A numerical method for multiphysics simulations based on hierarchical Cartesian grids

Gonzalo BRITO GADESCHI∗, Lennart SCHNEIDERS∗, Matthias MEINKE∗ and Wolfgang SCHRÖDER∗

∗ Institute of Aerodynamics, RWTH Aachen University
Wüllnerstraße 5a, 52062 Aachen, Germany
E-mail: office@aia.rwth-aachen.de

Received 18 September 2014

Abstract
We present a numerical method based on hierarchical Cartesian grids for the simulation of multiphysics problems, here in particular for the conjugate heat transfer between a fluid and a solid. The data structures developed for this simulation method allow to account for moving objects and are especially well suited for massive parallelization. The major problem of a suitable domain decomposition for a coupled fluid and heat conducting domain is solved by discretizing both domains on a joint hierarchical Cartesian mesh, where the individual cells can be marked according to the underlying physics, i.e. to be a fluid cell, a heat conducting cell or both. Individual solvers for the Navier-Stokes equations and the heat conduction equation are implemented which only operate on the Cartesian cells belonging to the fluid or solid subset of the joint hierarchical mesh. The solution strategy is validated against the analytical solution of the convective heat transfer between a heat-conducting solid flat plate and a laminar incompressible boundary layer. The applicability of the method for moving objects is then demonstrated by solving a conjugate heat transfer problem for a heated and moving cylinder in a laminar flow.

1. Introduction

The motivation of this paper is the development of numerical methods for studying particle-laden turbulent flows. These are of high engineering interest in applications as diverse as the combustion process of pulverized coal (Bühre et al., 2005), or the debris deposition during electrical discharge machining (Li et al., 2012). These applications involve not only the fluid force prediction on the particles themselves, but also require the computation of the heat exchange between the particles and the fluid to correctly determine e.g. the ignition point of the coal powder, or the debris cooling during deposition. A resolved particle simulation (RPS) for such problems requires to predict both the fluid phase as well as the heat exchange between the turbulent flow and the freely moving particles with usually non-spherical shape. Since the particles can be smaller than the Kolmogorov length scale, this results in an over-resolved Direct Numerical Simulation (DNS) of the turbulent flow-field involving multiple moving boundaries. Such a resolved particle simulation with a particle number on the order of O(1000) poses several constraints on the solution method. First, an immersed boundary method which adaptive grid refinement is of great advantage to avoid a frequent regeneration of meshes which is inevitable for boundary fitted grids. Second, the coupling between the fluid and the heat conducting domain must be implemented efficiently on high-performance, i.e. massively parallel computer hardware, to enable a simulation of a flow with a large number of particles. Special care must be taken to minimize the amount of data to be communicated for the heat transfer between the solid and the fluid phase.

In this work, we present a method for RPS based on hierarchical Cartesian grids which is well suited for multiphysics applications with moving objects in general. Cartesian meshes have been shown to be well suited for adaptive mesh refinement (Hartmann et al., 2011), the application of multilevel methods (Hartmann et al., 2008), and implementation on massively parallel supercomputers (Burstedde et al., 2011). In addition, an accurate and fully conservative formulation of a cut cell boundary condition for moving surfaces is available (Schneiders et al., 2013).
This paper is organized as follows: first, we present a brief review of the state of the art in different approaches for coupling multi-physics problems, conjugate heat transfer, and hierarchical Cartesian methods as well as the treatment of moving boundaries. In section 3, the conjugate heat transfer problem is introduced. Subsequently, we show in section 4 how to use the topology of hierarchical grids to solve the coupled system of PDEs with different solvers when moving bodies are present. In section 5 the proposed method is validated and finally, the main achievements are summarized in section 6.

2. Review of the state of the art

Numerical methods for conjugate heat transfer problems in particle-laden turbulent flows need to address orthogonal issues regarding the coupling strategy and the treatment of moving boundaries. In this section, these aspects are considered independently and the trade-offs involved are discussed.

Coupling strategies are typically categorized into monolithic and partitioned approaches. The partitioned approach (Park et al., 1977) couples the solution of separately solved PDEs by means of an iterative procedure such as, e.g., a fixed-point iteration. This approach is conceptually simple and allows to combine different spatial and temporal discretizations for each PDE involved (Kittler and Wall, 2008). Since different PDEs can have different mathematical properties, this flexibility allows the best suited discretization to be selected for each PDE. Most partitioned schemes aim to reduce the number of iterations or to eliminate iterations completely. This is typically achieved by extrapolating the solution of each PDE in time and using this information during time integration (Farhat et al., 2006). There are, however, pathological cases for which eliminating or reducing the number of coupling iterations inevitably leads to numerical instabilities (Causin et al., 2005). For these types of problems, large number of iterations might be required if convergence can be achieved at all (Degroote et al., 2009). The monolithic approach (Le Tallec and Mouro, 2001) aims to avoid these instabilities by solving the complete system of PDEs simultaneously. This requires a specific spatial and temporal discretization for the PDEs being considered, resulting in a tightly coupled software which cannot be neither easily extended nor reused.

