Improving the Parallel Performance of Overlapping Schwarz Methods by Using a Smaller Energy Minimizing Coarse Space

Alexander Heinlein[1], Axel Klawonn[1], Oliver Rheinbach[2], and Olof Widlund[3]

1 Introduction

The GDSW preconditioner (Generalized Dryja, Smith, Widlund; see also [8]) is a two-level additive overlapping Schwarz preconditioner with exact local solvers (cf. [16]) using a coarse space constructed from energy-minimizing functions. It can be written in the form

$$M_{\mathrm{GDSW}}^{-1} = \Phi K_0^{-1} \Phi^T + \sum_{i=1}^{N} R_i^T \widetilde{K}_i^{-1} R_i, \tag{1}$$

where $K_0 = \Phi^T K \Phi$ is the coarse space matrix and the $\widetilde{K}_i = R_i K R_i^T$ represent the overlapping local problems; cf. [4]. The matrix Φ is the essential ingredient of the GDSW preconditioner. It is composed of coarse space functions which are discrete harmonic extensions from the interface into the interior degrees of freedom of nonoverlapping subdomains. The values on the interface are restrictions of the elements of the nullspace of the operator to the edges, vertices, and faces of the decomposition. Therefore, for a scalar elliptic problem, the coarse basis functions form a partition of unity on all subdomains which do not touch the Dirichlet boundary.

For $\Omega \subset \mathbb{R}^2$ being decomposed into John subdomains, the condition number of the GDSW preconditioner is bounded by

$$\kappa\left(M_{\mathrm{GDSW}}^{-1} A\right) \leq C \left(1 + \frac{H}{\delta}\right) \left(1 + \log\left(\frac{H}{h}\right)\right)^2, \tag{2}$$

[1]Mathematisches Institut, Universität zu Köln, Weyertal 86-90, 50931 Köln, Germany. e-mail: {alexander.heinlein,axel.klawonn}@uni-koeln.de
[2]Institut für Numerische Mathematik und Optimierung, Fakultät für Mathematik und Informatik, Technische Universität Bergakademie Freiberg, Akademiestr. 6, 09596 Freiberg, Germany. e-mail: oliver.rheinbach@math.tu-freiberg.de
[3]Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, New York 10012, USA. e-mail: widlund@cims.nyu.edu

cf. [3, 4]. Here, H is the size of a subdomain, h the size of a finite element, and δ is the overlap.

GDSW-type preconditioners have succesfully been developed for almost incompressible elasticty, e.g., [5] and problems in $H(\mathrm{curl})$ [2]. An efficient parallel implementation of the GDSW preconditioner based on Trilinos [13] was recently introduced by the authors in [10]. Although the preconditioner can use geometric information, a focus in [10] was to make use of the Trilinos infrastructure to construct the preconditioner algebraically from the assembled sparse stiffness matrix.

A coarse space for Overlapping Schwarz methods in two dimensions related to but smaller than the standard GDSW coarse space has been considered in [6]. Following [7], in this paper, we consider two reduced versions of the GDSW coarse space in three dimensions denoted by *Option 1* and *Option 2.2* in [7]. These spaces are also smaller than the standard GDSW coarse space. In the following, we will denote this reduced GDSW coarse space as RGDSW. Our reduced coarse spaces have a relation to discretization methods such as Multiscale Finite Element Methods (MsFEM), which also use harmonic extensions; see, e.g., [14, 17].

2 A Reduced GDSW Coarse Space

We have implemented the RGDSW coarse space in our parallel preconditioner [10] since, among the proposed options in [7], it is the most algebraic. As in the standard version, we introduce coarse basis functions that form a partition of unity on the interface of the domain decomposition. Again, we extend the values on the interface as discrete harmonic functions into the interior of the nonoverlapping subdomains.

Let $\mathscr{S}_n$ be the index set of all subdomains which share the node n. A node n_i is called an ancestor of n_j if $\mathscr{S}_{n_j} \subset \mathscr{S}_{n_i}$. If no other node is an ancestor of a node n_j, it is called a coarse node. Using this definition, we can construct for each coarse node n_i a coarse basis function φ_i such that

$$\sum_{n_i \text{ coarse node}} \varphi_i = 1$$

on all subdomains which do not touch the Dirichlet boundary. A coarse basis function φ_i is constructed as follows:

$$\varphi_i(n) = \begin{cases} \frac{1}{|\mathscr{C}_n|} & \text{if } n_i \in \mathscr{C}_n, \\ 0 & \text{otherwise,} \end{cases}$$

with $\mathscr{C}_n$ being the set of all ancestors of the interface node n; cf. Fig. 1 (top). On the Dirichlet boundary, we set all coarse basis functions to zero.

