An Implicit and Parallel Chimera Type Domain Decomposition Method

B. Eguzkitza[1], G. Houzeaux[1], R. Aubry[2], and O. Peredo[1]

[1] Barcelona Supercomputing Center (BSC-CNS),
Dept. Computer Applications in Science and Engineering,
Edificio NEXUS I, Campus Nord UPC,
beatriz.eguzkitza@bsc.es
[2] CFD Center, Dept. of Computational and Data Science
M.S. 6A2, College of Science, George Mason University
Fairfax, VA 22030-4444, USA
US Naval Research Laboratory
4555 Overlook Ave SW
Washington DC 20375, USA

1 Introduction

The Chimera Method developed originally in [1, 19, 20] simplifies the construction of computational meshes about complex geometries. This is achieved by breaking the geometries into components and generating independently a series of different meshes. This enables one a great flexibility on the choice of the type of elements, their orientations and local mesh refinement. The components are further coupled by transmitting information from one mesh to the other to obtain a global solution.

The Chimera Method is a very efficient tool to treat moving objects [3, 16] as the different meshes can move as rigid bodies in an independent way. Nevertheless, we will focus in this work on fixed subdomains. The main application in this context is optimization analysis, where different configurations can be tested without having to remesh the whole geometry. In order to achieve this, we have developed a versatile strategy based on the Chimera Method.

Usually, in the Chimera Method, the mesh is divided into a background mesh, which covers all the computational domain, and patch (overset) meshes attached to the different components (objects) which are located upon the background mesh. First, we apply a proper preprocessing consisting in removing elements of the background mesh located inside the patch meshes to create apparent interfaces between the background and the patches. The present algorithm requires in addition to smooth the interfaces. This is achieved using a smoothing strategy of the interfaces and the neighboring volume mesh. Then a new coupling algorithm is carried out in order to obtain a "continuous solution" across the interfaces. In the literature, the Chimera coupling has generally been implemented as an iterative algorithm (see [2] for a

R. Bank et al. (eds.), *Domain Decomposition Methods in Science and Engineering XX*,
Lecture Notes in Computational Science and Engineering 91,
DOI 10.1007/978-3-642-35275-1__68, © Springer-Verlag Berlin Heidelberg 2013

Schwarz coupling or [9] for a Dirichlet/Neumann coupling). Here the coupling is implicit. The implementation properties of the proposed coupling facilitate its parallel implementation and makes it a versatile method to be used on general PDE's.

In the following we explain the two basic steps of the Chimera method. The preprocessing step which consists in creating the interfaces between the subdomains. This is a purely geometrical task. We then present the coupling step which couples the solution from the different meshes. Finally we show a numerical examples.

2 Interface Creation Process

The first task of the Chimera method is to create apparent interfaces between the background and the patch meshes. This is achieved by the hole cutting step of the Chimera method. As will be explained in next section, our coupling strategy requires smooth interfaces. After the hole cutting, smoothing of the interfaces are also necessary. We now explain these two points.

2.1 Hole Cutting

The hole cutting tasks consists in removing elements (the hole elements) from the background mesh to form interfaces with the patches. We start by identifying the hole nodes. The hole nodes are those nodes of the background mesh that are located inside the patch mesh. To do this we have used a *skd-tree* strategy, as explained in [12]. Skd-trees are used to find efficiently the signed shortest distance between a point and a surface. In our case, the surfaces are the patch outer boundaries. In practice we obtain a better efficiency if we use the search algorithm described in [18], which is a slightly modified version of the above reference. Having found the hole nodes, we identify the hole elements which are the background elements of which all nodes are hole nodes. The fringe nodes are defined as the nodes located on the outer boundaries of the hole elements. They are the hole nodes having non-hole neighbor nodes. The fringe nodes are used to form the interface of the backgound with the patches.

2.2 Smoothing

The domain decomposition coupling we propose is geometrical, as will be shown in next section. It is therefore important to ensure a minimum regularity of the interfaces and the mesh nearby, as this will affect the quality of the results. Figure 1 (Left) shows an example of typical background interface resulting from the previous hole cutting process. The proposed strategy consists in smoothing first the interface and then the volume mesh in the vicinity.

In this article, we are interested in mesh smoothing techniques that relocate the nodes to improve the mesh without changing its topology. The particular method we consider here is based on local mesh smoothing algorithms, since they have shown to be efficient in repairing distorted elements. The most common smoothing technique is Laplacian smoothing (see [13]), which moves a given node to the barycenter of all its connected nodes. This method is not computationally expensive but does not guarantee an improvement in mesh quality. In addition, it can create invalid elements or poor quality elements resulting in convergence and shrinkage problems. To overcome this shortcoming, different variations of Laplacian smoothing have been proposed like [5, 22].

