Tenth International Conference on Computational Fluid Dynamics (ICCFD10), Barcelona,Spain, July 9-13, 2018

ICCFD10-2018-381

Dynamic and implicit coupling of non-matching meshes

G, Houzeaux*, R. Borrell*, J.C. Cajas* and M. Vázquez*
Corresponding author: guillaume.houzeaux@bsc.es

* Barcelona Supercomputing Centre (BSC), Barcelona, Spain.

Abstract: Many fluid applications involve bodies in relative motion for which, if a remeshing is not possible or not computationally efficient, a coupling algorithm must be considered. One common application is the coupling of rotating and fixed fluid domains, represented by the interaction of the wheels with a car, rotor/stator interactions in turbines, wind turbines or chemical reactors among others. The different numerical approaches to attack these problems are referred to as sliding mesh methods in the literature. In previous papers, we have developed such a strategy to couple the solution of PDEs on non-conforming meshes. The proposed method is implemented at the algebraic level, as an extension of the classical sparse matrix-vector product (SpMV). This is done through the introduction of transmission matrices that ensure transmissions of Dirichlet and Neumann data in a conservative way. The resulting coupling is implicit and can be interpreted as a generalization of classical parallelization techniques for non-matching meshes. We will present the implementation aspects of the methodology, with particular emphasis on parallelization issues.

Keywords: Sliding mesh, Implicit coupling, Non-matching meshes, Navier-Stokes equations.

1 Introduction

Many fluid applications involve bodies in relative motion, like the interaction of the wheels with a car, rotor/stator interactions in turbines, wind turbines or chemical reactors among others. If a remeshing is not possible, that is if you are not lucky enough to have an efficient mesh adaptation tool and a parallel redistribution algorithm in your code, a coupling method is thus necessary to obtain a global solution. The coupling method aims at imposing continuity of the unknowns and their fluxes across the interface of the fixed and moving subdomains.

Such a coupling method can be explicit or implicit. On the one hand, an explicit method involves solving the governing equations in each subdomain independently, and reconnecting both solutions with a coupling method in a later operation. The main drawback of explicit methods is that the coupling delays the convergence of the global system, with the risk of preventing it. This convergence depends also on the kind of transmission conditions imposed (Dirichlet, Neumann, Robin). On the other hand, implicit methods consist in solving a monolithic system for both subdomains, thus avoiding any convergence issues... at the cost of complicating the implementation.

Different methods based on mesh operations have been presented in the literature to achieve an implicit coupling, basically aiming at re-conforming the mesh in the interface vicinity. Let us mention the Shear-Slip Mesh Update Method [1], the DRAGON method [2] (Direct Replacement of Arbitrary Grid Overlapping by Non-structured grid), the Mesh Matching method presented in [3], the mesh merging technique [4, 5], the conforming joining method presented in *Code_Saturne* [6], and the overlapping HERMESH method which connects non-conforming meshes [7, 8]. Such methods are complex to implement in parallel, as new elements may be created, data movement between CPUs must be implemented efficiently, and possibly mesh redistribution to control load balance is required.

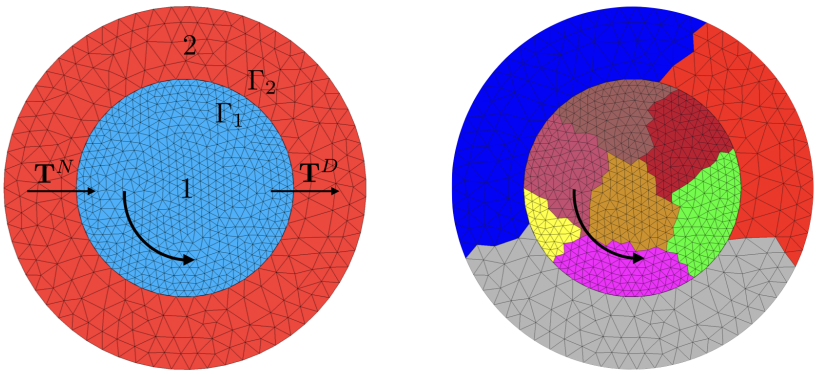

Figure 1: (Left) Coupling of rotating and fixed part. (Right) Coupling in parallel.

