Università della Svizzera italiana

Faculty of Informatics

Institute of Computational Science ICS

Parallel Surface and Volume Projection for FEM Simulation on Unrelated Arbitrarily Distributed Meshes

Prof. Dr. Rolf Krause Patrick Zulian

Introduction

Objective

Development of a parallel computing framework for stable information transfer between meshes in the context of *non-conforming domain decomposition* techniques, *adaptive mesh refinement* and *multilevel solvers*. Ability to handle arbitrarily distributed and unrelated meshes in parallel environments in a transparent and efficient way.

Applications

Computational Mechanics, *Multigrid* methods.

Transfer Operator

In order to transfer a quantity **u** between non-nested *FEM* spaces

$$\mathcal{X}^S = \text{span}\,\{\psi_p^S\}, \qquad \mathcal{X}^M = \text{span}\,\{\psi_p^M\},$$

the construction of a L^2-like projection operator **T** is needed. The weak coupling condition can be written as

$$\int_Y (\mathbf{u}^M - \mathbf{u}^S)\lambda = 0 \qquad \forall \lambda \in \mathcal{M},$$

For suitably chosen multiplier spaces M, in the case of *non-conforming domain decomposition* methods, this coupling condition allows for proving optimal error estimates. The domain Y is either surface or volume. By expressing quantities as a linear combination of the finite element basis,

$$\int_Y \left(\sum_p \mathbf{u}_p^S \psi_p^S - \sum_q \mathbf{u}_p^M \psi_p^M \right) \lambda_r = 0 \Leftrightarrow \sum_p \mathbf{u}_p^S \int_Y \psi_p^S \lambda_r = \sum_q \mathbf{u}_p^M \int_Y \psi_p^M \lambda_r,$$

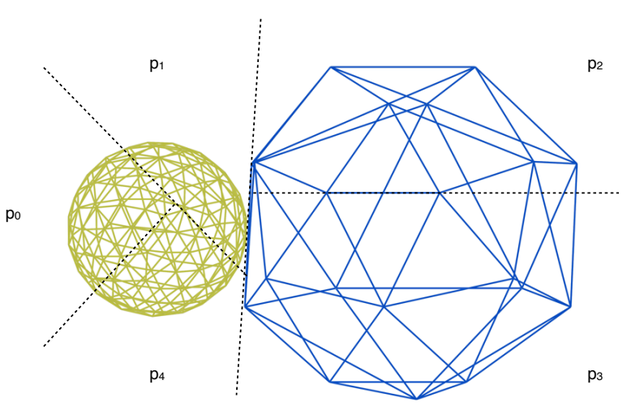
we obtain the coupling operators

$$M_{pr} = \int_Y \psi_p^S \lambda_r, \qquad B_{qr} = \int_Y \psi_p^M \lambda_r$$

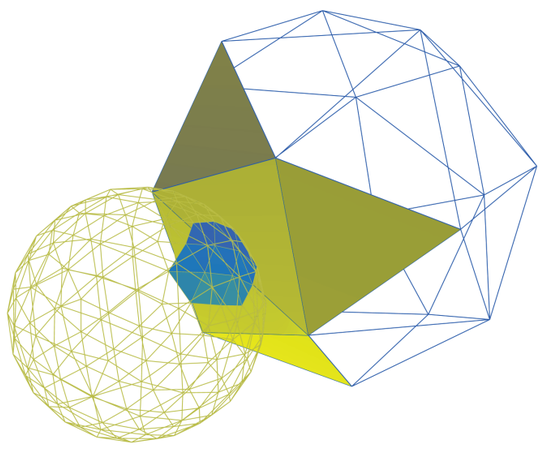
and $\mathbf{T} = \mathbf{M}^{-1}\mathbf{B}$. The transfer is performed by $\mathbf{u}^S = \mathbf{T}\mathbf{u}^M$. For more details see [1] and [2].