Another option to define a reduced coarse space, using basis function based on an inverse distance weighting approach, has been introduced in [7, eq. (5)]. In particular, according to [7, eq. (5)], the values of the coarse basis function on the interface are chosen as

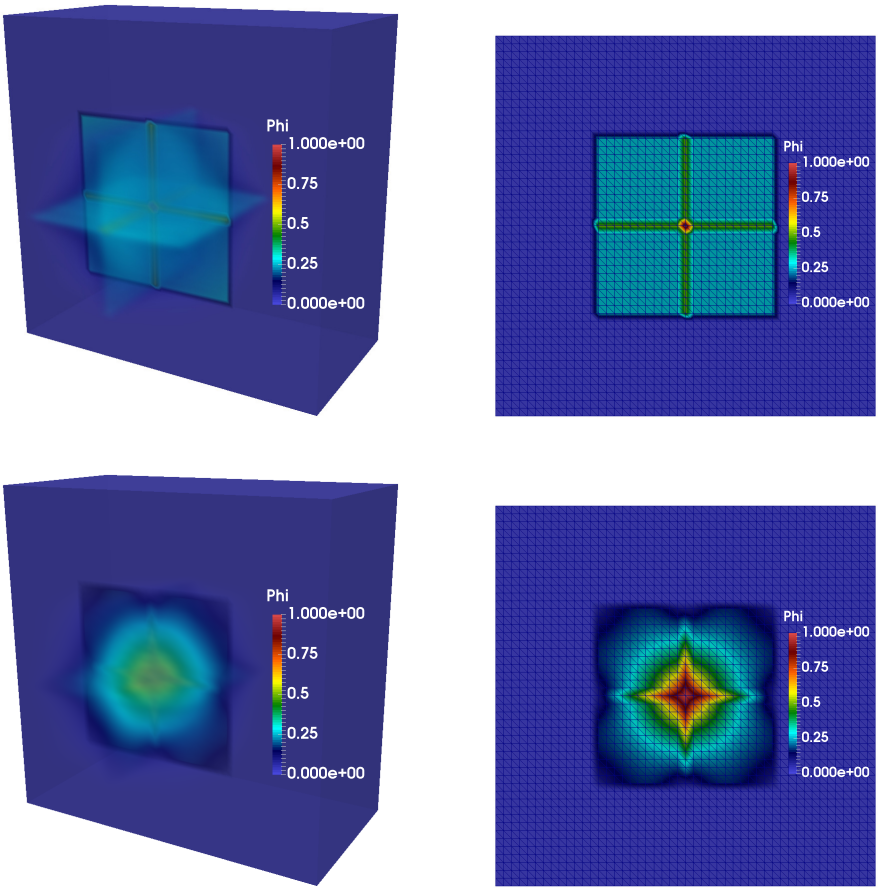


Fig. 1 Plot of the coarse basis function corresponding to the center node for the reduced GDSW coarse spaces, denoted *Option 1* (top) and *Option 2.2* (bottom) in [7]. Here, we assume the structured decomposition of a cube into 4x4x4 cubic subdomains.

$$\varphi_i(n) = \begin{cases} \frac{1/d_i(n)}{1/d_1(n)+1/d_2(n)+1/d_3(n)+1/d_4(n)} & \text{if } n_i \in \mathscr{C}_n, i \in \{1,2,3,4\} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

for components with four coarse nodes. Here, $d_i(n), i = 1,\ldots,4$ is the distance to the coarse node n_i. For components with any number of coarse nodes, we set

$$\varphi_i(n) = \begin{cases} \frac{1/d_i(n)}{\sum_{n_j \in \mathscr{C}_n} 1/d_j(n)} & \text{if } n_i \in \mathscr{C}_n, \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

on the interface; cf. Fig. 1 (bottom). This construction is denoted as *Option 2.2* in [7]. As we will observe in section 4, this choice leads to a better convergence,

