Efficient Coupling of Fluid and Acoustic Interaction on Massive Parallel Systems

Verena Krupp, Kannan Masilamani, Harald Klimach and Sabine Roller

Abstract We present and compare two coupling approaches for direct aeroacoustic simulations. Direct aeroacoustic simulations pose a multi-scale problem, as the generation of sound in a flow field occurs at small spatial scales with high energy, while its propagation in the farfield has to be observed on a large spatial scale with only low energy. The challenge of different scales can be addressed by employing different numerical schemes in the individual spatial areas with an interaction between them on the surfaces. Two implementation strategies of this coupling approach are presented. The first coupling strategy employs a library that allows a wide range of different applications to be coupled with minimal changes to the individual solvers. Hence, this is a very flexible approach but limited access to information and therefore cope with loss of potential performance. Further this strategy involves the handling of multiple executables on today supercomputer. This multi-solver approach requires data interpolation at the coupling interface which introduce another numerical error. In contrast, the second approach is fully integrated within one numerical framework. Thereby the solvers are invoked as a library by the coupling application and only one single applications must be handled. Tethering high order solvers, fully access to the data implies that no additional data interpolation is required which promise better numerical results. This tight integration allows for the exploitation of knowledge about internal data structures and therefore yield performance benefits accompany with less flexibility. Both strategies will be compared with respect to numerical error due to data interpolation at the coupling interface as well as scalability and performance on modern supercomputer.

V. Krupp (✉) · K. Masilamani · H. Klimach · S. Roller
University of Siegen, Hölderlinstr. 3, Siegen, Germany
e-mail: verena.krupp@uni-siegen.de

K. Masilamani
e-mail: kannan.masilamani@uni-siegen.de

H. Klimach
e-mail: harald.klimach@uni-siegen.de

S. Roller
e-mail: sabine.roller@uni-siegen.de

M.M. Resch et al. (eds.), *Sustained Simulation Performance 2016*,
DOI 10.1007/978-3-319-46735-1_6

1 Introduction

Increasing computational resources allow the simulation of a new range of multi-physics and multi-scale problems that were unfeasible with former compute resources. Such simulations have the potential to provide more insight into applications from various fields, as, for example, the sound design of aircrafts or wind turbines. With an increased awareness for noise pollution such consideration get more and more important in the design process of industrial applications.

In this work, we focus on the coupling of fluid flows and acoustic sound propagation. The main challenge of this coupled application is that both phenomena include different length and energy scales. The multi-scale nature of the fluid-acoustic interactions is best described in the example of a wind turbine: Noise is generated by the vortices of the rotating geometry at a length scale in the order of centimeters. The whole turbine size is in the scale of meters, while the noise emission is of relevance in a distance of hundreds of meters up to a few kilometers from the sound source. Simulating the entire domain while resolving the smallest turbulent scales and resolving the boundary layer adequately would require approximately 10^{18} degrees of freedom, which is out of reach even with the larges computing facilities in the forseeable future. For fluid-acoustic interactions the phenomena can often be clearly separated in the different areas of the domain. Different sets of equations and different discretization resolutions and schemes can be used for each part individually. The fluid-acoustic coupling interface is rather large and, therefore, needs to be efficient and fully parallelized.

We describe a partitioned coupling approach, i.e., we split the physical space into smaller domains, each covering a so-called single-physics subdomain. These subdomains can be solved with numerical methods and resolutions tailored to the local physical requirements. This allows for the re-use of existing scalable software based on decades of experience in each single-physics discipline, thus enabling acceptable software development times along with efficiency and performance optimization. The interaction between the domains is realized by exchanging data at the boundary. By the adaptation of numerical approximations in the individual domains, the computation of complete interactions between fluid mechanics and acoustic wave propagation becomes feasible.

In this paper, we investigate two different partitioned coupling approaches. One makes use of individual solvers that run as independent executables and use a coupling library to exchange data. The other approach uses a more integrated approach, where a single application is used and the individual solvers are incorporated as libraries. This tight integration on the basis of a common framework allows for the exploitation of knowledge about internal data structures and therefore potentially a faster coupling mechanism. However, this comes at the cost of reduced flexibility. The presented work focuses on establishing both approaches within the simulation framework *APES* and compares numerical as well as performance results. First, we briefly recapitulate the governing equations for fluid mechanics and acoustic wave propagation in Sect. 2 followed by Sect. 3 describing the methodology of the flow and acoustic solver *Ateles*. Section 3.2 describes the partitioned coupling approach in general including the multi-solver approach using the open-source coupling library *preCICE*, and the integrated coupling approach *APESmate* within the numerical framework *APES*. Finally, Sect. 4 presents the results of numerical simulations of two academic testcases as well as performance results for both approaches.

2 Governing Equations

Acoustic phenomena are based on the same principals as fluid motion. However, while for general fluid motion nonlinear equations have to be considered, acoustic phenomena can be represented in linearized equations, as only small perturbations need to be considered. The linearization reduces the numerical effort drastically and, therefore, is a necessity for large computational domains as required for the computation of acoustic far fields.

2.1 Fluid Equations

Frictionless flow is governed by the compressible Euler equations based on the conservation of mass, momentum and energy. We use the superscript f to indicate variables in the flow field. The conservation of mass can be written as

$$\frac{\partial \rho^f}{\partial t} + \nabla \cdot (\rho \mathbf{v})^f = 0, \qquad (1)$$

the conservation of momentum is given by

$$\frac{\partial\,(\rho \mathbf{v})^f}{\partial t} + \nabla \cdot \left((\rho \mathbf{v})^f \mathbf{v}^f\right) + \nabla p^f = 0, \qquad (2)$$

and the conservation of energy yields

$$\frac{\partial}{\partial t}\left(\rho^f\left(e^f + \frac{1}{2}\mathbf{v}^f \cdot \mathbf{v}^f\right)\right) + \nabla \cdot \left((\rho \mathbf{v})^f\left(e^f + \frac{1}{2}\mathbf{v}^f \cdot \mathbf{v}^f + \frac{p^f}{\rho^f}\right)\right) = 0. \qquad (3)$$

The velocity field is denoted by $\mathbf{v}^f$, pressure is denoted with p^f, and the density is given as ρ^f. The internal energy of the flow is e^f. The Euler equations are derived from the Navier-Stokes equations by neglecting viscous effects, heat flow and external forces. We only consider ideal gases here to close the system:

$$p^f = \rho^f R T = (\gamma - 1)\left(e^f - \frac{\rho^f \mathbf{v}^f \cdot \mathbf{v}^f}{2}\right)$$

which yields a relation between pressure p and energy e, where R is the ideal gas constant, T is the temperature and γ is the isentropic coefficient.

