



Proceedings of the Seventh International Conference on
Parallel, Distributed, GPU and Cloud Computing for Engineering
Edited by: P. Iványi, F. Magoulès and B.H.V. Topping
Civil-Comp Conferences, Volume 4, Paper 1.2
Civil-Comp Press, Edinburgh, United Kingdom, 2023
doi: 10.4203/ccc.4.1.2
©Civil-Comp Ltd, Edinburgh, UK, 2023

Enhancing the Global/Local Coupling Method: An Asynchronous Parallel Framework

A. El Kerim¹, P. Gosselet² and F. Magoulès^{1,3}

¹ Laboratory of Mathematics in Interaction with Computer
Science, CentraleSupélec, Paris-Saclay University,
Gif-sur-Yvette, France

² Lamcube, University of Lille, CNRS, Centrale Lille, Lille, France

³ Faculty of Engineering and Information Technology,
University of Pècs, Pècs, Hungary

Abstract

A novel approach is being developed to introduce a parallel asynchronous implementation of non-intrusive global-local coupling. This study examines scenarios involving numerous patches, including those covering the entire structure. By leveraging asynchronous, the method aims to minimize reliance on communication, handle failures effectively, and address load imbalances. Detailed insights into the methodology are presented, accompanied by a demonstration of its performance through an academic case study.

Keywords: asynchronous iterations, global/local coupling, RDMA-MPI, asynchronous domain decomposition methods, non intrusive coupling, linear problem

1 Introduction

In the industry, the submodeling method [1,12] has been widely employed for studying local complex behaviors in simulated structures. However, despite its non-intrusive nature, this approach often results in significant errors, which are evident as imbalances at the interfaces of the submodel patches in the reference model. To overcome these limitations and enhance the accuracy of the submodeling method, the global/local coupling technique [7] has been developed. This technique adopts an iterative approach to refine the submodeling outcomes and achieve greater precision. The global/local coupling method can be classified as a domain decomposition technique, specifically a Schwarz decomposition [3, 4, 10, 11]. These methods are well-suited for parallel computation, and recent studies [5, 6, 9, 13] have demonstrated their compatibility with asynchronous iterations. Asynchronous iterations provide enhanced tolerance to network latencies, load imbalances, and highly heterogeneous computing architectures. In the paper [4], the first asynchronous version of the global/local coupling method is introduced for linear elliptic problems. Building upon this work, the current paper briefly presents the coupling method and its asynchronous variant. Furthermore, it discusses the implementation aspects and showcases novel numerical results obtained through this approach.

2 Principle of the method

To illustrate the study of local complex behaviors in simulated structures, we consider an approximation of a 2D turbine blade, as shown in Figure 1. The model comprises three distinct zones: a global zone (depicted in blue), and two zones of interest (highlighted in yellow and green) that exhibit specific and intricate geometries.

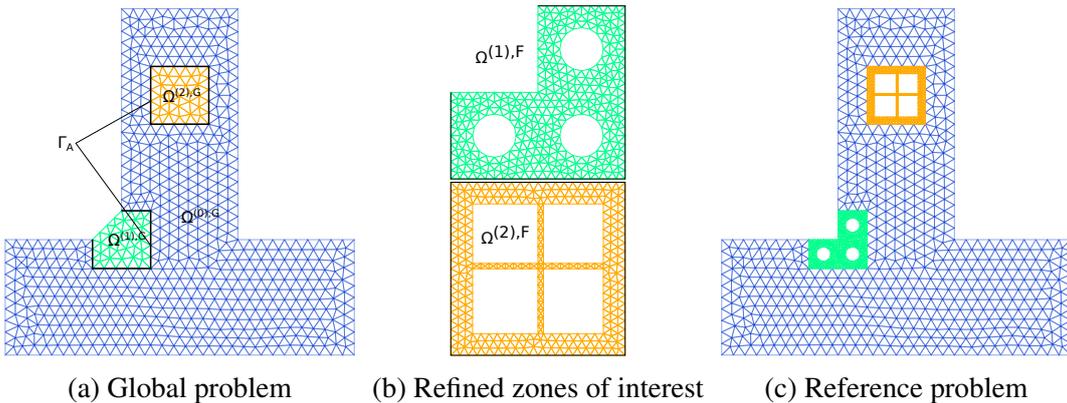


Figure 1: Coarse and refined meshes

In Figure 1a, we present the coarse-scale representation of the 2D turbine blade, referred to as the global model. The coarse mesh used in this model cannot accurately

capture the behaviors occurring in the zones of interest. Therefore, highly refined meshes are required to simulate these zones effectively, as shown in Figure 1b. The refined zones of interest, denoted as $\Omega^{1,F}$ and $\Omega^{2,F}$, are where geometrical modifications or nonlinear behaviors, such as plasticity, viscoplasticity, fatigue, and cracking, are of particular interest.