	Dimension of the Coarse Space		
# subdomains	Standard GDSW	RGDSW (Option 1&2.2)	Reduction
2^3	19	1	94.74 %
4^3	279	27	90.32 %
8^3	2 863	343	88.02 %
16^3	25 695	3 375	86.87 %
24^3	89 999	12 167	86.48 %
32^3	217 279	29 791	86.29 %
40^3	429 039	59 319	86.17 %
80^3	3 507 679	493 039	85.94 %
100^3	6 880 599	970 299	85.90 %
$1\,000^3$	$7.0 \cdot 10^9$	$1.0 \cdot 10^9$	85.73 %
$10\,000^3$	$7.0 \cdot 10^{12}$	$1.0 \cdot 10^{12}$	85.72 %

Table 1 Dimension of the coarse spaces and the reduction due to the use of the reduced coarse spaces in percent. We use one subdomain for each processor core.

	scalar elliptic		compressible linear elasticity
	face paths	edge paths	face paths
Option 1	$\alpha = 1$	$\alpha = 2$	$\alpha = 1$
Option 2.2	$\alpha = 0$	$\alpha = 1$	$\alpha = 0$

Table 2 Values of α in the condition number bound (5). For the definition of quasi-monotone paths, see [7].

however, it relies on additional geometric information to allow for the computation of the distance between interface nodes and the relevant coarse nodes. Therefore, it can be regarded as less algebraic compared to *Option 1*.

The advantage of these two reduced GDSW coarse problems over the classical GDSW coarse problem is their smaller size; cf. Fig. 2. Indeed, in 3D, for structured decompositions, they are smaller by more than 85 percent; cf. Table 1. This can be a significant advantage when striving for better parallel scalability on larger supercomputers.

For the reduced coarse spaces, for scalar elliptic problems in 3D as well as elasticity, the condition number of the preconditioned operator satisfies

$$\kappa(M_{\mathrm{RGDSW}}^{-1}A) \leq C \left(1 + \frac{H}{\delta}\right) \left(1 + \log\left(\frac{H}{h}\right)\right)^{\alpha}, \tag{5}$$

where α is given in Table 2; cf. [7] and also see Fig. 4.

3 Implementation

Our parallel implementation of the GDSW preconditioner and its more recent version with a reduced coarse space size (here denoted by RGDSW) is based on the implementation described in [10, 9, 12, 11]. We use Trilinos version 12.0; cf. [13].

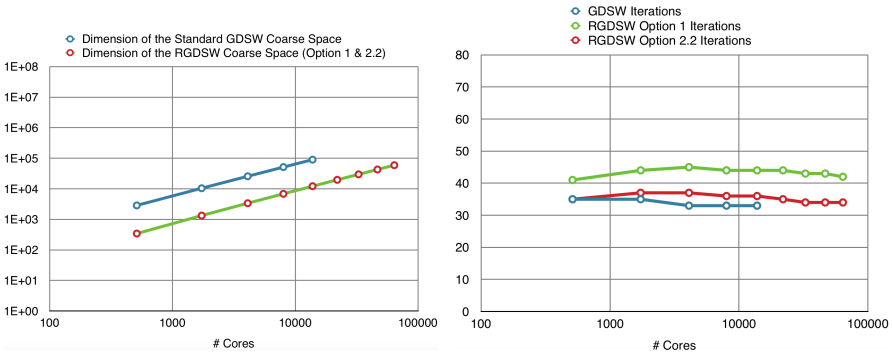


Fig. 2 We compare for a Laplace model problem in three dimensions: dimension of the coarse spaces (left) and corresponding numbers of iterations for the standard and the reduced GDSW coarse space (right); we use $H/h = 30$ and two layers of overlap. Computations run on the JUQUEEN supercomputer.

In our experiments presented here, for simplicity, we use a structured decomposition of our cubic computational domain into cubic subdomains. The overlapping subdomain problems and the coarse problem are solved using Mumps 4.10.0 (cf. [1]) in sequential mode. On the JUQUEEN BG/Q supercomputer, we use the IBM XL compilers 12.1 and the ESSL 5.1 when compiling Trilinos and the GDSW preconditioner. On the magnitUDE supercomputer at Universität Duisburg-Essen, we use the Intel compiler and the Intel MKL 2017.1.132.

