V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

SIMULATION OF AUTOMOTIVE INJECTOR NOZZLE NOISE WITH FULLY COUPLED CFD/CAA SOLVER

Harald Klimach[*], Sabine P. Roller[*], Jens Utzmann [†], Claus-Dieter Munz[†]

[*]German Research School for Simulation Sciences GmbH and RWTH Aachen University,
Applied Supercomputing in Engineering, Schinkelstr. 2A, D-52062 Aachen
e-mail: s.roller@grs-sim.de, h.klimach@grs-sim.de

[†]Institute for Aerodynamics and Gasdynamics, Universität Stuttgart
Pfaffenwaldring 21, D-70550 Stuttgart
e-mail: {utzmann,munz}@iag.uni-stuttgart.de

Key words: Fluid Dynamics, Aero-acoustics, Multi-scale coupling, Industrial applications

Abstract. *Aero-acoustics simulation in complex geometry still is difficult to perform due to high computational costs. On the other hand, noise generation is an important problem in industry. The development of injector nozzles e.g. has to take into account the noise generated by the flow of the fuel through the nozzle. In this contribution, we describe the coupled CFD/CAA simulation of the nozzle flow with its noise generation and propagation which enables us to solve real world industrial problems.*

The idea is to apply a heterogeneous domain decomposition approach which allows for reducing the computational effort by applying different schemes in different parts of the computational domain. Thus, in each domain, only the necassary phenomena are covered and the best suited numerical scheme is applied. Each domain in itself is than again parallelized in itself, which in total leads to a 2-level parallelization approach.

The entire scheme than consists of the different modules for the different parts plus a coupling module which has to take into account the different properties. The coupling module itself is a compute-intensive part since it does not only deliver data from one domain to the other. It also has to determine the values at Gaussian points, i.e. evaluate functions and inter-/extrapolate variables. It also has to take into account the different time steps which are gone by the different regions. Special emphasize has to be paid to the spatial and temporal coupling.

Computations need computers. When talking about large applications, this typically means to run the simulation on supercomputers. But supercomputers often are highly specialized machines, which need adaption of the software codes. Not all machines are suited for all types of applications. Thus, another important point in coupled simulations is to map the entire scheme properly to computer systems.

With that, we are able to show the simulation of large industrial applications as the aeroacoustics in an automotive injector nozzle.

1 INTRODUCTION

Aero-acoustics simulation in complex geometry still is difficult to perform due to high computational costs. Aero-acoustics is a typical multi-scale problem where on the one side the small structures have to be resolved which generate the noise, while on the other side the propagation of the sound waves has to be covered over long distances. Besides amplitude damping, also the phase errors have to be kept low. This typically results in high order schemes at least for the sound propagation. Together with the large domains, this leads to high computational costs.

On the other hand, noise generation is an important problem in industry. The development of injector nozzles in automotive industry for example has to take into account the noise generated by the flow of the fuel through the nozzle.

In this contribution, we describe the coupled CFD/CAA simulation of the nozzle flow with its noise generation and propagation which enables us to solve real world industrial problems. We describe the coupling of different solvers via heterogeneous domain decomposition. With this approach, we can solve the viscous Navier-Stokes equations on unstructured grids with a suitable discretization technique like Discontinuous Galerkin in the region of noise generation around the complex geometry. As soon as possible we switch to a structured mesh to save the overhead of unstructured mesh handling and allow numerical schemes which are not easily deployed on unstructured meshes. When the physical area is reached, where viscosity is not relevant anymore, this effect can be dropped from the computed equation system. Even further, we can linearize the equations and apply e.g. a Finite Differences scheme on a much coarser grid, also allowing for much larger time steps, when nonlinear effects in the flow get neglictible. This approach allows to reduce the computational costs since in each part of the computational domain only the necessary effort has to be spent.

The layout of the paper is the following: in section 2 we describe the basic idea of heterogeneous domain decomposition. Section 3 deals with the spatial and temporal coupling mechanisms to maintain the high overall order in space as well as in time. Section 4 discusses the mapping of the scheme and its modules to different types of computer architectures to make use of the best suitable hardware system for each module. Section 5 describes the test case and discusses the results of the simulation. The last section draws the conclusions.

2 Heterogenous Domain Decomposition

The heterogeneous domain decomposition technique is mainly developed for multi-scale and multi-physics applications. Aero-acoustics is a typical multi-scale example, where the different scales can easily be separated between the flow and the acoustics.