Co-simulation uses the partitioned approach to externally couple already existing applications for solving each PDE independently. Examples of generic co-simulation couplers include e.g. MpCCI (Joppich and Kirschner, 2006), OpenPALM (Jaure et al., 2013), and CHIMPS (Schlüter et al., 2005). Since the standalone programs usually have been validated previously, the development process focuses on the implementation and validation of new coupling conditions. Each solver, however, is typically parallelized separately, i.e., each domain is partitioned independently from the others. As a consequence, a complex initialization process is required to identify the connectivity between the different domains. Furthermore, the coupling conditions can easily result in many-to-many communication, degrading scalability, which becomes important when a large number of computing cores has to be used. Since current supercomputers generally require the different applications to run on different computing nodes, solving these issues has proven to be challenging at the application level (Jaure et al., 2013). In the case of RPS, the situation becomes even more complex due to the moving boundaries, which require a frequent re-computation of inter-grid connectivity, making co-simulation less attractive than the alternatives.

Another strategy is to implement a monolithic multi-physics program that solves all PDEs at once (Le Tallec and Mouro, 2001). The large upfront investment required as well as the lack of flexibility for adapting these programs to different problems in the future often do not compensate for the possible efficiency improvements or the improved stability properties. For certain strongly coupled problems, however, the monolithic approach is the only viable alternative since partitioned approaches might not converge at all (Causin et al., 2005, Degroote et al., 2009).

Finally, the third alternative formulating multi-physics methods is to solve each PDE using individual algorithms that share a common framework of data-structures facilitating the coupling with the partitioned approach as well as the communication between the different solvers. The aim is to maintain the flexibility of the co-simulation strategy without sacrificing scalability. Re-implementing possibly already-existing applications within a common framework as well as performing their validation incurs, however, large development costs.

The advantages of the partitioned approach have motivated a large number of practical applications in the field of conjugate heat transfer using both co-simulation as well as special purpose frameworks. Examples thereof include a LES of an aeronautical combustor (Jaure et al., 2013), unsteady conjugate natural convection in an enclosure (Kazemi-Kamyab et al., 2014), and the modeling of an inertial-confinement-fusion hohlraum target as well as the flow of coolant past an hexagonal array of heated fuel rods (Henshaw and Chand, 2009).

The direct numerical simulation of particle laden turbulent flow deals with multiple moving particles undergoing
arbitrary large displacements. Satisfying the geometric conservation law while using moving mesh approaches such as Arbitrary Lagrangian-Eulerian methods requires expensive remeshing and projection operations that, as the number of particles increases, dominate the cost of the simulation. Non-conforming methods avoid remeshing all together. They differ from each other in the way in which the effect of the boundary is considered. The original immersed boundary method (Peskin, 1972) captures the effect of the boundary on the flow with a forcing term in the momentum equation that is imposed using a discrete delta function. On the other hand, the immersed interface method accounts for the known jumps across the interface via correction terms in the numerical discretization (Li and Lai, 2001). These methods are, however, non-conservative. Furthermore, they are exclusively developed for incompressible flows. The cut-cell finite-volume method is a conservative sharp interface method for compressible flows (Hartmann et al., 2008, 2011) that has been successfully extended to moving boundary problems (Schneiders et al., 2013). Cells intersected by the boundary are dynamically reshaped to maintain a conforming mesh as the moving interfaces evolve.

Non-boundary conforming methods are typically implemented for structured grids as well as hierarchical grids with or without anisotropy. In particular, hierarchical Cartesian grids offer a simplicity close to that of structured grids but with the flexibility of unstructured grids while being highly suited for adaptive mesh refinement and multigrid (Hartmann et al., 2008) as well as for parallelization and dynamic load balancing using space-filling curves (Burstedde et al., 2011). These are also the main reasons, why this approach is extended for a multiphysics simulation, i.e. conjugate heat transfer in this paper.

