ESAIM: PROCEEDINGS, February 2007, Vol.16, 1-15
Eric Cancès & Jean-Frédéric Gerbeau, Editors
DOI: 10.1051/proc:2007003

A COUPLING SCHEME FOR DIRECT NUMERICAL SIMULATIONS WITH AN ACOUSTIC SOLVER

A. Babucke, M. Dumbser, J. Utzmann[1]

Abstract. A code for direct numerical simulations (DNS) has been coupled with an acoustic solver for aeroacoustic applications. The DNS code is intended to compute the noise generation while the acoustic far-field is evaluated using the linearized Euler equations. Both codes run on different computers on which they show their best computational performance. As a test case, we show a pressure pulse emitting from the DNS domain and being propagated by the acoustic solver.

1. Introduction

Noise reduction is an important issue for a wide range of technical problems but the mechanisms of aeroacoustic noise generation are not yet understood properly. Aeroacoustic simulations are a relatively new field in computational fluid dynamics showing several difficulties. On the one hand, one is interested in a large acoustic domain while on the other hand, a relatively small flow field with high resolution in space and time is needed to simulate the noise generation. Due to the small amplitudes of the sound compared to the hydrodynamic fluctuations, boundary conditions as well as the discretization have to be chosen carefully to avoid that the numerical scheme itself acts as a relevant acoustic source. Two main approaches exist for these computations:
1.) The flow field is computed without the acoustic field. Then an acoustic analogy, e.g. Lighthill [9] or an acoustic solver is applied using source terms from the flow field or values from the boundaries.
2.) The computation includes the flow field and a relevant part of the acoustic field, computing the acoustic field directly with the same computational scheme. The typical assumption for sound propagation is that viscous terms can be neglected for acoustic wave transport. Therefore it seems reasonable to compute the region of sound generation with a DNS code and the acoustic far field with the linearized Euler equations. The method presented here couples the two codes in both directions with data exchange at every common timestep. Due to that, the acoustic field can still interact with the flow field and no artificial boundary condition at the exchange plane is needed. As both codes contain different numerical schemes, they show best computational performance on different types of computers. Therefore the coupling mechanism uses the regular network protocol [13] instead of intra-machine communication like MPI [11].

2. Numerical Methods

2.1. DNS code

On the DNS side, we use the NS3D code based on the three-dimensional unsteady compressible Navier-Stokes equations. It uses the conservative formulation solving for the conservative variables $\mathbf{Q} = [\rho, \rho u, \rho v, \rho w, E]$

[1] Institut für Aerodynamik und Gasdynamik, Universität Stuttgart, Pfaffenwaldring 21, 70550 Stuttgart

containing the density, the three momentum densities and the total energy per volume

$$E = \rho \cdot c_v \cdot T + \frac{\rho}{2} \cdot \left(u^2 + v^2 + w^2\right) \ . \tag{2.1}$$

The continuity equation, the three momentum equations and the energy equation can be written in vector notation

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z} = 0 \tag{2.2}$$

with the flux vectors **F**, **G** and **H**:

$$\mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p - \tau_{xx} \\ \rho uv - \tau_{xy} \\ \rho uw - \tau_{xz} \\ u(E+p) + q_x - u\tau_{xx} - v\tau_{xy} - w\tau_{xz} \end{bmatrix} \tag{2.3}$$

$$\mathbf{G} = \begin{bmatrix} \rho v \\ \rho uv - \tau_{xy} \\ \rho v^2 + p - \tau_{yy} \\ \rho vw - \tau_{yz} \\ v(E+p) + q_y - u\tau_{xy} - v\tau_{yy} - w\tau_{yz} \end{bmatrix} \tag{2.4}$$

$$\mathbf{H} = \begin{bmatrix} \rho w \\ \rho uw - \tau_{xz} \\ \rho vw - \tau_{yz} \\ \rho w^2 + p - \tau_{zz} \\ w(E+p) + q_z - u\tau_{xz} - v\tau_{yz} - w\tau_{zz} \end{bmatrix} \tag{2.5}$$

containing normal stresses

$$\tau_{xx} = \frac{\mu}{Re}\left(\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\frac{\partial v}{\partial y} - \frac{2}{3}\frac{\partial w}{\partial z}\right) \tag{2.6}$$

$$\tau_{yy} = \frac{\mu}{Re}\left(\frac{4}{3}\frac{\partial v}{\partial y} - \frac{2}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\frac{\partial w}{\partial z}\right) \tag{2.7}$$

$$\tau_{zz} = \frac{\mu}{Re}\left(\frac{4}{3}\frac{\partial w}{\partial z} - \frac{2}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\frac{\partial v}{\partial y}\right) \ , \tag{2.8}$$

shear stresses

$$\tau_{xy} = \frac{\mu}{Re}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \tag{2.9}$$

$$\tau_{xz} = \frac{\mu}{Re}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right) \tag{2.10}$$

$$\tau_{yz} = \frac{\mu}{Re}\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \tag{2.11}$$

and the heat flux

$$q_x = -\frac{\vartheta}{(\kappa-1)RePrMa^2}\frac{\partial T}{\partial x} \tag{2.12}$$

$$q_y = -\frac{\vartheta}{(\kappa-1)RePrMa^2}\frac{\partial T}{\partial y} \tag{2.13}$$

$$q_z = -\frac{\vartheta}{(\kappa-1)RePrMa^2}\frac{\partial T}{\partial z}\ . \tag{2.14}$$

Closure of the equation system is provided by the ideal gas law:

$$p = \frac{1}{\kappa Ma^2}\cdot\rho T\ . \tag{2.15}$$

The specific heats c_p and c_v are assumed to be constant and therefore their ratio $\kappa = c_p/c_v = 1.4$ is constant as well. Temperature dependence of viscosity μ is modeled using the Sutherland law:

$$\overline{\mu}(T) = \overline{\mu}(\overline{T}_\infty)\cdot T^{3/2}\cdot\frac{1+T_s}{T+T_s}\ , \tag{2.16}$$

where $T_s = 110.4K/\overline{T}_\infty$ and $\overline{\mu}(\overline{T}_\infty = 280K) = 1.735\cdot 10^{-5}kg/(ms)$. Thermal conductivity ϑ is obtained by assuming a constant Prandtl number $Pr = c_p\mu/\vartheta$. The most characteristic parameters describing a compressible viscous flow are the Mach number Ma and the Reynolds number $Re = \overline{\rho}_\infty\overline{U}_\infty\overline{L}/\overline{\mu}_\infty$. with $\overline{L}$ being a reference length and the overbar denoting dimensional values. Dimensionless quantities are obtained by normalizing time with $\overline{L}/\overline{U}_\infty$, velocities with the inflow velocity $\overline{U}_\infty$, p with $\overline{\rho}_\infty\cdot\overline{U}^2_\infty$, specific heats with $\overline{L}/\overline{U}^2_\infty/\overline{T}_\infty$ and all other quantities with their respective inflow values.