2.2 Acoustic Equations

Acoustic phenomena are also fluid motion and are, therefore, governed by the Euler equations (1)–(3). As there are only small changes in the flow, they can be linearized around a constant background flow. The constant background flow is denoted by the subscript 0 and the perturbation is denoted with the superscript a. In the following, we will treat only the primitive variables density ρ, velocity $\mathbf{v}$ and pressure p in the acoustic domain. The linearized Euler equations are given by the linearized equation of mass conservation

$$\frac{\partial \rho^a}{\partial t} + \nabla \cdot \left(\mathbf{v}_0 \rho^a + \rho_0 \mathbf{v}^a\right) = 0, \tag{4}$$

the linearized momentum equation

$$\frac{\partial \mathbf{v}^a}{\partial t} + \nabla \cdot \left(\mathbf{v}_0 \mathbf{v}^a + \frac{1}{\rho_0} p^a\right) = 0 \tag{5}$$

and linearized energy equation

$$\frac{\partial p^a}{\partial t} + \nabla \cdot \left(\mathbf{v}_0 p^a + \gamma \ p_0 \ \mathbf{v}^a\right) = 0. \tag{6}$$

Since the Euler equations require conservative variables for the coupling, the general transformation between primitive variables ρ^f, $\mathbf{v}^f$, p^f and conservative variables ρ^f, $\rho^f \mathbf{v}^f$, e^f is required

$$\rho^f = \rho^f, \quad \mathbf{v}^f = \frac{(\rho \mathbf{v})^f}{\rho^f}, \quad p^f = (\gamma - 1)[\rho^f e^f - \frac{1}{2\rho^f}((\rho \mathbf{v})^f)^2],$$

as well as vice versa

$$\rho^f = \rho^f, \quad (\mathbf{v}\rho)^f = \mathbf{v}^f \rho^f, \quad e^f = \frac{1}{(\gamma - 1)} \frac{p^f}{\rho^f} + \frac{1}{2}(\mathbf{v}^f)^2.$$

To compute the linearized variables in the acoustic domain, simple subtraction of the background state is sufficient to obtain the perturbations:

$$\rho^a = \rho^f - \rho_0, \quad \mathbf{v}^a = \mathbf{v}^f - \mathbf{v}_0, \quad p^a = p^f - p_0,$$

where the background flow is defined by the user.

3 Methodology

In this section, we describe the methodology of the established simulation approach. First we present the flow and acoustic solver *Ateles*. Then, we describe the partitioned coupling approach for fluid-acoustic interaction in detail including the two implementation strategies: the multi-solver approach which uses the open-source coupling library *preCICE*and the integrated approach *APESmate* which is incorporated within the framework *APES*.

3.1 High-Order Solver Ateles

For the flow as well as the acoustic domain, we use the high-order solver *Ateles* which is included in the end-to-end parallel framework *APES* [6, 10]. The *APES* framework is designed to take advantage of the massively parallel systems available in supercomputing today. Therefore, it provides additional tools for pre- and post-processing on the basis of a common mesh library.[1] The *TreElM* library [4] relies on an octree representation of the mesh and provides a distributed neighborhood search within that mesh. Using a space-filling curve for the domain decomposition of the octree mesh gives hierarchically structured data and maintains locality. This locality can be perfectly exploited by the high-order Discontinuous Galerkin solver *Ateles*.

Ateles is capable to solve various equation systems such as compressible flow, linear wave propagation and electro-dynamics, which are solved with an explicit Runge Kutta method in time and a modal discontinuous Galerkin method (DG) with arbitrary order in space [3]. The Discontinuous Galerkin method is based on a polynomial representation within an element and flux calculation between elements over their faces. Hence, there is a strong coupling of data within each element and only a loosely coupling between elements via element surfaces. The choice of the polynomial degree controls the spatial discretization order. By choosing a high degree for the polynomial function a high-order method is constructed. Exploiting modal basis functions has computational reasons, e.g. that the numerical flux can be directly evaluated in modal space, using cubical elements without any extra transformation to a reference element [9].

A higher order scheme has several advantages. First, it yields low numerical dissipation and dispersion errors, which is advantageous for approximating the wave propagation over long distances in the acoustic far field. Secondly, a higher order scheme shows high convergence rates in case of smooth solutions. Hence, a high order approximation provides a high accuracy with only few degrees of freedom. For nonlinear systems high-order schemes imply an increased computational cost, but for

[1] https://bitbucket.org/apesteam/treelm

the linear system of the acoustic domain, a modal scheme keeps the computational effort per degree of freedom constant over increased spatial orders and solves them efficiently.

The polynomial representation of the DG method also has an advantage in the coupling context. For data exchange at the coupling interface, the polynomial representation can be evaluated at any point on the surface up to the chosen order for the method. In general, the quadrature points of the polynomial on the surface are utilized as exchange points.

3.2 Partitioned Coupling

Partitioned coupling is based on the idea that an entire computational domain can be split into subdomains, where only single physics need to be considered in each subdomain. For the example of fluid-acoustic interaction, this means: we split the whole domain into a subdomain of flow and acoustic generation and a subdomain where only acoustic waves propagate. Small vortices with high energy occur typically around a structure or at high Mach number and generate acoustic waves. In this domain, the small scales of the flow must be resolved. Acoustic waves on the other hand live on larger scales, having less energy, and are transported into the acoustic far field. In this case, the phenomena have to be resolved over long distances. The interactions are realized by a surface coupling between the compressible fluid domain and the acoustic far field. To realize a full coupling which means including information travelling between both subdomains, a bidirectional coupling is deployed, i.e. both domains provide and receive data at the interface.

Figure 1 shows a partitioned coupling example using implicit coupling between structure and fluid and explicit coupling between fluid and acoustic subdomains.

