Finite Element Analysis of Multi - Component Assemblies: CAD - based Domain Decomposition

Kirill Pichon Gostaf[1], Olivier Pironneau[1], and François-Xavier Roux[1]

Summary: We apply domain decomposition to carry out finite element simulations of multi-component computer aided design (CAD) assemblies. The novelty of our research is the CAD-based domain decomposition. We consider design parts as independent sub-domains and reuse assembly topology to define regions, where the interface boundary conditions should be applied. The Dirichlet-Neumann [1], Neumann-Neumann [2] and FETI [3] methods for non-matching triangulations have been studied. We endorse the proposed framework with numerical experiments and we focus on the essence of its parallel implementation.

Key words: CAD/FEM interface, multi-component assemblies, non-matching triangulations, parallel computing.

1 Introduction

Computer aided design (CAD) and finite element (FE) modeling are standards in a concept to manufacture industrial chain. Realistic FE simulations require huge computational resources and may last unacceptably long. In this paper, we present a comprehensive framework that allows to automate and parallelize numerical simulations of multi-component CAD assemblies. We refer to the work of Pironneau [4] et al., where the authors have proposed to use constructive solid geometry modeling as a basis for spatial domain decomposition; see [6, 7, 8, 9] for related work on three dimensional contact problems in solid mechanics.

The novelty of our research is the CAD-based domain decomposition method. We consider design parts as independent sub-domains. Then we reuse assembly topology to define regions where the interface boundary conditions should be applied. Our motivation is to automate FE management of an existing CAD data, i.e. to update only the concerning meshes when CAD parts are modified. In addition, the method aims to regularize mathematical models when using various material properties (steel, cooper, rubber etc.). The method is inherently parallel and therefore perfectly suited for hight performance computing.

[1] Laboratoire Jacques-Louis Lions, UPMC, France.
e-mail: {gostaf}{pironneau}{roux}@ljll.math.upmc.fr

2 CAD-based Domain Decomposition

Generally, a FE model of an entire CAD prototype takes several days to be properly defined. Once a pre-processing stage is completed (a global mesh, loadings and constraints are generated), adaptive refinement procedure requires communication with CAD kernel at each computational iteration. Meanwhile, engineering design changes are made on a daily basis at the CAD level, and the mesh generally may not follow the changes. Hence, the FE model cannot be updated within such timespan.

Assembly-driven decomposition

The application of assembly driven domain decomposition allows to automate the above framework. We consider each component of a CAD assembly as an independent sub-domain. Triangulations are generated independently and could be further updated. Variational formulations are then explicitly written for each sub-domain. Inter-domain continuity conditions are set according to the domain decomposition algorithm.

Let $\{P_1,...,P_s\}$ refer to a set of assembly components (design parts), with $s \geq 2$. We define $\{\Omega_1,...,\Omega_s\}$ to be a set of the corresponding computational sub-domains. An illustration is given in Fig. 1.

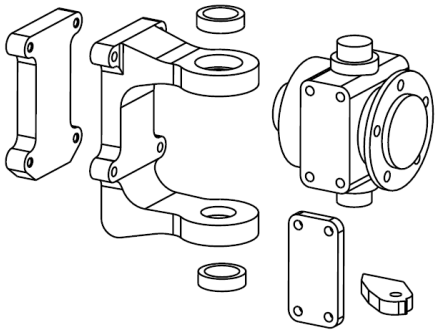

Fig. 1 An assembly-driven domain decomposition. Design parts are considered as independent computational sub-domains.

Modeling accuracy

Solid parts are generated independently of the FE process, yet they are manipulated by FE algorithms after discretization. For a manifold M (a boundary of a solid part), we define

$$H = \mathrm{diam}(M) = \sup_{x_1,x_2 \in M} |x_1 - x_2|$$

along with the "smallest feature length" l (the smallest hole, fillet, chamfer etc.). According to the CAD documentation [10], parts are initially created with the relative accuracy δ^r_{CAD} which satisfies

$$10^{-6} \leqslant l/H < \delta^r_{CAD} \leqslant 10^{-2}$$

CATIA modeling platform [11] allows to design parts with

$$10^{-6} \leqslant l, H \leqslant 10^{3}$$

however an option

$$10^{-8} \leqslant l, H \leqslant 1$$

is available to design small parts, but the module has limited implementation.