Spatial discretization in streamwise x and normal y direction is done by 6^{th}-order compact finite differences. High wavenumbers which are generated by the non-linear terms in the Navier-Stokes equations are damped by applying alternating up- and downwind-biased finite differences for the convective terms [7]. The second derivatives are evaluated directly resolving the viscous terms more precisely [2]. Arbitrary grid transformation in the x-y plane is provided by mapping the physical x-y grid on an equidistant computational ξ-η grid as described by Anderson [1]:

$$x = x(\xi,\eta)\ ,\ y = y(\xi,\eta)\ . \tag{2.17}$$

Since the flow is assumed to be periodic in spanwise direction, we apply a spectral ansatz for the $z-$direction:

$$f(x,y,z,t) = \sum_{k=-K}^{K}\hat{F}_k(x,y,t)\cdot e^{i(k\gamma z)}\ . \tag{2.18}$$

f denotes any flow variable, $\hat{F}_k$ its complex Fourier coefficient, K the number of spanwise modes and $i = \sqrt{-1}$. The fundamental spanwise wavenumber γ is given by the fundamental wavelength λ_z representing the width of the integration domain by $\gamma = \frac{2\pi}{\lambda_z}$. De- aliasing is provided by using only 2/3 of the maximum number of modes.

Time integration is done using the standard 4^{th}-order Runge-Kutta scheme (see e.g. [7]). The code is parallelized by domain decomposition in the ξ-η plane using MPI [11] and a shared memory parallelization in spanwise direction. Good vectorization properties provide a fast and efficient usage of vector computers. More details on the NS3D code can be found in [3].

2.2. **Acoustic Code**

The general approach for our acoustic code is to use explicit high order schemes. For Finite Volume (FV) schemes, Schwartzkopff et al. [14, 15, 17, 18] adopted and optimized the idea of the ADER schemes by Toro et al. [20–22] for linear and nonlinear equations on structured grids. The main advantage of the ADER-FV schemes on structured grids is that the first step in a FV scheme, the reconstruction, becomes rather simple, as all neighborhoods are known directly. Dumbser et al. [4–6] extended the ADER idea on Discontinuous Galerkin (DG) methods. The very compact ADER-DG formulation does not need a reconstruction and thus provides the possibility to achieve arbitrary high order of accuracy in space and time even on unstructured grids. This is particularly useful for accurate noise propagation in the time domain around complex obstacles or in complex geometries. Arbitrary high order discretization of curved wall boundaries on unstructured grids is also introduced using superparametric DG elements.
Both ADER-FV and ADER-DG schemes performed well either as stand-alone method or in the framework of a heterogeneous domain decomposition approach (Utzmann et al. [23]).
We chose the ADER-DG method as acoustic solver for the coupling with the DNS code because of several reasons. First of all, it is a good testbed for the coupling procedure, as it combines FV properties (elementwise solution) with a Finite Element (FE) discretization which contains several degrees of freedom per element. Hence both aspects can be examined. Second, the unstructured code offers high flexibility and can be used for a variety of test cases, e.g. also calculations in complex geometries. Finally, it allows an efficient parallelization for the acoustic domain due to its locality. On the other side, it is clear that the DG method is far more computationally expensive than e.g. the ADER-FV method for linear systems on structured grids and would not be the method of choice in a domain which is far away from complex geometries. However, this will be accepted in this first study.
Our implementation of the ADER-DG scheme is very general concerning the mesh, the equations to solve and the order of accuracy in space and time. The implemented scheme can solve general linear hyperbolic systems with variable coefficients and source terms on 2D and 3D unstructured domains and the user may chose *any* order of accuracy in space and time he needs. Arbitrary spatial accuracy is possible due to the availability of sets of hierarchical orthogonal basis functions for any desired polynomial degree. For linear systems, it is sufficient that the user provides the system Jacobian matrices together with their eigenvalues and left and right eigenvectors. In our case, the governing equations are the linearized Euler equations (LEE), here in primitive variables:

$$\mathbf{U}'_{\mathbf{t}} + \underline{\underline{\mathbf{A}}}_{\mathbf{0}}\mathbf{U}'_{\mathbf{x}} + \underline{\underline{\mathbf{B}}}_{\mathbf{0}}\mathbf{U}'_{\mathbf{y}} + \underline{\underline{\mathbf{C}}}_{\mathbf{0}}\mathbf{U}'_{\mathbf{z}} = \mathbf{0} \tag{2.19}$$

with $\mathbf{U}'$ denoting the perturbation around the background state $\mathbf{U_0}$:

$$\mathbf{U}' = \mathbf{U} - \mathbf{U_0} = \begin{bmatrix} \rho' \\ u' \\ v' \\ w' \\ p' \end{bmatrix}, \quad \underline{\underline{\mathbf{A}}}_{\mathbf{0}} = \begin{bmatrix} u_0 & \rho_0 & 0 & 0 & 0 \\ 0 & u_0 & 0 & 0 & \frac{1}{\rho_0} \\ 0 & 0 & u_0 & 0 & 0 \\ 0 & 0 & 0 & u_0 & 0 \\ 0 & \gamma p_0 & 0 & 0 & u_0 \end{bmatrix} \tag{2.20}$$

$$\underline{\underline{\mathbf{B}}}_0 = \begin{bmatrix} v_0 & 0 & \rho_0 & 0 & 0 \\ 0 & v_0 & 0 & 0 & 0 \\ 0 & 0 & v_0 & 0 & \frac{1}{\rho_0} \\ 0 & 0 & 0 & v_0 & 0 \\ 0 & 0 & \gamma p_0 & 0 & v_0 \end{bmatrix}, \quad \underline{\underline{\mathbf{C}}}_0 = \begin{bmatrix} w_0 & 0 & 0 & \rho_0 & 0 \\ 0 & w_0 & 0 & 0 & 0 \\ 0 & 0 & w_0 & 0 & 0 \\ 0 & 0 & 0 & w_0 & \frac{1}{\rho_0} \\ 0 & 0 & 0 & \gamma p_0 & w_0 \end{bmatrix}. \tag{2.21}$$