3. Conjugate heat transfer between a heat-conducting moving rigid solid and an ideal gas

Let the dimensionless density, pressure, fluid particle velocity, and specific total energy per unit mass be denoted by \( \rho, \ p, \ u, \) and \( E, \) respectively. Stagnation variables are denoted with \((,)_{0}\), dimensioned variables with an overline \((,)\); \( I \) is the identity matrix. Fluid and solid variables are distinguished by \((,)_{s}\), and tensors with an underline \((,)_{f}\), respectively. The Navier-Stokes equations for compressible flow are

\[
\begin{pmatrix}
\rho \\
\rho u \\
\rho E
\end{pmatrix} + \nabla \cdot \begin{pmatrix}
\rho u \\
\rho u \otimes u + p I \\
\rho E + p u
\end{pmatrix} + \frac{1}{Re_0} \nabla \cdot \begin{pmatrix}
\tau \\
t u + q
\end{pmatrix} = \begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix},
\]

(1)

where the shear-stress tensor for a Newtonian fluid \( \tau \) and the heat flux \( q \) are given by

\[
\tau = -\mu \left( \nabla u + (\nabla u)^T \right) + \frac{2}{3} \mu (\nabla \cdot u) I \quad \text{and} \quad q = -\frac{\lambda_f \nabla T}{Pr_0 (\gamma - 1)},
\]

(2)

respectively. Here \( \mu \) is the dynamic viscosity, \( \gamma = \frac{s_{p, f}}{s_{V, f}} \) is the ratio of specific heats, \( T \) is the temperature, and \( \lambda_f \) is the thermal conductivity. The Reynolds and Prandtl numbers are

\[
Re_0 = \frac{\overline{a_0} \overline{L}_{rel} \overline{\rho}_{0,f}}{\overline{\mu}_0} \quad \text{and} \quad Pr_0 = \frac{\overline{s}_{p, f} \overline{\rho}_{0,s}}{\lambda_{0,f}},
\]

(3)

respectively, where \( \overline{a}_0 \) is the stagnation speed-of-sound, and \( \overline{L}_{rel} \) is the characteristic length. For constant Prandtl number flows \( \lambda_f(T) = \mu(T) \). Sutherland’s law,

\[
\mu = \tau^{3/2} \frac{1 + S/T_0}{T + S/T_0},
\]

(4)

provides a valid approximation of the dynamic viscosity for gases at moderate temperatures. For an ideal gas the relation

\[
p = (\gamma - 1) \left( \rho E - \rho \frac{\|u\|^2}{2} \right)
\]

(5)

closes the system of equations. For air \( \gamma_{air}=1.4, \) \( Pr_{air,0}=0.72, \) and the Sutherland’s constant \( S_{air,0}=110.4K \) are chosen. Here, the reference temperature is \( T_0=273.15K. \) The unsteady heat equation for a solid of constant density and temperature-independent thermal conductivity is

\[
\frac{\partial}{\partial t} T_s - \nabla \cdot (\nabla T_s) = \frac{\overline{L}_{rel}}{\overline{\rho}_{0,s} \overline{T}_0} \frac{\partial T_s}{\overline{\rho}_{0,s} \overline{T}_0},
\]

(6)
where the thermal diffusivity \( \alpha = \lambda_{cs}/\rho s c_p \) and \( \tau \) is a source term. The acoustic time scale of the fluid and the diffusive time scale of the heat transfer within the solid are related by

\[
\frac{t_{\text{solid}}}{t_{\text{fluid}}} = \frac{\alpha_{\text{fluid,0}}}{\alpha_{\text{solid}}} \frac{T_{\text{ref, solid}}^2}{T_{\text{ref, fluid}}} \text{Re}_0 \text{Pr}_0
\]

where the right hand side has the form of Brun’s number (Luikov, 1974). At the fluid-solid interface \( \Gamma_{f/s} \), the following compatibility conditions ensure mass, momentum, and energy conservation,

\[
\begin{align*}
    u_f|_{\Gamma_{f/s}} &= u_s|_{\Gamma_{f/s}}, \\
    \frac{\partial u_f}{\partial t}|_{\Gamma_{f/s}} &= \frac{\partial u_s}{\partial t}|_{\Gamma_{f/s}}, \quad \text{and} \\
    -\lambda_f \nabla T_f|_{\Gamma_{f/s}} \cdot \mathbf{n} &= -\lambda_s \nabla T_s|_{\Gamma_{f/s}} \cdot \mathbf{n}.
\end{align*}
\]

4. Numerical Method

In this section, a partitioned approach using a special purpose multi-physics solver based on non-conforming methods on hierarchical Cartesian grids is introduced. First, the common framework of data structures is explained in 4.1. Then, the immersed boundary finite-volume method used is introduced in 4.2. Finally, the boundary and coupling conditions are given in 4.3.

4.1. A hierarchical Cartesian framework for multi-physics problems

First, the main idea behind the coupling strategy using hierarchical Cartesian grids is introduced. Then, the layout of the data structure is explained in detail. Finally, the cost of typical operations on the grid is discussed briefly.

![Fig. 1](a) The resulting solid (top, light grey) and fluid (bottom, dark grey) meshes are shown in (Fig. 1a). The octrees containing the solid and fluid grids are illustrated in (Fig. 1b) and (Fig. 1c), respectively, where the nodes belonging to the final grids are colored.