Initialization of inter-component contact regions

We propose to reuse "assembly constrains" (data on parts relative position stored in a CAD assembly file) in order to generate an initial list of the contact regions (called contact faces). Let $\mathbb{S}$ denote a set of initial contact faces between all adjacent assembly components (solid parts)

$$\mathbb{S} = \{P_i \cap^* P_j\} \qquad 1 \leq i \neq j \leq s$$

where $\cap^*$ stands for a Boolean cut operator (intersection of manifolds). The number of all possible contact pairs is bounded by the binomial coefficient

$$\dim(\mathbb{S}) \leq \begin{pmatrix} s \\ 2 \end{pmatrix}$$

In practice, for CAD assemblies, the number of inter-component contact faces is much smaller than the binomial coefficient and often satisfies

$$\dim(\mathbb{S}) \sim \mathscr{O}(s)$$

Definition 1. Two objects $A \subset \mathbb{R}^d$ and $B \subset \mathbb{R}^d$, $d \geq 1$ are geometrically equal if the set A is equivalent to the set B.

Definition 2. Topological equivalence - Two objects are topologically equivalent if there is a homeomorphism between them.

In the following, a contact face $\mathscr{F}$ is a set of patches; a patch is defined by four NURBS or B-spline curves. Let $\mathscr{F}_{i,j}$ and $\mathscr{F}_{j,i}$ be the opposite contact faces belonging to the adjacent components P_i and P_j, respectively. Then, $\mathscr{T}_{i,j}$ and $\mathscr{T}_{j,i}$ be a discretization of the above contact faces. In order to build

$$\mathscr{T}_{i,j} = \mathscr{T}_{j,i} \tag{1}$$

we require both geometrical and topological equivalence of $\mathscr{F}_{i,j}$ and $\mathscr{F}_{j,i}$. However, (1) is hard to achieve, since solid models are built with only a fixed accuracy.

Remark: Obviously, matching triangulations $\mathscr{T}_{i,j} = \mathscr{T}_{j,i}$ might be generated within an additional computational cost. Unfortunately, for simulations involving sliding, mixed finite elements (shape, order) or discontinuities in material coefficients matching triangulations are hard to maintain.

Geometric discontinuities across contact faces

In practice, most contact regions are either non-planar or have curved boundaries. When meshes are generated independently, $\mathscr{T}_{i,j}$ and $\mathscr{T}_{j,i}$ often appear different, owing to round-off errors. As a result, geometric discontinuities are certain for non-matching triangulations, which is clearly seen in Fig. 2.

Let u_{1h} and u_{2h} be discrete functions defined on the triangulations $\mathscr{T}_{1,2}$ and $\mathscr{T}_{2,1}$, respectively. For $\mathscr{T}_{1,2} \neq \mathscr{T}_{2,1}$ the computation of a jump operator

$$u_{2h}(x) - u_{1h}(x)$$

on contact faces is not properly defined (not unique). Indeed, when Ω_h is a polygonal approximation of Ω, numerical integration of boundary integrals will not be equal on $\mathscr{T}_{1,2} \neq \mathscr{T}_{2,1}$. In this context, we are interested to compute the value of a finite element function $u_h(y)$ slightly outside its domain of definition, namely at $y \in \mathbb{R}^3$ close to Ω_h in the sense

$$\min_{x \in \Omega_h} |x - y| < c\,h$$

where $c \in \mathbb{R}_+$, and h is the discretization parameter (mesh size).

Assume that Ω_h is triangulated into tetrahedral elements. Let $\{v_0, ..., v_3\}$ be the vertices of a tetrahedral element T close to y. The barycentric coordinates $\{\lambda_0, ..., \lambda_3\}$ of y with respect to T satisfy

$$\sum_{i=0}^{3} \lambda_i = 1 \qquad \text{and} \qquad y = \sum_{i=0}^{3} \lambda_i\, v_i$$

When a point y does not belong to the discrete domain, we shall define

$$u_h(y) = \sum_{i=0}^{3} \lambda_i\, u_h(v_i)$$

where the vertices v_i are those of the nearest tetrahedral element. We use the same approach for a $\mathbb{P}_2$ or higher Lagrangian finite element.