3. Domain Coupling

The aim of the present work was to couple two stand-alone solvers at their common boundary in order to make a direct aeroacoustic simulation possible. The coupling procedure allows the change of equations, grid types, numerical methods and timesteps. First of all, the DNS code calculates the noise generated in the flow by solving the nonlinear Navier-Stokes equations. For the sound propagation, we switch to the linearized Euler equations in the acoustic domain. Second, we allow the mesh to change at a domain interface. As a Finite Difference scheme is used in the DNS domain, a fine, structured, node based grid is used in order to resolve the small scale flow phenomena. In the acoustic domain, much coarser Discontinuous Galerkin elements can be used in order to calculate the propagation of the sound waves which have a rather large wave length. The numerical methods used are always explicit time integration methods with a time step that is chosen as large as possible in each domain. As a consequence, not only the spatial grid size has a jump at an interface, but also the time step. In the following paragraphs we explain some basic parts of the coupling method.

3.1. Coupling of Different Equations

The coupling of different equations is a demanding issue. The coupling of nonlinear and linear Euler equations has been discussed by Schwartzkopff et al. [12, 16, 23]. It could be shown, that reflections at the coupling boundaries can be reduced drastically by prescribing rather the continuity of the state variables at the coupling boundary than the continuity of the flux. The coupling of the nonlinear Navier-Stokes equations (DNS domain) with the linearized Euler equations (acoustic domain) follows this principle and allows a discontinuous interface flux while preserving the continuity of the state.

3.2. Coupling of Different Grids

The coupling of arbitrary FV- and DG- grids (Utzmann [23]) has been extended for FD grids. The DNS domain Ω_{FD} and the acoustic domain Ω_{DG} are coupled at their common boundary only over data in their ghostelements. There are several rows of ghostpoints (their number depends on the chosen order of accuracy) for the FD scheme, for the DG scheme only one row of ghostcells is needed (Fig. 1 and 3). These ghostelements are then used by the numerical methods to update the inner elements in each domain. In the following, the basic coupling mechanisms between the two discretizations shall be described. The interpolation procedure is illustrated for 2D elements while the actual implementation is 3-dimensional.

(1) **Interpolation from the FD domain onto the DG ghostcells:** As the domains are aligned along their common boundary, the DG ghostcells lie completely within the FD domain (Fig. 1). In order to obtain the values for the elements, a 3-dimensional Lagrange interpolation is performed: The state vector $\mathbf{U}$ is interpolated from the regular FD mesh onto the position of each Gauss point iGP in the element's volume. Then the Gauss point values $\mathbf{U_{iGP}}$ are integrated and projected onto the DG cell's degrees of freedom $\hat{\mathbf{U}}_{\mathbf{Ghostcell}}$:

$$\hat{\mathbf{U}}_{\mathbf{Ghostcell}} = \sum_{\mathbf{iGP=1}}^{\mathbf{nGP}} \sum_{\mathbf{iDegFr=1}}^{\mathbf{nDegFr}} \frac{\omega_{\mathbf{iGP}} \cdot \phi_{\mathbf{iGP,iDegFr}} \cdot \mathbf{U_{iGP}}}{\mathbf{M_{iDegFr,iDegFr}}}, \tag{3.22}$$

with the integration weights ω_{iGP}, the mass matrix $M_{iDegFr,iDegFr}$ and the value of the base function $\phi_{iGP,iDegFr}$ at the Gauss point position. The interpolation stencils are taken from the FD grid and their order can be arbitrary. However, odd numbers of interpolation points in each direction provide more symmetrical stencils and are preferred. Figure 2 depicts a close-up of one of the ghostcells: In this case, fifth order stencils are used for the interpolation.

(2) **Interpolation from the DG domain onto the FD ghostpoints:** A unique neighbor element from the DG domain can be assigned to every ghostpoint of the FD coupling boundary (Fig. 3). As the DG cell's degrees of freedom contain the complete polynomial information, there is no need for an

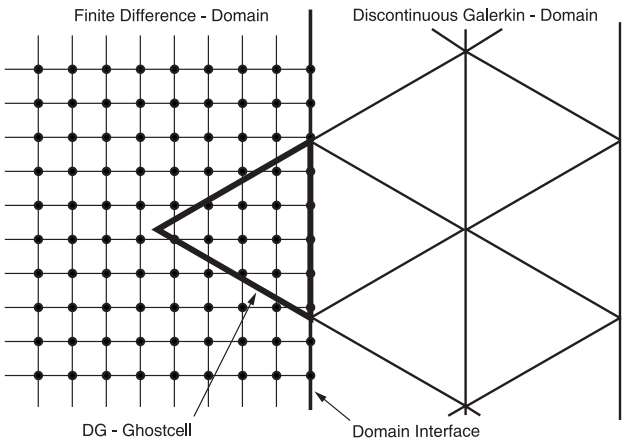

FIGURE 1. A DG ghostcell connecting with the FD domain.

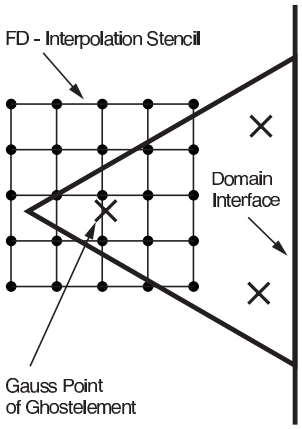

FIGURE 2. Interpolation onto the element's Gauss points.

