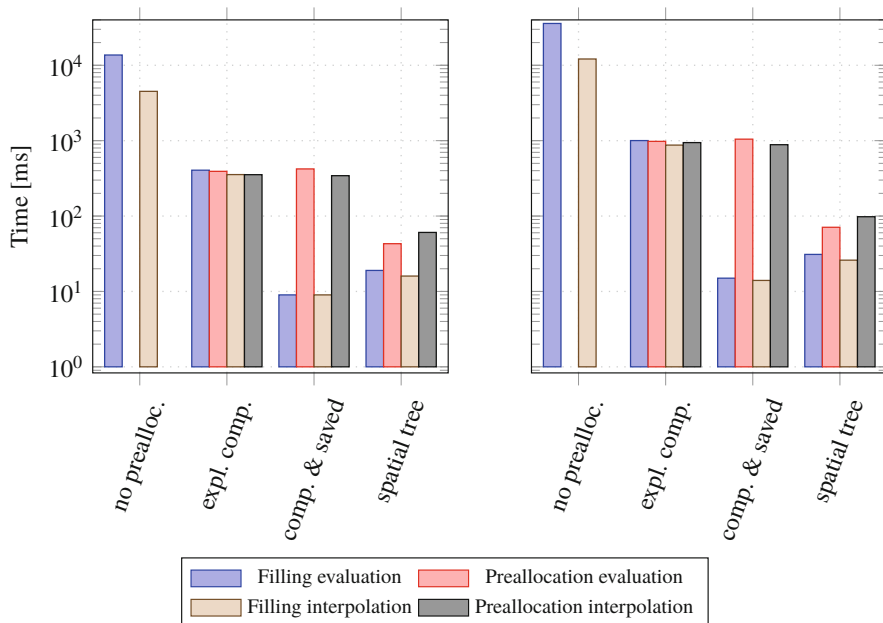


Fig. 3 Mapping initialization. Comparison of different preallocations methods for mesh sizes 6400 (left sub-figure) and 10,000 (right sub-figure) on two ranks per participant. The plot compares times spent in the stages of preallocation and filling of matrices for both the evaluation matrix and the interpolation matrix of an RBF mapping with localized Gaussian basis functions including 6 vertices of the mesh. The total time required is the sum of all bars of one measurement. Note the logarithmic scaling of the y-axis. The measurements were performed on one node of the `sgsc11` cluster, using $4 \times$ Intel Xeon E3-1585 CPUs

mapping. Each method can be considered as an enhancement of the previous one. As it becomes obvious from Fig. 3, the spatial tree was able to provide us with an acceleration of more than two orders of magnitude.

Communication For communication and its initialization, we only present results for the new single-communicator MPI based solution. For TCP socket communication that still requires the exchange of many connection tokens by means of the file system, we only give a rough factor of 2.5 that we observed in terms of acceleration of communication initialization. Note that this factor can be potentially higher as the number of processes and, thus, connections grows larger, and that the hash-based approach removed the hard limit of ranks per participant inherent to the old approach.

In Figs. 4, 5 and 6, we compare performance results for establishing an MPI connection among different ranks using many-communicators for 1-to-1 connections with using a single communicator representing an n -to- m connection. In our academic setting, both Artificial Solver Testing Environment (ASTE) participants run on n cores. On SuperMUC, each rank connects to $0.4n$ ranks, on HazelHen, with

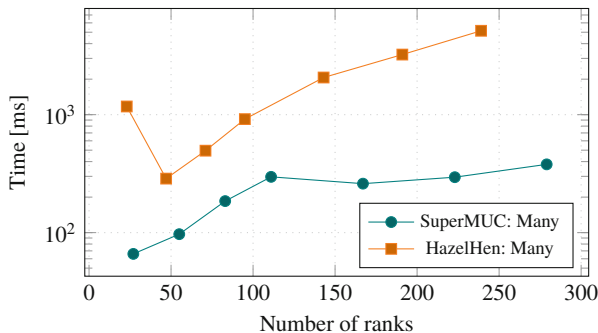


Fig. 4 Communication. Publishing of MPI connection information from participant A for the many-communicator approach. The timings of the new single-communicator approach are not shown, as they are almost negligible with a maximum of 2 ms

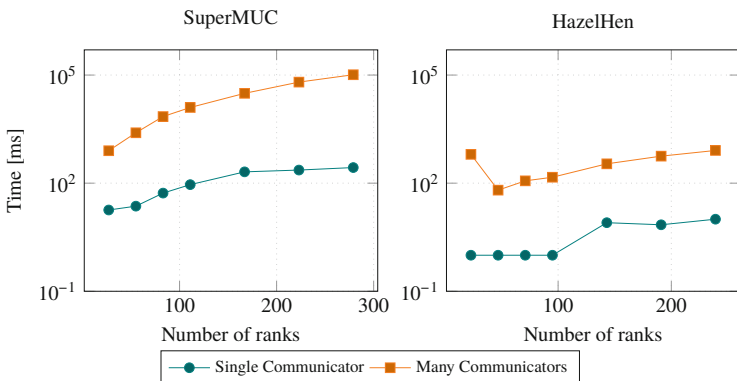


Fig. 5 Communication. Runtime for establishing the connection between the participants using `MPI_Comm_accept` and `MPI_Comm_connect`

a higher number of ranks per node, each rank connects to $0.3n$ ranks. The amount of data transferred between each connected pair of ranks is held constant with 1000 rounds of transfer of an array of 500, and 4000 double values from participant B to participant A. Each measurement is performed five times of which the fastest and the slowest runs are ignored and the remaining three are averaged. We present timings from rank zero, which is synchronized with all other ranks by a barrier, making the measurements from each rank identical. Note, that the measurements are not directly comparable between SuperMUC and HazelHen due to the different number of cores per node and that the test case is even more challenging than actual coupled simulations. In an actual simulation, the number of partner ranks per rank of a participant is constant with increasing number of cores on both sides.