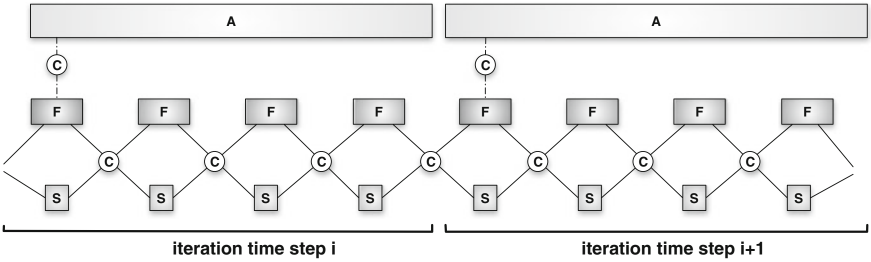


Fig. 1 Overview of parallel execution of the fluid (F) -structure (S) -acoustic (A) simulations. Implicit coupling (C) for fluid-structure interaction presenting an iterative method and explicit coupling (C) at the beginning of the timestep

When coupling several solvers, there are major tasks involved in a coupled setup:

- Steering of the individual single physic solvers
- Data interpolation between non-matching exchange points
- Communication of primitive variables at the interface

These tasks should be handled efficiently in parallel by the coupling tool.

Steering of individual solvers

To control the simulation and the correct update of information at the coupling interface in time, the coupling device should steer the individual solver. The major challenge here is the definition of the synchronization time step.

Data interpolation

For a general setup, allowing individual resolution in each subdomain, the exchange points at the interface do not require to coincide. A non-matching coupling mesh at the interface can occur when e.g. coupling of a higher order Discontinuous Galerkin method which requires information at non-equidistant quadrature points. Figure 2 gives an example of such a non-matching coupling interface, when coupling the same grid resolution but an 8th order Discontinuous Galerkin scheme with a 4th Discontinuous Galerkin scheme where both yield 16 points at a 1d surface. Therefore, an efficient interpolation method is required to transfer the primitive variables of one coupling interface to the other.

Communication

The exchange of data between the solvers is also a task of the coupling device. We aim for large scale simulations on massive parallel systems. Therefore, direct MPI

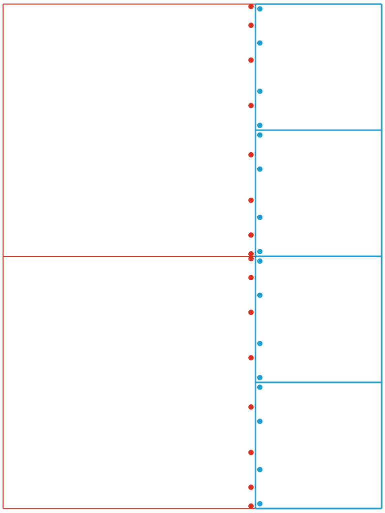


Fig. 2 Example of non-matching exchange points at the coupling interface when coupling the same grid resolution but a 8th order Discontinuous Galerkin scheme (*red*) with a 4th order Discontinuous Galerkin scheme (*blue*) yielding both 16 points at a 1d surface

communication between processes that host coupling elements is essential. This communication takes place at each synchronization time step.

In our approach, explicit coupling is exploited. For the first attempt, we do not allow for adaptive time stepping and sub-cycling of one solver to avoid non-consistent coupling in time. Hence, all subdomains use the same timestep limited by the CFL condition of the explicit timestepping within the solver. This is clearly a lack of ideal performance and it is part of future work. Assuming no adaptive time stepping and a fixed coupling interface, a static load balancing based on heuristics can be achieved by choosing an appropriate number of processes for each subdomain, such that solving each domain takes approximatively the same computational time.

3.2.1 Multi-solver Approach Using Coupling Library *preCICE*

For the multi-solver approach, the focus is on using the solvers as ‘black box’ which means that the solvers are accessible only via their interfaces for input and output values. Therefore, the aforementioned major tasks of the coupling device are more challenging: Steering between individual solvers, communication of data between executables and accurate interpolation methods between non-matching interfaces. The open-source coupling library *preCICE*[2] offers methods for all these building blocks while allowing for a minimally invasive integration into existing solvers [1]. Additionally, for implicit coupling, which is not part of this paper but a key benefit of *preCICE*, efficient solvers for fixed-point equations derived from coupling conditions are implemented in *preCICE*. Clearly, the major tasks of the coupling device need to work efficiently and should be scalable for distributed data. In [2, 7] development and achievements of *preCICE*working on distributed data are presented.

In *preCICE*, the initialization of the communication is done via exchanging the entire coupling interface via master processes. The communication of coupling participant and coupling library during the time loop is done with point-to-point communication realized via TCP/IP (based on `Boost.Asio`[3]). Coupling different numerical resolution in space requires data at position on the interface which might not be provided by one participant as illustrated in Fig. 2. Therefor interpolation methods between non-matching coupling meshes are required. *preCICE*provides two standard interpolation methods: low order projection-based mapping (nearest neighbor, nearest projection) and second order radial basis function mapping. Both mappings work on pure geometric information assuming ‘black box’ solvers.

Flexibility is the key benefit of using a coupling tool like *preCICE*. The application programming interface (API) is concise and enables an easy coupling of individual solvers. Additionally, it implements several sophisticated coupling methods, which are required to improve numerical stability at the coupling interface. The advantages are only clouded by the decrease in performance due to generality of a ‘black-box’ approach.

[2] www.precice.org

[3] www.boost.org

Furthermore, the handling of a coupled simulation involves several executables. Porting software, establishing the correct pinning of MPI ranks in this setup, and compiling the job script on a supercomputer is more challenging compared to running a single application.

3.2.2 Integrated Approach Using *APESmate*

The integrated coupling approach *APESmate* is fully implemented in the previously presented framework *APES* [6, 10]. Here, finding a synchronization time step is similar to using the multi-solver approach but the steering of the coupled simulation is direct by accessing the data structure explicitly instead of providing and returning information from a library. Also, communication can be done in a direct way: all components are implemented in a single application which efficiently distributes domains across several processes. Starting with a global communicator, each subdomain gets its own MPI sub-communicator for domain-internal communications. Therefore, a global communicator is used only for domain-domain communication. During the initialization step, all coupling requests of one subdomain are locally gathered such that only one large communication is necessary instead of multiple small ones. Then this information is exchanged in a round robin fashion. Since every solver in *APES* is based on an Octree data structure and uses a space-filling curve for partitioning, it is easy to get information about the location of the individual exchange points. The identified ranks which accommodate exchange points are provided to the requested domains and these ranks are then used to build communication buffers for data exchange between domains. Point coordinates are only exchanged at the initial step, the point values are evaluated and exchanged via the global communicator once within every time step.