A balanced octree data-structure is used to generate and store both the fluid and the solid meshes, see Fig. 1. This results in a compressed representation of the connectivity information between cells of the same grid as well as cells of the fluid and solid grid. Furthermore, this enables an efficient retrieval of connectivity information between grid interfaces, since the inter-grid connectivity is maintained implicitly by the invariants of the octree. That is, even if the inter-grid connectivity changes frequently it does not need to be recomputed. This property is especially attractive for simulations involving multiple moving boundaries or unsteady solution adaptive mesh refinement. A discussion about the interaction of the proposed data structure with multi-grid methods and its advantages is delayed to future work.

The one-dimensional data layout follows a breadth-first Morton z-curve, see Fig. 2, right. This space-filling curve is simple and far from optimal, but ensures that cells lying close to each other in space remain close to each other in memory, allowing massive parallelization of the data structure and dynamic load balancing (Burstedde et al., 2011).
The main idea of this work is to generate all hierarchical Cartesian grids from a common octree node, storing all grids within the same balanced octree, Fig. 3. Each grid node stores links to cell data located within the internal data-structures of the different solvers. The octree is then used to efficiently query spatial and topological relationships between the different grid cells. In particular, grid interfaces and overlapping regions are identified by testing neighbor and overlapping nodes for links to the cell data of any given solver. An example is given in Fig. 3, where the neighbor □ of the fluid cell * is found in the solid domain with just three edge traversals. Furthermore, finding that the fluid * and solid ■ boundary cells overlap requires only one edge traversal.

Fig. 3: A cross section of the fluid and the solid mesh is displayed along with the octree data structure. The fluid (dark grey) and the solid (light grey) are shown. The arrows illustrate the traversal required to identify common boundary cells, e.g., from * to ■, and to find a neighboring cell in a different domain, e.g., from * to □.
particular, the node length is recursively computed with \( h_0(N) = h_0(E_P(N))/2 \) which terminates at the root node since \( h_0(N_0) \) is given. The cell coordinates are recursively computed in an analogous way. Finally, links between cell data and grid nodes are stored to form a bidirectional map. These links allow to traverse from a given tree node \( N \) to a solver cell \( C_G \) at the grid \( G \) and vice-versa via \( M_{G,N}(C_G) = N \) and its inverse \( M_{N,G}(N, G) = C_G \). Storing the cell data separately allows to tune the data-structure of the different solvers independently both from each other and the grid. This results in significant performance improvements since most of the computation takes places in the solvers for the PDEs. The memory requirements of the bidirectional map are \( N_G \) words per node where \( N_G \) is the number of grids and one word per grid cell.

The proposed data structure offers excellent complexity guarantees. Typical operations in such a tree include finding connectivity information between nodes and grids as well as finding overlapping grid regions or computing a nodes coordinates and length. If the maximum difference of level between neighboring nodes is one, the octree is said to be balanced. A balanced tree containing \( N_t \) nodes has order \( O(\log N_t) \) levels. Since these operations require \textit{at worst} a traversal from a given node to the root node and back down to a different node, they take, in general, \( O(\log N_t) \) time.

4.2. Finite-volume method

Two Cartesian cell-centered immersed boundary finite-volume methods are used to discretize the Navier-Stokes and the heat conduction equation. Here (7) is used to denote the cell-center averaged values of variables. Let the vector of cell averaged variables at the \( i \)-th cell be denoted by \( \mathbf{Q}_i \), the \( i \)-th cell volume by \( V_i \), and the numerical flux through the \( j \)-th surface of the \( i \)-th cell by \( F_{i,j} \). The semi-discrete system of ODEs resulting from the finite-volume method is

\[
\frac{d\mathbf{Q}}{dt} = -\frac{1}{V_i} \sum_{j=1}^{N_c} \left( \mathbf{F}_{i,j} \cdot \mathbf{n}_j \right) \equiv \mathbf{L}(\mathbf{Q}) .
\]

The numerical flux used for the viscous terms is a second-order accurate central difference. The inviscid flux of the Navier-Stokes equations is discretized with the AUSM scheme (Liou and Steffen, 1993). The system is integrated in time using an explicit two-step low-storage second-order accurate Runge-Kutta method:

\[
\begin{align*}
\mathbf{Q}^{n+1/2} &= \mathbf{Q}^n - \Delta t_f \mathbf{L}(\mathbf{Q}^n) , \\
\mathbf{Q}^{n+1} &= \frac{1}{2} \mathbf{Q}^n + \frac{1}{2} \left( \mathbf{Q}^{n+1/2} - \Delta t_f \mathbf{L}(\mathbf{Q}^{n+1/2}) \right) ,
\end{align*}
\]

where the time-step \( \Delta t \) is computed according to the CFL conditions

\[
\Delta t_f = \min_{c \in \mathbb{C}} \left( \min_{d \in \mathbb{D}, a_d} \left( \frac{C_h}{|u_{d,a_d}| + c\ell_c} \right) \right), \quad \Delta t_s = \min_{c \in \mathbb{C}} \left( \frac{C_h^2}{2|\mathbf{C}_t|} \right), \quad C_f = 0.8, \quad \text{and} \quad C_s = 0.5 .
\]

Here \( a_c = \sqrt{\frac{E_c}{\rho_c}} \) is the speed of sound and \( C \) is the Courant number. The simulation is advanced using the same physical time-step in both domains: \( \Delta t = \min(\Delta t_f, \Delta t_s) \). Each time-step is, however, in a different time scale due to the different non-dimensionalization. The different time scales are related through (7).