In previous papers, the authors have developed an implicit coupling strategy to couple the solutions of PDEs on non-conforming meshes. The proposed method is implemented at the algebraic level, as an extension of the classical sparse matrix-vector product (SpMV). This is done through the introduction of transmission matrices that ensure transmissions of Dirichlet and Neumann data in a conservative way, namely $\mathbf{T}^D$ and $\mathbf{T}^N$, respectively. The resulting coupling is implicit and can be interpreted as a generalization of classical parallelization techniques for non-matching meshes [9, 10].

The proposed coupling can be divided in two phases. Firstly, the setup or preprocess phase to compute proper interpolation or projection from one side of the interface to the other, for both the Dirichlet and Neumann data. The main outputs of this phase consists of transmission matrices for each node involved in the coupling as well as a communication strategy to further exchange the solutions. Secondly, the coupling itself to be implemented in SpMV subroutines, similar to classical parallelization techniques which exchange the result between neighbors.

In the general case, special attention is required to the second phase as the coupling is carried out at each SpMV, which occur tens of thousands of times in fluid simulations. In the particular case of moving subdomains, the setup phase is critical as it should be carried out at each time step of the simulation, in order to actualize the transmission matrices to account for the relative mesh motion. These phases are carried out in parallel, so a careful implementation is required. We will describe in this work the optimization of the first an second phases, in the context of distributed memory systems.

2 Coupling strategy

The implicit coupling method considered in this work was previously introduced in [9, 10]. Let us consider two meshes, 1 and 2, with a disjoint and non-matching interface, as shown in Figure 1.

We then distinguish the unknowns into interior and interface unknowns in subdomains 1 and 2, namely $\mathbf{u}_1$, $\mathbf{u}_{\Gamma_1}$ and $\mathbf{u}_2$, $\mathbf{u}_{\Gamma_2}$, respectively. The independent discretization of a PDE on these subdomains leads to two uncoupled algebraic systems of the form:

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{1\Gamma_1} \\ \mathbf{A}_{\Gamma_1 1} & \mathbf{A}_{\Gamma_1\Gamma_1} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_{\Gamma_1} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_{\Gamma_1} \end{pmatrix}, \qquad \begin{pmatrix} \mathbf{A}_{22} & \mathbf{A}_{2\Gamma_2} \\ \mathbf{A}_{\Gamma_2 2} & \mathbf{A}_{\Gamma_2\Gamma_2} \end{pmatrix} \begin{pmatrix} \mathbf{u}_2 \\ \mathbf{u}_{\Gamma_2} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_2 \\ \mathbf{b}_{\Gamma_2} \end{pmatrix}.$$

A possible coupling of these systems consists in imposing the continuities of Dirichlet and Neumann data on the interface. The Dirichlet data is the unknown of the system on the interface of subdomain i, $\mathbf{u}_{\Gamma i}$, while

the Neumann data consists of the residual of the equation on the interface $\mathbf{r}_{\Gamma i}$, such that:

$$\mathbf{r}_{\Gamma_i} = \mathbf{b}_{\Gamma_i} - \mathbf{A}_{\Gamma_i i}\mathbf{u}_i - \mathbf{A}_{\Gamma_i\Gamma_i}\mathbf{u}_{\Gamma_i}.$$

Now let us introduce transmission matrices $\mathbf{T}^D$ and $\mathbf{T}^N$ to pass Dirichlet and Neumann data, respectively, from one subdomain to another. Arbitrarily, we impose:

$$\begin{aligned} \text{Dirichlet:} \quad & \mathbf{u}_{\Gamma_1} = \mathbf{T}^D\mathbf{u}_{\Gamma_2}, \\ \text{Neumann:} \quad & \mathbf{r}_{\Gamma_2} = \mathbf{r}_{\Gamma_2} + \mathbf{T}^N\mathbf{r}_{\Gamma_1}. \end{aligned}$$