Within the integrated coupling, the application can access solver specific data. Tethering high order DG solver *Ateles*, obtaining data at arbitrary exchange points on the coupling interface can be done via direct evaluation of the polynomial representations. Hence, coupling non-matching grids with different numerical resolution, as shown in Fig. 2, does not involve additional interpolation. This is a key benefit compared to using a multi-solver approach. In the case of coupling other solvers within *APESmate*, e.g. a Lattice-Boltzmann scheme which does not provide a polynomial representation of the solution, the solver is required to provide an interpolation method using its data representation and mathematical formulation. i.e. even if interpolation is necessary, it is done by the data-providing solver, making use of all the knowledge regarding its data and data structure.

In general, *APESmate* is implemented in a way such that surface as well as volume coupling can be realized to increase the range of applications, e.g. the coupling of multi component flow and the electro-dynamic field [5].

Naturally, with this integrated approach, we can only couple solver and methods which are included within *APES* and operating on the underlying data structure of the common mesh library *TreElM*. Up to now, only explicit coupling via data exchange at every time step is available in *APESmate*. However, for fluid-acoustic interaction addressed in this paper, single physics solvers with explicit time step are sufficient.

The performance benefits of *APESmate* as a single application is superior to multi-solver approach due to communications over global MPI communicator and direct control over *Ateles* solver. With respect to load balancing, assuming no adaptive time stepping and not changing the coupling interface, the same static load balancing based on heuristic as presented for the multi-solver approach can be applied. Also, a dynamic load balancing can be deployed easier. From the user perspective, the handling of an integrated approach with one executable is facilitated.

4 Results

In this section, we show the comparison of the two presented coupling approaches, using the external library *preCICE*as well as the integrated approach *APESmate*. We setup two different scenarios, one coupling the same equations system on both sides, but using different mesh sizes and approximation orders, and the other coupling different equation systems, on the same and on different meshes and orders. When using *preCICE*, we also vary the interpolation method between first order nearest neighbor interpolation and second order radial basis functions.

The second part of this section describes the performance scalability of both established coupling strategies on modern supercomputer.

4.1 Simulation Setup

We show two dedicated test cases:

(a) Gaussian distribution in density on a 2-dimensional domain (Fig. 3a)
(b) Gaussian distribution pressure on a 3-dimensional domain (Fig. 3b)

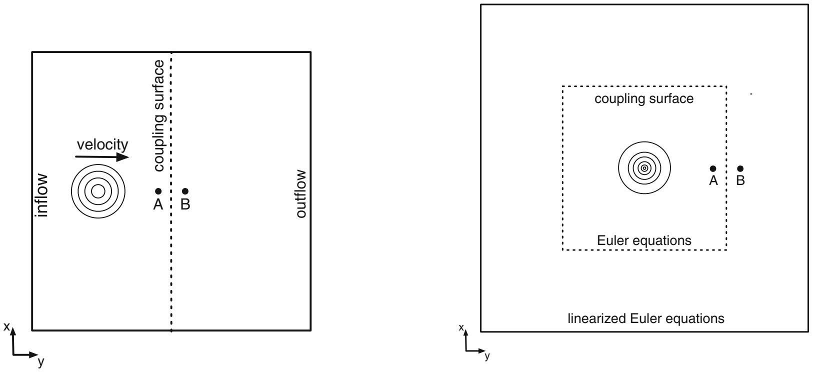
Testcase (a) is used for coupling the same equations systems and (b) to couple two different equations e.g. a non-linear flow subdomain with a linearized Euler domain.

For test case (a) we will refer in the following to as left and right subdomain as illustrated, and for test case (b) as flow domain and acoustic domain.

4.2 Numerical Results

4.2.1 Bidirectional Coupling of the Same Equations Systems: Flow with Flow

To test the coupling of twice the same equation systems, we deploy a 2-dimensional Gaussian density distribution which travels from left to right due to advection of the flow in positive x-direction, see Fig. 3a. The whole domain is a two dimensional



(a) Gaussian pulse in density, where the whole flow domain is split into two subdomains left and right. The • marks represents the measurement positions.

(b) Gaussian pulse in pressure, where the inner box is the flow domain (Euler equations) and the surrounding is the acoustic domain (linearized Euler equations). The • marks represents the measurement positions.

Fig. 3 Sketch of the two dedicated simulation setups

4 × 4 xy-plane, which is split into a left and a right subdomain. As described in Sect. 2, for the Euler equations (1)–(3) the ideal gas is considered. Here, the isentropic coefficient is chosen to be $\gamma = 1.4$ and the ideal gas constant is $R = 296.0$. The density is initially given as a Gaussian pulse shifted by $(x_0 = -1.0, y_0 = 0.0)$ to be fully located in the left subdomain:

$$\rho = \rho_0 + \rho_{pulse} \cdot \exp\left(-[(x + x_0)^2 + (y + y_0)^2]/d \cdot \log(2)\right)$$

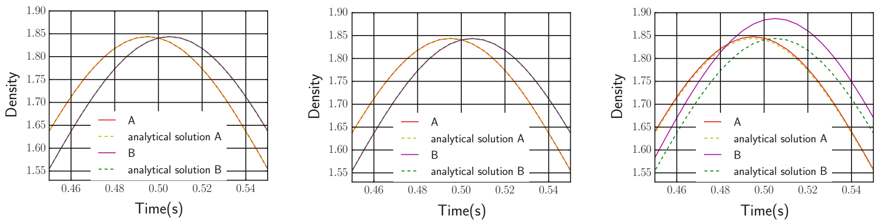
with background density $\rho_0 = 1.0$, amplitude of the pulse $\rho_{pulse} = 1.0$ and half width of the pulse $d = 0.02$. The flow is initialized with a constant velocity field, $\mathbf{v}_{t=0} = \left[2.0\ 0.0\right]^T$ and pressure, $p_{t=0} = 8.0$. As shown in Fig. 3a, the left and right boundary conditions are inflow and outflow respectively, whereas the upper and lower boundaries are set to the full state $[\rho_0, \mathbf{v_0}, p_0]$. The analytical solution of the density pulse traveling through the flow at time t is

$$\rho_{ref} = \rho_0 + \rho_{pulse} \cdot \exp\left(-[(x + x_t)^2 + (y + y_t)^2]/d \cdot \log(2)\right) \tag{7}$$

with the location of the pulse $x_t = \mathbf{v}_{0x}t - x_0$, $y_t = \mathbf{v}_{0y}t - y_0$.