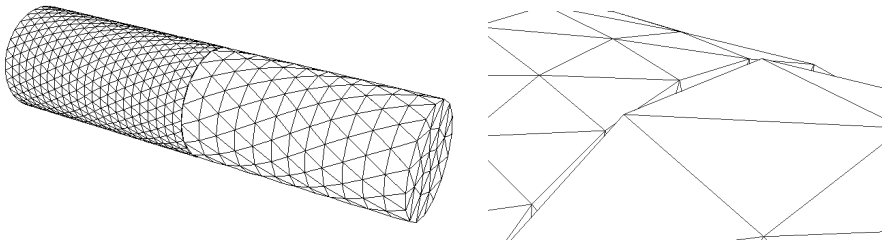

Fig. 2 Geometric discontinuity across the contact region in case of non-matching triangulations.

Parallel implementation

Depending on the computer architecture, we propose two implementation schemes: the first is suitable for small commodity clusters; the second fits massively parallel architecture (HPC). Let N_{cpu} be the number of available CPUs, processor cores, that are used for a FE simulation. Let s be the number of assembly components. Reasonably, one can expect $N_{cpu} \sim s$ for small or intermediate commodity clusters and $N_{cpu} >> s$ for HPC machines. Fig. 3 (left) illustrate the case, where three sub-domains are assigned to a single process, i.e. `cpu2`, and solved in sequence; the right chart in Fig. 3 shows the case, where each sub-domain is treated in parallel. Inside one sub-domain either algebraic mesh partitioning or multi-threading is used to parallelize a local solver.

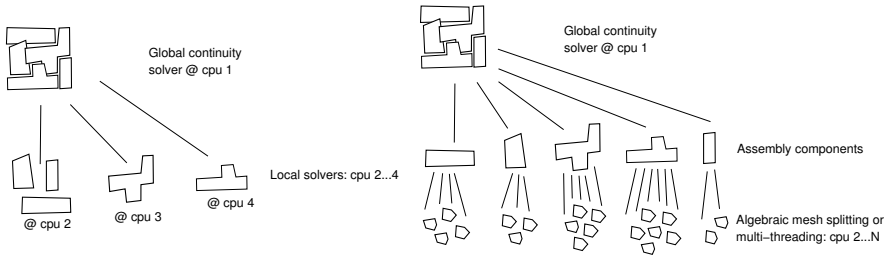

Fig. 3 Parallel implementation for small commodity clusters (left) and HPC systems (right).

Assume that one MPI process lives on each multi-core unit, and OpenMP parallelization occurs below, i.e. inside the multi-core NUMA unit [12, 13]. Actually, a good practice for computational performance is to set the number of OpenMP threads equal to the physical number of cores inside one NUMA node. Fig. 4 depicts the scalability results of a multi-threaded CG solver running on a Cray XE6 node (left) and SGI Altix UV 100 shared memory cluster. We observe almost linear speed-up, $\times 6$ and $\times 8$, respectively, for threads placed inside a single multi-core die.

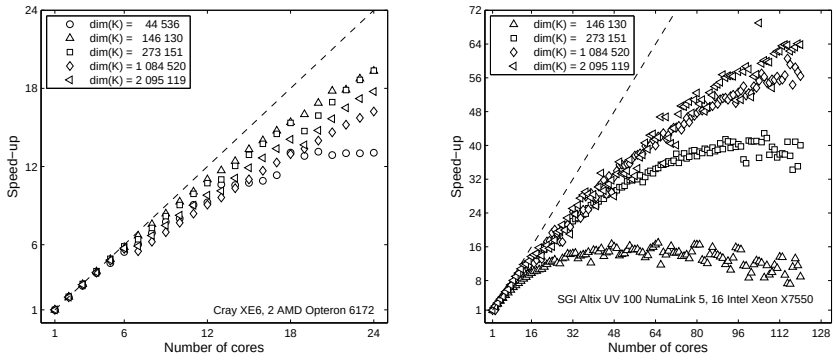

Fig. 4 Scalability results of a multi-threaded CG solver: Cray XE6 (left), SGI Altix UV (right).