Figure 4 shows the time to publish the connection token. The old approach requires to publish many tokens, which obviously becomes a performance bottleneck as the simulation setup moves to higher number of ranks. The new approach,

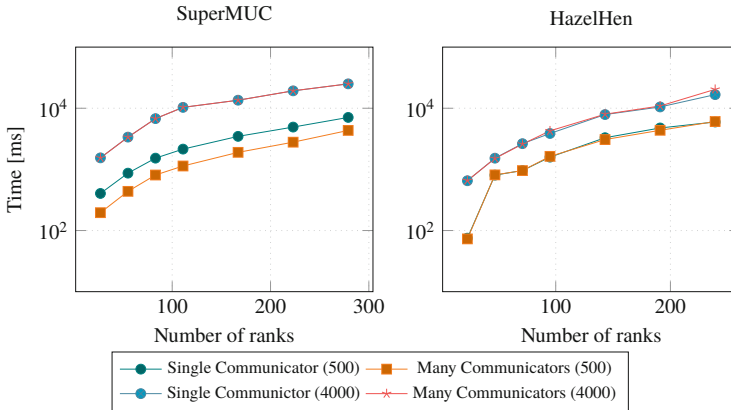


Fig. 6 Communication. Times for 1000 rounds of data transfer of a vector of 500 or 4000 doubles, respectively, from participant B to A. For the transfer, the synchronous MPI routines (`MPI_Send` and `MPI_Recv`) have been used

on the other hand, only publishes one token. It is omitted in the plot, as the times are negligible (< 2 ms). In Fig. 5, the time for the actual creation of the communicator is presented. The total number of communication partners per communicator is smaller with the old many-communicator concept (as the communication topology is sparse). However, the creation of many 1-to-1 communicators is substantially slower than the creation of one all-to-all communicator for both HPC systems. Finally, in Fig. 6 the performance for an exchange of data sets of two different sizes is presented. The results for single- and many-communicator approaches are mostly on par with the notable exception of the SuperMUC system. There, the new approach suffers a small but systematic slow-down for small message sizes. We argue that this is a result of vendor specific settings of the MPI implementation.

Data Mapping As described above, we have further improved the mapping initialization, in particular by applying a tree-based approach to identify data dependencies induced by the mapping between grid points of the non-matching solver grids and to assemble the interpolation matrix for RBF mapping. Accordingly, we show both the reduction of the matrix assembly runtime (Fig. 3) and the scalability of the mapping, including setting up the interpolation system and the communication initialization.

These performance tests of preCICE are measured using a special testing application called ASTE.⁷ This application behaves like a solver to preCICE but provides artificial data. It is used to quickly generate input data and decompose it for upscaling tests. ASTE generates uniform, rectangular, two-dimensional meshes on $[0, 1] \times [0, 1]$ embedded in three-dimensional space with the z-dimension always

⁷<https://github.com/precice/aste>.

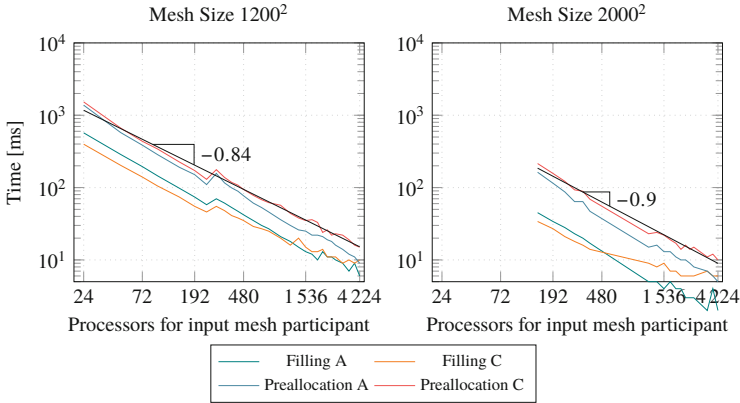


Fig. 7 Data mapping initialization. Strong scaling of the initialization of the interpolation matrix C and the evaluation matrix A for RBF interpolation on mesh sizes 1200^2 and 2000^2 with Gaussian ($m = 6$) basis functions on up to 4224 processors on the HazelHen HPC system

set to zero. The mesh is then decomposed using a *uniform* approach, thus producing partitions of same size as far as possible. Since we mainly look at the mapping part which is only executed as one of the participants, we limit the upscaling to this participant. The other participant always uses one node (28 resp. 24 processors). The mesh size is kept constant, i.e., we perform a strong scaling. The upscaling of an RBF mapping with Gaussian basis functions is shown in Fig. 7.