The classic scenario of the global/local coupling method is illustrated in Figure 2. Initially, a linear global coarse model represents the overall structure (Figure 1a). Subsequently, specific zones of interest Ω_G^s ($s > 0$) are identified based on criteria or the need for more detailed modeling. In the example shown, adapted meshes and geometrical details are introduced in the fine modeling of these zones, and modifications to material laws may also be incorporated. The fine computations are performed in parallel on the patches, utilizing the global solution as a Dirichlet boundary condition. Although the fine and global subdomains may differ ($s > 0$), their interface Γ^s remains consistent, defined as $\Gamma^s = \Omega \cap \partial\Omega^{s,F} = \Omega \cap \partial\Omega^{s,G}$. The error in the flux balance

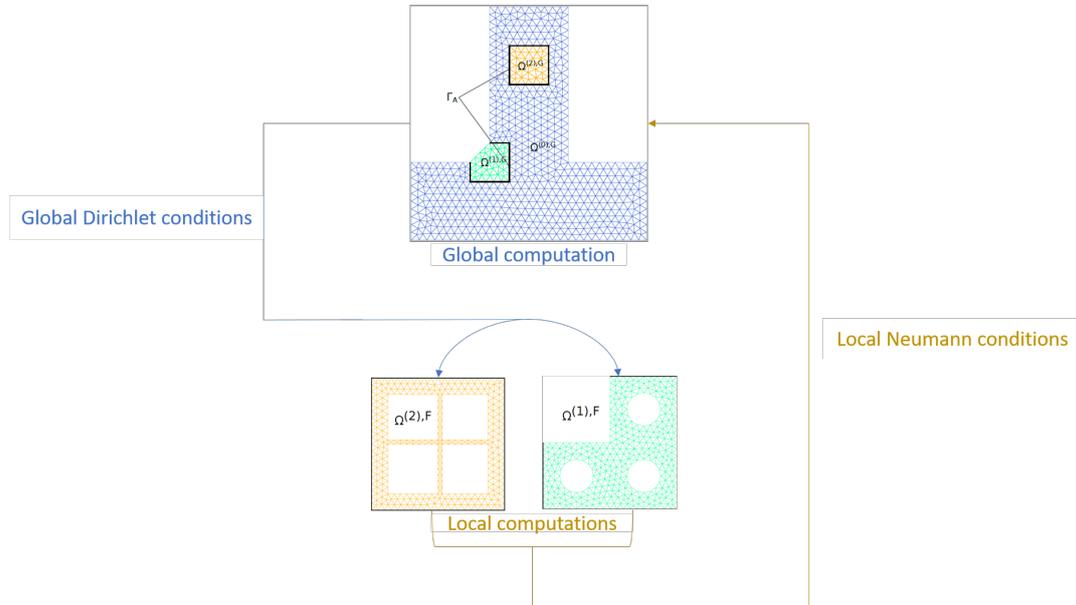


Figure 2: Global Local non-invasive coupling iteration

between the global zone Ω^0 (not covered by patches) and the fine models is quantified as a lack of equilibrium. This error, known as the residual, is incorporated into the global model as an immersed Neumann condition on the interface. The corrected global model, accounting for this interface load, is then solved, followed by the solution of local Dirichlet problems. This iterative process continues until convergence is achieved.

The global/local coupling method can be viewed as a simple iterative technique, specifically a Richardson iteration in its simpler form. Its objective is to obtain the reference solution (Figure 1c) by utilizing computations performed on the global and fine models, without the need for creating a potentially cumbersome reference model or making significant interventions in the models and software.

It is worth noting that if the computation is halted after the first iteration without sending back the nodal reactions to the global model, a submodeling method is performed.

3 Asynchronous coupling iterations

The Global/Local coupling method has been recognized as a robust and non-invasive technique. However, its applicability to large-scale simulations and modern parallel computing architectures is limited due to its alternating nature, as discussed in [4]. This alternating approach, illustrated in Figure 3a, introduces waiting and inactivity times on both sides, leading to reduced performance, especially when faced with load imbalances, communication delays, or machine failures.

