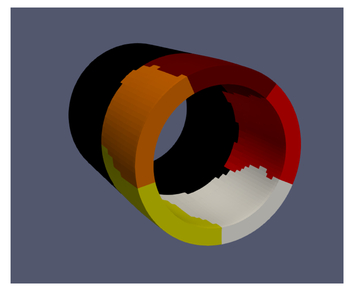


A GENERIC FRAMEWORK FOR SCHWARZ DECOMPOSITION METHODS.

{ PRATIK.NAYAK AND HARTWIG.ANZT }@KIT.EDU

Karlsruher Institut für Technologie

PROBLEM

Exascale computing requires resilience, scalability and efficiency. A test bed is required to:

1. Test and solve large general problems.
2. Test the capability of the Schwarz methods, one-level and multilevel methods.
3. Experiment with different interface boundary conditions for the Optimized Schwarz methods.
4. Test the benefits of an asynchronous communication layer, especially at exascale.

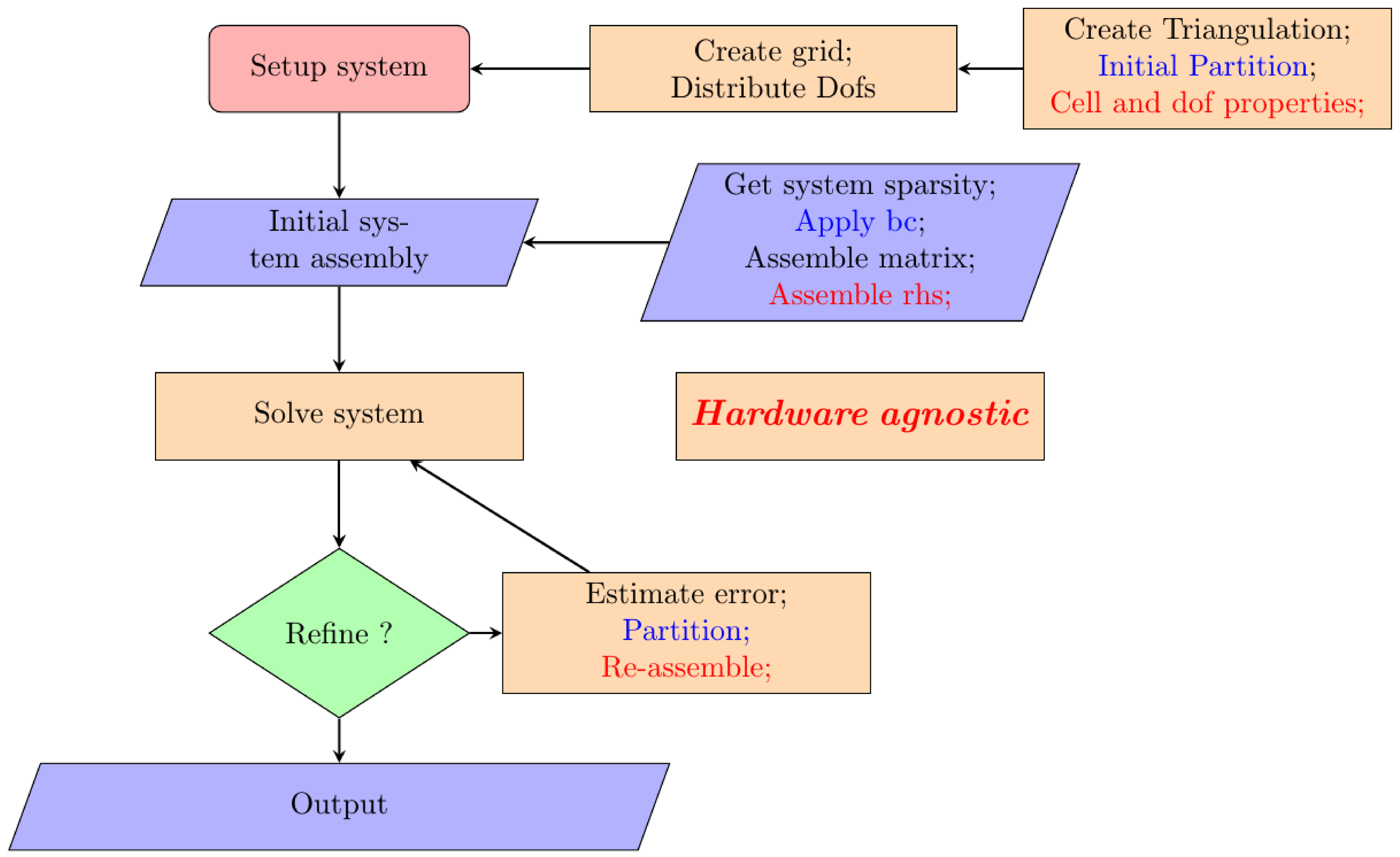
CONTRIBUTIONS

A generic framework to solve Schwarz methods for very large problems using `deal.ii` [1]. Solve for all permutations below:

1. Optimized Schwarz, Restricted Schwarz and Multi-level Schwarz methods.
2. Different partitioning methods: `Trilinos - Zoltan`, `METIS`, `p4est` with adaptive meshing.
3. Different finite element problems and techniques.

METHOD

1. Partition can be replaced by different modules: `METIS`, `Zoltan`, `KaHIP`, `p4est`, `userdefined`
2. System solution is hardware agnostic: CPU, GPU, KNL and indifferent to communication layers: Synchronous or asynchronous.



EXAMPLE - THE POISSON PROBLEM - OPTIMIZED SCHWARZ

Generic interface condition

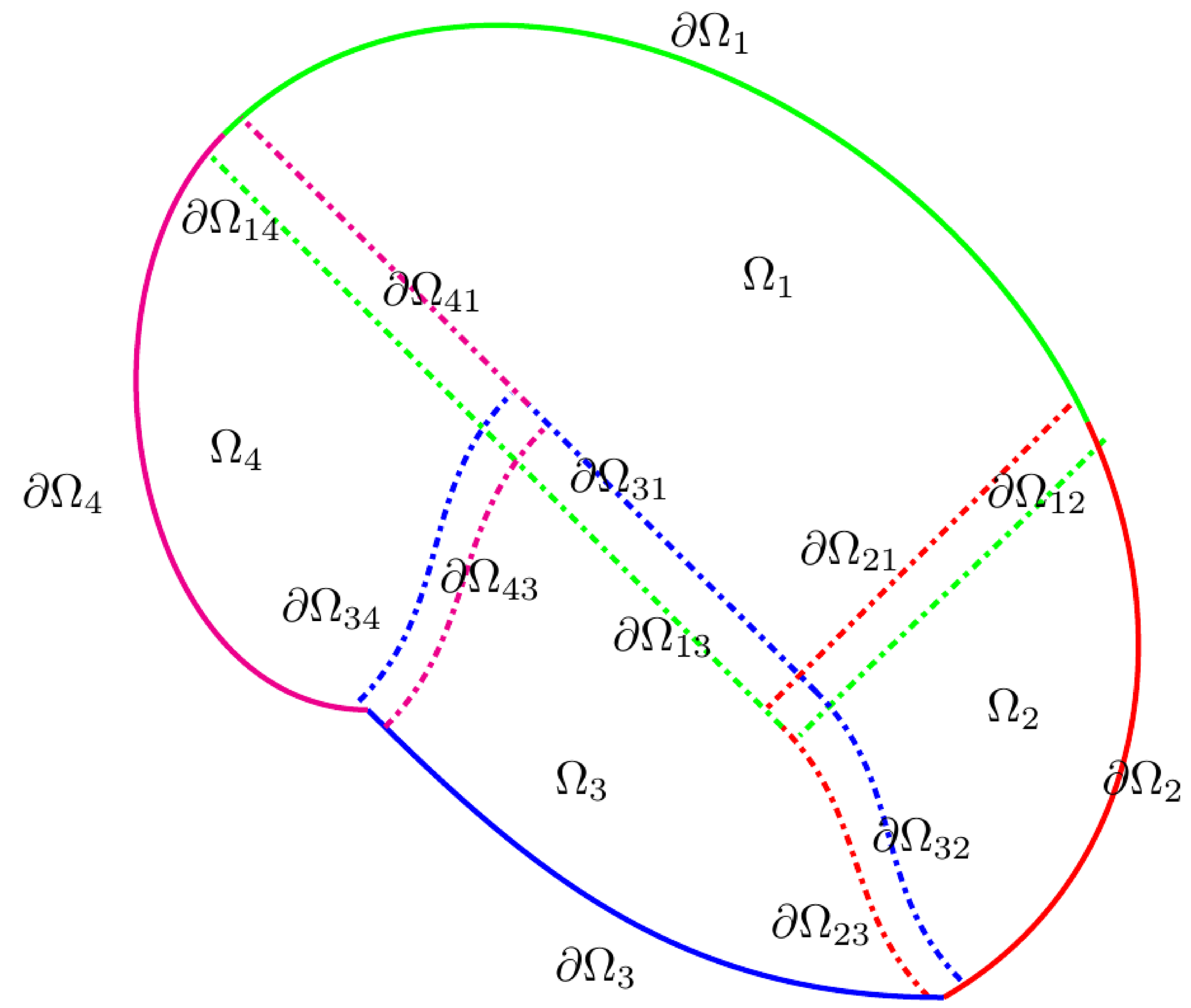
$$-\Delta u(\mathbf{x}) = f(\mathbf{x}), \forall \mathbf{x} \in \Omega; \mathcal{B}(u) = g, \text{on } \partial\Omega.$$