By applying these transmission conditions, we end up with the following coupled system:

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{1\Gamma_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{T}^D \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{22} & \mathbf{A}_{2\Gamma_2} \\ \mathbf{T}^N\mathbf{A}_{\Gamma_1 1} & \mathbf{T}^N\mathbf{A}_{\Gamma_1\Gamma_1} & \mathbf{A}_{\Gamma_2 2} & \mathbf{A}_{\Gamma_2\Gamma_2} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_{\Gamma_1} \\ \mathbf{u}_2 \\ \mathbf{u}_{\Gamma_2} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{0} \\ \mathbf{b}_2 \\ \mathbf{b}_{\Gamma_2} + \mathbf{T}^N\mathbf{b}_{\Gamma_1} \end{pmatrix} \tag{1}$$

We note that if the meshes coincide on the interface, the transmission matrices are identity, and we can easily check that the latter system is equivalent to the monolithic system on would obtain by collapsing the interface unknowns. In addition, if the transmission matrices are chosen such that

$$\mathbf{T}^N = (\mathbf{T}^D)^t,$$

we can easily check that the coupling method preserves symmetry of the problem.

The construction of system (1) can be very complex in parallel. Another way this coupling can be achieved implicitly, is to extend the parallel sparse-matrix vector product (SpMV) to account for non-matching meshes, by introducing the transmission matrices, as explained in [10]. The resulting algorithm is implemented just like the classical parallel SpmV, the only difference being the use of transmission matrices whenever the nodes do not coincide.

3 Dynamic treatment

The coupling method introduced previously requires the construction of the transmission matrices. These matrices can be build in different ways, using interpolation or projection techniques. The choice depends on the relative size of the meshes of the subdomains and the conservation properties sought. In any case, the matrices relate the values on one side of the interface to those on the other side.

In a parallel code for distributed memory systems, the calculation of the matrices should be carried out in parallel. In this case, the interface nodes are in general unlikely to be located in the same MPI partition. Thus, a specific implementation is required with respect to the sequential case, and a communication strategy is required. For the sake of simplicity, let us assume that we are building the transmission matrix $\mathbf{T}^D$ to pass the Dirichlet data from subdomain 2 to subdomain 1. We define the interface nodes of subdomain 1 as the target nodes (also referred to in the literature as wet nodes). Fig. 2 illustrates the procedure. The different colors represent the different MPI partitions, obtained with METIS. The partitioning in METIS is based on the element connectivity graph. As the meshes of subdomain 1 and 2 are not connected, we can observe that no partition overlaps subdomain 1 and 2 at the same time. This is generally the case when partitioning non-connex meshes with METIS.

On the left part of the figure, the target nodes of the yellow subdomain are depicted. A straightforward implementation would consist in broadcasting the target nodes coordinates to all the partitions. Then, each partition would compute a possible interpolation, for example based on boundary interpolation of the projection of the nodes onto the boundaries. Then, the result would be sent back to the yellow subdomain which, at its turn, would have to choose between the different proposals of the partitions. Finally, the yellow

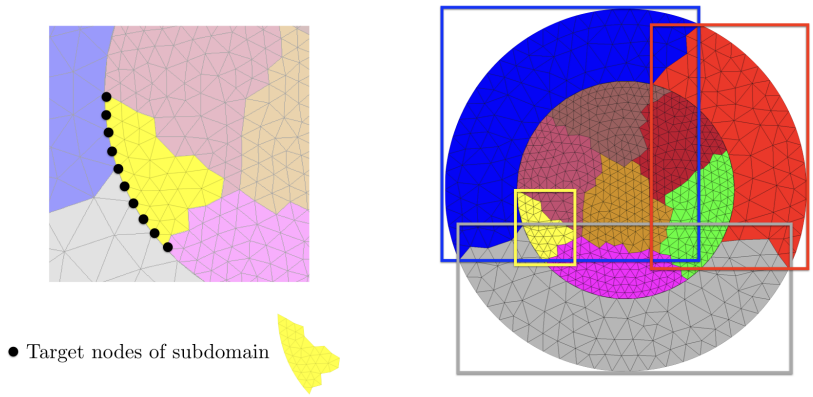

Figure 2: Construction of the transmission matrices in parallel. (Left) Target nodes. (Right) Bounding boxes to minimize communications.

subdomain would return its decision to the other partitions, and the communication could then be set up. To reduce the possible communications, bounding boxes are created, thus limiting the number of candidate partitions for the target nodes. The resulting Algorithm is shown in Alg. 1.