To address these challenges and improve the efficiency of the Global/Local coupling method, an asynchronous parallel version has been proposed. This approach allows each processor to work independently and at its own pace, utilizing the most recent available data. The time sequence depicted in Figure 3b illustrates the asynchronous approach, where processors only wait when they have no new data to process.

By adopting asynchronous iterations, the method gains several advantages. It becomes more resilient to network latencies, load imbalances, and heterogeneous computing architectures. Furthermore, by reducing waiting and inactivity times, the overall performance and efficiency of the Global/Local coupling method are enhanced, making it well-suited for high-performance computing environments.

The asynchronous approach enables processors to make progress independently and asynchronously, effectively utilizing computational resources and minimizing potential bottlenecks. This adaptability to modern parallel computing architectures and the ability to handle large-scale simulations make the asynchronous parallel version of the Global/Local coupling method a favourable choice.

By embracing the benefits of the asynchronous parallel version, we can overcome the limitations of the synchronous approach and achieve improved performance and scalability in high-performance computing environments. This advancement opens up new possibilities for effectively applying the Global/Local coupling method in a wide range of complex simulations and analyses.

4 Implementation

RDMA (Remote Direct Memory Access) has emerged as a promising approach for implementing asynchronous communication protocols. Unlike traditional two-sided communication based on MPI, RDMA utilizes one-sided communication, which has shown significant efficiency and adaptability in recent research. Works in [15] and [9] highlight the effectiveness of one-sided communication, also known as MPI-RDMA,

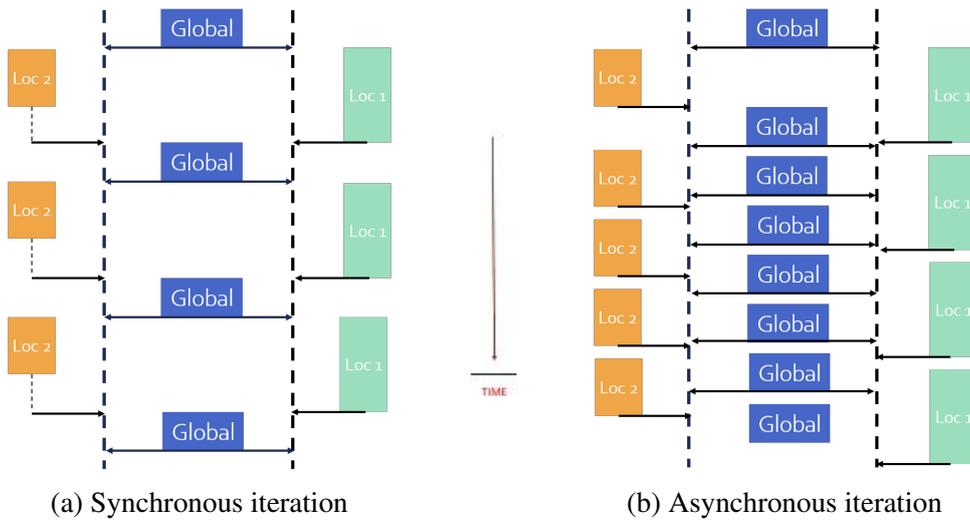


Figure 3: Time course of the Global/Local coupling in the case of two patches

in achieving efficient asynchronous communication.

To implement RDMA-based asynchronous communication, a memory region called a window is created. This window allows direct access to data on remote machines without involving the target machine. PUT and GET operations can be performed on the window, enabling efficient data transfer and retrieval. This approach eliminates the need to interrupt computations for send or receive operations, making it highly suitable for asynchronous calculations.

The experiments reveal that network-specific factors, MPI version variations, and the inherent asynchronicity of communication influence the overall performance of RDMA-based asynchronous protocols.

Synchronization is an essential aspect of RDMA communication. Two synchronization techniques are commonly used: active synchronization and passive synchronization. Active synchronization involves performing collective operations using **MPI_WIN_Fence()** to synchronize all processes before transitioning between iterations. In contrast, passive synchronization is employed in asynchronous scenarios. Processors individually synchronize by opening an epoch with **MPI_win_lock**, executing desired **PUT** and **GET** operations, and then closing the epoch with **MPI_win_Unlock**. The **MPI_win_Flush** operation ensures the completion of send operations within the epoch.

