MOX-Report No. 82/2022

A PETSc Parallel Implementation of Substructured One- and Two-level Schwarz Methods

Ciaramella, G.; Gander, M.; Van Criekingen, S.; Vanzan, T.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it https://mox.polimi.it

A PETSc Parallel Implementation of Substructured One- and Two-level Schwarz Methods

Gabriele Ciaramella, Martin J. Gander, Serge Van Criekingen and Tommaso Vanzan

1 Introduction

Substructured Schwarz methods are interpretations of volume Schwarz methods as algorithms on interface variables. We here consider the substructured version of the Parallel Schwarz Method (PSM) as presented in [GH12, p.24] and recently extended to a two-level (i.e. coarse-corrected) framework in [CV22b] and [CV22a], using a geometric and spectral approach for the definition of the coarse space.

The expected gain of substructured methods relies on the smaller size of the resulting problems, notably with Krylov-type acceleration techniques when the dimension of the subspace of approximants becomes large [Saa03]. Moreover, with the coarse functions being defined only on interfaces, the size of the resulting coarse matrices may be smaller than their volume counterparts, giving a competitive advantage to two-level substructured methods. However, note that the local subdomain solves appearing in substructured methods (already at one-level) require an exact solution to guarantee convergence [CV22b].

We here present an implementation of the substructured PSM based on the PETSc (Portable, Extensible Toolkit for Scientific Computation) linear algebra package [BAA+22a, BAA+22b, BGMS97], successively considering one- (section 2) and two- (section 3) level methods. For the latter, four coarse spaces are introduced, all based on a geometric approach. Note that, at this time, spectral approaches still

Gabriele Ciaramella
Politecnico di Milano, e-mail: gabriele.ciaramella@polimi.it

Martin J. Gander
University of Geneva, e-mail: martin.gander@unige.ch

Serge Van Criekingen
CNRS/IDRIS and Université Paris-Saclay, UVSQ, CNRS, CEA, Maison de la Simulation, e-mail: serge.van.criekingen@idris.fr

Tommaso Vanzan
CSQI Chair, EPFL Lausanne, e-mail: tommaso.vanzan@epfl.ch

require further investigations and are therefore not presented here (- the reason being that the eigenvectors on which spectral coarse spaces are based in [CV22a] are in general complex and in turn necessitate a PETSc installation adapted to complex arithmetic, which has a negative influence on the resulting computational times). We compare our substructured numerical results to the RAS volume method [CS99] for which a two-level PETSc implementation has already been presented by the authors [GV19, GV21] using various coarse spaces.

2 The one-level substructured formulation

We consider the system $Au = f$ for the Laplace problem with Dirichlet boundary conditions discretized with finite differences. We first derive the substructured system

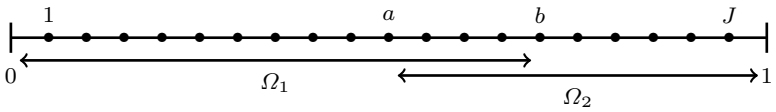


Fig. 1: Two subdomain decomposition in the 1-D case.

for the 1-D case, namely the [0,1] interval subdivided into $J+1$ mesh cells of size h as depicted in Fig. 1 in the two-subdomain case. Following [GH12], we decompose $A \subset \mathbb{R}^{(J-1)\times(J-1)}$ in two different ways as

$$A = \begin{pmatrix} A_1 & B_1 \\ C_1 & D_1 \end{pmatrix} = \begin{pmatrix} D_2 & C_2 \\ B_2 & A_2 \end{pmatrix}, \tag{1}$$

where $A_1 \subset \mathbb{R}^{(b-1)\times(b-1)}$ and $A_2 \subset \mathbb{R}^{(J-a)\times(J-a)}$. Our starting point is the discretized Parallel Schwarz Method (PSM) for $Au = f$ which reads

$$A_1 u_1^{n+1} = f_1 - \tilde{B}_1 u_2^n, \tag{2}$$
$$A_2 u_2^{n+1} = f_2 - \tilde{B}_2 u_1^n, \tag{3}$$

where $\tilde{B}_1 = [0_{b-1,d-1} B_1]$ and $\tilde{B}_2 = [B_2 0_{J-a,d-1}]$ (with $d = b - a$ the overlap) are extensions by zeros of the B_1 and B_2 matrices of (1) such that

$$\tilde{B}_1 u_2 = (0, ..., 0, -\frac{1}{h^2}(u_2)_b) \subset \mathbf{R}^{b-1},$$
$$\tilde{B}_2 u_1 = (-\frac{1}{h^2}(u_1)_a, 0, \ldots, 0) \subset \mathbf{R}^{J-a}.$$