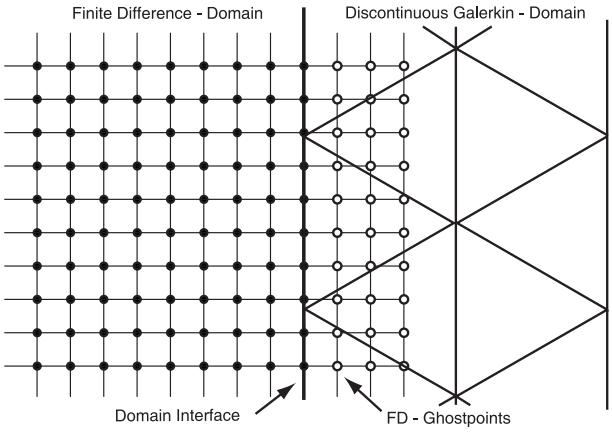

FIGURE 3. FD ghostpoints connecting with the DG domain.

interpolation stencil and the value of the state vector $\mathbf{U_{Ghostpoint}}$ at the position of each FD ghostpoint can be obtained easily:

$$\mathbf{U_{Ghostpoint}} = \sum_{\mathbf{iDegFr=1}}^{\mathbf{nDegFr}} \phi_{\mathbf{iDegFr}}(\mathbf{X_{FD}}) \cdot \hat{\mathbf{U}}_{\mathbf{iDegFr}}, \tag{3.23}$$

with the degrees of freedom $\hat{\mathbf{U}}_{\mathbf{iDegFr}}$ in the DG neighborcell and the value $\phi_{iDegFr}(\mathbf{X_{FD}})$ of the base function at the position $\mathbf{X_{FD}}$ of the ghostpoint. The spatial derivatives of the state vector are obtained in the same way by using the derivatives of the base functions.

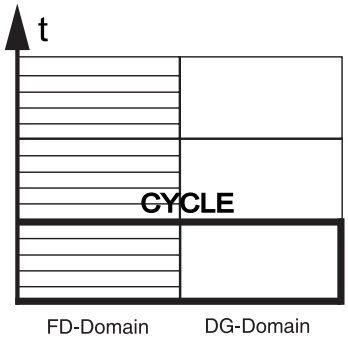

FIGURE 4. Data exchange cycle of the two domains.

3.3. **Coupling of Different Time Steps**

The idea of multi size meshes with different time steps has been examined by Tam et al. [19] for DRP schemes on particularly designed grids. This basic idea has been extended for FV- and DG-schemes on arbitrary meshes (Utzmann et al. [23]) and can be also applied for the considered coupling between the Finite Difference DNS domain and the DG acoustic domain.
The idea is illustrated in Fig. 4:

The acoustic domain's timestep is constant due to the linear problem. It is considered greater than or equal to the DNS domain's timestep. In order to ensure an equidistant temporal output for postprocessing (e.g. Fourier analysis), the DNS timestep is set constant as well for the simulation. The local CFL number (which defines the stability limit for convective problems) of the acoustic domain is adjusted such, that the timestep is a multiple of the neighboring domain. Hence, after the acoustic domain performed one timestep, all domains have the same time level again. Meanwhile, the other domain has been updated n times with $n \geq 1$. If a Runge-Kutta time integration scheme is chosen for the FD domain, also intermediate timesteps are needed and the overall number of timesteps for the FD domain will be $2 \cdot n + 1$. The question is how to treat the ghostpoints that need to be provided with an updated value for every small timestep. As it has been shown [23] for FV- and DG-schemes, the key idea is here to use the Cauchy-Kovalevskaja (CK) procedure to update the ghostpoints. The starting point is a Taylor series in time

$$\mathbf{U}(\mathbf{X_0}, \mathbf{t^n} + \tau) = \mathbf{U}(\mathbf{X_0}, \mathbf{t^n}) + \tau \frac{\partial \mathbf{U}(\mathbf{X_0}, \mathbf{t^n})}{\partial \mathbf{t}} + \frac{\tau^{\mathbf{2}}}{\mathbf{2}} \frac{\partial \mathbf{U^2}(\mathbf{X_0}, \mathbf{t^n})}{\partial^{\mathbf{2}} \mathbf{t}} + \ldots + \frac{\tau^{\mathbf{k}}}{\mathbf{k!}} \frac{\partial \mathbf{U^k}(\mathbf{X_0}, \mathbf{t^n})}{\partial^{\mathbf{k}} \mathbf{t}} \tag{3.24}$$

The truncation order matches the chosen order of accuracy, e.g. third order for a $\mathcal{O}3$ method in space and time. Time derivatives are now replaced by spatial derivatives with the Cauchy-Kovalevskaja procedure [8, 23]. Once the derivatives at the point $\mathbf{X_0}$ are known, the state at an arbitrary time $t^n + \tau$ can be calculated easily. Figure 5 shows a whole cycle between two data exchanges including the updating of the ghostpoints via the CK procedure.