Optimization-based smoothing algorithms are alternative local smoothing strategies. These algorithms depend on the type of mesh, the optimization method used and a measure of the mesh quality, and require an objective function to be optimized. The objective function should include a good representation of the mesh quality. A good summary of measures for the quality of tetrahedra and a global definition of the tetrahedron shape measure is given in [4]. Besides the geometrical objective functions described in the above reference, there exist other quality interpretations based on matrices and matrix norms. This matrix perspective suggests several different objective functions as, for example, the smoothness objective function in terms of the condition number of the Jacobian matrix; see [6].

Our smoothing process consists first of a surface Laplacian-smoothing algorithm based on [21] for the interface. An example is shown in Fig. 1. As a consequence,

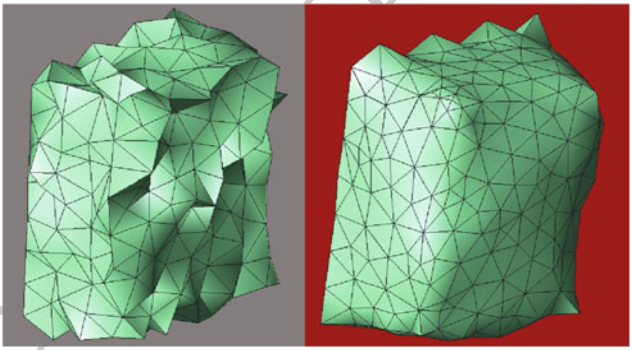

Fig. 1. (*Left*) Original interface after hole cutting. (*Right*) Smoothed interface

we need to relocate the volume nodes in order to repair the bad quality elements. To tackle this problem, we have applied a tetrahedra mesh improvement via optimization of the element condition number developed in [6]. This optimization uses a steepest descent method with a modified line search adapted to the geometrical constraints of the sub-mesh associated to the node we want to move. The implemented line search satisfies the Armijo rule which guarantees the local convergence of the method. For more details about this issue the reader can refer to [14]. Besides, a structured strategy is applied to perform the line search. The descent direction is obtained using the gradient of the objective function $f(\mathbf{x})$, in which the free vertex (node) $\mathbf{x}$ is the unknown: $f(\mathbf{x}) = \|K(\mathbf{x})\|_2 = \left[\sum_{m=0}^{M-1} \kappa_m(\mathbf{x})^2\right]^{1/2}$, where κ_m represents the condition number associated to the tetrahedron m, the moving node having M

sub-mesh elements. We then compute the steepest descent $\mathbf{p} = -\nabla f$ and find the position which gives minimum $f(\mathbf{x})$.

3 DD-Coupling

The Chimera method can be viewed as an overlapping domain decomposition technique, where transmission conditions are imposed on the interfaces of the subdomains, see [17]. A key point of the Chimera method is the way the information on the artificial boundaries is transferred, that is, the coupling. The different classical options depends on the type of the transmission conditions imposed on the interfaces. The most typical are Dirichlet/Dirichlet (D/D) coupling, also known as Schwarz' method, Dirichlet/Neumann (D/N) coupling, Dirichlet/Robin (D/R) coupling, Robin/Robin(R/R) coupling. In the litterature, the coupled system is usually solved iteratively. In each subdomain Ω_i local problems are solved by using as boundary conditions (of Dirichlet or Robin type) the values form its neighbours Ω_j until convergence is achieved. Relaxation is often needed to obtain this convergence and depends on the local character of the equation. In [8], the equivalence between the one-domain formulation and overlapping domain decomposition methods of Dirichlet/Neumann(Robin) type is shown at the continuous level. The equivalence is no longer true at the discrete level.

We have developed in this work a new way of coupling the subdomains that we refer to as Extension-Dirichlet (Ext+D). The advantage of the method is that it is implicit and parallel. Therefore, no additional iterative loop is introduced and a-fortiori the convergence of the method has no relation with the overlap. The idea consists in extending the subdomains from their interfaces to their neighboring subdomains, and imposing the Dirichlet condition implicitly, by connecting their extension to the nodes of the neighbors. This method is equivalent, in practice, to imposing Dirichlet boundary condition and eliminating it.