Thus $\tilde{B}_1$ maps a vector defined on Ω_2 into one defined on Ω_1, extended by zero out of Ω_2 (and similarly for $\tilde{B}_2$). We introduce the trace operators

$$G_1 : (v_1, \ldots, v_a, \ldots, v_{b-1}) \to v_a,$$
$$G_2 : (v_{a+1}, \ldots, v_b, \ldots, v_J) \to v_b,$$

such that $G_1 u_1 = (u_1)_a$ and $G_2 u_2 = (u_2)_b$, as well as the extension by zero operators

$$E_1 : v_b \to (0, \ldots, 0, v_b) \subset \mathbf{R}^{b-1},$$
$$E_2 : v_a \to (v_a, 0, \ldots, 0) \subset \mathbf{R}^{J-a},$$

such that $\tilde{B}_1 u_2 = -\frac{1}{h^2} E_1 (u_2)_b$ and $\tilde{B}_2 u_1 = -\frac{1}{h^2} E_2 (u_1)_a$. Applying the trace operators to the PSM system (2)-(3) then yields

$$(u_1^{n+1})_a = \frac{1}{h^2} G_1 A_1^{-1} E_1 (u_2^n)_b + G_1 A_1^{-1} f_1,$$
$$(u_2^{n+1})_b = \frac{1}{h^2} G_2 A_2^{-1} E_2 (u_1^n)_a + G_2 A_2^{-1} f_2.$$

Defining interface unknowns $g^T = (g_1, g_2) = ((u_1)_a, (u_2)_b)$, this is the block Jacobi method applied to the substructured system

$$Tg = f^g, \tag{4}$$

where

$$T = \begin{pmatrix} I & -\frac{1}{h^2} G_1 A_1^{-1} E_1 \\ -\frac{1}{h^2} G_2 A_2^{-1} E_2 & I \end{pmatrix} \text{ and } f^g = \begin{pmatrix} G_1 A_1^{-1} f_1 \\ G_2 A_2^{-1} f_2 \end{pmatrix}. \tag{5}$$

This system can also be solved using a Krylov method (`GMRES` here).

From a parallel data transfer point of view, in the two-subdomain case of Fig.1, we have that Ω_1 sends u_a to Ω_2, while Ω_2 sends u_b to Ω_1. In the three subdomain case (Fig.2), two trace operators are necessary for the central subdomain Ω_2, ex-

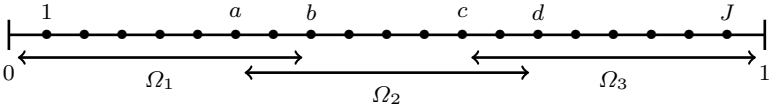
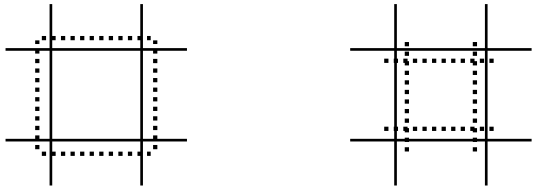


Fig. 2: Three subdomain decomposition in the 1-D case.

tracting respectively u_b and u_c and sending them to Ω_1 and Ω_3, again respectively. Meanwhile, subdomain Ω_2 receives u_a from Ω_1 and u_d from Ω_3.