Moving boundaries are described using the level-set method. The level-set function \( \phi(x, t) \) is defined as the smallest signed Euclidean distance from the point \( x \) to the fluid-structure interface \( \Gamma \); positive in the fluid domain, negative in the solid domain, and zero on the interface:

\[
\phi(x, t) = \min_{\kappa \in \Omega} \|x - x_\kappa\|_2 \begin{cases} > 0 & \text{for } x \in \Omega_t , \\ = 0 & \text{for } x \in \Gamma , \\ < 0 & \text{for } x \in \Omega_s . \end{cases}
\]

One level-set field per boundary condition is used to describe the boundaries. This allows us to describe the boundaries of multiple moving particles by means of multiple level-set fields. For a given discrete field of level-set values at the cell-centers, boundary cells are identified by a change in the signed-distance between adjacent cells. To reconstruct the boundary surfaces within the boundary cells from the level-set field, first the level-set values are interpolated into the cell corner points. Then, linear interpolation along the edges provides the intersection points of the boundary surface with the cell edges. These points are used to represent surfaces as hyperplanes whose normal points into the solvers domain.

An immersed boundary ghost-cell method (Peskin, 1972) is used to impose boundary conditions, see Fig. 4. The line spanned by the surface normal and the ghost-cell center \( x_{gc} \) is used to find a mirror point lying at a constant distance \( d_m = \)}
Fig. 4: Ghost cell treatment at moving boundaries. The computation of the mirror point $x_{mp}$ for the ghost-cell at $x_{gc}$ is displayed. The perpendicular distances between the ghost-cell and the closest surface $d_{gc,ip}$ and between the mirror point and the surface $d_{ip,mp}$ as well as the cells in the interpolation stencil for the mirror point are shown.

$h_i/\sqrt{2}$ from the boundary. The conservative variables at the mirror point $x_{mp}$ are then interpolated from the neighboring cells $N_{inter}$ (see Fig. 4) using multi-quadric radial-basis functions $R$ with width $\epsilon = (\sqrt{2} h_i)^{-1}$:

$$Q(x_{mp}) = \sum_{n \in N_{inter}} w_n R(\|x_{mp} - x_n\|_2), \quad \text{with} \quad R(d) = \frac{1}{\sqrt{1 + (d\epsilon)^2}}.$$  

The interpolation coefficients $w_n$ are obtained from

$$Aw = v, \quad \text{where} \quad A_{ij} = R(\|x_i - x_j\|_2) \quad \text{and} \quad v_i = Q^T(x_i) \quad \forall i \in N_{inter}, j \in N_{inter},$$

using a singular value decomposition. Here $v$ is the vector of values at the interpolation neighbors cell centers. The singular value decomposition is robust and allows the efficient solution of the system for multiple right hand sides, i.e., for each cell variable. The Dirichlet and Neumann boundary conditions for the generic quantity $\phi$ are imposed at the ghost-cell as

$$\phi_{gc} = (d_{gc,ip} + 1)(\phi_T - \phi_{mp}) + \phi_{mp} \quad \text{and} \quad \phi_{gc} = \phi_{mp} - d_{gc,mp} \nabla \phi_T \cdot n,$$

respectively, where $d_{gc,mp} = d_{gc,ip} + d_{ip,mp}$.

4.3. Boundary and coupling conditions

The free stream inflow and outflow conditions are

$$\rho_T = \rho_\infty, \quad u_T = u_\infty, \quad \text{and} \quad (\nabla p \cdot n)_T = 0,$$

and

$$\rho_T = \rho_\infty, \quad (\nabla u_i \cdot n)_T = 0 \quad \forall i \in [0..N_d], \quad \text{and} \quad (\nabla p \cdot n)_T = 0,$$

respectively, where the free stream values of the primitive variables are computed from the Mach number as follows:

$$T_\infty = (1 + (\gamma - 1)M_\infty^2/2)^{-1}, \quad \rho_\infty = (T_\infty)^{\gamma/\gamma}, \quad u_\infty = M_\infty \sqrt{T_\infty}, \quad \text{and} \quad \rho_\infty = (T_\infty)^{\gamma/\gamma}.$$

The symmetry condition imposes a zero-gradient Neumann boundary condition for all primitive variables except for the velocity component normal to the symmetry plane, which is set to zero at the surface.