3 Numerical Experiments

We consider the case of a linear elasticity. The model problem allows to describe the displacement $\mathbf{u} = (u_1, u_2, u_3)^t$ of an elastic body in its equilibrium position under the action of an external body force $\mathbf{f} = (f_1, f_2, f_3)^t$ and a surface charge $\mathbf{g}_N = (g_{N_1}, g_{N_2}, g_{N_3})^t$ distributed on $\partial\Omega_N$. Without getting technical about the spaces involved, i.e. the displacement weighting and trial solution W and V, for details see [15], the weak formulation for a problem of linear elasticity reads: find $\mathbf{u} \in V$ such that for all $\mathbf{w} \in W$

$$a(\mathbf{u}, \mathbf{w}) = F(\mathbf{w})$$

with

$$a(\mathbf{u}, \mathbf{w}) = \int_\Omega \lambda\, (\nabla \cdot \mathbf{u})(\nabla \cdot \mathbf{w})\, dx + \int_\Omega 2\,\mu\, \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{w})\, dx$$
$$F(\mathbf{w}) = \int_\Omega \mathbf{f} \cdot \mathbf{w}\, dx + \int_{\partial\Omega_N} \mathbf{g}_N \cdot \mathbf{w}\, ds$$

where λ and μ are the Lamé parameters, and $\varepsilon(\mathbf{u})$ is the infinitesimal strain tensor.

We have discretized the above problem using a $\mathbb{P}_2$ finite element. The FE model consists of three sub-domains, each triangulated independently, see Fig. 5. When working with fine meshes, the finest sub-domain contains roughly 3.6 million unknowns. We have used 4 computational nodes of a Cray XE6, with a total of 96 cores. The tasks were executed by 4 MPI processes each with 24 OpenMP threads, see Fig. 3 (right) (one MPI for a global continuity solver, one MPI per sub-domain). Three domain decomposition algorithms for non-matching meshes (Dirichlet-Neumann, Neumann-Neumann and FETI) have been implemented using a modified version of the integrated environment `FreeFem++` [14].

For simplicity reasons (to avoid floating sub-domains), we have set that each sub-domain has a part of its boundary belonging to a Dirichlet datum $\mathbf{u} = \mathbf{g}_D$ on $\partial\Omega_D$

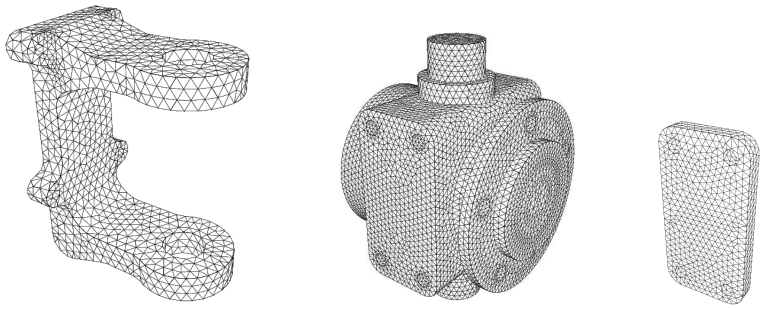

Fig. 5 A three component assembly. Non-matching triangulations are clear.

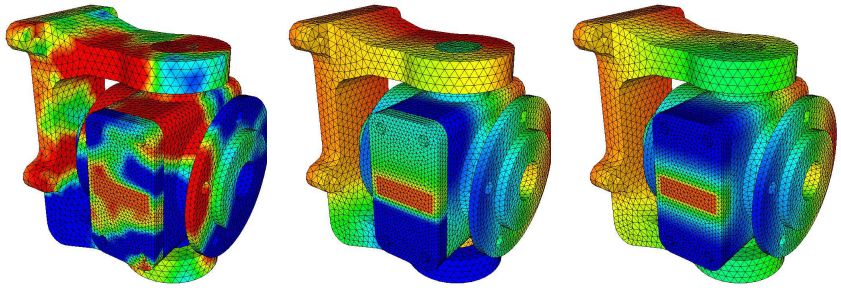

Fig. 6 The FETI method for non-matching triangulations. Lineal elasticity problem. Displacement field at iterations: 1, 2, 10.

(bolt holes of the left and right components, a back face of the middle component). All components are subject to the gravitational force. Portion of a front face of the right component is subject to a surface charge. We have set E=210 GPa, $\nu = 0.3$, E=105 GPa, $\nu = 0.34$ and E=117 GPa, $\nu = 0.33$ for the left, middle and right component, respectively; recall: $\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}$ and $\mu = \frac{E}{2(1+\nu)}$. For each sub-domain, we have set the initial solution $\mathbf{u}_{ih}^{(0)} = (0,0,0)^t$.

On Fig. 6, we have visualized the computed displacements at iterations 1, 2 and 10; the computational time was 73 seconds per a single global iteration in the FETI method (fine meshes). The rate of convergence is shown in Fig. 7 for the Dirichlet-Neumann and Neumann-Neumann methods, respectively. The FETI method exhibits performance similar to the Neumann-Neumann method. The computations have been repeated for quasi-uniform coarse, medium and fine triangulations; for the mixed test we have used coarse, fine, medium triangulations for the left, middle and right component, respectively.