4.2.2 Results and Comparison

The investigation is done for both established methods, the integrated approach *APESmate* and the multi-solver approaches using *preCICE*. The two approaches differ mainly in the way they obtain the data required for the one side from the data



(a) Integrate dapproach *APESmate*.

(b) Multi-solver approach *preCICE* using Radial basis function.

(c)Multi-solver approach *preCICE* using nearest neighbor.

Fig. 4 Comparison of numerical and analytical result in both subdomains for coupling of different resolutions: *Left* $\mathscr{L} = 4$, $\mathscr{O} = 16$; *Right* $\mathscr{L} = 4$, $\mathscr{O} = 22$

provided by the other. When looking at Fig. 2, the left domain delivers and expects data on the red points, while the right side delivers and expects data at the blue points. When using the multi-solver approach, the coupling tool *preCICE*interpolates from red to blue and from blue to red points. Two different interpolation methods, nearest neighbor and radial basis function, are available in *preCICE*. The integrated approach in *APESmate*—as it is able to access directly the high-order polynomials within the left as well as within the right part of the domain—directly evaluates the polynomials at the points requested and thus does not interpolate at all.

For validation purposes, a first run of the simulation is performed using the same mesh and the same order of the DG scheme on both sides. Thus, the data exchange points match on both sides, and interpolation reduces to pure injection. The two different interpolation schemes which are tested against each other within the multi-solver approach with *preCICE*should not show any difference, neither compared to each other nor compared to the integrated approach with *APESmate*. For this pre-testcase, all results coincide as expected.

The next variation now checks the influence of the interpolation in the case of non-matching grids as in Fig. 2. Non-matching grids are obtained when using different mesh sizes or different approximation orders in the DG scheme. We refer to the grid resolution as refinement Level $\mathscr{L}$ of the Octree mesh and $\mathscr{O}$ for the numerical order of the DG scheme in space.

Figure 4 shows the comparison of the different coupling strategies when coupling two different discretizations, i.e. left: $\mathscr{L} = 4$, $\mathscr{O} = 16$; right: $\mathscr{L} = 4$, $\mathscr{O} = 22$. It is measured at positions A and B (Fig. 3a) at the point in time when the maximum amplitude of density pulse is reached. The integrated approach (Fig. 4a) as well as the multi-solver approach using second order radial basis functions for data interpolation at the exchange points (Fig. 4b) give good results and are identical with the analytical solution. The first order nearest neighbor interpolation in the multi-solver approach (Fig. 4c) produces an overshooting in point B (to the right of the coupling interface), compared to the solution in point A (left to the coupling interface), and the analytical solution.

4.2.3 Error Analysis

For the error analysis, we compare the simulation result to the analytical solution and present the error, i.e. the difference between analytical solution and simulation at the maximum density of the Gaussian pulse in both subdomains. Table 1 gives an overview of the simulation error at points A (left to the coupling interface) and B (right to the coupling interface).

As mentioned before, testing the same numerical resolution in both subdomains ensures matching exchange points at the coupling interface and avoids influences

Table 1 Comparison of simulation error at maximum of the Gaussian distribution at points ±0.01 distance from the coupling surface

Left domain			Right domain		
$\mathscr{L}$	$\mathscr{O}$	Error	$\mathscr{L}$	$\mathscr{O}$	Error
(a) Integrated approach APESmate					
4	16	−2.3833e-4	4	8	1.8816e-4
4	16	2.7661e-4	4	12	−7.9379e-4
4	16	5.7715e-5	4	16	4.1064e-5
4	16	1.1531e-5	4	22	4.5484e-6
4	16	−1.7762e-5	4	32	−2.0103e-5
4	16	−2.1650e-5	3	32	−1.8538e-5
4	16	1.1151e-4	5	8	−9.2129e-4
3	64	−2.9949e-6	4	32	−3.5793e-7
(b) Multi-solver approach with preCICEusing 2nd order radial Basis function interpolation					
4	16	−2.7129e-3	4	8	1.9446e-3
4	16	2.820e-4	4	12	1.0909e-3
4	16	5.7715e-5	4	16	4.1064e-5
4	16	4.0299e-5	4	22	−3.1967e-4
4	16	−1.4563e-5	4	32	−3.0865e-4
4	16	3.4437e-5	3	32	3.3906e-4
4	16	1.4550e-4	5	8	1.0778e-3
3	64	−7.6588e-6	4	32	−2.6305e-5
(c) Multi-solver approach with preCICEusing Nearest Neighbor interpolation					
4	16	1.5106e-2	4	8	−1.3613e-1
4	16	1.3116e-2	4	12	2.7013e-3
4	16	5.7715e-5	4	16	4.1064e-5
4	16	3.6480e-3	4	22	4.3454e-2
4	16	5.6850e-3	4	32	−3.7787e-2
4	16	5.0766e-4	3	32	3.6137e-2
4	16	7.2353e-4	5	8	8.0711e-4
3	64	−8.3883e-4	4	32	2.3469e-3

of the data interpolation required for the multi-solver approach. Comparing the simulation error for the same numerical resolution, i.e Left: $\mathscr{L} = 4$, $\mathscr{O}(16)$; Right: $\mathscr{L} = 4$, $\mathscr{O}(16)$, 3 line in Table 1, demonstrates a good agreement of all strategies as expected.

Comparing different numerical resolutions demonstrates the important influence of the data mapping strategy on the coupling interface. Table 1 indicates that a first order nearest neighbor interpolation is no option for coupling non-matching grids since disproportionally large numerical errors arise. Only one out of the 8 combinations results in errors comparable to the other approaches, which is line 7, Left: $\mathscr{L} = 4$, $\mathscr{O}(16)$; Right: $\mathscr{L} = 5$, $\mathscr{O}(8)$. This might be due to the exact position of the exchange points on the surface. In this setting, the distance between the coupling points on both sides of the coupling interface is nearly minimal.

In general, it can be stated that the integrated *APESmate* as well as the multi-solver approach *preCICE*with a second order radial basis function (RBF) interpolation give good results, whereby the simulation error for *APESmate* yields a simulation error which is even one or two orders of magnitude lower than for the *preCICE* and, order RBF approach.