A partitioned loosely-coupled scheme is used to impose the coupling condition

$$T_{cl} = T_{sl}.$$
for the heat transfer at the fluid-structure interface. A Dirichlet boundary condition for the temperature is used to impose the coupling condition on the solid domain. On the fluid domain, an isothermal moving wall is used:

\[
\rho_r = \frac{\gamma}{T_r}, \quad \mathbf{u}_r = \mathbf{u}_r, \quad \text{and} \quad \nabla p \cdot \mathbf{n}_r = -\rho_r (\mathbf{a}_r \cdot \mathbf{n}),
\]

where \(\mathbf{a}_r\) is the acceleration at the moving interfaces.

5. Applications

5.1. Convective conjugate heat transfer between a solid wall and a fluid

The coupling procedure is validated against the analytical solution for the convective conjugate heat transfer between a flat plate and an incompressible laminar flow as studied by Luikov (Luikov, 1974). The problem is depicted in Fig. 5. The boundary conditions on the fluid domain are free-stream inflow \(\Gamma_{\text{Inflow}}\) (16), free-stream outflow \(\Gamma_{\text{Outflow}}\) (17), symmetry condition upstream of the plate, and the coupling condition with the solid \(\Gamma_{t/s}\) (19). On the solid domain, a Dirichlet boundary condition for the temperature \(\Gamma_{\text{Isothermal}}\) is imposed on the bottom surface with \(T = T_b\). The plate is coupled with the fluid through its top surface \(\Gamma_{f/s}\) (19), and zero-gradient Neumann boundary conditions are applied on the sides.

The plate’s length and thickness are \(L_{\text{Plate}} = 1.0\) and \(h_{\text{Plate}} = 0.1\) respectively. The analytical solution is expressed as \(\Theta = T - T_b\) where \(T_b\) is the constant temperature at the boundary. The coordinate origin for the analytical solution \(x = (x, y)\) is at the tip of the plate (see Fig. 5); the solution itself is given in the domain \(x \in [0, L_{\text{Plate}}], y \in [-b, \delta_T]\).

The thermal boundary layer thickness \(\delta_T\) is computed from the free-stream Reynolds number as follows:

\[
\delta_T = \delta \sqrt{\frac{13}{14P_0}}, \quad \delta = \frac{4.64x}{\sqrt{Re_x}}, \quad \text{and} \quad Re_x = Re_{\infty},
\]

The temperature distribution perpendicular to the plate \(\Theta\) is

\[
\Theta(x, y) = \begin{cases} 
\frac{\delta_{t/s}(x) + \frac{3}{2} \frac{\theta_{0,0}(x)}{\delta_{t/s}(x)} y - \frac{1}{2} \frac{\theta_{0,-1}(x)}{\delta_{t/s}(x)} y^3}{z + 1}, & y \in [0, \delta_T], \\
\frac{\delta_{t/s}(x) + \frac{1}{b} \theta_{b,0}(x)}{b}, & y \in [-b, 0],
\end{cases}
\]

where \(\delta_{t/s}\) is the temperature at the fluid-structure interface:

\[
\delta_{t/s} = \frac{\theta_0 z}{z + 1}, \quad T_0 = T_\infty \left(1 + \frac{1}{2} (y - 1) \sqrt{Pr_0 M_\infty^2}\right), \quad z = \frac{3}{2} \frac{\lambda_t}{h_{\text{plate}}} \frac{1}{\delta_T(x)}.
\]

For a boundary temperature \(T_b = 0.9\) the the iso-lines of the temperature are also shown in Fig. 6 (left). The comparison between the analytical and numerical solution shows good agreement. As expected, a small jump occurs in the gradient of the temperature profile at the interface.
Fig. 5: (Right) Temperature distribution $T$ on the fluid and the solid. The solid black lines are temperature isolines at $\{0.9, 0.91, ..., 1.0\}$. The dashed black line is the fluid-solid interface $\Gamma_{f/s}$.
(Left) Temperature distribution in the solid (red) and fluid (blue) domains perpendicular to the plate superposed with the analytical solution.