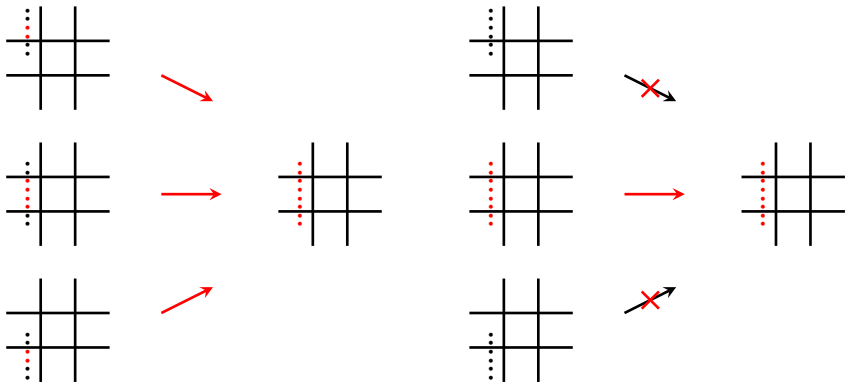
In 2-D, for a typical non-boundary subdomain, data exchange consists in receiving data on a square skeleton obtained by extending the domain by the size of the overlap (Fig. 3a) and sending local data from four "portions" within the domain, at overlap distance from the interface (Fig. 3b). Furthermore, in 2D a partition of unity is



(a) (b)

Fig. 3: Dotted are the substructure values to be received (a) or sent (b) by the central subdomain.

required and we investigated two data exchange options, with or without transfers from diagonal neighbours, as illustrated in Fig. 4 for the left-to-right data exchange.



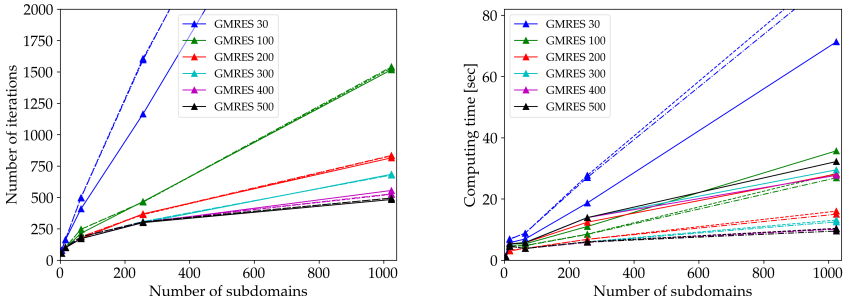
(a) (b)

Fig. 4: Schematic representation of left-to-right data exchange with (a) or without (b) transfers from diagonal neighbours. The transferred data are in red.

The T substructured system matrix defined in (5) is implemented matrix-free in our `PETSc` implementation, using the `MatCreateShell` and `MatShellSetOperation` tools. Each multiplication by T implies data transfer (with or without diagonal transfers), extension by zero (E_i), *exact* solve by the local matrices A_i (direct solver with LU decomposition computed only once) and taking the trace in the subdomain (G_i). To solve the substructured system (4), we apply `GMRES` without preconditioner, since this system is in fact already preconditioned by the Schwarz method.

We compare our substructured method to the (volume) RAS method [CS99] (implemented in `PETSc` as `PCASM`) on a weak scaling experiment for the 2-D Laplace problem on the unit square with 5-point finite difference scheme, using square decompositions into 2×2 to 32×32 subdomains (one processor per subdomain)

and a 256×256 fine mesh within each subdomain. Several observations can be



(a) Iteration count (b) Timing results

Fig. 5: Weak scaling results for the 2×2 to 32×32 square decompositions, using various `GMRES` restart parameters. Volume methods (solid lines) and substructured methods with (dashed lines) or without (dashdot lines) diagonal transfer are used.

made from the results displayed in Fig.5. First, there is virtually no difference in the number of iterations with or without diagonal transfers, so that the extra cost of the diagonal transfers is not compensated by a decrease in iterations. Consequently, we stick to the no diagonal transfer option in the remainder of our study. Second, when looking at computational times, the optimal `GMRES` restart parameter for the substructured method (here 500, which in fact means no restart since a bit less than 500 iterations are then performed) appears to be larger than for the volume method (here 400 with 200 being very close), the smaller size of the substructured problem thus making a larger Krylov space profitable. Third, and most importantly, at high restart parameters and in particular at the optimal one, substructured methods yield better timing performances than volume methods. This appears to be due to the smaller size of the substructured systems since the number of iterations with both methods is similar.