Algorithm 1 Implementation for each MPI partition.

1: I Compute my bounding box.
2: All gather all the bounding boxes.
3: I broadcast the target node coordinates to the MPI partitions if they fall in their bounding boxes. These MPI partitions are referred to as *candidate MPI sources*.
4: All candidate MPI sources compute a candidate interpolation.
5: All candidate MPI sources send back their results, together with a distance criterion (e.g. distance of the target node to the boundary).
6: Based on this distance criterion, I decide which MPI partitions is the final source for each target node.
7: I send my decision to all candidate MPI sources.
8: I setup the communication between myself and the final MPI sources.

In the case of a moving subdomain, this algorithm should be repeated at each time step, to account for the new relative positions of the target nodes. In addition, all geometrical arrays should be updated as well, such like the mass matrix or the exterior normals.

4 Result

We consider an idealized chemical reactor on which we solve the Navier-Stokes equations. The solver is implicit and extensively described in [11], while details on the parallelization can be found in [12]. In brief, the Navier-Stokes equations are first assembled, using a finite element method. The solver is based on the solution of the pressure Schur complement. For this, the momentum equations are then solved (GMRES), then the pressure equation (Deflated CG), and finally the momentum equations are solved a second time. As the equation is non-linear, the scheme is repeated for each time step until a convergence criterion is satisfied.

The geometry as well as the partitioning into 39 subdomains are shown in Fig. 3. We observe that METIS has partitioned independently the fixed and rotating subdomain.

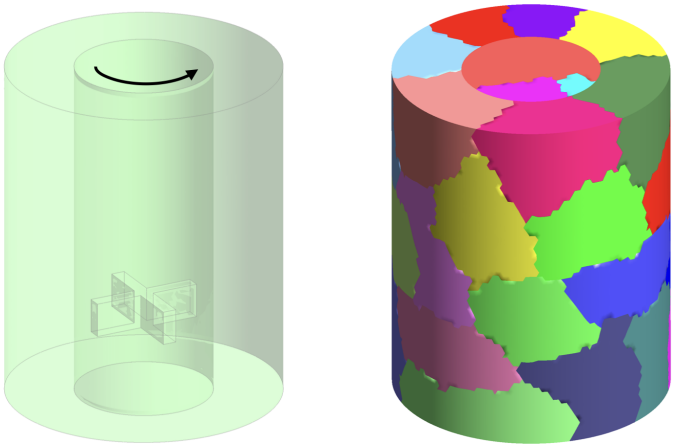

Figure 3: Idealized chemical reactor. (Left) Geometry and partitions. (Right) Partitions.

Fig. 4 shows a trace of the execution. The different colors show the different steps carried out during two time steps. We observe that the coupling update (red, blue and white) represents a very short time when compared to the solution of the Navier-Stokes equations. This shows that, at least for this application, the proposed strategy is efficient.

5 Conclusion and Future Work

We have briefly introduced an implicit coupling for moving subdomains. The method is based on connecting the nodes on both sides of the interface through the introduction of transmission matrices in the SpMV. The method was tested on an implicit Navier-Stokes solver, showing a relatively low cost. For semi-implicit solvers like the fractional step method, the relative cost of to the coupling update will increase. A possibility to lower this cost is to combine this strategy with an ALE method, until the mesh gets too distorted. This will be part of the future work carried out to optimize this implicit coupling method.

References

[1] M. Behr and T. Tezduyar. The shear-slip mesh update method. *Comp. Meth. Appl. Mech. Eng.*, 174:261–274, 1999.