The basic idea behind heterogeneous domain decomposition is to apply different schemes in the different regimes of the computational domain, taking into account as many physical phenomena as necessary but as less as possible. All physical effects which are important in a given part of the domain are covered but only where they play an essential role. In other regions of the computational domain, they often can be neglected. A complete separation of the involved domains and external coupling over the boundary conditions allows different underlying equations to be solved in each domain.

With the heterogeneous domain decomposition the computational domain is decomposed into problem specific parts, where in contrast to more common domain partitioning, the different domains can use individual solution methods. It is thus possible to reduce the computational effort by adapting the

- grid types (structured / unstructured)
- mesh sizes (coarse / fine mesh)
- time steps (larger / smaller time steps)
- spatial and/or temporal order of the scheme (p-adaptivity)
- discretization method (Finite Differences, Finite Volumes, Discontinuous Galerkin)
- equations (Navier-Stokes, Euler, LEE)

Such a decomposition with heterogeneous domains is shown in figure 1 for a 2D problem. This is a typical setup with the unstructured domain around a geometric obstacle, and rectangular cartesian meshes for the obstacle free area.

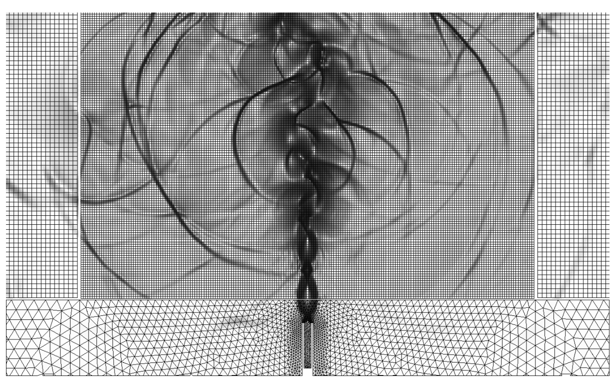

Figure 1: Heterogenous Domains in 2D

Our application is mainly developed for high speed flows and thus provides a coupling mechanism for explicit time marching solvers only. For the industrial nozzle flow acoustics described below we apply basically 4 domains:

- Inner Flow domain: fine grid, unstructured, small time step, $P_N P_M$ scheme [6], nonlinear Navier-Stokes, around obstacles

- Outer Flow domain: fine grid, structured, small time step, high order FV, non-linear Navier-Stokes, no obstacles
- Inner acoustic domain: coarser grid, structured, larger time step, FV, non-linear Euler equations
- Outer acoustic domain: even coarser mesh, structured, large time step, FD, linearized Euler equations (LEE)

3 Coupling boundary treatment

An important point in the development of a heterogeneous domain decomposition method is the treatment of the coupling boundaries. We have to consider the spatial coupling as well as the temporal coupling, with the emphasize on the goal to keep the global high order of the scheme even across the boundaries.

3.1 Spatial coupling

The spatial coupling needs to exchange values in given discrete points which are then used by the neighbouring domain to set the ghost values. Coupling with entire ghost cells is not used since it would need to find intersections of several unstructured cells. While this was a working approach in 2D, it became unfeasably expensive in 3D. Allowing for arbitrary unstructured meshes this leads to complex intersection scenarios especially in corner cases with several different domains adjacent to a single point. Because the coupling via entire ghost cells has shown to be much to costly, the coupling is now done via discrete points which are used by the underlying numerical schemes anyway.

The construction of the ghost cells can be reduced to values at discrete points for all three used spatial discretization methods. By the restriction to discrete points the coupling becomes independent of the discretization, as each domain only needs to interpolate its own solution onto an exchanged set of coordinates to provide all necessary data to its neighbors. In each domain the ghost cells can then be reconstructed by just using the exchanged values at the appropriate points. The discrete points are equally easy to handle in 3D as in 2D and provide a path to data encapsulating for the parallel computation. The coupling ghost cells of one domain interacting with two neighbors are depicted in figure 2. For the ghost cell overlapping both neighbors, the constructing discrete points are indicated.

The coupling used, preserves the high order of the numerical scheme across domain interfaces, as the appropiate underlying discretisation and schemes are used on either side.