Overall, RDMA-based asynchronous communication, leveraging the advantages of one-sided communication and appropriate synchronization techniques, offers a promising approach for efficient and scalable implementations in various MPI versions and network configurations.

5 Numerical results

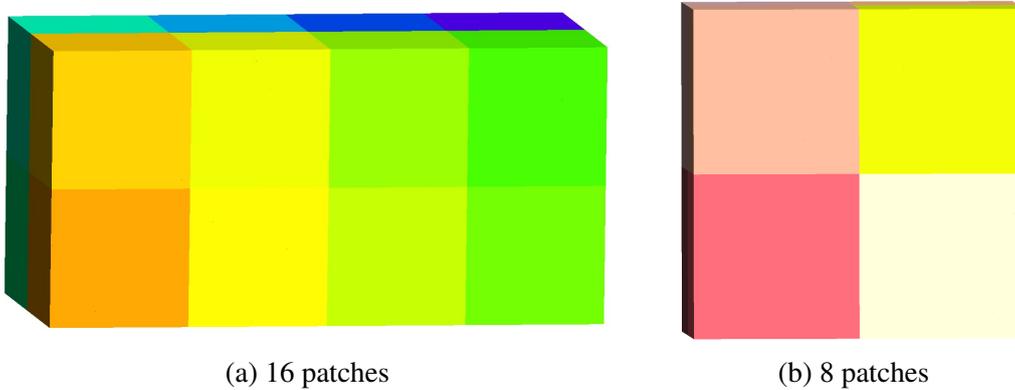
5.1 Codes

In our Python code implementation, we utilize various tools and software to facilitate different aspects of the study. Firstly, we employ GMSH [8] for generating geometries and meshes for the cases under investigation. For the finite element approximation, we rely on the Getfem library [14]. To enable parallel computing, we utilize the mpi4py library [2].

The study was conducted using the cluster at the LMPS simulation center, utilizing multiple workstations interconnected via an Ethernet network. The machines employed in the cluster exhibit heterogeneity, featuring four different generations of CPUs: Intel(R) Xeon(R) CPU E5-1660 v3 (Haswell) @ 3.00GHz, Intel(R) Xeon(R) CPU E5-2630 v4 (Broadwell) @ 2.20GHz, Intel(R) Xeon(R) Silver 4116 CPU (Sky-lake) @ 2.10GHz, and Intel(R) Xeon(R) W-2255 CPU (Cascade Lake) @ 3.70GHz.

5.2 Studied case

The aim of the study is to generate a flexible number of patches, specifically non-overlapping contiguous patches that cover the entire global model. To achieve this, a cuboid domain is chosen, with cuboid patches applied. An example of an 8-patch configuration can be seen in Figure 4b, while Figure 4a illustrates a 16-patch scenario.



Two representations of cubes are considered in the study. The cubes in the global model are homogenous and have a coarse mesh, as depicted in Figure 5a. On the other hand, the cubes in the local models have refined meshes (shown in Figure 5b). They are heterogeneous and contain a spherical inclusion within each cube. By default, the sphere is centered and has a radius equal to a quarter of the side of the cube, although these parameters may vary in certain studies (refer to Figure 5c). It should be noted that the fine meshes are independently constructed within the patches and are not required to match at the interface. In contrast, the global mesh conforms to the interface.

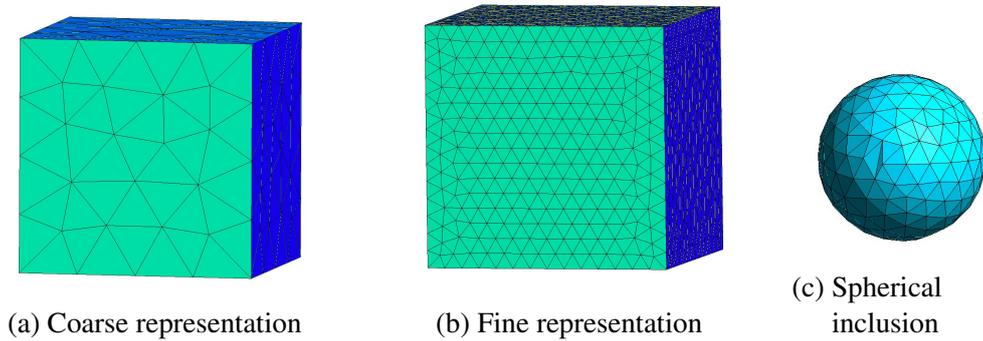


Figure 5: Example of one unit cube with different representations

There are two linear problems to consider: the Poisson equation, which is used to model thermal problems, and the linear elasticity equation. In the case of the Poisson equation, it represents the conduction of heat and assumes homogeneous boundary conditions. In some scenarios, there may be a contrast in the conductivity coefficient, denoted as α . The source term in this equation is simply set to 1. The linear elasticity equation, on the other hand, is used to model mechanical deformation. Here, a contrast in Young's modulus is introduced. The source term is a vector with all components equal to 1, representing a uniform force or load applied to the system.