4 Numerical Results

Based on the infrastructure given by our parallel implementation [10], we compare the reduced coarse space (denoted by RGDSW) to the standard coarse space (denoted by GDSW) for a scalar elliptic problem in 3D. Our numerical results in Fig. 2, and 3 show that the smaller dimension of the new coarse spaces *Option 1* and *Option 2.2* proposed in [7] indeed help to increase the parallel efficiency of the method significantly; see also Tables 3 and 4. By "Total Time", we denote the total time to solution including the assembly of the problem. The "Setup Time" includes the assembly of the problem and the setup of the preconditioner. This includes the factorization of the subdomain matrices. Finally, "Solver Time" only denotes the time spent in the GMRES iteration. The number of Krylov iterations for the new methods increases but only slightly in comparison with the standard GDSW preconditioner (cf. Fig. 2, right), as also demonstrated in [7]; the increase is too small to be reflected in the computation times. Indeed, as shown in Fig. 3, the total time to solution is always smaller for the new coarse spaces.

Acknowledgements
This work was supported in part by the German Research Foundation (DFG) through the Priority

	GDSW					RGDSW Option 1					RGDSW Option 2.2				
# cores = # subdomains	# it.	Setup Time	Solver Time	Total Time	Effic.	# it.	Setup Time	Solver Time	Total Time	Effic.	# it.	Setup Time	Solver Time	Total Time	Effic.
512	35	45.94 s	16.75 s	62.69 s	96%	41	43.84 s	18.82 s	62.66 s	96%	35	43.94 s	15.97 s	59.91 s	100%
1 728	35	47.47 s	19.49 s	67.23 s	89%	44	44.96 s	20.98 s	65.94 s	91%	37	45.15 s	17.55 s	62.70 s	96%
4 096	33	52.46 s	23.49 s	75.95 s	79%	45	45.66 s	22.56 s	68.22 s	88%	37	45.78 s	18.35 s	64.13 s	93%
8 000	33	63.55 s	31.73 s	92.28 s	65%	44	48.91 s	23.89 s	72.80 s	83%	36	49.35 s	19.35 s	68.70 s	87%
13 824	33	81.61 s	46.77 s	128.38 s	47%	44	51.08 s	26.28 s	77.36 s	77%	36	50.92 s	21.30 s	72.22 s	83%
21 952	Out of memory					44	53.98 s	30.26 s	84.24 s	71%	35	53.88 s	23.89 s	77.77 s	77%
32 768	Out of memory					43	57.71 s	34.27 s	91.98 s	65%	34	57.93 s	27.12 s	85.05 s	70%
46 656	Out of memory					43	64.48 s	41.60 s	106.08 s	56%	34	65.41 s	32.59 s	98.00 s	61%
64 000	Out of memory					42	74.18 s	48.17 s	122.35 s	49%	34	76.26 s	38.90 s	115.16 s	52%

Table 3 Detailed timers corresponding to Fig. 3 (left). Base line for the efficiency ("Effic.") is the fastest "Total Time" for 512 processor cores.

	GDSW				RGDSW Option 1				RGDSW Option 2.2			
# cores =	First Level		Second Level		First Level		Second Level		First Level		Second Level	
# subdomains	Setup	Apply	Setup	Apply	Setup	Apply	Setup	Apply	Setup	Apply	Setup	Apply
512	13.54 s	11.36 s	12.95 s	0.30 s	14.69 s	13.19 s	9.78 s	2.52 s	14.68 s	11.36 s	9.81 s	2.15 s
1 728	13.71 s	11.57 s	14.51 s	0.55 s	15.45 s	14.44 s	10.01 s	3.12 s	15.61 s	12.20 s	10.04 s	2.64 s
4 096	14.71 s	10.97 s	18.22 s	10.50 s	15.79 s	14.89 s	10.34 s	4.24 s	15.90 s	12.25 s	10.35 s	3.48 s
8 000	16.97 s	11.16 s	26.62 s	18.71 s	18.14 s	14.81 s	11.06 s	5.74 s	18.37 s	12.04 s	11.13 s	4.74 s
13 824	17.21 s	11.15 s	44.58 s	34.22 s	18.53 s	14.77 s	12.71 s	8.14 s	18.40 s	12.13 s	12.60 s	6.69 s
21 952	Out of memory				18.61 s	14.98 s	15.40 s	12.05 s	18.57 s	11.97 s	15.35 s	9.62 s
32 768	Out of memory				18.48 s	14.39 s	19.39 s	16.89 s	18.72 s	11.42 s	19.43 s	13.52 s
46 656	Out of memory				18.70 s	14.66 s	25.56 s	23.98 s	18.73 s	11.66 s	26.61 s	18.92 s
64 000	Out of memory				18.96 s	14.36 s	34.94 s	31.16 s	18.92 s	11.71 s	37.13 s	25.34 s

Table 4 Detailed timers corresponding to Fig. 3 (right).