5 Black-Box Coupling Versus White-Box Coupling with APESMate

In the above section, we have evaluated the performance of the black-box coupling tool preCICE. In this section, we introduce an alternative approach that allows to couple different solvers provided within the framework APES [31]. Black-box data mapping in preCICE only requires point values (nearest neighbor and RBF mapping), and in some cases (nearest projection) connectivity information on the coupling interface. The white-box coupling approach of APESmate [25] has knowledge about the numerical schemes within the domain, since it is integrated in the APES suite, and has access to the common data-structure TreELM [22]. APESmate can directly evaluate the high order polynomials of the underlying Discontinuous Galerkin scheme. Thus, the mapping in preCICE is more generally applicable, while the approach in APESmate is more efficient in the context of high order scheme. Furthermore, APESmate allows the coupling of all solvers of

the APES framework, both in terms of surface and in terms of volume coupling. The communication between solvers can be done in a straightforward way as all coupling participants can be compiled as modules into one single application. Each subdomain defines its own MPI sub-communicator, a global communicator is used for the communication between the subdomains. During the initialization process, coupling requests are locally gathered from all subdomains and exchanged in a round-robin fashion. As all solvers in APES are based on an octree data-structure and a space-filling curve for partitioning, it is rather easy to get information about the location of each coupling point on the involved MPI ranks. In the following, we compare both accuracy and runtime of the two coupling approaches for a simple academic test case that allows to control the ‘difficulty’ of the mapping by adjusting order and resolution of the two participants.

Test Case Setup We consider the spreading of a Gaussian pressure pulse over a cubic domain of size $5 \times 5 \times 5$ unit length, with an ambient pressure of 100,000 Pa and a density of 1.0 kg/m^3 . The velocity vector is set to 0.0 for all spatial directions. To generate a reference solution, this test case is computed monolithically using the inviscid Euler equations.

For the coupled simulations, we decompose the monolithic test case domain into an inner and an outer domain. The resolution and the discretization order of the inner domain are kept unchanged. In the outer domain, we choose the resolution and the order such that the error is balanced with that of the inner domain. See [15] for the respective convergence study. To be able to determine the mapping error at the coupling interface between inner and outer domain, we choose the time horizon such that the pressure pulse reaches the outer domain, but is still away from the outer boundaries to avoid any influences from the outer boundaries. The test case is chosen in a way, that the differences between the meshes at the coupling interface increase, thus increasing the difficulty to maintain the overall accuracy in a black-box coupling approach. Table 1 provides an overview of all combinations of resolution and order in the outer domain used for our numerical experiments, where the total number of elements per subdomain is given as nElements, the number of coupling points with nCoupling points and the scheme order by nScheme order, respectively. For time discretization, we consider the explicit two stage Runge-Kutta scheme with a time step size of 10^{-6} for all simulations.

Table 1 White-box coupling test scenario with Gaussian pressure pulse combinations of orders and resolution used for the evaluation of the mapping methods

	Test case a		Test case b		Test case c		Test case d	
	Inner	Outer	Inner	Outer	Inner	Outer	Inner	Outer
nElements	32,768	124,000	32,768	7936	32,768	992	32,768	124
nCoupling points	55,296	9600	55,296	3456	55,296	1536	55,296	1176
nScheme order	3	4	3	6	3	8	3	14

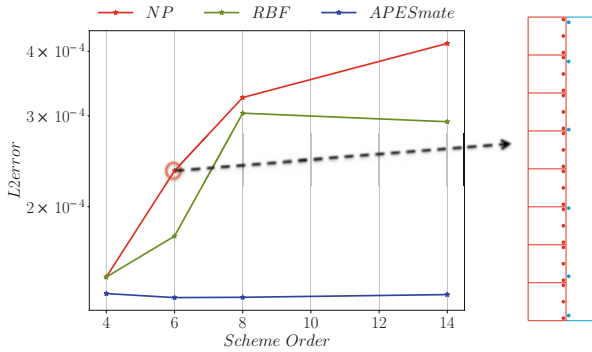


Fig. 8 Mapping accuracy black-box (APESmate) versus white-box (preCICE) approach. Comparison of the L2 error (with the analytical solution as reference) for the Gaussian pressure pulse test case variants and exemplary illustration of the coupling point distribution when using DG for test case (right figure). We compare the black-box data mapping methods Radial-Basis Functions (RBF) interpolation and Nearest Projection (NP) with the direct white-box evaluation of APESmate. The RBF mapping uses local Gaussian basis functions covering three mesh points in every direction

Mapping Accuracy In terms of mapping accuracy, it is expected, that the APESmate coupling is order-preserving, and by that not (much) affected by the increasing differences between the non-matching grids at the coupling interface, while preCICE should show an increasing accuracy drop when the points become less and less matching. This is the case for increasing order of the discretization in the outer domain. Figure 8 illustrates first results. As can be clearly seen, the white-box coupling approach APESmate provides outstanding results by maintaining the overall accuracy of the monolithic solution for all different variations of the coupled simulations, independent of the degree to which the grids are non-matching (increasing with increasing order used in the outer domain). For the interpolation methods provided by preCICE, the error increases considerably with increasing differences between the grids at the interface. As the error of the interpolation methods depends on the distances of the points (see Fig. 8), the error is dominated by the large distance of the integration points in the middle of the surface of an octree grid cell in the High Order Discontinuous Galerkin discretization.