5.3 Rate of communication study

To gain a better understanding of the impact of synchronization on communication and waiting time, a first study is proposed. The study focuses on a thermal problem set on a well-balanced case with 64 subdomains. The case has a heterogeneity ratio of 100, where the thermal diffusion coefficient in the spherical inclusion is 100 times lower than in the rest of the cube. The dimensions of the problems are provided in Table 1.

Different numbers of MPI processes are employed, which means that for fewer than 65 processes, a single MPI rank is responsible for handling multiple subdomains. The study analyzes the percentage of communication time in the total simulation time, considering a convergence tolerance of 10^{-7} for the norm of the residual.

Problem	Global	Local (One subdomain)
Number of nodes	1449	1858

Table 1: Mesh data

Table 2 illustrates that the synchronous approach with the Aitken accelerator outperforms the asynchronous approach in terms of overall speed. However, it is noteworthy that the proportion of time dedicated to communication significantly increases in the synchronous case, reaching up to 97%. In contrast, the asynchronous case experiences a more moderate increase, never exceeding 50%. As a result, the asynchronous

#ranks	Aitken #iter #time (s) [% communication time]	Asynchronous #iter. glob.[#loc. sol. [min, max]] #time (s) [% communication time]
9	25 & 11.72s [30%]	334[49, 54] & 22.4s[10%]
17	25 & 8.08s [80%]	182[56, 77] & 13.25s[10%]
33	25 & 4.53s [71%]	104[65, 124]& 8.13s[16%]
65	25 & 8.57s [97%]	105[81, 160] & 8.40s[46%]

Table 2: Analysis of the time spent in communication (64-subdomain thermal case).

approach proves to be faster in the 65-process case. Specifically, the transition from computations involving a single node (9 subdomains) to two nodes (17 subdomains) leads to a substantial increase in communication time for the synchronous case, while the asynchronous case remains unaffected.

5.4 Load imbalance study

In contrast to the previous studies that focused on nearly perfect load balance, this section examines the case of load imbalance to demonstrate the impact of synchronization and highlight the advantages of the asynchronous model.

The study is conducted using a 128-patch geometry (Figure 6) with dimensions provided in Table 3. The Young’s modulus in the spherical inclusion is 100 times lower than in the remaining part of the cube. The objective is to achieve a global residual norm below 10^{-7} .

Problem	Global	Local (One subdomain)
Nombre of nodes	2769	1254

Table 3: Mesh data

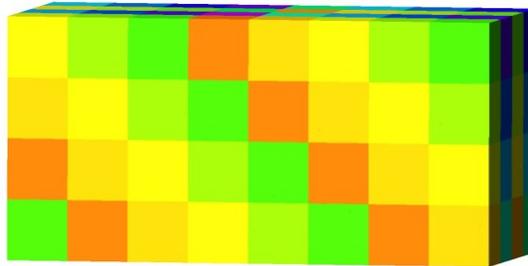


Figure 6: 128 subdomains with $\Omega^0 = \emptyset$

In this scenario, the objective is to assign varying computational loads to each core, assuming a limited number of cores available, specifically 65 cores. The task distribution is random, and two cases are considered:

1. Access to all machines in the cluster is granted, but only a specific number of cores are allocated per machine. As a result, the machines operate at a reduced capacity. However, the exchange of information becomes more extensive as it involves multiple machines.

Model	Aitken	Relaxed asynchronous
Time(s)	150	130.07
Async. #iter. glob.	24	239
Async. #loc. sol. [min, max]	-	[70 - 338]

In the asynchronous case, despite a large number of iterations performed both globally and locally, it still exhibits faster performance. The noticeable discrepancy between the minimum and maximum number of iterations, caused by load imbalance, is evident. This condition, which poses a disadvantage for the synchronous approach, highlights the adaptability of the asynchronous method to load imbalance situations.