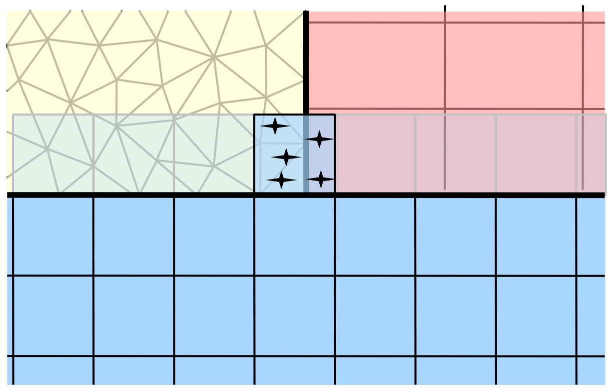

Figure 2: Spatial coupling. The values to be exchanged are computed at Gaussian integration points

3.2 Temporal coupling

Domains with different mesh sizes should be solved with different time steps as well. Otherwise, all domains would have to stick with the same smallest time step of the finest domain. This results on the one hand in unnecessary small time steps in all regions of the domain with coarser grid cells. On the other, it increases the numerical errors when to many too small time steps are done. Thus, adapting the time steps in the different regions of the computational domain reduces the computational costs and increases the accuracy of the solution.

In order to allow this, a sub-cycling algorithm is deployed, where the domains step with integer multiples of a common smallest time step and therefore meet at exact predictable common timelevels. Time integration is done using the "Arbitrary high order using derivatives" (ADER) scheme in the domains. This requires the Cauchy-Kowalevsky procedure [4] (also called Lax-Wendroff procedure) to obtain higher order time approximations. During the larger time step the domain with the smaller time step has to approximate the value of its ghost cells at each time level within the longer time step of the neighbor. This can is also done by using the Cauchy-Kowalevsky procedure, preserving the time order of the approximation for the values on intermediate time levels. Only at common time levels, data is exchanged. Thus the communication effort in a parallel simulation is usually less between two domains of different time steps than within the domains themselves. A more detailed description of the complete coupling scheme is given by[5].

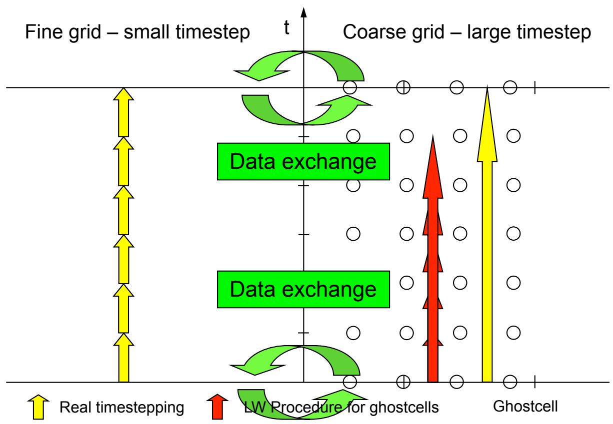

Figure 3: Temporal coupling

For parallel computation of the domains, a list of all the points, which can be found in a neighbor, has to be created for each neighbor and then exchanged over MPI Point-to-Point communication.

Due to the nature of the coupling between different domains, the computation of those domains can be realized even on systems with a quite poor network interconnection. This makes it well suited for inter-cluster communication in the heterogeneous network.

4 Mapping to different hardware systems

The heterogeneous approach as described above is necessary to enable the simulation of complex real life problem, but it still requires the power of supercomputers. To be even more efficient, we try to map the different parts of the simulation to different types of computers. This allows us to make use of the best suited machine for each module. In the current example, the structured parts of the simulation run best on a vector supercomputer, while the unstructured parts are much better suited for cache-based x86 architecture. We therefore coupled two different supercomputers to fit the different requirements of the structured and the unstructured solver[1]. The simulations run on up to 8 nodes of vector supercomputer NEC SX-9 with a peak performance of 12.8 TFlops and about 2048 cores of Intel Nehalem with another 22.7 TFlops nominal performance. This computational resources from HLRS where provided to us through the DAOSA project in the DEISA Extreme Computing Initiative.

To enable the simulation to be run on different machines at the time, we make use of the PACX-MPI library[2,3]. The PACX-MPI library was developed to enable the usage of a heterogeneous set of supercomputers without leaving the MPI context. It is a library sitting on top of the "native" MPI libraries within the individual machines. For the application, it is transparent in the sense that each domain communicates by calling a standard MPI call. The PACX library takes this call up and checks whether it remains within one machine. In this case, it is handed over to the native MPI library without any further action. Otherwise, it opens a communication to the other machine. With that, it allows even for different network protocols within each involved machine and between the machines. In the setup for this paper, the vector part uses IXS network, the Intel Nehalem cluster is connected via Infiniband, whereas the network between both machines is gigabit ethernet.