Surface Projections

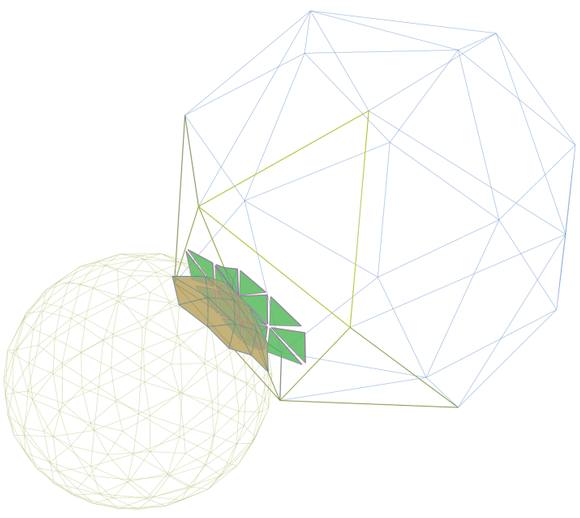
Here we illustrate the procedural pipeline for the assembly of the surface transfer operator.

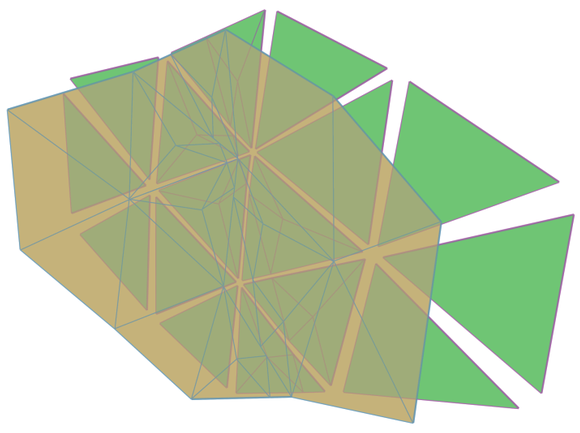


Arbitrary domain decomposition on a grid of processes p_i, i=0, . . . , 4.



Proximity detection. Filled faces depict the candidates for the projection.





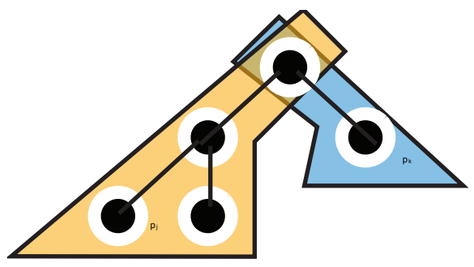
Boundary elements are geometrically projected onto each other, intersected, triangulated and used to assemble the transfer operator between the two domains.

In the case of volume projections the pipeline is similar but without geometrical surface-projections.

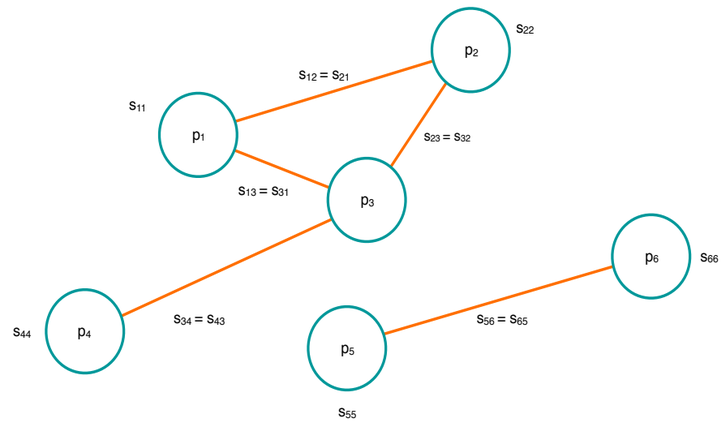
Parallelization

The procedural pipeline of the parallelization consists of candidates detection, scheduling and load balancing, data distribution, processing, and data redistribution.

Candidates Detection

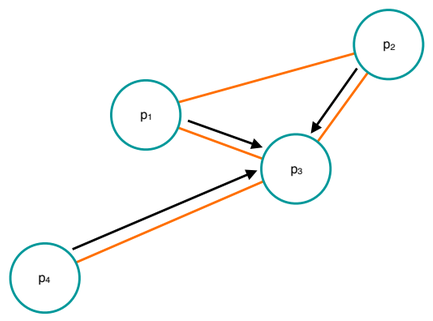


Proximity/Intersection detection with parallel *octree*. Parallel iterative breadth-first traversal with only structural information matching.