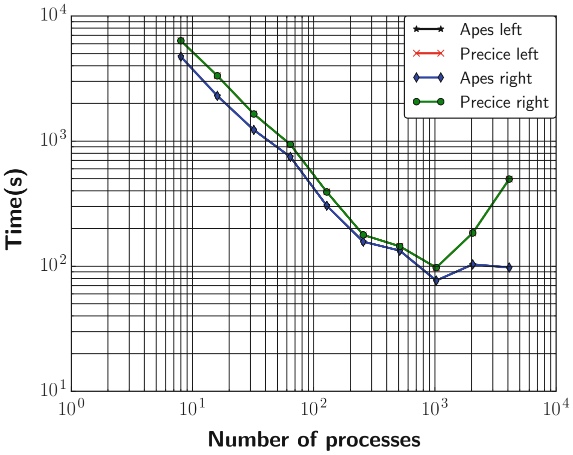
4.2.4 Performance Results

In this section, we present the performance of integrated coupling *APESmate* and multi-solver coupling *preCICE*using nearest neighbor interpolation only on the SuperMuc Phase 1 IBM system at LRZ, Munich. This system comprises a total of 9216 nodes on 18 islands with 2 Sandy Bridge-EP Xeon E5-2680 processor with 8 cores per node resulting in 147,456 cores. The nodes are connected with Infiniband FDR10. For performance measurements, both coupling approaches are scaled up to a single island, i.e 512 compute nodes or 8192 cores. Using more than one island is not possible at the moment due to limitations of MPI-IO on SuperMuc. Only MPI parallelism is considered here.

A 3D version of the test case presented in the previous section (Gaussian pulse in density traveling from left to right domain) is used with a total problem size of 8192 elements, i.e 4096 elements per coupling domain. The total problem size of 8192 elements is chosen such that there is at least one element per core and the polynomial order, $\mathscr{O}$ is chosen to fit maximally to the memory per node which is found to be $\mathscr{O}(20)$. The simulations are run for 100 iterations. The number of degrees of freedom per element is 109760, resulting in 163840000 DoF for problem size of 4096 elements per domain.

In Fig. 5, the strong scaling of both strategies *APESmate* and *preCICE*coupling on left and right domain are shown together with the number of processes on the X-axis and the total run time in seconds on Y-axis. Both approaches have good scalability up to 1024 processes per domain. Beyond that, the integrated approach *APESmate* does not scale anymore and gets flat where as the scalability of the multi-solver approach *preCICE*gets worse, i.e run time increases with number of processes. The increase in run time might be due to load imbalances stemming from only few processes

Fig. 5 Strong scaling of integrated coupling *APESmate* and multi-solver coupling *preCICE*

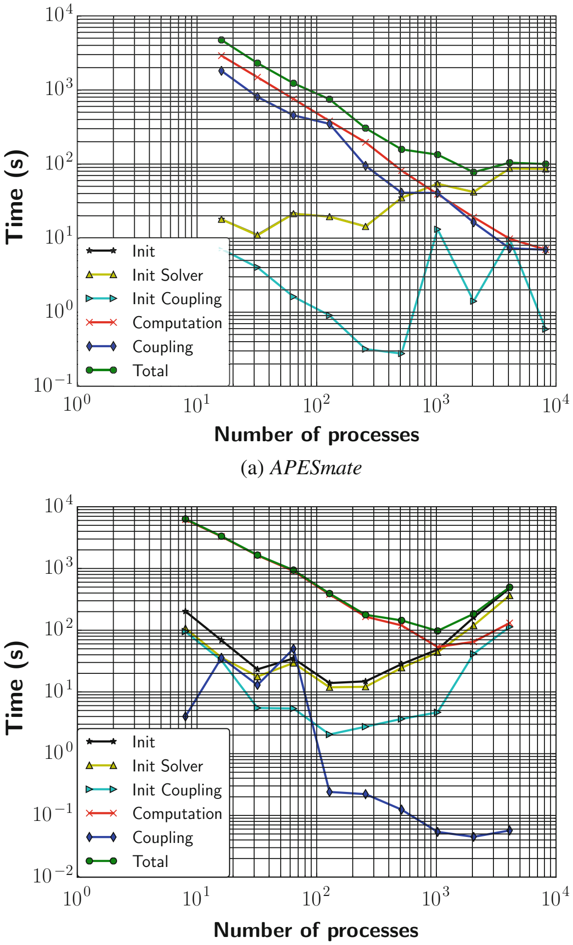


participating in the coupling. Nevertheless, at all points, the integrated approach *APESmate* is roughly 20 % faster then the multi-solver approach using the external coupling library *preCICE*. To determine the performance critical step, the overall run time is split into initialization, computation and coupling. Figure 6 shows this breakdown of overall run time for *APESmate* in Fig. 6a and *preCICE*in Fig. 6b. For both approaches, the initialization step is further split into initialization of solver and coupling. In Fig. 6, the total initialization time increases with the number of processes for both approaches but *preCICE*initialization time is much higher than *APESmate*. As stated in [2], the initialization step is not fully parallelized and work in progress. In *APESmate*, we can measure the time spent on computation and initialization separately where as in *preCICE*coupling initialization time is part of computation time and difficult to calculate explicitly. This can be seen from the computation time in Fig. 6a since both approaches uses the *Ateles* solver which is scalable on its own. The coupling step involving the evaluation of point values and data exchange is faster with *preCICE*than *APESmate*. The coupling in *APESmate* involves the evaluation point values using polynomial which is expensive but more accurate than the fast, but inaccurate nearest neighbor approach used in *preCICE*. Also, the*APESmate* coupling approach shows better scalability than *preCICE*. From Fig. 6b, we can conclude that for *preCICE*, the increase in run time beyond 1024 processes per domain is mainly due to the initialization of coupling.