Figure 4 depicts the general PACX-MPI communication setup:

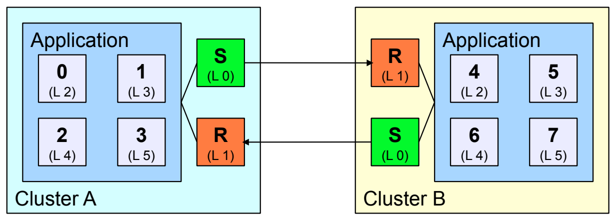

Figure 4: Transparent machine coupling using PACX-MPI

The local MPI ranks (L X) get remapped to global ranks and PACX reserves two local processes on each side for its communication. The application just "sees" the global ranks and has not to care about how communication between all its processes is done.

4.1 Domains and partitions

The entire parallelization approach then results in a 2-level parallelism: We decompose the computational domain into parts of different equations, mesh types, and discretization methods as described above according to the physical needs in the simulation. Thus, between these parts we deal with the heterogeneity. Each part is then parallelized in itself. To allow this an extra MPI Communicator is used to organize the communication within each simulation part. The two levels in parallelism are therefore represented by a global communicator for communications between all domains, and communicators local to each domain. This hierarchy allows the distribution of parts to different machines while within each partion a common (homogeneous) domain partitioning approach is applied.

The general scheme then also forms a 2-step algorithm: First do the domain calucu-lations (inside each domain), afterwards do the coupling calculations (interpolation and mapping between the domains). If this scheme would be applied in a naive way, it would lead to large portions of idle time due to the different runtimes of the computation parts on different architectures. A proper choice of parallel processes is difficult when on the one machine the calculation is fast, but coupling expensive and on the other machine vice versa:

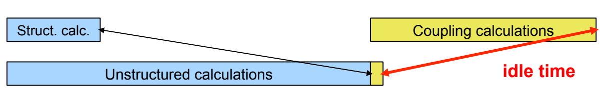

Figure 5: Synchronization issues when synchronizing between domain calculation and coupling

To obtain a better load balancing, the synchronization between coupling and domain calculations had to be shifted to the end of the entire block of domain calculation and coupling. The remaining idle times can then be eliminated by proper choice of processes on the different machines.

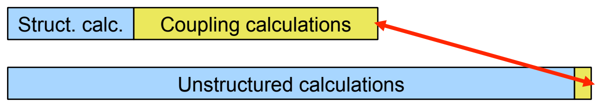

Figure 6: Synchronization issues when synchronizing only once, after domain calculation and coupling

To summarize this section, we can state that PACX-MPI, which is used for this distribution allows the coupling of different architectures without leaving the MPI context in the application itself. This makes the usage of a heterogeneous infrastructure very convenient from the applications point of view. The load balancing between the different machines can be obtained when the synchronization points are chosen carefully.

Thus, we now are able to use different machines for heterogeneous decompositions of the computational domain. By chosing the appropriate set of equation, discretization and machine reduces the computational costs to a point which is suitable for large real life simulations.

5 Showcase and results

The underlying scheme is a heterogenous solver for direct aero-acoustic simulation of the flow itself and the sound waves propagation through the entire domain of interest. The difficulty lays in the different scale of the injector nozzle and the surrounding area. The 4 nozzle outlets themselves are of size 0.15 mm, but the entire device is of size $1 \times 1 \times 1\text{m}^3$. The field around the nozzle outlets requires a fine discretization and an unstructured grid to resolve the geometry and the flow properties, while it is not possible – due to the computational costs – to resolve the entire device with such a fine mesh.

The setting therefore is a heterogeneous scenario: the computational domain for the surrounding near field which has to be discretized by a small-scale unstructured grid to resolve vorteces and flow structures is of size $8 \times 8 \times 10\text{mm}^3$ This unstructured domain is discretized by 15 million elements, the solver uses a $P_N P_M$ discretization of medium order in space and time.

As soon as possible, the mesh is switched to a structured mesh and the solution method now can be switched to a Finite Volume scheme, of high order in space as well as in time. This is solver is much cheaper with respect to CPU time as well as memory requirements. Therefore, a second – structured – domain of size which $2 \times 2 \times 5$ cm^3 with up to 1 billion cells surrounds the innermost domain, and a third domain, again with a structured, but even coarser mesh with only 350000 cells fills the computational domain up to the entire device size of $1 \times 1 \times 1\text{m}^3$.

The Mach number M=1.4, the Reynolds number Re=$17,000-30,000$. Figure 7a shows the setup of the innermost unstructured domain and the first surrounding structured domain. Figure 7b first simulation results for the acoustic waves.