3 Two-level substructured methods

We model our two-level substructured method on the (volume) two-level RAS methods ("RAS2") developped in [GV19], namely

$$u^{n+1/2} = u^n + \sum_{j=1}^{J} \tilde{R}_j^T \, A_j^{-1} \, R_j \, (f - Au^n),$$
$$u^{n+1} = u^{n+1/2} + R_c^T \, A_c^{-1} \, R_c \, (f - Au^{n+1/2}),$$

where R_j are restriction operators to the (possibly overlapping) Ω_j subdomains decomposing the global domain Ω, $\tilde{R}_j$ are the equivalents for a non-overlapping decomposition of Ω into $\tilde{\Omega}_j$, and R_c is the restriction operator to the coarse space. Moreover, we have defined the local matrices as $A_j = R_j\, A\, R_j^T$ and the coarse matrix as $A_c = R_c\, A\, R_c^T$. In our `PETSc` implementation, this is implemented as a multiplicative composition (`PCCOMPOSITE`) of RAS (`PCASM`) with a hand-made second-level correction (`PCSHELL` framework). The coarse solve A_c^{-1} is performed with the direct solver `MUMPS` with agglomeration of the coarse unknowns. A `GMRES` acceleration can be applied to the (full) iteration. The volume RAS2 coarse correction chosen here is `Q1`, a coarse space made out of linear functions with, in 2-D, four coarse nodes placed around each cross-point [DGL$^+$12, GHS14, GV19].

We proceed similarly for our two-level substructured implementation: for the system $Tg = f^g$, our two-level method reads

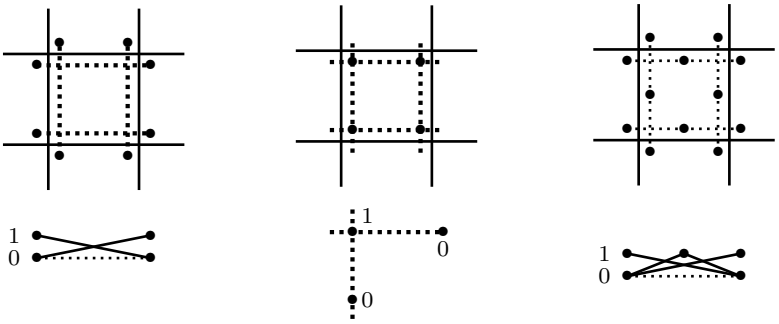
$$g^{n+1/2} = g^n + (f^g - Tg^n), \tag{6}$$

$$g^{n+1} = g^{n+1/2} + R_c^T\, T_c^{-1}\, R_c\, (f^g - Tg^{n+1/2}), \tag{7}$$

where R_c is again the restriction operator to the coarse space and $T_c = R_c T R_c^T$ is the coarse matrix. In `PETSc`, we proceed again with a multiplicative composition of, this time, `PCNONE` (no preconditioner) with a hand-made second-level correction. The T_c matrix is built once and for all at the begining of the calculation, as well as its LU decomposition using `MUMPS`. Here also `GMRES` can be applied to the full iteration.

Our substructured coarse space functions will be defined exclusively on the interfaces, more precisely, for each of them, on the four substructure portions of a typical non-boundary subdomain (Fig. 3b). We here consider four geometric substructured coarse spaces, namely `Constant` with one constant coarse function per portion (so 4 functions for a non-boundary subdomain), `Linear` (Fig. 6a) with two linear coarse functions per portion (so 8 coarse points and functions for a non-boundary subdomain), `Linear4` (Fig. 6b) with four linear functions (and as many coarse points) for a non-boundary subdomain (- this space can be seen as the volume `Q1` coarse space restricted to the substructure) and `Enriched` (Fig. 6c) with three linear coarse functions per portion (so 12 coarse points and functions for a non-boundary subdomain). Thus, for an $N \times N$ decomposition, the coarse space sizes asymptotically behave as $4N^2$ with `Constant` and `Linear4`, $8N^2$ with `Linear` and $12N^2$ with `Enriched`.