To illustrate the method, let us solve a diffusion equation, $\Delta u = 0$ using the Galerking method in domain $[0,1]$ discretized in 4 linear elements, with the boundary conditions, $u(0) = 1$ and $u(1) = 3$. The analytical solutions is $u = -2x + 1$. Figure 2 (Left) shows the two unconnected subdomains and the corresponding assembled global matrix. Then, Fig. 2 (Center) shows, for the same example, the results of an implicit Dirichelt/Dirichlet coupling. To achieve this, $u_3 - u_5 = 0$ substitutes line 3 and $u_4 - u_2 = 0$ subsitutes line 4. The $(\text{Ext+D})^2$ method we propose is illustrated in Fig. 2 (Right). Starting with the matrix of Fig. 2 (Left), we perform the following:

- Extend node 3 shape function to node 6 of the second subdomain. This provides additional terms in the equation for node 3.
- Extend node 4 shape function to node 1 of the second subdomain. This provides additional terms in the equation for node 4.

We can observe that in practice the $(\text{Ext+D})^2$ method creates new elements. In this example the new elements are 3–6 and 4–1. The element matrices and RHS's are

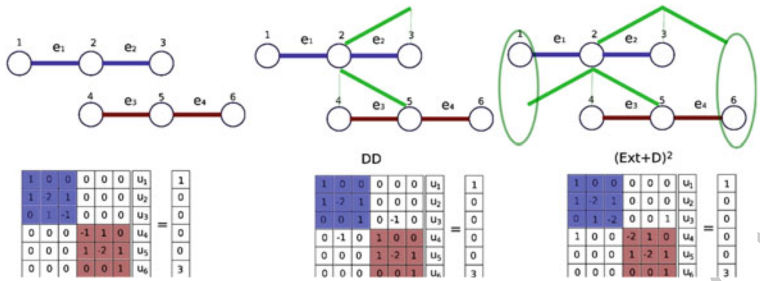

Fig. 2. (*Left*) Problem statement and domain. (*Center*) Dirichlet/Dirichlet assembled. (*Right*) (Ext+D)^2

computed as any other elements of the mesh, but only the lines of node 3 and node 4 of these matrices and RHS's are assembled into the global matrix, respectively.

The main difficulty of the method is to be able to construct a proper extension from one interface node to the other subdomain. This task is specially complex in the 3D case, mainly due to the restriction that the extension must be closed. In variational terms, this means that the extension has a compact support. We are going to describe the way to create the extensions on the interface Γ_{ij} between subdomain Ω_i and subdomain Ω_j in the 2D case. The process, illustrated in Fig. 3, consists in the

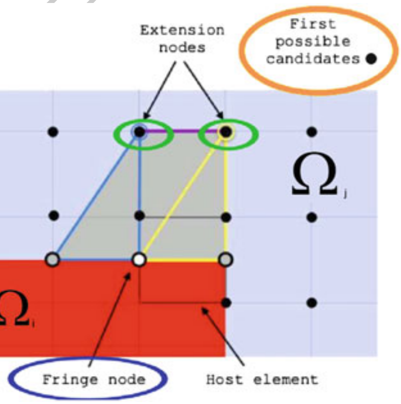

Fig. 3. 2D extensions

following.

- For a fringe node of Ω_i, identify the host element in Ω_j.
- The nodes connected to this host element are the possible candidates to create the triangles that form the associated extension. They are the black nodes.
- Construct two triangles (blue and yellow) connected to the boundaries of the fringe node.

- Close the result with a third one (purple).

The choice of the *extension nodes* (blue and yellow circled) is based on a quality criterion of the resulting triangles [7], among all the possibilities for the previous list. The third node of the triangle is the other node that forms the interface boundary.

4 Numerical Example

Figure 4 shows some results obtained for a flow around a boat. The Navier-Stokes equations are solved together with a level set function and one-equation Spalart-Allmaras turbulence model. The space discretization is a variational multiscale finite element method. The complete description of the algorithm can be found in [10, 11, 15] This complex case computed with 256 CPU's reflects the versatile property of our method and its parallel capacity. The first figure shows the extension elements while the second one the velocity module.