4.2.5 Bidirectional Coupling of Differing Equation Systems: Euler with Linearized Euler

To test the coupling of differing equation systems e.g. Euler equations with linearized Euler equations, we use an acoustic pulse initialized at time t = 0 with a Gaussian pressure distribution which is spreading spherically symmetric with respect to the

Fig. 6 Strong scaling breakdown of overall time into initialization, computation and coupling of integrated approach *APESmate* and multisolver approach *preCICE*



(a) *APESmate*

(b) Precice.

origin of the pulse as described in [8], sketched in Fig. 3. The 3-dimensional flow domain in which the pulse is located is a 20 × 20 × 20 box with a surrounding acoustic domain of size 60 × 60 × 60. For both domains, the isentropic coefficient is set to $\gamma = 1.4$ and the ideal gas constant is $R = 296.0$. Additionally, for the acoustic domain, treating the linearized Euler equations (4)–(6), the background flow is set to $\rho_0 = 1.0$, $\mathbf{v_0} = \left[0.0, 0.0, 0.0\right]^T$, $p_0 = \frac{1}{\gamma}$ yielding a speed of sound $c = 1.0$. For the inner flow domain, the initial condition for the Euler domain is a Gaussian pressure distribution:

$$p = p_0 + p_{pulse} \cdot \exp\left(-[(x + x_0)^2 + (y + y_0)^2 + (z + z_0)^2]/d \cdot \log(2)\right)$$

with amplitude of the pulse $p_{pulse} = 0.001$ and half width set to $d = 3$. The background for the flow is set to background of the acoustic domain. The Euler domain is initialized with density $\rho_{t=0} = 1.0$ and velocity $\mathbf{v}_{t=0} = \left[0.0, 0.0, 0.0\right]^T$. For the surrounding acoustic domain, the initial condition $\left[\rho^a, \mathbf{v}^a, p^a\right]^T$ is specified to 0, since at the start of the simulation, no acoustic perturbation should occur. The outer boundaries for the acoustic domain are set to a Dirichlet boundary condition for all state variables i.e. $\rho^a = 0.0$, $\mathbf{v}^a = \left[0.0, 0.0, 0.0\right]^T$, $p^a = 0.0$. The analytical solution for a Gaussian pressure distribution spreading spherically symmetric with respect to the origin (0.0, 0.0, 0.0) and the radial distance $r = \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2}$ is :

$$\begin{aligned} p = p_0 + p_{pulse} \cdot \; & \Bigg[\tfrac{r-c\cdot t}{2\cdot r} \cdot \exp\left(-\log(2) \cdot \left(\tfrac{r-c\cdot t}{b}\right)^2\right) \\ & + \tfrac{r+c\cdot t}{2\cdot r} \cdot \exp\left(-\log(2) \cdot \left(\tfrac{r+c\cdot t}{b}\right)^2\right) \Bigg] \end{aligned} \tag{8}$$

with speed of sound defined by the material $c = \sqrt{\frac{\gamma \cdot p_0}{\rho_0}}$.

4.2.6 Results and Comparison

We look at the temporal evolution at the specific points A and B close (±0.01) to the coupling interface as sketched in Fig. 3a. Figure 7 shows a coupled simulation using the non-linear Euler equations in the inner domain, and the linearized Euler

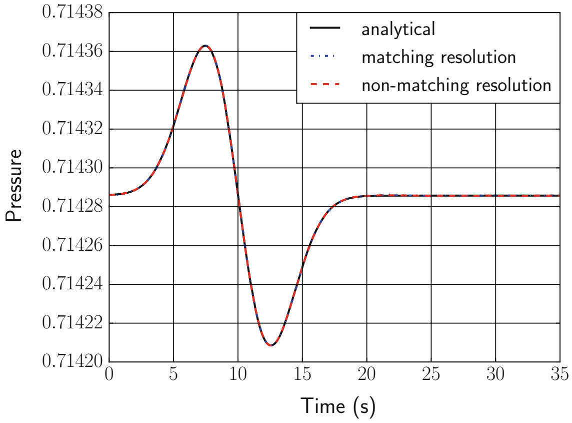


Fig. 7 Time evolution of pressure at measure positions, see Fig. 3b. Comparing a coupled simulation with same numerical resolution—matching [flow domain: 8,000 elements, element size = 1, $\mathcal{O}(6)$, acoustic domain: 208,000 elements, element size = 1, $\mathcal{O}(6)$] with different numerical resolution—non-matching [flow domain: 8,000 elements, element size = 1, $\mathcal{O}(6)$, acoustic domain: 3,250 elements, element size = 5, $\mathcal{O}(12)$]

Table 2 Absolute and relative error for the different simulations of the Gaussian pulse distribution measured at positions ±0.1 off from the coupling surface in the flow respectively in the acoustic domain. Relative error is normalized to the acoustic perturbation in the pressure since this is the travelling information

	Flow domain	Acoustic domain		
	Absolute error	Relative error	Absolute error	Relative error
Maximum of pressure distribution				
Matching coupling	1.5e-07	1.9e-03	7.25e-08	9.37e-04
Non-matching coupling	1.53e-07	1.98e-03	8.07e-08	1.04e-03
Minimum of pressure distribution				
Matching coupling	1.14e-07	1.471e-03	3.73e-08	4.83e-04
Non-matching coupling	1.16e-07	1.497e-03	4.27e-08	5.53e-04

equations in the outer domain. In the first setup, only the equations are switched, but mesh level and order of the scheme are kept the same in both parts of the domain (matching resolution). In a second setup, the effort for the outer domain is decreased by using a coarser mesh, but a higher order in the DG scheme compared to the inner domain (non-matching). The numerical configuration for the matching simulation is flow domain: 8,000 elements, element size = 1, $\mathcal{O}(6)$, acoustic domain: 208,000 elements, element size = 1, $\mathcal{O}(6)$. The configuration for the non-matching setup is flow domain: 8,000 elements, element size = 1, $\mathcal{O}(6)$, acoustic domain:3,250 elements, element size = 5, $\mathcal{O}(12)$.

Table 2 shows the comparison of the results for matching and non-matching grid in terms of absolute and relative error in pressure. The table shows the good accordance of the results, i.e. the quality of the solution is the same for the finer mesh with lower order (matching configuration) as for the coarser mesh with higher resolution. We will now investigate the gain in performance by this variation.