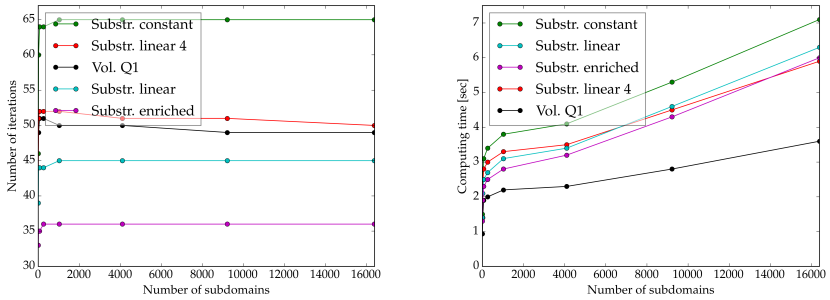
Fig. 7 displays iteration count and computational (wall-clock) times for the weak scaling experiment described above using the two-level volume and substructured methods, with square decompositions up to 128×128 subdomains (- the solving time results, not shown here, exhibit a very similar behavior). There is no `GMRES` restart performed here. We observe that all our two-level methods achieve scalability in terms of number of iterations. Scalability in terms of computational times is quite well achieved even though not perfectly, with performances slightly below the two-level volume `Q1` method. Various implementation optimizations might in the future improve these already encouraging results. For example, the two-level iteration (6)-(7) requires the computation of two actions of the operator T, which correspond to two steps of the one-level PSM. Notably, one of the two actions can be eliminated



(for the upper-left coarse point, similarly for the three others.)

(a) `Linear` (b) `Linear` 4 (c) `Enriched`

Fig. 6: Schematic view of substrucutred coarse space options, with coarse point positions (above) and coarse function sketch (below).



(a) Iteration count (b) Wall-clock times

Fig. 7: Two-level numerical results up to 16,384 processors.

using, e.g., the strategy proposed in [CV22b, CV22a]. However, this requires extra work on the PETSc implementation. Note also the particularly interesting behavior of the `Linear4` coarse space, yielding less iterations than the `Constant` one with asymptotically the same number of coarse functions.

4 Conclusions

A `PETSc` implementation of the substructured one-and two-level PSM has been presented. Our one-level results show that the smaller size of the substructured system compared to the volume one makes the use of larger Krylov spaces (i.e.,

using larger `GMRES` restart parameters, or no restart at all) profitable, resulting in better computational times. Furthermore, we introduced four new substructured geometric coarse spaces defined exclusively on the interfaces and our numerical results up to 16,384 cores show that the resulting two-level methods achieve a perfect scalability in terms of number of iterations and a very decent scalability in terms of computational solving and wall-clock times.

Acknowledgements This work was performed using HPC resources from GENCI-IDRIS.

References

[BAA+22a] S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. Brown, P. Brune, K. Buschelman, E. M. Constantinescu, L. Dalcin, A. Dener, V. Eijkhout, J. Faibussowitsch, W. D. Gropp, V. Hapla, T. Isaac, P. Jolivet, D. Karpeev, D. Kaushik, M. G. Knepley, F. Kong, S. Kruger, D. A. May, L. Curfman McInnes, R. Tran Mills, L. Mitchell, T. Munson, J. E. Roman, K. Rupp, P. Sanan, J. Sarich, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, and J. Zhang. PETSc Web page. https://petsc.org/, 2022.

[BAA+22b] S. Balay, S. Abhyankar, M. F. Adams, S. Benson, J. Brown, P. Brune, K. Buschelman, E. M. Constantinescu, L. Dalcin, A. Dener, V. Eijkhout, J. Faibussowitsch, W. D. Gropp, V. Hapla, T. Isaac, P. Jolivet, D. Karpeev, D. Kaushik, M. G. Knepley, F. Kong, S. Kruger, D. A. May, L. Curfman McInnes, R. Tran Mills, L. Mitchell, T. Munson, J. E. Roman, K. Rupp, P. Sanan, J. Sarich, B. F. Smith, S. Zampini, H. Zhang, H. Zhang, and J. Zhang. PETSc/TAO users manual. Technical Report ANL-21/39 - Revision 3.18, Argonne National Laboratory, 2022.

