Delft University of Technology
Friedrich-Alexander-Universität Erlangen-Nürnberg

Project: Efficient Inexact Subdomain Solvers for Domain Decomposition Methods with RACE

Supervisor(s):
Filipe Cumaru, Alexander Heinlein (Numerical Analysis, EEMCS)
Dane Lacey (FAU), Gerhard Wellein (FAU, TU Delft)

September 22, 2025

Project Description

Partial differential equations (PDEs) are central to many scientific and engineering problems. Their numerical solution requires discretization, leading to large sparse linear systems:

$$Ax = b. \tag{1}$$

Higher accuracy demands finer meshes, resulting in larger systems. Direct solvers become infeasible for large A due to superlinear growth in memory usage and runtime.

To address this challenge, iterative methods such as Krylov subspace methods [5] are used, with the cost of each iteration scaling linearly with problem size. However, convergence slows as the system grows. Domain decomposition-based preconditioners can accelerate convergence.

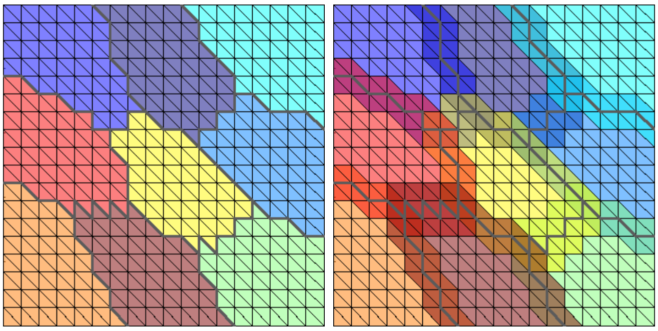

Figure 1: Non-overlapping (left) and overlapping (right) domain decomposition of a structured triangulation. Different colors indicate different subdomains.

Here, we consider overlapping Schwarz preconditioners. To this end, we partition A based on a non-overlapping domain decomposition and extend subdomains by layers of nodes to obtain overlapping subdomains; see figure 1. This allows us to define the preconditioner

$$M^{-1} = \sum_{i=1}^{N} R_i^\top A_i^{-1} R_i,$$

where R_i restricts to the i-th overlapping subdomain and $A_i = R_i A R_i^\top$; see [6]. Employing A_i^{-1} in the preconditioner is referred to as using *exact solvers*. Then solving

$$M^{-1}Ax = M^{-1}b,$$

converges faster than the original system equation (1). However, solving the local subproblems with A_i^{-1}, typically with direct solvers, quickly becomes the performance bottleneck, especially as larger subdomains are employed or memory is limited.

The goal of this project is to investigate the use of inexact solvers for the local subdomain systems and compare them to the capabilities of standard direct solvers. Inexact solvers typically reduce memory demands significantly, while the benefits in computational cost are less clear. During the project, we will systematically investigate memory and computational costs, depending on the choice of the inexact solver as well as the properties of the underlying system matrix A. In addition to algorithmic aspects, we will also consider the hardware efficiency of the implementations. Here, new developments such as cache-blocking matrix-power kernels provided by the RACE framework [1, 3] will be taken into account. For the implementation of the domain decomposition preconditioners, we will employ the FROSch (Fast and Robust Overlapping Schwarz) package [2] from the Trilinos library [4].

Tasks

The tasks of this project are:

- Literature review on domain decomposition methods and inexact subdomain solvers.
- Installation of FROSch on DelftBlue (TU Delft) and/or Fritz (NHR@FAU) supercomputers.

- Interfacing FROSch with the RACE framework.
- Systematic performance evaluation of the inexact subdomain solvers on DelftBlue and/or Fritz supercomputers, including hardware-efficiency aspects.
- Comparison with exact subdomain solvers.

Contact

Are you interested, or do you have any questions? Send an email to Alexander Heinlein (a.heinlein@tudelft.nl) and/or Gerhard Wellein (gerhard.wellein@fau.de).

References

[1] C. L. Alappat, G. Hager, O. Schenk, and G. Wellein. Level-based Blocking for Sparse Matrices: Sparse Matrix-Power-Vector Multiplication. *IEEE Transactions on Parallel and Distributed Systems*, 34(2):581–597, Feb. 2023. ISSN 1045-9219, 1558-2183, 2161-9883. doi: 10.1109/TPDS.2022.3223512. arXiv:2205.01598 [math].

[2] A. Heinlein, A. Klawonn, S. Rajamanickam, and O. Rheinbach. FROSch: a fast and robust overlapping Schwarz domain decomposition preconditioner based on Xpetra in Trilinos. In *Domain decomposition methods in science and engineering XXV*, volume 138 of *Lect. Notes Comput. Sci. Eng.*, pages 176–184. Springer, Cham, 2020. doi: 10.1007/978-3-030-56750-7_19.

[3] D. Lacey, C. Alappat, F. Lange, G. Hager, H. Fehske, and G. Wellein. Cache blocking of distributed-memory parallel matrix power kernels. *The International Journal of High Performance Computing Applications*, 39 (3):385–404, May 2025. ISSN 1094-3420. doi: 10.1177/10943420251319332. Publisher: SAGE Publications Ltd STM.

[4] M. Mayr, A. Heinlein, C. Glusa, S. Rajamanickam, M. Arnst, R. Bartlett, L. Berger-Vergiat, E. Boman, K. Devine, G. Harper, M. Heroux, M. Hoemmen, J. Hu, B. Kelley, K. Kim, D. P. Kouri, P. Kuberry, K. Liegeois, C. C. Ober, R. Pawlowski, C. Pearson, M. Perego, E. Phipps, D. Ridzal, N. V. Roberts, C. Siefert, H. Thornquist, R. Tomasetti, C. R. Trott, R. S. Tuminaro, J. M. Willenbring, M. M. Wolf, and I. Yamazaki. Trilinos: Enabling Scientific Computing Across Diverse Hardware Architectures at Scale, Mar. 2025. arXiv:2503.08126 [cs].

[5] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Other Titles in Applied Mathematics. Society for Industrial and Applied Mathematics, Jan. 2003. ISBN 978-0-89871-534-7. doi: 10.1137/1.9780898718003.

[6] A. Toselli and O. Widlund. *Domain decomposition methods—algorithms and theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2005. ISBN 978-3-540-20696-5. doi: 10.1007/b137868.