EXASTEEL-II

Energy efficiency of nonlinear domain decomposition methods

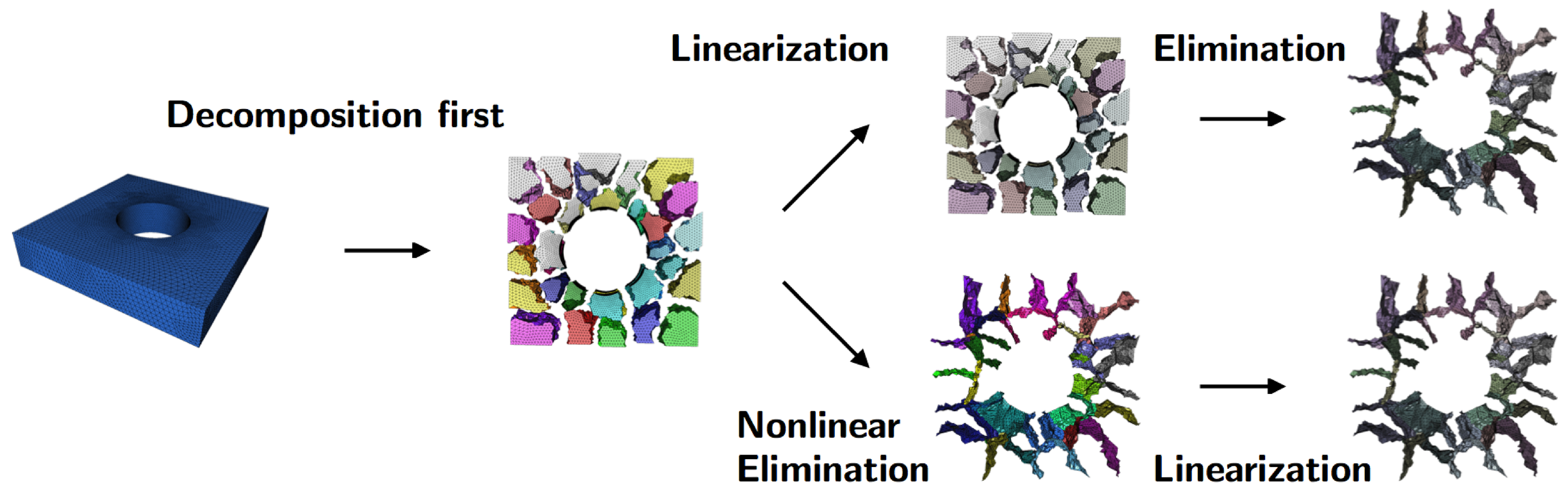
Martin Lanser*, Axel Klawonn*, Oliver Rheinbach†, Gerhard Wellein††, and Markus Wittmann††

*Universität zu Köln, †TU Bergakademie Freiberg, ††Friedrich-Alexander-Universität Erlangen-Nürnberg

1. Introduction

Nonlinear domain decomposition (DD) approaches are solution methods for nonlinear finite element problems based on a decomposition of the nonlinear problem into concurrent nonlinear problems. In recent years, new nonlinear DD approaches have been introduced and their superiority over the classical combination of a nonlinear solver, e.g., Newton's method, with a linear DD approach has been shown for many model problems. Nevertheless, in nonlinear DD methods, many decoupled local nonlinear problems have to be solved in parallel, which can lead to a load imbalance. We propose to use a non-blocking MPI_Ibarrier and to set idle cores to sleep, in order to save energy [1]. We show that nonlinear DD methods can save a significant amount of energy compared to classical approaches and even a better power efficiency can be obtained. We prove this effect for the example of **NL-FETI-DP** (Nonlinear Finite Element Tearing and Interconnecting - Dual Primal) and compare to classical **NK-FETI-DP** (Newton-Krylov-FETI-DP).