[BGMS97] S. Balay, W.D. Gropp, L. Curfman McInnes, and B.F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, Modern Software Tools in Scientific Computing, pages 163–202. Birkhäuser Press, 1997.

[CS99] X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. SIAM J. Sci. Comp., 21(2):239–247, 1999.

[CV22a] G. Ciaramella and T. Vanzan. Spectral coarse spaces for the substructured parallel Schwarz method. J. Sci. Comput., 91(69), 2022.

[CV22b] G. Ciaramella and T. Vanzan. Substructured two-grid and multi-grid domain decomposition methods. Numerical Algorithms, 91:413–448, 2022.

[DGL+12] O. Dubois, M.J. Gander, S. Loisel, A. St-Cyr, and D.B. Szyld. The optimized Schwarz methods with a coarse grid correction. SIAM J. Sci. Comp., 34(1):A421–A458, 2012.

[GH12] M. J. Gander and L. Halpern. Méthodes de décomposition de domaine, encyclopédie électronique pour les ingénieurs. Technical report, 2012.

[GHS14] M.J. Gander, L. Halpern, and K. Santugini. A new coarse grid correction for RAS/AS. In Domain Decomposition Methods in Science and Engineering XXI, Lecture Notes in Computational Science and Engineering, pages 275–284. Springer-Verlag, 2014.

[GV19] M.J. Gander and S. Van Criekingen. New coarse corrections for restricted additive Schwarz using PETSc. In Domain Decomposition Methods in Science and Engineering XXV, Lecture Notes in Computational Science and Engineering, pages 483–490. Springer-Verlag, 2019.

[GV21] M.J. Gander and S. Van Criekingen. Coarse corrections for Schwarz methods for symmetric and non-symmetric problems. In Domain Decomposition Methods in Science and Engineering XXVI, Lecture Notes in Computational Science and Engineering, pages 589–596. Springer-Verlag, 2021.

[Saa03] Y. Saad. Iterative Methods for Sparse Linear Systems. SIAM, 2003.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

81/2022 Bonizzoni, F.; Hauck, M.; Peterseim, D.
A reduced basis super-localized orthogonal decomposition for reaction-convection-diffusion problems

80/2022 Balduzzi, G.; Bonizzoni, F.; Tamellini, L.
Uncertainty quantification in timber-like beams using sparse grids: theory and examples with off-the-shelf software utilization

79/2022 Antonietti, P. F.; Farenga, N.; Manuzzi, E.; Martinelli, G.; Saverio, L.
Agglomeration of Polygonal Grids using Graph Neural Networks with applications to Multigrid solvers

78/2022 Bucelli, M.; Gabriel, M. G.; Gigante, G.; Quarteroni, A.; Vergara, C.
A stable loosely-coupled scheme for cardiac electro-fluid-structure interaction

77/2022 Ziarelli, G.; Dede', L.; Parolini, N.; Verani, M.; Quarteroni, A.
Optimized numerical solutions of SIRDVW multiage model controlling SARS-CoV-2 vaccine roll out: an application to the Italian scenario.

76/2022 Spreafico, M.; Ieva, F.; Fiocco, M.
Longitudinal Latent Overall Toxicity (LOTox) profiles in osteosarcoma: a new taxonomy based on latent Markov models

71/2022 Calabrò, D.; Lupo Pasini, M.; Ferro, N.; Perotto, S.
A deep learning approach for detection and localization of leaf anomalies

70/2022 Andrini, D.; Balbi, V.; Bevilacqua, G.; Lucci, G.; Pozzi, G.; Riccobelli, D.
Mathematical modelling of axonal cortex contractility

68/2022 Orlando, G.; Benacchio, T.; Bonaventura, L.
An IMEX-DG solver for atmospheric dynamics simulations with adaptive mesh refinement

73/2022 Spreafico, M.; Gasperoni, F.; Barbati, G.; Ieva, F.; Scagnetto, A.; Zanier, L.; Iorio, A.; Sinagra, G.; Di Lenarda, A.
Adherence to disease-modifying therapy in patients hospitalized for HF: findings from a community-based study