[2] M.S. Liou and K.H. Kao. Progress in grid generation: From Chimera to DRAGON grids. Technical Report NASA Technical Memorandum 106709, ICOMP-94-19, NASA, 1994.

[3] Matthew L Staten, Jason F Shepherd, Franck Ledoux, and Kenji Shimada. Hexahedral mesh matching: Converting non-conforming hexahedral-to-hexahedral interfaces into conforming interfaces. *Int. J. Num. Meth. Eng.*, 82(12):1475–1509, 2010.

[4] S.H. Lo and W.X. Wang. A fast robust algorithm for the intersection of triangulated surfaces. *Engineering with Computers*, 20(1):11–21, 2004.

[5] J.R. Cebral, R. Löhner, P.L. Choyke, and P.J. Yim. Merging of intersecting triangulations for finite element modeling. *Journal of biomechanics*, 34(6):815–819, 2001.

[6] Code_Saturne.

[7] B. Eguzkitza, G. Houzeaux, R. Aubry, and M. Vázquez. A parallel coupling strategy for the Chimera and domain decomposition methods in computational mechanics. *Comput. Fluids*, 80:128–141, 2013.

[8] B. Eguzkitza. *HERMESH: A Geometrical Domain Composition Method in Computational Mechanics*. PhD thesis, Universitat Politècnica de Catalunya, Barcelona (Spain), 2014.

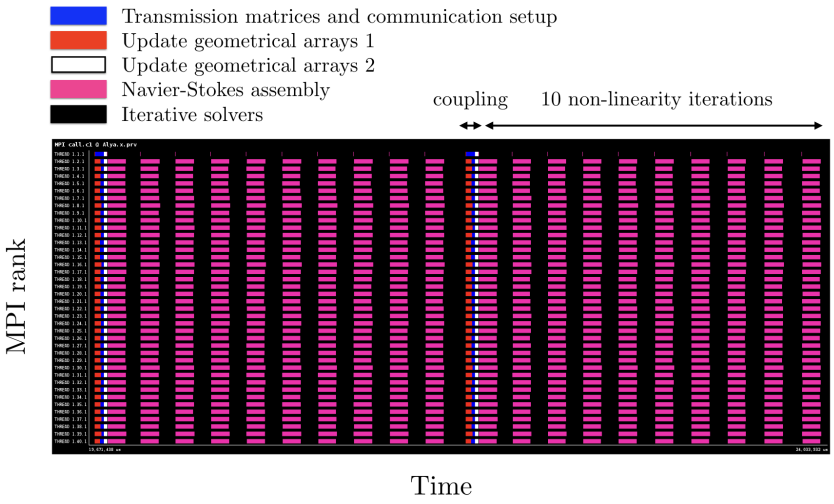

Figure 4: Trace showing the different steps of the solver.

[9] G. Houzeaux, J.C. Cajas, M. Discacciati, B. Eguzkitza, A. Gargallo-Peiró, M. Rivero, and M. Vázquez. Domain decomposition methods for domain composition purpose: Chimera, overset, gluing and sliding mesh methods. *Arch. Comp. Meth. Eng.*, 24(4):1033–1070, 2017.

[10] G. Houzeaux, R. Borrell, J.C Cajas, and M. Vázquez. Extension of the parallel sparse matrix vector product (spmv) for the implicit coupling of pdes on non-matching meshes. *Submitted to Computers & Fluids*, 2018.

[11] G. Houzeaux, R. Aubry, and M. Vázquez. Extension of fractional step techniques for incompressible flows: The preconditioned orthomin(1) for the pressure schur complement. *Comput. Fluids*, 44:297–313, 2011.

[12] M. Vázquez, G. Houzeaux, S. Koric, A. Artigues, J. Aguado-Sierra, R. Arís, D. Mira, H. Calmet, F. Cucchietti, H. Owen, A. Taha, E. Dering Burness, J. M. Cela, and M. Valero. Alya: Multiphysics engineering simulation towards exascale. *J. Comput. Sci.*, 14:15–27, 2016.