Fig. 7: Conjugate heat transfer between a moving solid cylinder and a laminar flow. The fluid (white) and solid (dark grey) domains are shown as well as the boundary conditions for the fluid and the solid.
5.2. Conjugate heat transfer between a moving cylinder and a laminar flow

The applicability of the proposed approach is demonstrated by computing the conjugate heat transfer between a solid moving cylinder with diameter \( D = \frac{D}{L_{rel}} = 1 \) and a fluid, see Fig. 7. The flow is assumed to be almost incompressible \( M = 0.1 \) and laminar \( Re_{\infty} = 150 \). The properties of air at \( T_0 = 273.15 K \) are chosen and the thermal diffusivity ratio \( \alpha_{f,0}/\alpha_s = 10^{-4} \). An oscillating movement with the amplitude \( A_y = 2D \) perpendicular to the flow direction and angular frequency \( \omega_h = 2/5\pi \) is prescribed for the cylinder where \( \omega_h = 2\pi f L_{rel}/u_\infty \). A temperature \( T_D = 1.25T_\infty \) is imposed at the center of the solid cylinder over a circular surface of radius 0.1\( D \). Within this simulation the cells in which the coupling surface of the PDEs for the fluid and solid is located, are continuously changing with the movement of the solid cylinder. This update of the cells with coupling interface is straightforward to formulate, due to the connectivity information inherently stored in the joint hierarchical Cartesian mesh.

The results are shown in Fig. 8, where instantaneous snapshots of the temperature field are plotted. Since the cylinder generates a von Karman vortex street, warm fluid heated by the cylinder is entrained into the vortices, which are then shed and traveling downstream. Within the solid cylinder the temperature field shows the cooling effect of the fluid, which is larger at the front stagnation point compared to the separated flow in the rear of the cylinder.

![Instantaneous temperature field](image)

**Fig. 8:** Instantaneous temperature field \( T/T_0 \) over the fluid and solid phases at two different hydrodynamic times \( t_h = t/(L_{rel}/u_\infty) \): \( t_h = 655.2 \) (left) and \( t_h = 868.4 \) (right). The solid black circle indicates the surface of the solid moving cylinder.

5.3. Conjugate heat transfer between multiple moving cylinders and a laminar flow

Since moving boundaries are described implicitly by the level-set method, multiple moving boundaries can be considered by including them in the level-set field. In Fig. 9 the set-up shown in Fig. 7 is extended with another three moving particles, only this time the particles are all moving in phase. The same physical parameters and boundary conditions as in the single-particle simulation are used.

6. Summary and outlook

A finite-volume immersed boundary method for resolved particle simulation including conjugate heat transfer on hierarchical Cartesian grids has been presented and validated against the analytical solution of convective heat transfer between a solid heat conducting plate and a laminar incompressible flow. The approach using hierarchical Cartesian grids has been described in detail and, in particular, the extension of hierarchical Cartesian grids to handle multi-physics problems has been discussed. Finally, the applicability of the method was demonstrated by computing the conjugate heat transfer between a solid moving cylinder and a laminar flow field. Future work will deal with the implementation of the presented approach on massively parallel supercomputers and the simulation of conjugate heat transfer in resolved...
Fig. 9: Instantaneous temperature field $T/T_0$ over the fluid and solid phases at two different hydrodynamic times $t_h = t/(L_{ref}/u_\infty)$: $t_h = 124.1$ (left) and $t_h = 143.5$ (right). The solid black circle indicates the surface of the solid moving cylinder.

Acknowledgments

This work has been financed by the research cluster Fuel production with renewable raw materials (BrenaRo) at RWTH Aachen University as well as by the German Research Foundation (DFG) within the framework of the SFB/Transregio 129 and the SFB/Transregio 136. The support is gratefully acknowledged. Computing time at the Jülich Supercomputing Center (JSC) on the Juqueen and at the High Performance Computing Center Stuttgart (HLRS) on the Hermit is gratefully acknowledged.

References

Buhre, B.J.P., Elliott, L.K., Sheng, C.D., Gupta, R.P., and Wall, T.F., Oxy-fuel combustion technology for coal-fired power generation, Progress in Energy and Combustion Science, Vol. 31, No. 4 (2005), pp. 283 – 307, <http://dx.doi.org/10.1016/j.pecs.2005.07.001>.

Burstedde, C., Wilcox, L.C., and Ghattas, O., p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees, SIAM Journal on Scientific Computing, Vol. 33, No. 3 (2011), pp. 1103 – 1133, <http://dx.doi.org/10.1137/100791634>.

Causin, P., Gerbeau, J.F., and Nobile, F., Added-mass effect in the design of partitioned algorithms for fluid-structure problems, Computer Methods in Applied Mechanics and Engineering, Vol. 194, No. 42-44 (2005), pp. 4506 – 4527, <http://dx.doi.org/10.1016/j.cma.2004.12.005>.

Degroote, J., Bathe, K.J, and Vierendeels, J., Performance of a new partitioned procedure versus a monolithic procedure in fluidstructure interaction, Computers & Structures, Vol. 87, No. 11-12 (2009), pp. 793 – 801, <http://dx.doi.org/10.1016/j.compstruc.2008.11.013>.