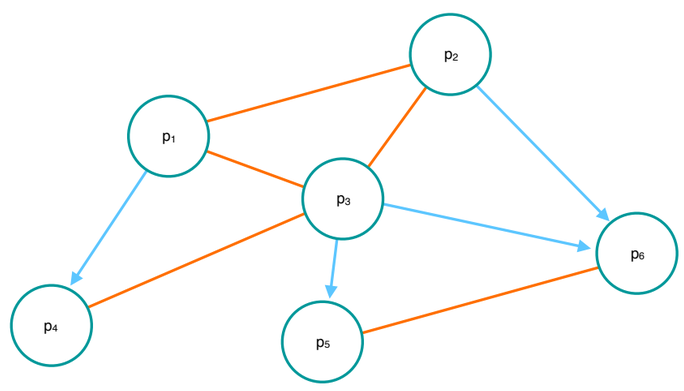


An example of data dependency graph resulting from the detection algorithm with the shared work s_{ij} on the edges and local work s_{ii} on the nodes.

Scheduling and Load Balancing



First phase



Second phase

From the dependency graph we derive a communication graph and a balanced scheduling. The scheduling algorithm works on the graph and it is divided in two phases. In the *first phase*, it optimizes locality by trying to put the desired amount of work on the less connected nodes and by putting the surplus on the most connected ones. In the *second phase* new connections (edges) are created in order to move the extra work from overloaded processes to idles ones. Once the communication graph as been set up the actual data needed for processing generally liying on different physical nodes has to be exchanged. The procedure ensures that the amount of communicated data is minimal.

Processing and Dynamic Scheduling

Each thread has its queue. During processing idle threads can steal tasks from other threads. Similarly idle processes can steal tasks and their data from other overloaded processes by sending a steal request and obtaining a task when available.

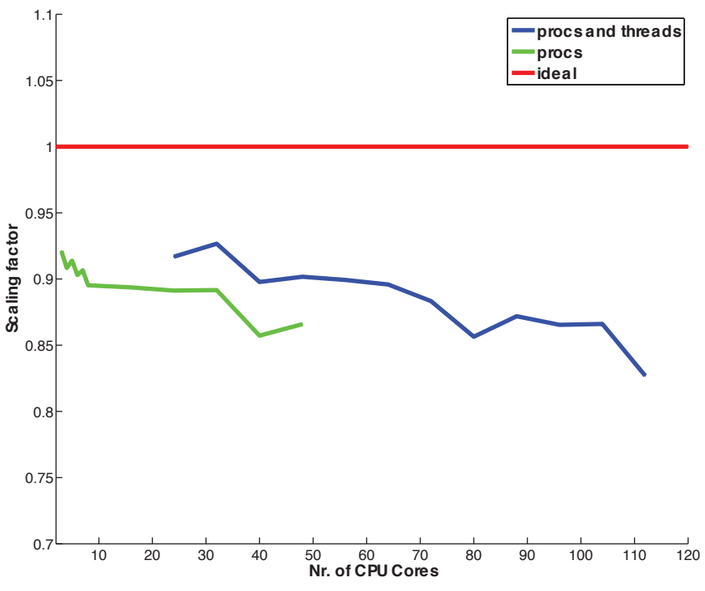
Numerical Experiments

Weak Scaling

We investigated how the framework behaves, with respect to computational time, when increasing the number of processes and keeping the amount of data per process fixed. *MPI* processes are reffered to as **procs** .

Mesh data

Two parallelepipedal domains in a grid of *MPI* processes. Each color represents a different process.



The experiment consists of assemblying the surface transfer operator in the possible contact boundary of two parallelepipedal domains. The meshes describing the domain have different resolutions in order to create imbalance.
The preliminary results are encouraging and we intend to investigate more thoroughly with different scenarios, use cases and data distributions.

References

[1] Thomas Dickopf and Rolf Krause. Efficient simulation of multi-body contact problems on complex geometries: A flexible decomposition approach using constrained minimization. International Journal for Numerical Methods in Engineering, 77(13):1834–1862, 2009.

[2] C. Bernardi, Y. Maday, and A. T. Patera, A new nonconforming approach to domain decomposition: the mortar element method, Collège de France Seminar, Vol. XI (Paris, 1989–1991) Pitman Res. Notes Math. Ser., vol. 299, Longman Sci. Tech., Harlow, 1994, pp. 13–51.