$$\int_{\Omega} \nabla\varphi_i \cdot \nabla\varphi_j \, d\mathbf{x} - \int_{\partial\Omega} \varphi_i \mathbb{B}\varphi_j \, d\mathbf{x} = \int_{\Omega} \varphi_i f \, d\mathbf{x} + \int_{\partial\Omega} \varphi_i g \, d\mathbf{x} \qquad i = 1, \cdots, N$$

Robin interface condition

$$\begin{aligned} -\Delta u &= f, && \Omega_m. \\ u &= a, && \partial\Omega_m \setminus \cup\partial\Omega_{mk}, k = \{(1...N) - m\}, \forall k \times m \notin \emptyset \\ \frac{\partial u}{\partial n} + \alpha u &= g_k, && \partial\Omega_{mk}, k = \{(1...N) - m\}, \forall k \times m \notin \emptyset \\ g_k &= \left(\frac{\partial u}{\partial n} + \alpha u\right)\Bigg|_k, && k = \{(1...N) - m\}, \forall k \times m \notin \emptyset \end{aligned} \tag{1}$$

$$m : \int_{\Omega_m} \nabla\varphi_i \cdot \nabla\varphi_j + \sum_{k}^{k \times m \notin \emptyset} \int_{\partial\Omega_{mk}} \varphi_i(\alpha\varphi_j) = \int_{\Omega_m} \varphi_i f + \sum_{k}^{k \times m \notin \emptyset} \int_{\partial\Omega_{mk}} \varphi_i g_k \tag{2}$$



REFERENCES

[1] G. Alzetta et.al, The `deal.ii` library: Version 9.0 In *J. Num. Math, 2018*

[2] A. Buchanan and A. Fitzgibbon. Interactive Feature Tracking using K-D Trees and Dynamic Programming. In *CVPR '06*

A FUTURE DIRECTION

Using `deal.ii` add an asynchronous communication layer to study the benefits of asynchronicity especially for large problems. Can resilience be proved for a general problem at large scales ?

Experiment with different interface boundary conditions and compare the benefits of these conditions for different physical problems compared and including multi-level Schwarz methods.

SOURCE CODE

The source code of the modified fork of the `deal.ii` library is available at `https://github.com/pratikvn/dealii`