Accuracy Improvement by Regular Subsampling We can decrease the L2 error of NP and RBF and improve the solution of the coupled simulation by providing values at equidistant points on the Ateles side as interpolation support points. The number of equidistant points is equal to the number of coupling points, hence as high as the scheme order. With this new implementation, the error shown in Fig. 9a decreases considerably compared to the results in Fig. 8. We achieve an acceptable accuracy for all discretization order combinations. However, the regular subsampling of values in the Ateles solver increases the overall computational time substantially as can be seen in Fig. 9b.

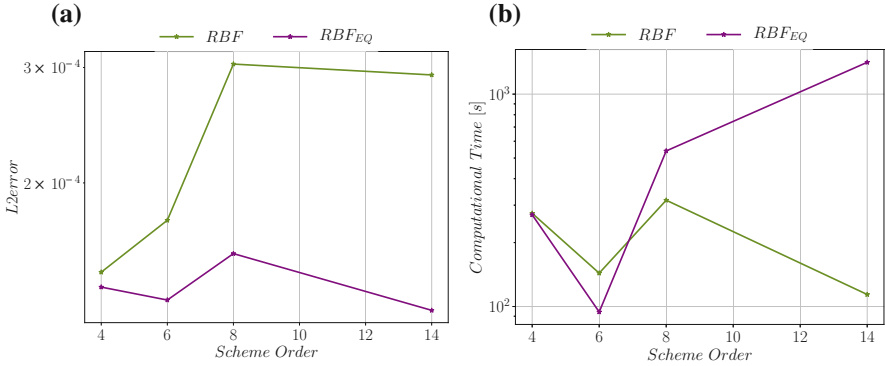


Fig. 9 Mapping accuracy and runtime of black-box with equidistant subsampling versus white-box approach. L2 error behavior (a) and computational time (b) for the RBF interpolation, when using equidistant (RBF_{EQ}) and non-equidistant (RBF) point distributions for data mapping

To improve the NP interpolation, it turned out that in addition to providing equidistant points, oversampling was required to increase the accuracy. Our investigation showed, that an oversampling factor of 3 is needed to achieve almost the same accuracy as APESmate. In spite of the additional cost of many newly generated support points, the runtime does not increase as much as for RBF, since for the RBF a linear equation system has to be computed, while for NP a simple projection needs to be done.

Summary and Runtime Comparison Figure 10 shows a summary of all tested methods for the interpolation/evaluation before and after improvements. The integrated coupling approach APESmate provides not just very accurate results, but also low runtimes. At this point, we want to recall that this is as expected—the white-box approach makes use of all internal knowledge, which gives it advantages in

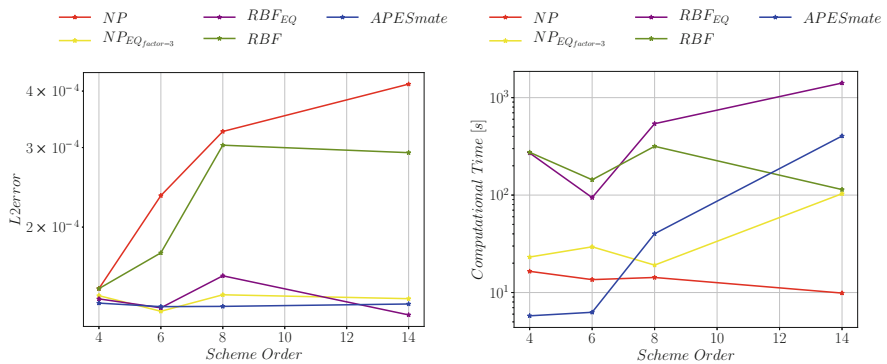


Fig. 10 Mapping accuracy and runtime summary of black-box versus white-box approach. L2 error and computational time for all methods

terms of accuracy and efficiency. On the other hand, this internal knowledge binds it to the solvers available in the framework, while preCICE can be applied to almost all available solvers. Further details regarding this investigation can be found in [15, 16].

6 Results

This section presents a more realistic test case, the turbulent flow over a fence, to assess the overall performance of our approach. Analyses for accuracy and specific isolated aspects are integrated in the sections above.

6.1 *Flow over a Fence Test Case Setup*

As a test case to assess the overall scalability, we simulate the turbulent flow over a (flexible) fence and the induced acoustic far-field as already shown schematically in Fig. 1. The FSI functionality of FASTEST has been demonstrated earlier many times, e.g. in [34]. Thus we focus on the acoustic coupling.

As boundary conditions, we use a no-slip wall at the bottom and the fence surface, an inflow on the left with u_{bulk} , outflow convective boundary conditions on the right, periodic boundary conditions in y -directions, and slip conditions at the upper boundary for the near-field flow. For the acoustic perturbation, we apply reflection conditions at the bottom and the fence surface, zero-gradient condition at all other boundaries. The acoustic far-field solver uses Dirichlet boundary conditions at its lower boundary (see also Eq. (9)). Therefore, the upper near-field boundary is not the coupling interface, but we instead overlap near-field and far-field as shown in Fig. 11. Figures 12 and 13 show a snapshot of the near-field flow and the near-field and far-field acoustic pressure, respectively.