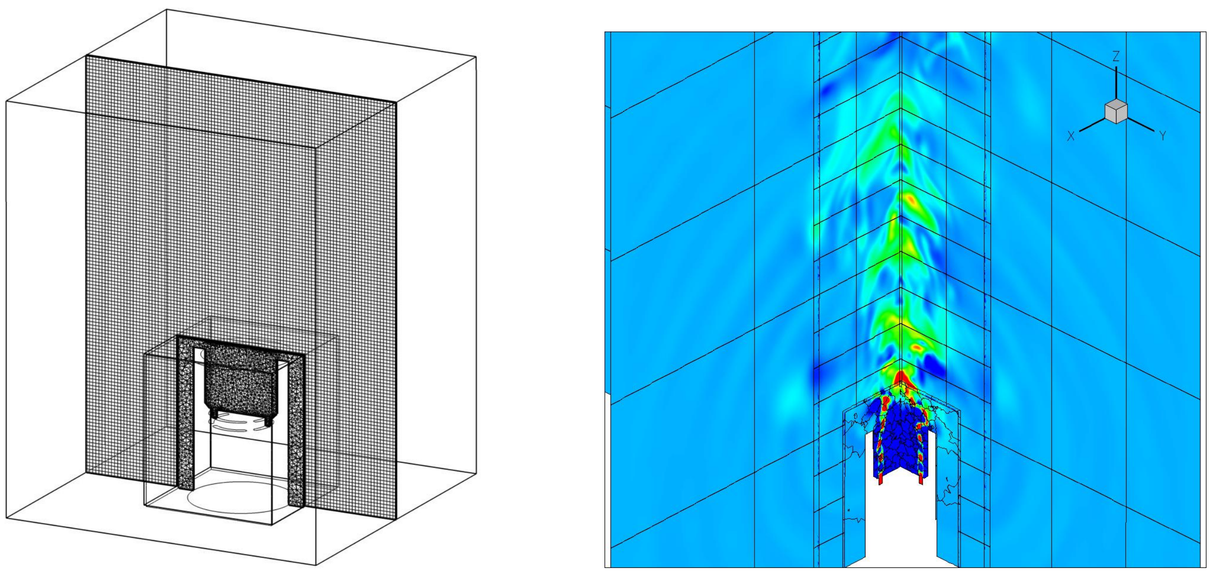

Figure 7: a) Domain decomposition, b) simulation results of aero-acoustics in 3D injector nozzle

6 CONCLUSIONS

In this paper, we have shown the simulation of a real world industrial aero-acoustics problem, and discuss the results as well as conclusions on computational requirements.

We can summarize the results as follows:

- Heterogeneity of both simulation and computing infrastructure does not have to be burden. With a proper mapping, advantages from all sides can be achieved
- The coupled application performs well in combination with PACX-MPI and different platforms
- Coupling library PACX-MPI
  - Library to enable communication in a heterogeneous cluster environment
  - Uses the "native" MPI library on each side
  - Other network protocol between clusters than within clusters
  - Transparent for the application
  - Needs to processes dedicated to inter-cluster communication on each side

- Bigger (3D) Navier-Stokes testcases require large computational power to cover the complex sound generating region as well as the domain where the propagation of the sound waves propagate

We thank the DEISA Consortium (www.deisa.eu), co-funded through the EU FP6 project RI-031513 and the FP7 project RI-222919, for support within the DEISA Extreme Computing Initiative.

REFERENCES

[1] H. Klimach, S. Roller and C.-D. Munz. Heterogeneous Parallel Aero-Acoustics Using PACX-MPI. *High Performance Computing on Vector Systems 2008*, pp. 215–222.

[2] E. Gabriel, M. Resch, Th. Beisel, R. Keller. Distributed computing in a heterogeneous computing environment. *Lecture Notes in Computer Science* **1497**, pp. 180 –187 (1998)

[3] R. Keller, M. Müller. PACX-MPI project homepage. http://www.hlrs.de/organization/amt/rojects/pacx-mpi

[4] S. v. Kowalevsky. Zur Theorie der partiellen Differentialgleichungen. *Ph.D. thesis, Philosophische Fakultät zu Göttingen* (1874)

[5] J. Utzmann, Th. Schwartzkopf, M. Dumbser, C.-D. Munz: Heterogeneous domain decomposition for computational aeroacoustics. *AIAA Journal* **44(10)**, pp. 2231 – 2250 (2006)

[6] M. Dumbser, D. Balsara, E.F. Toro, C.-D. Munz. A unified framework for the construction of one-step finite-volume and discontinuous Galerkin schemes. *Journal of Computational Physics*, 227:8209-8253, 2008.