2. Nonlinear-FETI-DP



Nonlinear-FETI-DP methods - developed in EXASTEEL - apply a decomposition-first paradigm (vs. classic linearization-first) and are based on the nonlinear saddle point problem [2],

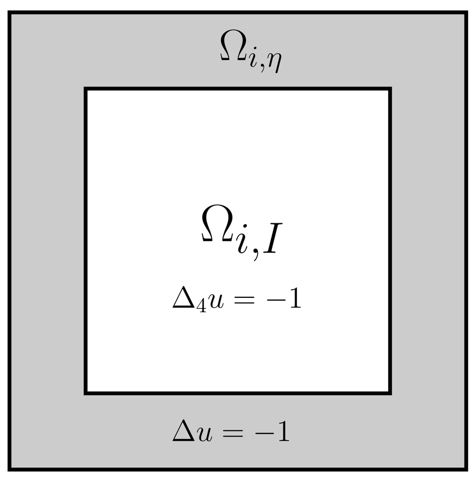
$$\begin{aligned} \widetilde{K}(\tilde{u}) + B^T\lambda &= \tilde{f} \\ B\tilde{u} &= 0\,. \end{aligned}$$

The tangent $D\widetilde{K}(\tilde{u})$ is almost block diagonal. Nonlinear FETI-DP methods benefit from **increased local work**, **reduced communication & synchronization**. Different variants have been developed and a superior parallel scalability compared to classical NK-FETI-DP approaches has been achieved [3]. Here, we discuss the variant NL-FETI-DP-3, where a certain subset of degrees of freedom is eliminated nonlinearly. We also provide a comparison to classical NK-FETI-DP.

3. Simple Model Problem

We consider simple model problems based on the nonlinear p-Laplace equation and artificially introduce localized nonlinearities. Therefore, we use $p = 4$ in **inclusions** inside of subdomains and use a linear matrix matrial ($p = 2$). We distinguish between:

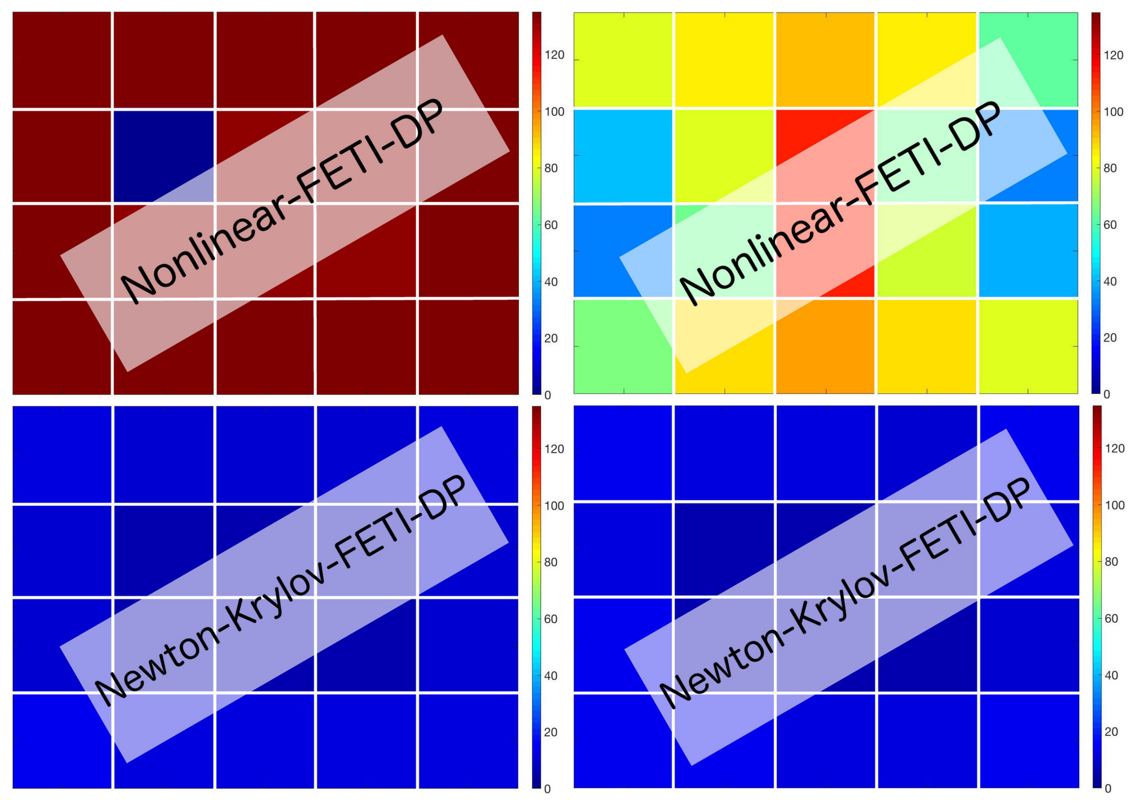
- **Single Inclusion Problem** with a single inclusion in one subdomain
- **Many Inclusions Problem** with an inclusion in each subdomain



4. Load Balance of NL-FETI-DP

We measured the time each subdomain spents in an MPI_Barrier. The load imbalance in the NL-FETI-DP method is caused by the solution of the local nonlinear problems. As expected, NK-FETI-DP is well balanced in contrast to NL-FETI-DP.