2. Considering a limited number of machines that can accommodate up to 65 cores, the focus is on utilizing the machines intensively while keeping the exchange network less burdened.

Model	Aitken	Relaxed asynchronous
Time(s)	286.9	177.27s
Async. #iter. glob.	24	353
Async. #loc. sol. [min, max]	-	[69 - 417]

In this scenario, the overall computation times are longer compared to the previous situation. The high demand placed on the machines adversely affects the synchronous algorithm, resulting in slower performance. On the other hand, the asynchronous algorithm demonstrates significantly faster execution in this context.

6 Conclusion

We have introduced an asynchronous version of the non-intrusive global/local computation method and proposed its implementation using MPI RMA parallelization techniques. The global/local framework is advantageous for asynchronism as we can easily prove and control its convergence.

The presented performance results are promising: in terms of computation time, the asynchronous method approaches the formidable Aitken variant of synchronous computation.

Acknowledgements

This work was partly funded by the French National Research Agency as part of project ADOM, under grant number ANR-18-CE46-0008.

References

- [1] Nathan Cormier, Brian S. Smallwood, Gleen B. Sinclair, and G. Meda. "Aggressive submodelling of stress concentrations". *International Journal for Numerical Methods in Engineering*, 46(6):889–909, 1999.
- [2] Lisandro Dalcin and Yao-Lung L. Fang. "mpi4py: Status update after 12 years of development". *Computing in Science & Engineering*, 23(4):47–54, 2021.
- [3] Mickaël Duval, Jean-Charles Passieux, Michel Salaün, and Stéphane Guinard. "Non-intrusive coupling: recent advances and scalable nonlinear domain decomposition". *Archives of Computational Methods in Engineering*, 23(1):17–38, 2014.
- [4] Ahmed El Kerim, Pierre Gosselet, and Frédéric Magoulès. "Asynchronous global–local non-invasive coupling for linear elliptic problems". *Computer Methods in Applied Mechanics and Engineering*, 406:115910, 2023.
- [5] José C. Garay, Frédéric Magoulès, and Daniel B. Szyld. "Synchronous and asynchronous optimized Schwarz methods for Poisson's equation in rectangular domains". *ETNA - Electronic Transactions on Numerical Analysis*, 2022.
- [6] Guillaume Gbikpi-Benissan and Frédéric Magoulès. "Resilient asynchronous primal schur method. Applications of Mathematics", 2022.
- [7] Lionel Gendre, Olivier Allix, Pierre Gosselet, and François Comte. "Nonintrusive and exact global/local techniques for structural problems with local plasticity". *Computational Mechanics*, 44(2):233–245, 2009.
- [8] Christophe Geuzaine and Jean-François Remacle. "gmsh : a threedimensional nite element mesh generator with built-in pre- and postprocessing facilities". *International Journal for Numerical Methods in Engineering*, 0:1–24, 2009.
- [9] Christian Glusa, Erik G. Boman, Edmond Chow, Sivasankaran Rajamanickam, and Daniel B. Szyld. "Scalable asynchronous domain decomposition solvers". *SIAM Journal on Scientific Computing*, 42(6):384–409, 2020.
- [10] Pierre Gosselet, Maxime Blanchard, Olivier Allix, and Guillaume Guguin. "Non-invasive global-local coupling as a Schwarz domain decomposition method: acceleration and generalization". *Advanced Modeling and Simulation in Engineering Sciences*, 5(4), 2018.
- [11] Frédéric Hecht, Alexei Lozinski, and Olivier Pironneau. Numerical zoom and the Schwarz algorithm. In *Proceedings of the 18th conference on domain decomposition methods*, 2009.
- [12] FS Kelley. "Mesh requirements for the analysis of a stress concentration by the specified boundary displacement method". In *Proceedings of the Second International Computers in Engineering Conference*, ASME, pages 39–42, 1982.

- [13] Frédéric Magoulès and Cédric Venet. "Asynchronous iterative substructuring methods". *Mathematics and Computers in Simulation*, 145:34–49, 2018.
- [14] Yves Renard and Konstantinos Poullos. "Getfem: Automated fe modeling of multiphysics problems based on a generic weak form language". *Advances in Engineering Software*, 47:1–31, 2021.
- [15] Ichitaro Yamazaki, Edmond Chow, Aurelien Bouteiller, and Jack Dongarra. "Performance of asynchronous optimized schwarz with one-sided communication". *Parallel Computing*, 86:66 – 81, 2019.