Farhat, C., van der Zee, K.G., and Geuzaine, P., Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity, Computer Methods in Applied Mechanics and Engineering, Vol. 195, No. 17-18 (2006), pp. 1973 – 2001, <http://dx.doi.org/10.1016/j.cma.2004.11.031>.

Hartmann, D., Meinke, M., and Schröder, W., An adaptive multilevel multigrid formulation for Cartesian hierarchical grid methods, Computers & Fluids, Vol. 37 (2008), pp. 1103 – 1125, <http://dx.doi.org/10.1016/j.compfluid.2007.06.007>.

[DOI: 10.1299/jfst.2015jfst0002] © 2015 The Japan Society of Mechanical Engineers
Hartmann, D., Meinke, M., and Schröder, W., A strictly conservative Cartesian cut-cell method for compressible viscous flows on adaptive grids, Computer Methods in Applied Mechanics and Engineering, Vol. 200 (2011), pp. 1038 – 1052, <http://dx.doi.org/10.1016/j.cma.2010.05.015>.

Henshaw, W.D., and Chand, K.K., A composite grid solver for conjugate heat transfer in fluid-structure systems, Journal of Computational Physics, Vol. 228, No. 10 (2009), pp. 3708–3741, <http://dx.doi.org/10.1016/j.jcp.2009.02.007>.

Jaure, S., Duchaine, F., Staffelbach, G., and Gicquel, L.Y.M., Massively parallel conjugate heat transfer methods relying on large eddy simulation applied to an aeronautical combustor, Computational Science & Discovery, Vol. 6, No. 1 (2013), <http://dx.doi.org/10.1088/1749-4699/6/1/015008>.

Joppich, W., and Kürschner, M., MpCCI - a tool for the simulation of coupled applications, Concurrency and Computation: Practice and Experience, Vol. 18, No. 2 (2006), pp. 183–192, <http://dx.doi.org/10.1002/cpe.913>.

Kazemi-Kamyab, V., van Zuijlen, A. H., and Bijl, H., Accuracy and stability analysis of a second-order time-accurate loosely coupled partitioned algorithm for transient conjugate heat transfer problems, International Journal for Numerical Methods in Fluids, Vol. 74, No. 2 (2014), pp. 113–133, <http://dx.doi.org/10.1002/fld.3842>.

Küttler, U., and Wall, W.A., Fixed-point fluid structure interaction solvers with dynamic relaxation, Computational Mechanics, Vol. 43 (2008), pp. 61–72, <http://dx.doi.org/10.1007/s00466-008-0255-5>.

Le Tallec, P., and Mouro, J., Fluid structure interaction with large structural displacements, Computer Methods in Applied Mechanics and Engineering, Vol. 190 (2001), pp. 3039–3067, <http://dx.doi.org/10.1016/S0045-7825(00)00381-9>.

Li, L., Gu, L., Xi, X., and Zhao, W., Influence of flushing on performance of EDM with bunched electrode, The International Journal of Advanced Manufacturing Technology, Vol. 58, No. 1-4 (2012), pp. 187–194, <http://dx.doi.org/10.1007/s00170-011-3357-8>.

Li, Z., and Lai, M.C., The immersed interface method for the Navier-Stokes equations with singular forces, Journal of Computational Physics, Vol. 171, No. 2 (2001), pp. 822 – 842, <http://dx.doi.org/10.1006/jcph.2001.6813>.

Liou, M.S., and Steffen, C.J., A New Flux Splitting Scheme, Journal of Computational Physics, Vol. 107 (1993), pp. 23–39, <http://dx.doi.org/10.1016/0021-9991(93)90112-2>.

Luikov, A.V., Conjugate convective heat transfer problems, International Journal of Heat and Mass Transfer, Vol. 17, No. 2 (1974), pp. 257 – 265, <http://dx.doi.org/10.1016/0017-9310(74)90087-8>.

Park, K.C., Felippa, C.A., and DeRuntz, J.A., Stabilization of staggered solution procedures for fluid-structure interaction analysis, volume 26 (1977), ASME Applied Mechanics Symposia Series, Editors: T. Belytschko and T. L. Geers.

Peskin, C.S., Flow patterns around heart valves: a numerical method, Journal of Computational Physics, Vol. 10 (1972), pp. 252–271, <http://dx.doi.org/10.1016/0021-9991(72)90065-4>.

Schlüter, J. U., Wu, X., v. d. Weidez, E., Hahnx, S., Alonso, J.J., and Pitsch, H., Multi-code simulations: A generalized coupling approach, In 17th AIAA Computational Fluid Dynamics Conference (2005), p.4997, Toronto, Ontario Canada.

Schneiders, L., Hartmann, D., Meinke, M., and Schröder, W., An accurate moving boundary formulation in cut-cell methods, Journal of Computational Physics, Vol. 235 (2013), pp. 786 – 809, <http://dx.doi.org/10.1016/j.jcp.2012.09.038>.