Single Inclusion Problem | **Many Inclusions Problem**



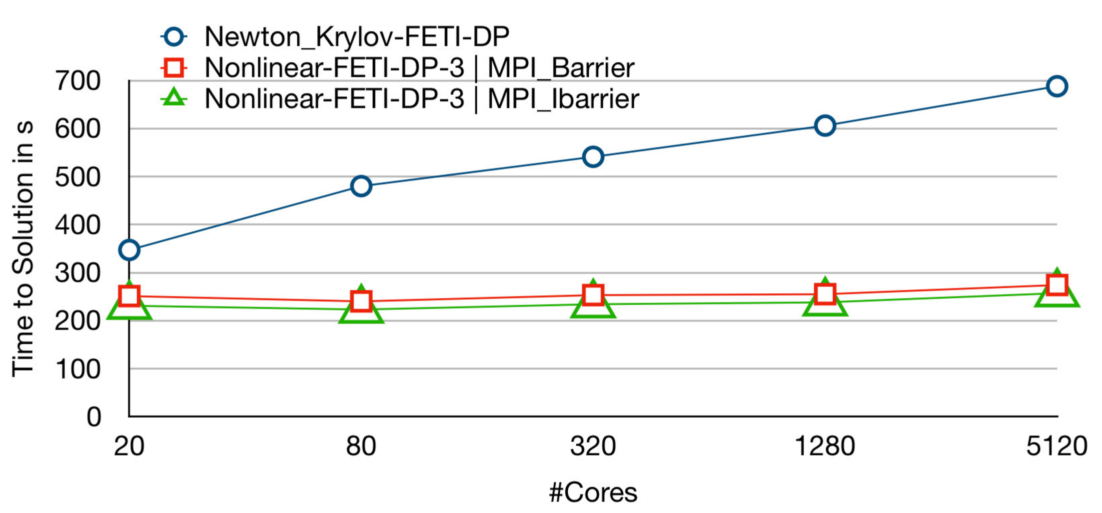
5. MPI_Ibarrier and Sleep

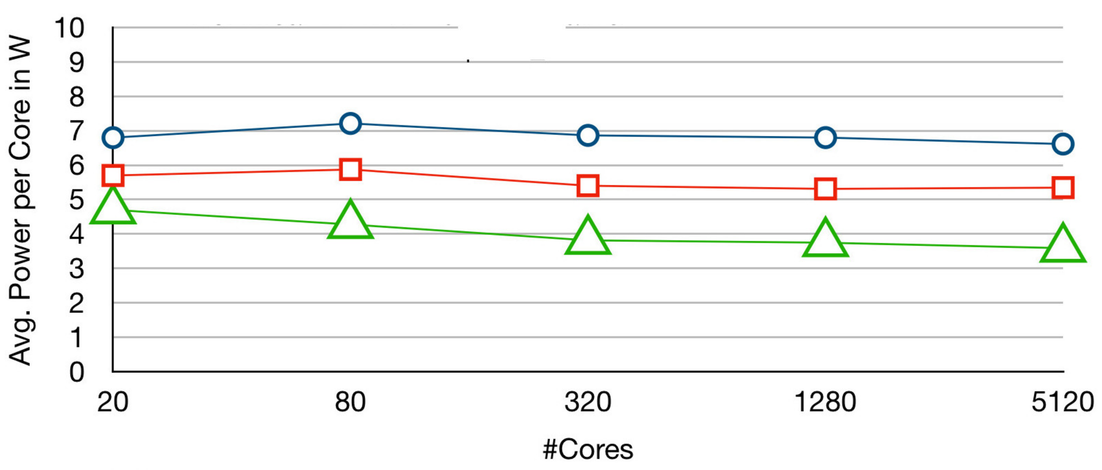
- Cores which have finished their local nonlinear solve are kept in an MPI_Barrier.
- They consume less energy than working cores. But still a significant amount compared to sleeping cores.
- We therefore set the cores to sleep by replacing the MPI_Barrier by the following construction:

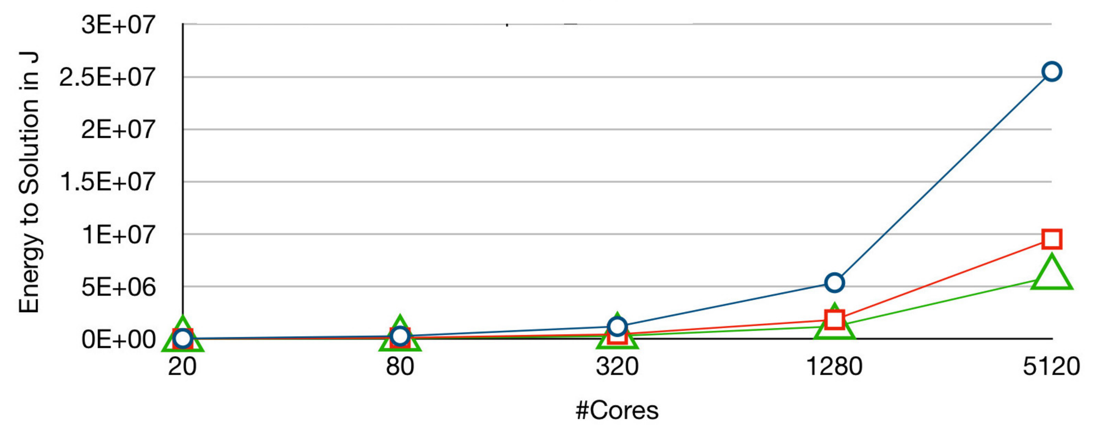
```
int flag;
MPI_Request r;

MPI_Ibarrier(comm,&r);

MPI_Test(r, &flag, MPI_IGNORE_STATUS);

while (!flag){
    usleep(sleep_duration);
    MPI_Test(r, &flag, MPI_IGNORE_STATUS);
}
```