6.2 *Fluid-Acoustics Coupling with FASTEST and Ateles*

To demonstrate the computational performance of our framework using FASTEST for the flow simulation in the near-field, the high-order DG solver Ateles for the far-field acoustic wave propagation, and preCICE for coupling, we show weak scalability measurements for the interaction between near-field flow simulation and far-field acoustics. We keep both the mesh and the number of MPI ranks in the near-field flow simulation fixed. In the far-field computed with Ateles, we refine the mesh to better capture the acoustic wave propagation. We use a multi-level mesh with a fine mesh at the coupling interface to allow a smooth solution at the coupling interface between the near-field and the far-field. We refine the far-field mesh in

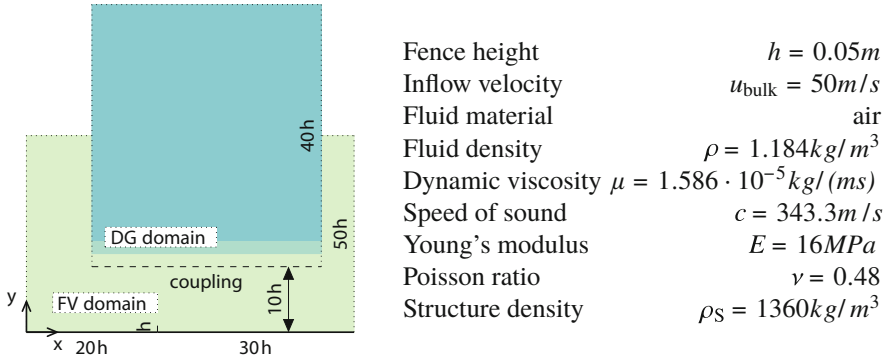


Fig. 11 Flow over a fence test case. Schematic view of the computational domain (left) and applied parameters (right). Colors indicate spatial discretization order in the various regions: The FV domain is completely second order, while the DG domain has second order at the coupling interface for reduced coupling interpolation errors, and subsequently increases the order in various layers for the far-field transport

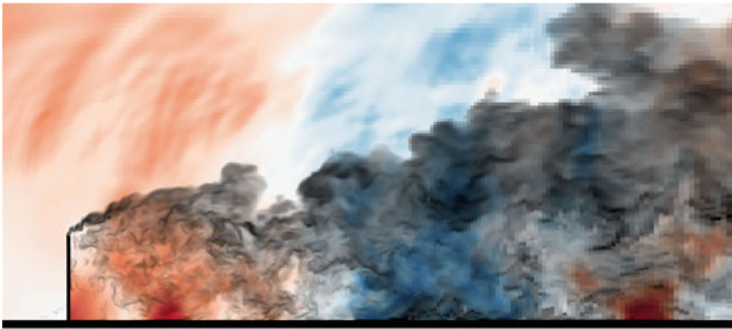


Fig. 12 Flow over a fence test case with FASTEST. Snapshot of the flow in the recirculation area behind the fence. Red/blue indicate acoustic pressure, grey shades show the modelled turbulent kinetic energy (for a $\zeta - f$ DES model)

two main steps: in the first step, we only refine the mesh at the coupling interface. In the next step, we first refine the whole mesh, and again the mesh at the coupling interface in the third and fourth step. Due to the refinement at the coupling interface, the number of Ateles ranks participating in the interface increases such that this study also shows that the preCICE communication does not deteriorate scalability. Table 2 gives an overview of the configurations used for the weak scaling study.

To find the optimal core distribution for all setups, the load balancing approach proposed in Sect. 4 is used. This analysis shows that for the smallest mesh resolution with 24,864 elements in the far-field, the optimal core distribution is 424 cores for the near-field domain and 196 cores for the far-field. For all other setups, we assume perfect scalability, i.e., we choose the number of cores proportional to the number of degrees of freedom in the weak scaling study and increase the number of cores

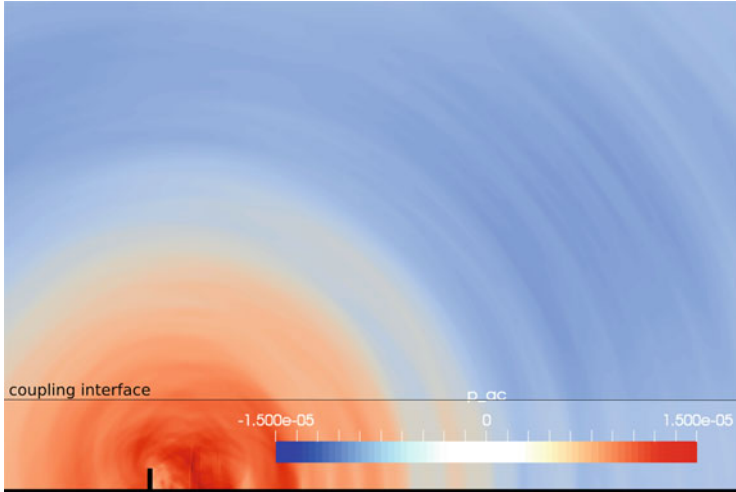


Fig. 13 Flow over a fence test case. Snapshot of the acoustic pressure in a coupled simplified setup

Table 2 Flow over a fence test case

Cores in FASTEST/Ateles	Ateles degrees of freedom	Ateles elements
424/196	16,116,480	24,864
424/756	62,535,840	89,376
424/1428	116,524,800	177,408
424/3136	254,150,400	607,488
424/15,372	1,245,054,720	3,704,064

Scalability study for the interaction between the near-field flow simulation and the far-field acoustics: Summary of mesh details and core numbers for weak scaling. In the FASTEST simulation of the near-field flow simulation, we use 52,822,016 elements. In the far-field, Ateles uses discretization order 9

simultaneously by a factor of two in both fields for strong scaling. The scalability measurements are shown in Fig. 14. The results show that the framework scales almost perfectly up to 6528 cores.