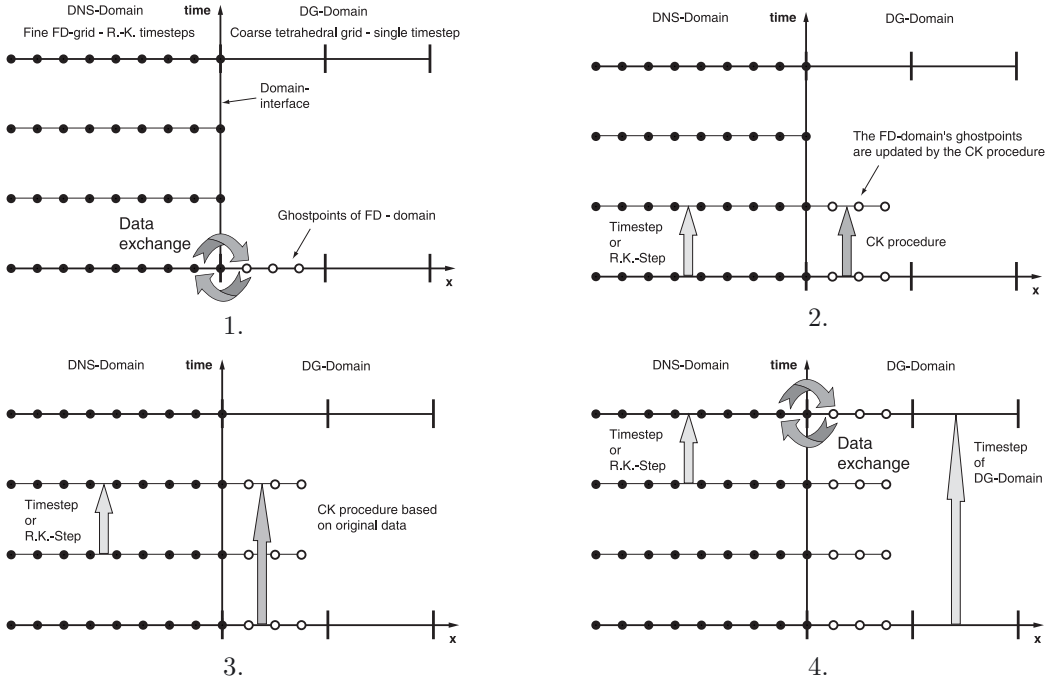

FIGURE 5. Coupling different time steps: 1. Data exchange at the common time level: The states are interpolated onto the ghostpoints of the FD domain and onto the ghostcells of the DG domain. For the FD ghostpoints, also the spatial derivatives are interpolated. 2. Time stepping of the FD domain. This can be understood as either an intermediate Runge-Kutta time step or a real time step. At the same time, its ghostpoints are lifted to the new time level by the CK procedure. 3. The FD domain makes another time step. Again, its ghostpoints are pushed to the new time level, based on the original data. 4. The DG domain performs one single time step and meets the FD domain at a common time level again. Then, a new data exchange takes place.

3.4. **Network connection**

As mentioned in the section above, both codes contain totally different numerics. Therefore they show a different behavior with respect to vectorization. The DNS code uses Cartesian grids showing a high vectorization ratio. This results in high computational performance on vector machines and a relative slow execution on scalar machines. On the other hand, the DG code is based on unstructured grids, having its best computational performance on scalar machines. Due to that we decided to implement a machine independent coupling mechanism which allows to run the codes on two different machines. Here, the DNS code runs on the NEC-SX8 vector machine of the High Performance Computing Center Stuttgart (HLRS) while the DG code is executed on its scalar front end machine TX7 containing 16 Itanium II processors.

The communication is based on the Transmission Control Protocol [13] consisting of a server and a client. The server opens a port and waits for the client to connect. To preserve the server being started before the client, the server must not run inside a queuing system. Therefore the DG code is the server, running interactively before the DNS code is submitted to the queue of the SX8 vector machine. For the client, one has to specify IP-address and port of the server. The communication itself is done by simply writing on or reading from the socket.

The initial communication defines the array sizes, the DNS code sends the baseflow to the DG code and the timestep ratio is determined. Then at every common timestep the set of primitive variables (ρ, u, v, w, p) near the boundary is exchanged. The communication is finished by closing the socket.

A big advantage of using the standard network [13] instead of MPI [11] is the fact, that the there are still two separate executables and the two codes do not have to be merged. Each code does not have to care about the internal parallelization concept of the other code, since only the first MPI process communicates with the other program. The data-distribution with the other internal processes is done by each program separately.

4. Numerical example

The example we have chosen is a uniform mean flow with $Ma_\infty = 0.5$ in streamwise direction, resulting in an ambient pressure of $p_\infty = 1/\kappa Ma^2 = 2.85714$. The temperature is $\overline{T}_\infty = 280K$, containing a temperature disturbance of $0.001 \cdot T_\infty$ with a radius of 1.0 located at $x = 39.0$ and $y = 0.0$ in the DNS domain. This leads to a circular pressure pulse being emitted in all directions and a temperature spot which is convected with the streamwise mean flow. The Reynolds number $Re = 500$ and the Prandtl number $Pr = 0.71$ which describe the viscous terms are of course only needed for the DNS domain.

The DNS grid has a range of $-15 \leq y \leq 15$ in normal and $0 \leq x \leq 115$ in streamwise direction. In order to prevent disturbances convected with the meanflow reaching the outflow, the grid is stretched in x direction from $x = 60.0$ onwards. The spanwise extent has been chosen with respect to favorable grid properties of the DG code and is $z = [0.0, 0.5]$ for the fine and $z = [0.0, 1.0]$ for the coarse grid, respectively. The DG domains are located above and below the DNS grid with the same extent in streamwise and spanwise direction and ranging to $y = \pm 25$. The wave propagation is evaluated by comparing discretization orders on different grids which are illustrated in Fig. 6 and 7.

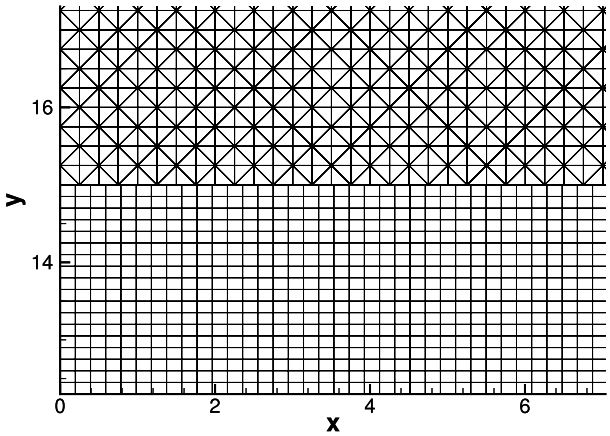

Figure 6. Detailed 2-d view of DNS- (bottom) and fine tetrahedral DG-grid (top)

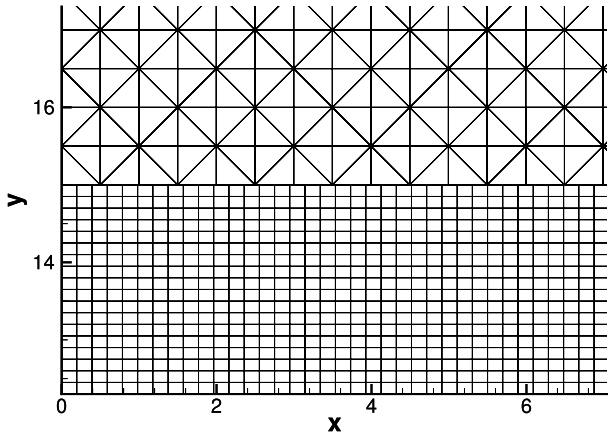

Figure 7. Detailed 2-d view of DNS- (bottom) and coarse tetrahedral DG-grid (top)

The following cases have been computed to evaluate the coupling scheme:

case	DNS grid (x y z)	DG elements	DG order
A	401 x 201 x 5	368800	2
B	401 x 201 x 5	368800	3
C	401 x 201 x 5	92400	4
D	401 x 201 x 5	92400	5
E	401 x 603 x 5	-	-

Case E is a larger DNS which has been performed to compare the results. The domain has three times the size of the original DNS ranging up to $y = \pm 45$. This gives enough space so that the wave does not reach the boundary for the considered time. As we can exclude errors due to boundary conditions, the DNS result may serve as reference data. The resulting pressure distribution in the x-y plane at time $t = 9.42477$ is shown in Fig. 8, illustrating the generated acoustic wave.