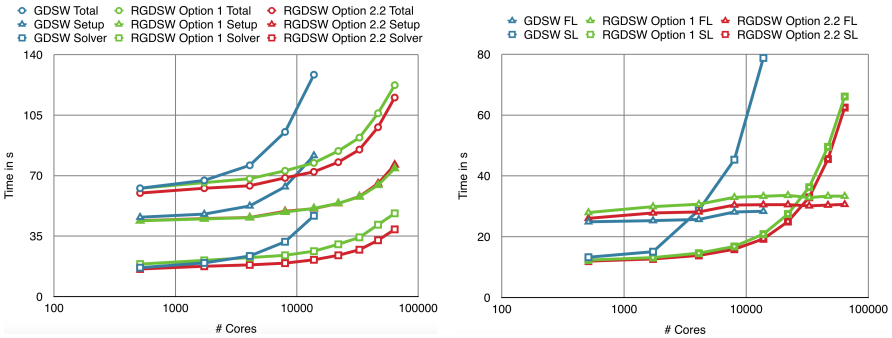
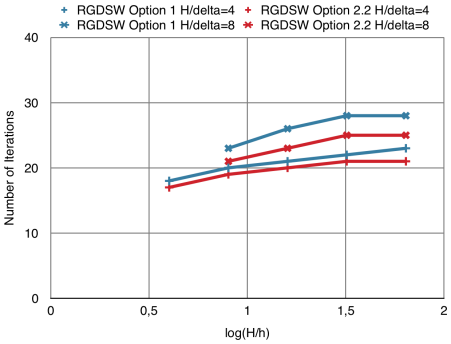


Fig. 3 Detailed times for the computations of a Laplace model problem in three dimensions using the standard GDSW coarse space and the reduced GDSW coarse space; we use $H/h = 30$ and two layers of overlap. Computations run on the JUQUEEN supercomputer.



H/h	Option 1		Option 2.2	
	$H/\delta = 4$	$H/\delta = 8$	$H/\delta = 4$	$H/\delta = 8$
4	18	-	17	-
8	20	23	19	21
16	21	26	20	23
32	22	28	21	25
64	23	28	21	25

Fig. 4 Numbers of iterations versus $\log(H/h)$ for the reduced GDSW coarse space and $1/H = 4$. Computations run on the magnitUDE supercomputer.

Programme 1648 “Software for Exascale Computing” (**SPPEXA**) under grants KL 2094/4-2 and RH 122/3-2. The work of the fourth author was supported by the National Science Foundation Grant DMS-1522736

The authors gratefully acknowledge the computing time granted by the Center for Computational Sciences and Simulation (CCSS) at Universität Duisburg-Essen and provided on the supercomputer magnitUDE (DFG grants INST 20876/209-1 FUGG, INST 20876/243-1 FUGG) at Zentrum für Informations- und Mediendienste (ZIM).

The authors also gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for providing computing time on the GCS Supercomputer JUQUEEN BG/Q supercomputer ([15]) at JSC Jülich. GCS is the alliance of the three national supercomputing centres HLRS (Universität Stuttgart), JSC (Forschungszentrum Jülich), and LRZ (Bayerische Akademie der Wissenschaften), funded by the German Federal Ministry of Education and Research (BMBF) and the German State Ministries for Research of Baden-Württemberg (MWK), Bayern (StMWFK) and Nordrhein-Westfalen (MIWF).