6.3 Fluid-Acoustics Coupling with Only Ateles

In Sect. 5 we investigated the suitability of different interpolation methods for our simulations. In this section, we present a strong scaling study for an Ateles-Ateles coupled simulation of the flow over a fence test case. The fence is modelled in

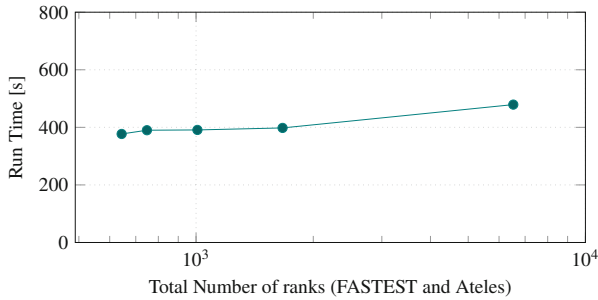


Fig. 14 Flow over a fence test case. Weak scalability measurement of the fluid-acoustic interaction simulation for the fence test case

Ateles using the newly implemented immersed boundary method, enabling high-order representation of complex geometries in Ateles [1]. We solve the compressible Navier-Stokes equations in the flow domain with a scheme order of 4 and a four step mixed implicit-explicit Runge-Kutta time stepping scheme, with a time step size of 10^{-7} . The total number of elements in the flow domain is 192,000. For the far-field, we use the same setup as for the FASTEST-Ateles coupling.

The linearized Euler equations in the far-field can be solved in a DG setting in the modal formulation, which makes the solver very cheap even for very high order. In the near-field domain, the non-linear Navier-Stokes equations are solved with a more expensive hybrid nodal-modal approach. Due to this, and the different spatial discretizations and scheme orders, both domains have different computational load, which requires load balancing. We use static load balancing, since neither the mesh nor the scheme order vary during runtime. As both solvers are instances of Ateles, we apply the SpartA algorithm [21], which allows re-partitioning of the workload according to weights per elements, which are computed during runtime. Those weights are then used to re-distribute the elements according to the workload among available processes (see [17] for more details). The total number of processes used for this test case are 14,336 processes, which is equal to one island on the system. As mentioned previously, the total workload per subdomain does not change, therefore we start our measurements by providing the lower subdomain 100 processes and the upper subdomain 12 processes, which is equal to 4 nodes on the system. This number per subdomain is then doubled for each run, the ratio is kept the same.

Figure 15 shows the strong scaling measurements for both coupling approaches (APESmate and preCICE) executed on the SuperMUC Phase2 system. As can be clearly seen, both coupling setups Ateles-APESmate-Ateles and Ateles-preCICE-Ateles scale almost ideally, however with a lower absolute runtime for the APESmate coupling as expected.

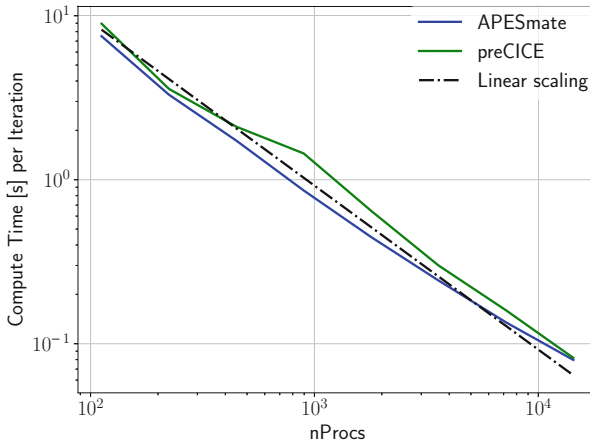


Fig. 15 Flow over a fence test case. Strong scaling measurement for Ateles-Ateles coupling using both APESmate and preCICE—and the ideal linear scaling as reference

7 Summary and Conclusion

We have presented a partitioned simulation environment for the massively parallel simulation of fluid-structure-acoustic interactions. Our setup uses the flow and acoustic solvers in the finite volume software FASTEST, the acoustic solvers in the discontinuous Galerkin framework Ateles as well as the black-box fully parallel coupling library preCICE. In particular, we could show that with a careful design of the coupling tool as well as of solver details, we can achieve a bottleneck-free numerically and technically highly scalable solution. It turned out that efficient initialization of point-to-point communication relations and mapping matrices between the involved participants, sophisticated inter-code load balancing and asynchronous communication using message buffering are crucial for large-scale scenarios. With these improvements, we advanced the limits of scalability of partitioned multi-physics simulations from less than a hundred cores to more than 10,000 cores. Beyond that, we reach a problem size that is not required by the given problem as well as scalability limits of the solvers. The coupling itself is not the limiting factor for the given problem size and degree of parallelism. To be able to use also vector architectures in an efficient sustainable way, we adapted our solvers with a highly effective code transformation approach.

Acknowledgments We thankfully acknowledge funding from SPPEXA, the German Science Foundation Priority Program 1648—Software for Exascale Computing and the European Union’s Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 754462, and compute time granted on the LRZ SuperMUC system and the HLRS HazelHen system.