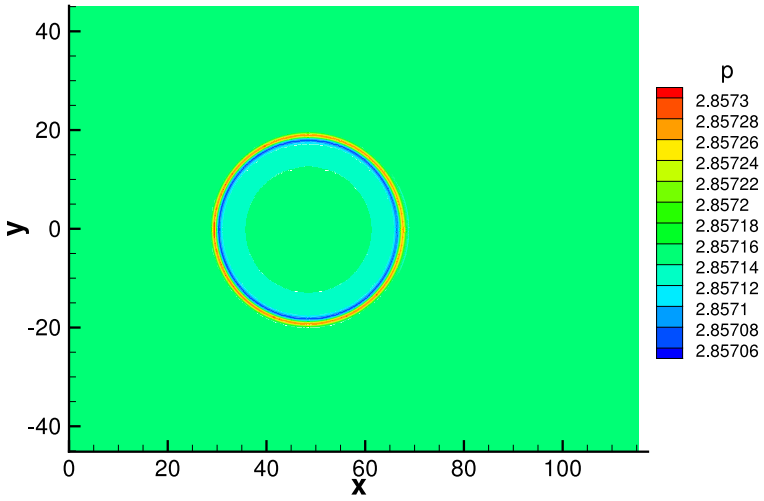

FIGURE 8. Instantaneous pressure field at time $t = 9.4247$, obtained by DNS without coupling.

The wave crossing the coupling interface is shown for case C in Fig. 9, using the polynomial ansatz of the elements in order to visualize the complete solution. Figure 10 illustrates the grid configuration visualizing only the integral mean values for the DG elements. Despite the different equations and discretization used in both codes, the wave crosses the coupling plane with almost no reflections.

The pressure distribution along a constant x position is shown in Fig. 11 and 13. The detailed views in Fig. 12 and 14 illustrate the differences between various DG orders and the DNS solution. At time $t = 7.8539$, when the pulse passes the coupling interface, a small amplitude error can be observed. This error may correspond to small reflections at the interface. Later in time, the amplitudes are the same for both DNS and DG scheme, as one can see in Fig. 14. The phase error compared to the DNS solution decreases with the higher order for the DG grid. It must be taken into account that the Navier-Stokes equations contain viscous terms while in the outer domain the linear Euler equations are solved. This might explain the better accordance of the amplitudes at the later timestep (Fig. 14).

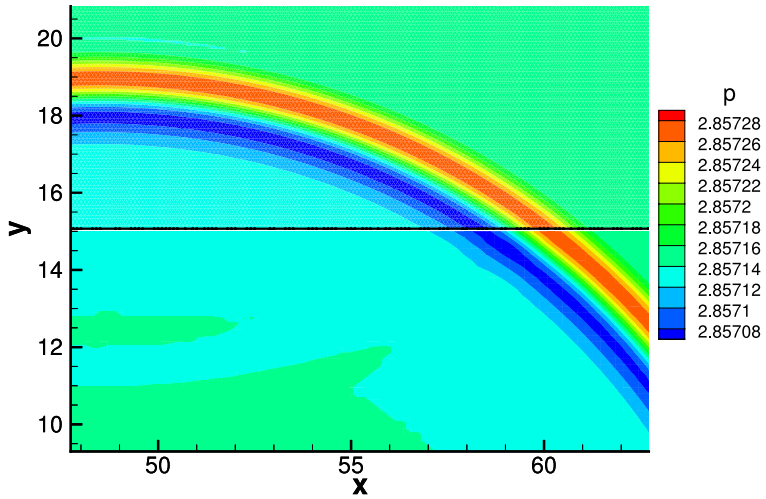

FIGURE 9. Acoustic wave crossing the coupling plane at time $t = 9.4247$ for 4^{th}-order DG scheme on the coarse grid.

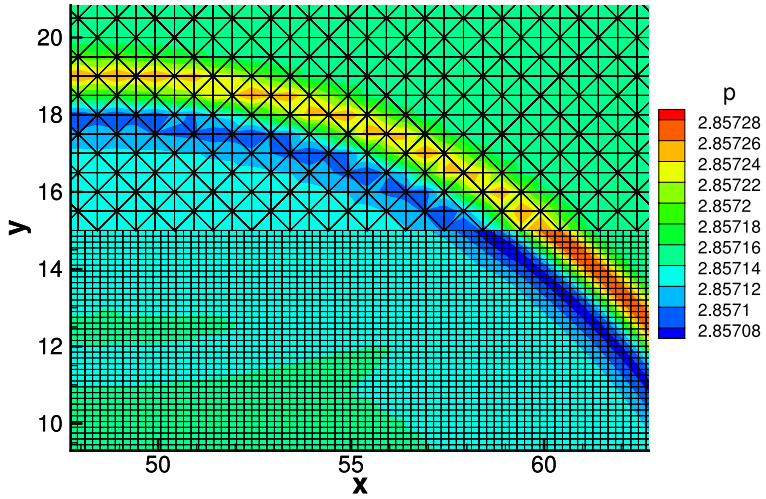

FIGURE 10. DNS and DG grids with integral mean values of the DG elements at time $t = 9.4247$ for 4^{th}-order DG scheme on the coarse grid.

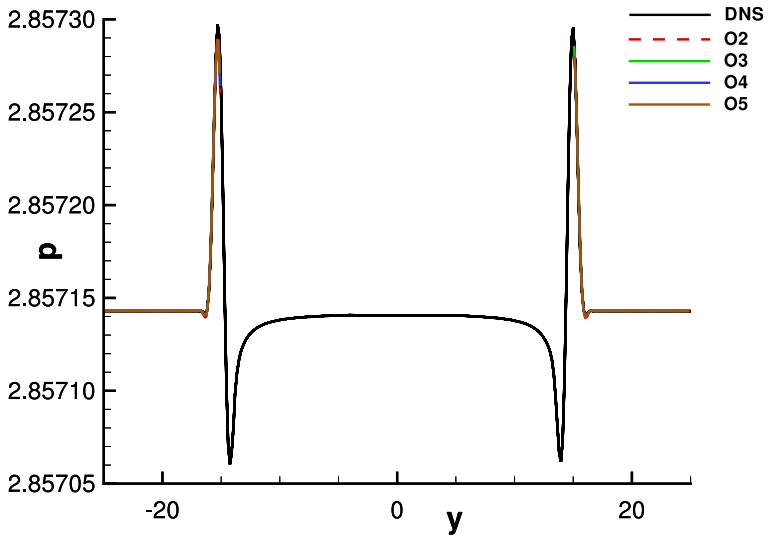

FIGURE 11. Pressure distribution at position $x = 52.03$ and time $t = 7.8539$.