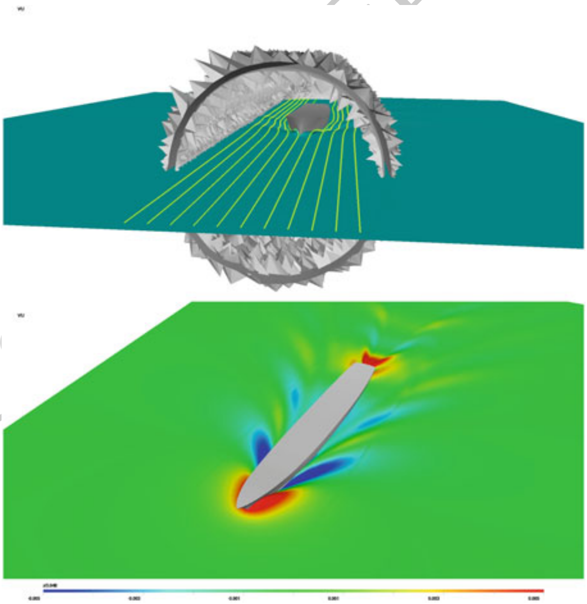

Fig. 4. (*Top*) Extension elements. (*Bottom*) Level set

5 Conclusions

We have devised in this paper a domain decomposition method, referred as $(\text{Ext}+\text{D})^2$ which is based on the explicit construction of extension elements assembled *almost* as any other element so that the implementation is straightforward. It consists in imposing implicitly Dirichlet transmission conditions and does not introduce any additional iterative loop to the algorithm. Another strength of the method is that it is naturally parallel. However, aspects like conservation should be treated in order to complete the analysis of the method.

Bibliography

[1] J.A. Benek, P.G. Buning, and J.L. Steger. A 3-d chimera grid embedding technique. *AIAA*, 1983.

[2] F. Brezzi, J.L. Lions, and O. Pironneau. Analysis of a chimera method. *C.R.A.S.*, 332:655–660, 2001.

[3] J.J. Chattot and Y. Wang. Improvement treatment of intersecting bodies with the chimera method and validation with a simple fast flow solver. *Computer & Fluids*, 27:721–740, 1998.

[4] J. Dompierre, P. Labbé, F. Guibault, and R. Camarero. Proposal of benchmarks for 3d unstructured tetrahedral mesh optimization. Technical report, Centre de Recherche en Calcul Appliqué, 1998.

[5] L. A. Freitag. On combining laplacian and optimization-based mesh smoothing techniques. In *TRENDS IN UNSTRUCTURED MESH GENERATION*, pages 37–43, 1997.

[6] L.A. Freitag and P.M. Knupp. Tetrahedral element shape optimization via the jacobian determinant and condition number. In *Proceedings of the 8th International Meshing Roundtable, South Lake Tahoe, CA, Sandia National Laboratories, Albuquerque, USA*, pages 247–258. Citeseer, 1999.

[7] P.L. George. Improvements on delaunay-based three-dimensional automatic mesh generator. *Finite Elements in Analysis and Design*, 25:297–317, 1997.

[8] G. Houzeaux. *A Geometrical Domain De Method in Computational Fluid Dynamics*. PhD thesis, CIMNE, 2002.

[9] G. Houzeaux and R. Codina. A chimera method based on a dirichlet/neumann(robin) coupling for the navier-stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 192:3343–3377, 2003.

[10] G. Houzeaux and J. Principe. A variational subgrid scale model for transient incompressible flows. *Int. J. CFD*, 22(3):135–152, 2008.

[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. *Computers & Fluids*, 44:297–313, 2011. ISSN 0045–7930. doi: DOI:10.1016/j.compfluid.2011.01.017.

[12] A. Khamayseh and A. Kuprat. Deterministic point inclusion methods for computational applications with complex geometry. *Computational Science & Discovery*, 1, 2008.

[13] S. Lo. A new mesh generation scheme for arbitraryplanar domain. *International Journal for Numerical Methods in Engineering*, 21:1403–1426, 1985.

[14] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2006.

[15] H. Owen, G. Houzeaux, C. Samaniego, A.C. Lesage, and M. Vázquez. Recent ship hydrodynamics developments in the parallel two–fluid flow solver alya. *Submitted to Computers & Fluids*, 2011.
[16] E. Pärt-Enanader. *Overlapping Grids and Applications in Gas Dynamics*. PhD thesis, Uppsala Universitet, Sweden, 1995.
[17] A. Quarteroni and A. Valli. *Domain decomposition methods for partial differential equations*. Oxford University Press, USA, 1999.
[18] C. Samaniego, G. Houzeaux, and M. Vázquez. A parallel fluid-particle solver. *Congreso de Métodos Numéricos em Engenharia*, 2011.
[19] J.L. Steger and F.C. Dougherty J.A. Benek. A chimera grid scheme. *Advances in Grid GEneration*, 5:59–69, 1983.
[20] J.L. Steger and J.A. Benek. On the use of composite grid schemes in computational aerodynamics. *Comp. Meth. Appl. Mech. Eng.*, 64:301–320, 1987.
[21] G. Taubin. A signal processing approach to fair surface design. Technical report, IBM Research, Febreary 1995.
[22] G. Taubin. Curve and surface smoothing without shrinkage. In *Fifth International Conference on Computer Vision*, 1995.