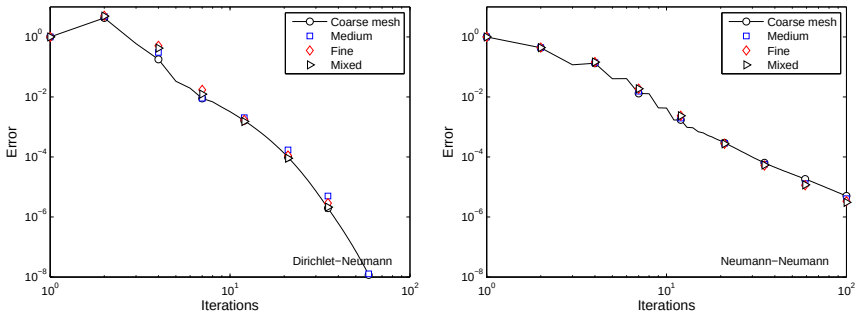

Fig. 7 The Dirichlet-Neumann method, relative L^2 error. The curves depict different levels of component mesh resolution H/h.

4 Conclusions

We have introduced a comprehensive framework that allows to automate numerical simulations of multi-component CAD assemblies in the sense that meshes can be independently updated for each component. This paper has presented the CAD-based domain decomposition method. We have implemented the Dirichlet-Neumann, Neumann-Neumann and FETI methods for non-matching triangulations. Numerical results have indicated that all above methods are highly accurate finite element approximations for problems of linear elasticity. We have compared convergence properties of the three methods. The Dirichlet-Neumann method exhibits better convergence and is the most simple to implement.

References

1. D. Funaro, A. Quarteroni and P. Zanolli: An iterative procedure with interface relaxation for domain decomposition methods. SIAM J. Numer. Anal. **25(6)**, 1213–1236 (1988)
2. P. Le Tallec, Y.H. De Roeck and M. Vidrascu: Domain decomposition methods for large linearity elliptic three-dimensional problems. J. Comput. Appl. Math. **34(1)**, 93–117 (1991)
3. C. Farhat and F.X. Roux: A method of finite element tearing and interconnecting and its parallel solution algorithm. Internat. J. Numer. Methods Engrg. **32**, 1205–1227 (1991)
4. F. Hecht, J.L. Lions and O. Pironneau: Domain decomposition algorithm for computer aided design. Applied nonlinear analysis, 185–198 (2002)
5. P. Heintz and P. Hansbo: Stabilized Lagrange multiplier methods for bilateral elastic contact with friction. Comput. Methods Appl. Mech. Engrg. **195(33-36)**, 4323–4333 (2006)
6. S. Hartmann and E. Ramm: A mortar based contact formulation for non-linear dynamics using dual Lagrange multipliers. Finite Elem. Anal. Des. **44(5)**, 245–258 (2008)
7. L. Nilsson and J. Forsberg: Evaluation of response surface methodologies used in crashworthiness optimization. International journal of impact engineering **32**, 759–777 (2006).
8. D.J. Benson and J.O. Hallquist: A single surface contact algorithm for the post-buckling analysis of shell structures. Comput. Methods Appl. Mech. Engrg. **78(2)**, 141–163 (1990)
9. M.A. Puso and T.A. Laursen: A 3D contact smoothing method using Gregory patches. Internat. J. Numer. Methods Engrg. **54**, 1161–1194 (2002)
10. Parametric Technology Corporation: PRO/ENGINEER Wildfire 4.0 Part modeling. Help topic collection, (2008)
11. SIMULIA: Realistic Simulation Inside CATIA V5. Abaqus for CATIA V5, R19 (2008)
12. J.M. Bull, J. Enright, X. Guo, C. Maynard and F. Reid: Performance Evaluation of Mixed-Mode OpenMP/MPI Implementations. Int. J. Parallel Prog. **38(5-6)**, 396–417 (2010)
13. L. Smith and M. Bull: Development of mixed mode MPI / OpenMP applications. Sci. Program. **9(2-3)**, 83–98 (2001)
14. O. Pironneau, F. Hecht, A. Le Hyaric and K. Ohtsuka: FreFem++. URL `http://www.freefem.org/`
15. A. Ern: Aide-mémoire, Eléments finis Dunod, Paris (2005)