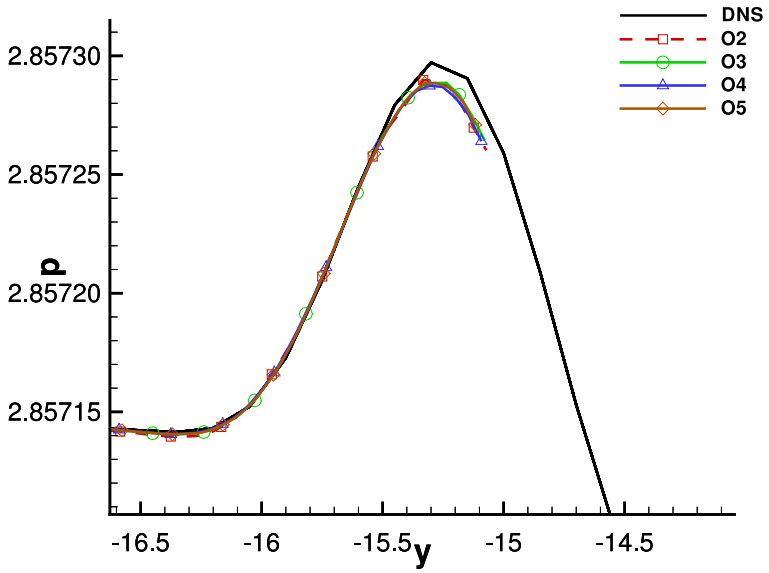

FIGURE 12. Detailed view of pressure distribution at position $x = 52.03$ and time $t = 7.8539$.

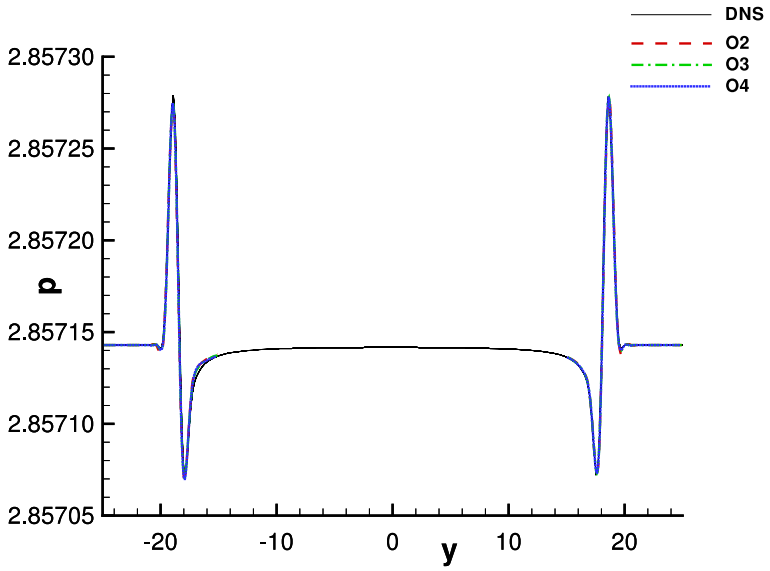

FIGURE 13. Pressure distribution at position $x = 52.03$ and time $t = 9.4247$.

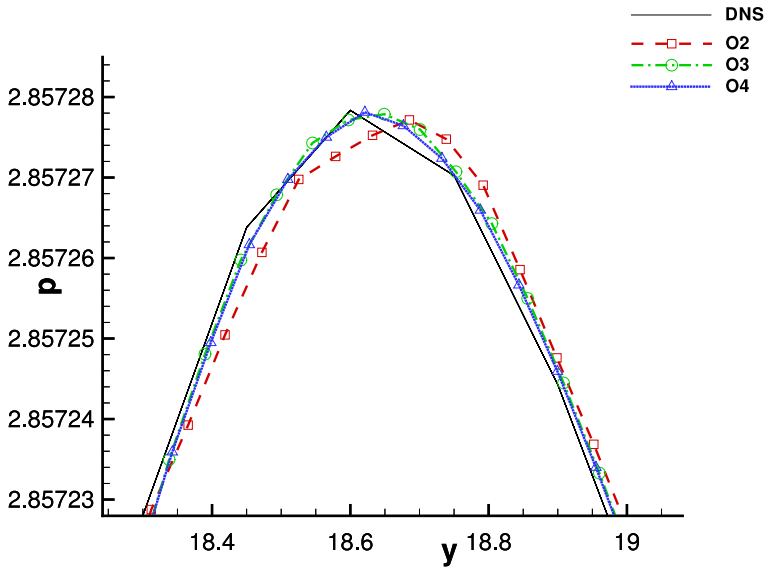

FIGURE 14. Detailed view of pressure distribution at position $x = 52.03$ and time $t = 9.4247$.

4.1. **Computational performance**

After comparing the results, we want to look at the computational performance. As mentioned above, the machines used here are a NEC-SX8 for the DNS code and an Itanium II machine for the DG code. The computational performance for the considered cases is summarized in the following table:

computational case	**A**	**B**	**C**	**D**	**E**
DG order	2	3	4	5	-
real time [s]	24079	60503	73869	195121	782
CPU time ADER-DG [s]	23560	59230	72869	193827	-
CPU time DNS-code [s]	576	571	507	503	2031
# CPU's ADER-DG	1	1	1	1	-
# CPU's DNS-code	1	1	1	1	3
Δt	0.003927	0.003927	0.003927	0.001964	0.015708
number of timesteps	800	800	800	800	800
DNS gridpoints	403005	403005	403005	403005	1209015
ADER-DG elements	368800	368800	92400	92400	-
comm. data [MB]	3.99	3.99	4.82	4.82	-

The CPU time of the DNS code varies slightly between approximately 500 to 600 seconds. This is due to the fact, that the computation did not use one complete node explicitly. The CPU time of the ADER-DG method grows exponentially with the order of the scheme and ranges from 6.5 to 53 hours for 800 timesteps. The total communication is less than five megabyte per timestep. Due to the good network connection between the SX8 and the Itanium II machine, the total communication per timestep takes less than one second.