4.2.7 Performance Benefits for Coupling Differing Equation Systems

Coupling non-linear Euler equations with linearized Euler equations yield different computational load on the corresponding subdomains. Therefore, a coupled simulation with properly chosen computational resources can reduce the computational cost. Besides the variation in the numerical parameters (matching/non-matching), also the distribution of the two subdomains to available compute resources can be optimized. Table 3 illustrates the overall runtime of the Gaussian pressure pulse simulation for different settings of the parallel distribution where the overall number of 512 MPI-ranks is used on the SuperMuc Phase 1 IBM system at LRZ, Munich. The particular number of MPI ranks is chosen to fill a full node with 16 processes. In the

Table 3 Load balancing for different distributions of total 512 MPI-ranks on SuperMUC of a coupled simulation of Gaussian pressure pulse against monolithic simulation. Using 64 ranks for the flow domain and 448 for the acoustic domain yield the faster computation for same resolution on both subdomains

Type	Number of MPI-ranks		Total computation time (s)
	Flow domain	Acoustic domain	
Matching coupling	256	256	1336
	128	384	886.65
	112	400	845.573
	96	416	824.695
	80	432	847.923
	64	448	832.06
	48	464	838.607
	32	480	902.342
	16	496	1535
Non-matching coupling	96	416	476.886

case of matching grids, we found that utilizing 96 ranks for the flow domain and 416 for the acoustic domain yields the fastest computation with 824s. Table 3 illustrates how the imbalance moves from the acoustic domain, when using a low number of MPI-ranks here, to the flow domain since when using more MPI-ranks in acoustics, less in the flow domain. Using a too small number of MPI-ranks for this domain, i.e. 16 MPI ranks for the flow domain as well using a too small number of MPI-ranks for the acoustics, leads to the longest runtimes of 1336 and 1535 s respectively. The optimal setting is neither to the one nor to the other extreme, but in a medium, best suited distribution which reduces the runtime by roughly a factor of 2.

The best setup nevertheless is the adapted configuration which uses a much coarser mesh for the acoustic domain, yet settled by an increased order for the DG scheme. By this change in the numerical parameters, the runtime can be decreased by another factor of 2. These performance benefits become even more crucial when enlarging the acoustic far field even more or when solving different numerical problem where the length scale in the flow domain are magnitudes smaller than in the presented testcase.

5 Conclusion

Partitioned coupling is a promising strategy to solve multi-scale as well as multi-physic problems on todays supercomputer. By splitting the whole domain into single physics subdomains and enabling interaction via surface coupling, each single physics domain can be solved by individual solvers using numerical methods which are perfectly tailored to the underlying physics. Hence, problems which might not be not feasible in a monolithic approach, due to e.g. too different length scales ending in large computational costs, can be accomplished.

We presented two different coupling approaches namely a multi-solver approach utilizing an external coupling library which takes care about steering, data mapping as well as data communication, but uses the individual solvers as black boxes, and an integrated approach, making use of all knowledge available on the solver, implemented within one numerical framework. This approach suffers from a loss of generality but gains performance.

Exploiting a higher order method in the solver has the advantage that polynomial approximations on the coupling surface are available and therefore, can be used within the integrated coupling approach *APESmate*. In contrast, the multi-solver approach with the coupling library *preCICE*requires an additional interpolation method for the data mapping. For non-matching grids, which typically occur when coupling different numerical resolution, using first order interpolation show unsatisfactory simulation error. Using direct polynomial evaluation for the data mapping, which is one key benefit of *APESmate* exhibits very good results when coupling high order which was shown with the example of coupling 64th order in space with and 32th order in space. For medium order, using *preCICE*with 2nd order radial basis functions and *APESmate* with direct evaluation, both yield satisfying numerical results.

Comparing the performance of the integrated approach *APESmate* with the multi-solver approach using *preCICE*on a modern supercomputer SuperMuc at LRZ, Munich, *APESmate* shows an advantage of 20 % lower overall computation time. This confirms the expected performance benefits gained by the tight integration of the coupling with the solver, which allows for exploitation of knowledge about internal data structures. But even that the multi-solver approach can not compete with a fully integrated approach in terms of overall runtime, the scalability is nevertheless satisfying.

Partitioned coupling leads to different work load in the single physics domains and hence, properly chosen number of compute resources can reduce the overall computational costs. This is shown on the example of a Gaussian pressure distribution, where a non-linear flow domain (Euler equations) is coupled to a surrounding acoustic domain (linearized Euler equations). Chosing the right distribution of MPI-ranks per subdomain, the computational cost is reduced by a factor of 2. Adaption of the numerical resolution in the individual domains, e.g. by coarsening the grid resolution and increasing the order in the acoustic domain can reduce the computational cost even more, in our example by a factor of 2 compared to the matching resolution coupling.

The focus of future work is on numerical challenges, in particular the coupling of different timesteps. Enabling subcycling of one solver by assuring a consistent timestep even for large differences in individual timesteps will give further performance benefits.

Acknowledgements The financial support of the priority program 1648 - Software for Exascale Computing 214 (www.sppexa.de) of the German Research Foundation. The performance measurements were performed on the Supermuc supercomputer at Leibniz Rechenzentrum (LRZ) der Bayerischen Akademie der Wissenschaften. The authors wish to thank for the computing time and the technical support.

References

1. 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 (2015). Accepted
2. 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)
3. Hesthaven, J.S., Warburton, T.: Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications, 1st edn. Springer Publishing Company, Incorporated (2007)
4. Klimach, H.G., Hasert, M., Zudrop, J., Roller, S.P.: Distributed octree mesh infrastructure for flow simulations. In: J. Eberhardsteiner (ed.) ECCOMAS 2012 - European Congress on Computational Methods in Applied Sciences and Engineering, e-Book Full Papers (2012)
5. Masilamani, K., Klimach, H., Roller, S.: Highly efficient integrated simulation of electro-membrane processes for desalination of sea water. In: W.E. Nagel, D.B. Kröner, M.M. Resch (eds.) High Performance Computing in Science and Engineering '13, pp. 493–508. Springer, New York (2013). doi:10.1007/978-3-319-02165-2
6. Roller, S., Bernsdorf, J., Klimach, H., Hasert, M., Harlacher, D., Cakircali, M., Zimny, S., Masilamani, K., Didinger, 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)
7. Shukaev, A.K.: A fully parallel process-to-process intercommunication technique for precice. Master's thesis, Institut für Informatik, Technische Universität München (2015)
8. Tam, C.K.W.: Computational Aeroacoustics. Cambridge University Press (2012). http://dx.doi.org/10.1017/CBO9780511802065. Cambridge Books Online
9. Zudrop, J.: Efficient numerical methods for fluid- and electro-dynamics on massively parallel systems. Ph.D. thesis, RWTH Aachen (2015)
10. Zudrop, J., Klimach, H., Hasert, M., Masilamani, K., Roller, S.: A fully distributed CFD framework for massively parallel systems. In: Cray User Group 2012. Stuttgart, Germany (2012)