References

1. Anand, N., Pour, N.E., Klimach, H., Roller, S.: Utilization of the brinkman penalization to represent geometries in a high-order discontinuous Galerkin scheme on octree meshes. *Symmetry* **11**(9), 1126 (2019). <https://doi.org/10.3390/sym11091126>
2. Banks, J.W., Henshaw, W.D., Schwendeman, D.W.: An analysis of a new stable partitioned algorithm for FSI problems. Part I: incompressible flow and elastic solids. *J. Comput. Phys.* **269**, 108–137 (2014)
3. Bazilevs, Y., Takizawa, K., Tezduyar, T.E.: *Computational Fluid-Structure Interaction: Methods and Applications*. Wiley, Hoboken (2012)
4. Blom, D., Ertl, T., Fernandes, O., Frey, S., Klimach, H., Krupp, V., Mehl, M., Roller, S., Sternel, D.C., Uekermann, B., Winter, T., van Zuijlen, A.: Partitioned fluid-structure-acoustics interaction on distributed data – numerical results and visualization. In: Bungartz, H.J., Neumann, P., Nagel, E.W. (eds.) *Software for Exa-scale Computing – SPPEXA 2013–2015*. Springer, Berlin (2016)
5. Boer, A.D., van Zuijlen, A., Bijl, H.: Comparison of conservative and consistent approaches for the coupling of non-matching meshes. *Comput. Methods Appl. Mech. Eng.* **197**(49), 4284–4297 (2008)
6. Bungartz, H., Mehl, M., Schäfer, M.: *Fluid Structure Interaction II: Modelling, Simulation, Optimization*. Lecture Notes in Computational Science and Engineering. Springer, Berlin (2010)
7. Bungartz, H.J., Lindner, F., Mehl, M., Uekermann, B.: A plug-and-play coupling approach for parallel multi-field simulations. *Comput. Mech.* **55**(6), 1119–1129 (2015)
8. Bungartz, H.J., Lindner, F., Gatzhammer, B., Mehl, M., Scheufele, K., Shukaev, A., Uekermann, B.: preCICE – a fully parallel library for multi-physics surface coupling. *Comput. Fluids* **141**, 250–258 (2016). *Advances in Fluid-Structure Interaction*
9. Bungartz, H.J., Lindner, F., Mehl, M., Scheufele, K., Shukaev, A., Uekermann, B.: Partitioned fluid-structure-acoustics interaction on distributed data – coupling via preCICE. In: Bungartz, H.J., Neumann, P., Nagel, E.W. (eds.) *Software for Exa-scale Computing – SPPEXA 2013–2015*. Springer, Berlin (2016)
10. Calotiu, A., Beckinsale, D., Earl, C.W., Hoefler, T., Karlin, I., Schulz, M., Wolf, F.: Fast multi-parameter performance modeling. In: *2016 IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 172–181 (2016). <https://doi.org/10.1109/CLUSTER.2016.57>
11. Caretto, L.S., Gosman, A.D., Patankar, S.V., Spalding, D.B.: Two calculation procedures for steady, three-dimensional flows with recirculation. In: *Proceedings of the Third International Conference on Numerical Methods in Fluid Dynamics*. Springer, Paris (1972)
12. Crosetto, P., Deparis, S., Fougere, G., Quarteroni, A.: Parallel algorithms for fluid-structure interaction problems in haemodynamics. *SIAM J. Sci. Comput.* **33**(4), 1598–1622 (2011)
13. Degroote, J., Bathe, K.J., Vierendeels, J.: Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction. *Comput. Struct.* **87**(11–12), 793–801 (2009)
14. Dhondt, G.: *The Finite Element Method for Three-Dimensional Thermomechanical Applications*. Wiley, Chichester (2004)
15. Ebrahimi Pour, N., Roller, S.: Error investigation for coupled simulations using discontinuous galerkin method for discretisation. In: *Proceedings of ECCM VI/ECFD VII*, Glasgow, June 2018
16. Ebrahimi Pour, N., Krupp, V., Klimach, H., Roller, S.: Coupled simulation with two coupling approaches on parallel systems. In: Resch, M.M., Bez, W., Focht, E., Gienger, M., Kobayashi, H. (eds.) *Sustained Simulation Performance 2017*, pp. 151–164. Springer International Publishing (2017). https://doi.org/10.1007/978-3-319-66896-3_10
17. Ebrahimi Pour, N., Krupp, V., Roller, S.: Load balancing for coupled simulations. In: Resch, M.M., Bez, W., Focht, E., Gienger, M., Kobayashi, H. (eds.) *Sustained Simulation Performance 2019*. Springer International Publishing, Cham (2019)