As the timestep for case E is four times larger than in cases A to C and eight times higher than in case D, the timestep is limited by the DG code for this problem. Therefore the timestep ratio is set to one for the cases regarded here. However, the Runge-Kutta method requires values at intermediate timelevels. Therefore, the Cauchy-Kovalevskaja procedure (see section 3.3) is necessary despite the common timestep. More details on stability limits of DG schemes can be found in [10].

5. Conclusions

We could show, that the developed coupling mechanism is capable of transporting an acoustic wave package from the interior DNS domain to the outer unstructured grid, where the linearized Euler equations are solved. Despite the different equations and different discretizations used in both codes, almost no reflection could be observed at the coupling plane. Nevertheless, the computational performance did not convince as the DG scheme required 40 to almost 400 times the CPU time of the DNS code. However, regarding the the testbed character of the simulation, the coupling procedure has been validated with good results. The more severe timestep limitation for the ADER-DG scheme is a minor problem for 'real world' computations where the spatial resolution of the DNS is much higher than in the DG domain where only long wave acoustics need to be resolved. Furthermore, while the unstructured DG method is highly suitable in the vicinity of complex geometries, a much cheaper structured code (e.g. linear ADER-FV) could be used for far field computations, based on the proposed coupling mechanism.

References

[1] J. D. Anderson. *Computational Fluid Dynamics*. McGraw-Hill, 1995.

[2] A. Babucke, M. Kloker, and U.Rist. DNS of a plane mixing layer for the investigation of sound generation mechanisms. *to appear in Computers and Fluids*, 2006.

[3] A. Babucke, J. Linn, M. Kloker, and U. Rist. Direct numerical simulation of shear flow phenomena on parallel vector computers. In *High performance computing on vector systems: Proceedings of the High Performance Computing Center Stuttgart, 2005*. SpringerVerlag Berlin, 2006.

[4] M. Dumbser and C.-D. Munz. Ader discontinuous galerkin schemes for aeroacoustics. *Comptes Rendus Mécanique*, 333:683–687, 2005.

[5] M. Dumbser and C.-D. Munz. Arbitrary high order discontinuous galerkin schemes. In S. Cordier, T. Goudon, M. Gutnic, and E. Sonnendrücker, editors, *Numerical Methods for Hyperbolic and Kinetic Problems*, IRMA Series in Mathematics and Theoretical Physics, pages 295–333. EMS Publishing House, 2005.

[6] M. Dumbser and C.-D. Munz. Building blocks for arbitrary high order discontinuous galerkin schemes. *Journal of Scientific Computing*, 2006. Going to appear.

[7] M. J. Kloker. A robust high-resolution split-type compact fd scheme for spatial dns of boundary-layer transition. *Appl. Sci. Res.*, 59:353–377, 1998.

[8] P. Lax and B. Wendroff. Systems of conservation laws. *Communications in Pure and Applied Mathematics*, 13:217–237, 1960.

[9] M. J. Lighthill. On sound generated aerodynamically. i. general theory. *Royal Society of London Proceedings Series A*, 211:564–587, 1952.

[10] F. Lörcher, G. Gassner, and C.-D. Munz. Arbitrary high order accurate time integration schemes for linear problems. *Proceedings of ECCOMAS CFD 2006*. Submitted.

[11] MPI-Forum. MPI a message-passing interface standard. Technical Report CS-94-230, University of Tennessee, 1995.

[12] C. Munz and T. Schwartzkopff. Direct simulation of sound generation and sound propagation using heterogenous domain decomposition. In *Proccedings of WCCM*, July 2002.

[13] J. Postel. RFC-793: Transmission Control Protocol, 1981.

[14] T. Schwartzkopff, M. Dumbser, and C. Munz. Fast high order ADER schemes for linear hyperbolic equations and their numerical dissipation and dispersion. Technical Report 2003/35, Preprint series of SFB404, Stuttgart University, 2003.

[15] T. Schwartzkopff, M. Dumbser, and C. Munz. Fast high order ADER schemes for linear hyperbolic equations. *J. Comp. Phys.*, 197:532–539, 2004.

[16] T. Schwartzkopff and C. Munz. Direct simulation of aeroacoustics. In W. Wendland and M. Efendiev, editors, *Analysis and Simulation of Multifield Problems*, volume 12 of *Lecture Notes in applied and computational Mechanics*, pages 337–342. Springer, 2003.

[17] T. Schwartzkopff, C. Munz, and E. Toro. ADER: A high order approach for linear hyperbolic systems in 2d. *Journal of Scientific Computing*, 17(1-4):231–240, 2002.

[18] T. Schwartzkopff, C. Munz, E. Toro, and R. Millington. The ADER approach in 2d. In T. Sonar, editor, *Discrete Modelling and Discrete Algorithms on Continuum Mechanics*, pages 207–216, Berlin, 2001. Logos Verlag.

[19] C. Tam and K. Kurbatskii. Multi-size-mesh multi-time-step dispersion-relation-preserving scheme for multiple-scales aeroacoustic problems. *International Journal of Computational Fluid Dynamics*, 17:119–132, 2003.

[20] V. Titarev and E. Toro. ADER: Arbitrary high order godunov approach. *Journal of Scientific Computing*, 17(1-4):609–618, December 2002.

[21] V. Titarev and E. Toro. ADER schemes for three-dimensional nonlinear hyperbolic systems. *Isaac Newton Institute for Mathematical Sciences Preprint Series*, 2003.

[22] E. Toro, R. Millington, and L. Nejad. Towards very high order godunov schemes. In E. Toro, editor, *Godunov Methods. Theory and Applications*, pages 907–940. Kluwer/Plenum Academic Publishers, 2001.

[23] J. Utzmann, T. Schwartzkopff, M. Dumbser, and C.-D. Munz. Heterogeneous domain decomposition for caa. *AIAA Journal*, 2006. Going to appear.