References

1. Patrick R. Amestoy, Iain S. Duff, Jean-Yves L'Excellent, and Jacko Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. SIAM J. Matrix Anal. Appl., 23(1):15–41, January 2001.
2. Juan G. Calvo. A two-level overlapping Schwarz method for $H(\mathrm{curl})$ in two dimensions with irregular subdomains. Electron. Trans. Numer. Anal., 44:497–521, 2015.
3. Clark R. Dohrmann, Axel Klawonn, and Olof B. Widlund. Domain decomposition for less regular subdomains: overlapping Schwarz in two dimensions. SIAM J. Numer. Anal., 46(4):2153–2168, 2008.
4. Clark R. Dohrmann, Axel Klawonn, and Olof B. Widlund. A family of energy minimizing coarse spaces for overlapping Schwarz preconditioners. In Domain decomposition methods in science and engineering XVII, volume 60 of Lect. Notes Comput. Sci. Eng., pages 247–254. Springer, Berlin, 2008.
5. Clark R. Dohrmann and Olof B. Widlund. Hybrid domain decomposition algorithms for compressible and almost incompressible elasticity. Internat. J. Numer. Methods Engrg., 82(2):157–183, 2010.
6. Clark R. Dohrmann and Olof B. Widlund. An alternative coarse space for irregular subdomains and an overlapping Schwarz algorithm for scalar elliptic problems in the plane. SIAM J. Numer. Anal., 50(5):2522–2537, 2012.
7. Clark R. Dohrmann and Olof B. Widlund. On the design of small coarse spaces for domain decomposition algorithms. SIAM Journal on Scientific Computing, 39(4):A1466–A1488, 2017.
8. Maksymilian Dryja, Barry F. Smith, and Olof B. Widlund. Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions. SIAM J. Numer. Anal., 31(6):1662–1694, December 1994.
9. Alexander Heinlein. Parallel Overlapping Schwarz Preconditioners and Multiscale Discretizations with Applications to Fluid-Structure Interaction and Highly Heterogeneous Problems. PhD thesis, Universität zu Köln, Germany, 2016.
10. Alexander Heinlein, Axel Klawonn, and Oliver Rheinbach. A parallel implementation of a two-level overlapping Schwarz method with energy-minimizing coarse space based on Trilinos. SIAM J. Sci. Comput., 38(6):C713–C747, 2016.
11. Alexander Heinlein, Axel Klawonn, and Oliver Rheinbach. Parallel two-level overlapping Schwarz methods in fluid-structure interaction. In Bülent Karasözen, Murat Manguoğlu, Münevver Tezer-Sezgin, Serdar Göktepe, and Ömür Uğur, editors, Numerical Mathematics and Advanced Applications ENUMATH 2015, pages 521–530, Cham, 2016. Springer International Publishing. Also Preprint 2015-15 at http://tu-freiberg.de/fakult1/forschung/preprints.
12. Alexander Heinlein, Axel Klawonn, and Oliver Rheinbach. Parallel overlapping Schwarz with an energy-minimizing coarse space. In Chang-Ock Lee, Xiao-Chuan Cai, David E. Keyes, Hyea Hyun Kim, Axel Klawonn, Eun-Jae Park, and Olof B. Widlund, editors, Domain Decomposition Methods in Science and Engineering XXIII, volume 116 of Lect. Notes Comput. Sci. Eng., pages 353–360, Cham, 2017. Springer International Publishing.
13. Michael A Heroux, Roscoe A Bartlett, Vicki E Howle, Robert J Hoekstra, Jonathan J Hu, Tamara G Kolda, Richard B Lehoucq, Kevin R Long, Roger P Pawlowski, Eric T Phipps, Andrew G Salinger, Heidi K Thornquist, Ray S Tuminaro, James M Willenbring, Alan Williams, and Kendall S Stanley. An overview of the Trilinos project. ACM Trans. Math. Softw., 31(3):397–423, 2005.
14. Thomas Y. Hou and Xiao-Hui Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. J. Comput. Phys., 134(1):169 – 189, 1997.
15. Michael Stephan and Jutta Docter. JUQUEEN: IBM Blue Gene/Q Supercomputer System at the Jülich Supercomputing Centre. Journal of large-scale research facilities, 1:A1, 2015.
16. Andrea Toselli and Olof Widlund. Domain decomposition methods—algorithms and theory, volume 34 of Springer Series in Computational Mathematics. Springer-Verlag, Berlin, 2005.
17. Jan Van lent, Robert Scheichl, and Ivan G. Graham. Energy-minimizing coarse spaces for two-level Schwarz methods for multiscale PDEs. Numer. Linear Algebra Appl., 16(10):775–799, 2009.