6. Measurements for Nonlinear-FETI-DP

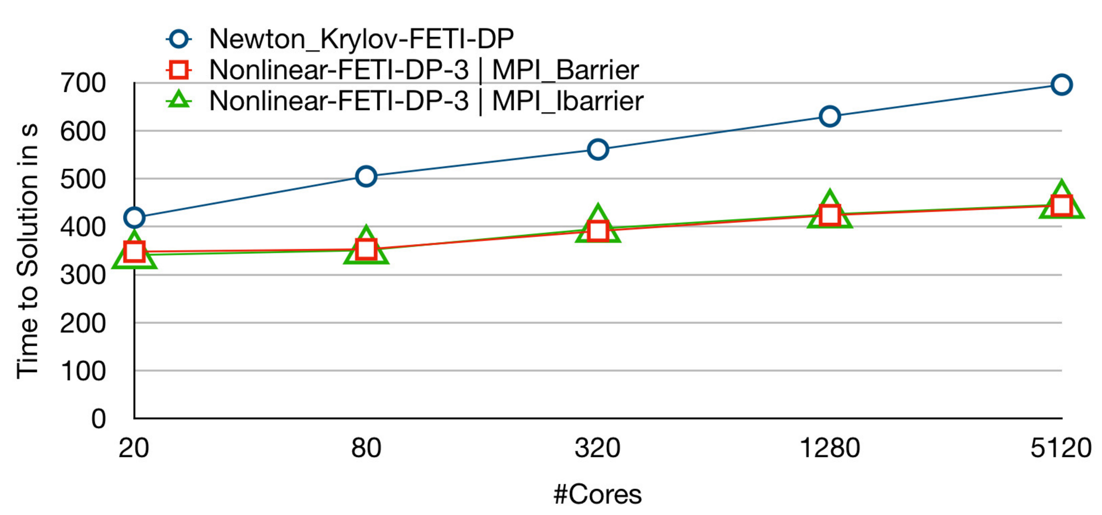
Single Inclusion Problem

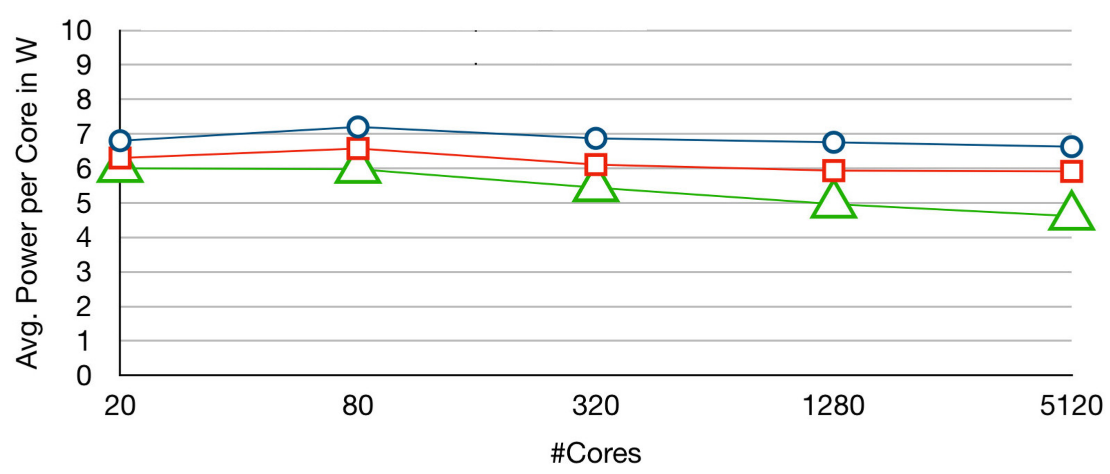


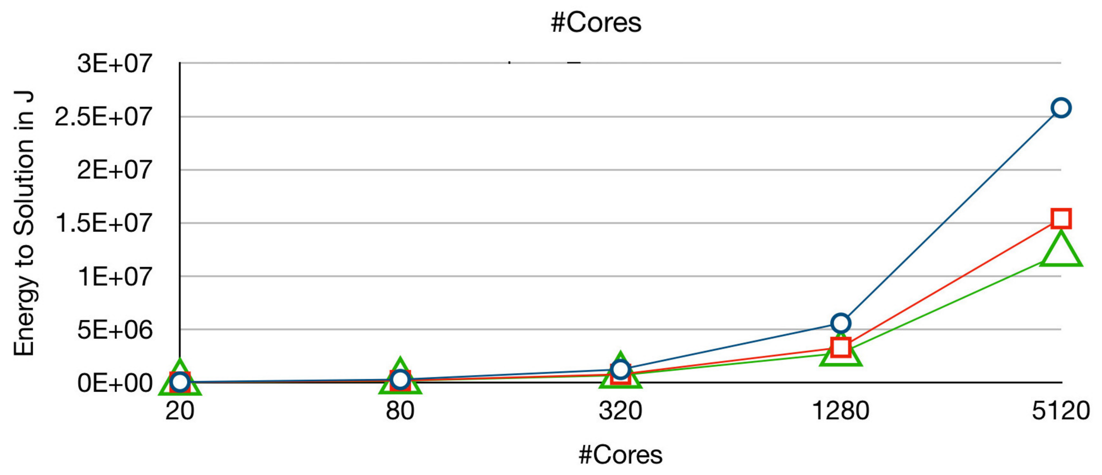




Many Inclusions Problem







All measurements are performed using LIKWID [4].

7. References

[1] Axel Klawonn, Martin Lanser, Oliver Rheinbach, Gerhard Wellein, and Markus Wittmann. Energy efficiency of nonlinear domain decomposition methods. In preparation.

[2] Axel Klawonn, Martin Lanser, and Oliver Rheinbach. Nonlinear FETI-DP and BDDC methods. *SIAM J. Sci. Comput.*, 36(2):A737–A765, 2014.

[3] Axel Klawonn, Martin Lanser, Oliver Rheinbach, and Matthias Uran. Nonlinear FETI-DP and BDDC methods: a unified framework and parallel results. *SIAM J. Sci. Comput.*, 39(6):C417–C451, 2017.

[4] J. Treibig, G. Hager, and G. Wellein. Likwid: A lightweight performance-oriented tool suite for x86 multicore environments. In *Proceedings of PSTI2010*, San Diego CA, 2010.