18. Fachgebiet für Numerische Berechnungsverfahren im Maschinenbau: FASTEST Manual (2005)
19. Gee, M.W., Küttler, U., Wall, W.A.: Truly monolithic algebraic multigrid for fluid–structure interaction. *Int. J. Numer. Methods Eng.* **85**(8), 987–1016 (2011)
20. Haelterman, R., Degroote, J., van Heule, D., Vierendeels, J.: The quasi-Newton least squares method: a new and fast secant method analyzed for linear systems. *SIAM J. Numer. Anal.* **47**(3), 2347–2368 (2009)
21. Harlacher, D.F., Klimach, H., Roller, S., Siebert, C., Wolf, F.: Dynamic load balancing for unstructured meshes on space-filling curves. In: 2012 IEEE 26th International on Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW), pp. 1661–1669 (2012)
22. Klimach, H.G., Hasert, M., Zudrop, J., Roller, S.P.: Distributed octree mesh infrastructure for flow simulations. In: Eberhardsteiner, J. (ed.) ECCOMAS 2012 - European Congress on Computational Methods in Applied Sciences and Engineering, e-Book Full Papers (2012)
23. Kong, F., Cai, X.C.: Scalability study of an implicit solver for coupled fluid-structure interaction problems on unstructured meshes in 3D. *Int. J. High Perform. Comput. Appl.* **32**, 207–219 (2016)
24. Kornhaas, M., Schäfer, M., Sternel, D.: Efficient numerical simulation of aeroacoustics for low Mach number flows interacting with structures. *Comput. Mech.* **55**, 1143–1154 (2015). <https://doi.org/10.1007/s00466-014-1114-1>
25. Krupp, V., Masilamani, K., Klimach, H., Roller, S.: Efficient coupling of fluid and acoustic interaction on massive parallel systems. In: Sustained Simulation Performance 2016, pp. 61–81 (2016)
26. Lindner, F.: Data transfer in partitioned multi-physics simulations: interpolation & communication. PhD thesis, University of Stuttgart (2019)
27. Lindner, F., Mehl, M., Uekermann, B.: Radial basis function interpolation for black-box multi-physics simulations. In: Papadarakakis, M., Schrefler, B., Onate, E. (eds.) VII International Conference on Computational Methods for Coupled Problems in Science and Engineering, pp. 1–12 (2017)
28. Link, G., Kaltenbacher, M., Breuer, M., Döllinger, M.: A 2D finite-element scheme for fluid-solid-acoustic interactions and its application to human phonation. *Comput. Methods Appl. Mech. Eng.* **198**(41–44), 3321–3334 (2009)
29. Mehl, M., Uekermann, B., Bijl, H., Blom, D., Gatzhammer, B., van Zuijlen, A.: Parallel coupling numerics for partitioned fluid-structure interaction simulations. *Comput. Math. Appl.* **71**(4), 869–891 (2016)
30. Reimann, T.: Numerische simulation von fluid-struktur-interaktion in turbulenten strömungen. Ph.D. thesis, Technische Universität Darmstadt (2013)
31. Roller, S., Bernsdorf, J., Klimach, H., Hasert, M., Harlacher, D., Cakircali, M., Zimny, S., Masilamani, K., Diding, L., Zudrop, J.: An adaptable simulation framework based on a linearized octree. In: Resch, M., Wang, X., Bez, W., Focht, E., Kobayashi, H., Roller, S. (eds.) High Performance Computing on Vector Systems 2011, pp. 93–105. Springer, Berlin (2012)
32. Ross, M.R., Felippa, C.A., Park, K., Sprague, M.a.: Treatment of acoustic fluid-structure interaction by localized Lagrange multipliers: formulation. *Comput. Methods Appl. Mech. Eng.* **197**(33–40), 3057–3079 (2008)
33. Schäfer, F., Müller, S., Uffinger, T., Becker, S., Grabinger, J., Kaltenbacher, M.: Fluid-structure-acoustic interaction of the flow past a thin flexible structure. *AIAA J.* **48**(4), 738–748 (2010)
34. Schäfer, M., Sternel, D.C., Becker, G., Pironkov, P.: Efficient numerical simulation and optimization of fluid-structure interaction. In: Bungartz, H.J., Mehl, M., Schäfer, M. (eds.) Fluid Structure Interaction II. Modelling, Simulation, Optimization. Lecture Notes in Computational Science and Engineering, vol. 73, pp. 131–158. Springer, Berlin (2010)
35. Scheufele, K., Mehl, M.: Robust multisecant quasi-Newton variants for parallel fluid-structure simulations – and other multiphysics applications. *SIAM J. Sci. Comput.* **39**(5), 404–433 (2017)
36. Stone, H.L.: Iterative solution of implicit approximations of multidimensional partial differential equations. *SIAM J. Numer. Anal.* **5**(3), 530–558 (1968)

37. Totounferoush, A., Ebrahimi Pour, N., Schroder, J., Roller, S., Mehl, M.: A new load balancing approach for coupled multi-physics simulations (2019). <https://doi.org/10.1109/IPDPSW.2019.00115>
38. Turek, S., Hron, J., Madlik, M., Razzaq, M., Wobker, H., Acker, J.: Numerical simulation and benchmarking of a monolithic multigrid solver for fluid-structure interaction problems with application to hemodynamics. In: Bungartz, H.J., Mehl, M., Schäfer, M. (eds.) *Fluid Structure Interaction II: Modelling, Simulation, Optimization*, p. 432. Springer, Berlin (2010)
39. Uekermann, B.: *Partitioned fluid-structure interaction on massively parallel systems*. Ph.D. thesis, Department of Informatics, Technical University of Munich (2019)
40. Uekermann, B., Bungartz, H.J., Cheung Yau, L., Chourdakis, G., Rusch, A.: Official preCICE adapters for standard open-source solvers. In: *7th GACM Colloquium on Computational Mechanics*, pp. 1–4 (2017)
41. Vierendeels, J., Lanoye, L., Degroote, J., Verdonck, P.: Implicit coupling of partitioned fluid-structure interaction problems with reduced order models. *Comput. Struct.* **85**(11–14), 970–976 (2007)
42. Zudrop, J., Klimach, H., Hasert, M., Masilamani, K., Roller, S.: A fully distributed CFD framework for massively parallel systems. In: *Cray User Group Conference*. Cray User Group, Stuttgart (2012)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

