A Performance Comparison of Classical Volume and New Substructured One- and Two-Level Schwarz Methods in PETSc

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 compare here the Parallel Schwarz Method (PSM, equivalent to RAS) in volume to the new substructured version of PSM in [11, p.24] and recently extended to a two-level (i.e. coarse-corrected) framework in [6] and [5], using a geometric and spectral approach for the definition of the coarse space. The expected gain of substructured methods is due to the smaller size of the resulting problems, notably with Krylov-type acceleration techniques when the dimension of the subspace of approximants becomes large [12].

While the numerical results in [5, 6] were obtained sequentially, we present here a parallel performance comparison of volume and substructured Schwarz methods using `PETSc` [1, 2, 3], successively considering one- (Section 2) and two- (Section 3) level methods. The substructured results are compared to the ones obtained by the RAS method in volume [4] for which two-level results with various coarse spaces were presented in [9, 10] also using `PETSc` . For the two-level substructured method, four coarse spaces are introduced here, all based on a geometric approach. Note that, at this time, spectral approaches still require further investigations and are therefore

Gabriele Ciaramella
MOX, Dipartimento di Matematica, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy, e-mail: gabriele.ciaramella@polimi.it

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

Serge Van Criekingen
Institut du Développement et des Ressources en Informatique Scientifique (IDRIS), CNRS, Université Paris-Saclay, F-91403, Orsay, France and Université Paris-Saclay, UVSQ, CNRS, CEA, Maison de la Simulation, 91191, Gif-sur-Yvette, France, e-mail: serge.van.criekingen@idris.fr

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

not presented here (- the reason being that the eigenvectors on which spectral coarse spaces are based in [5] 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).

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

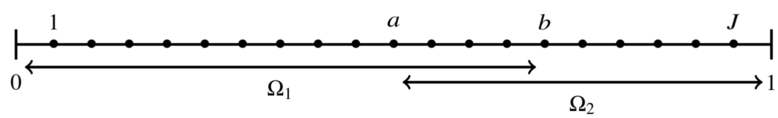


Fig. 1 Two subdomain decomposition in the 1-D case.

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 [11], 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

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

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

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

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

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

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-

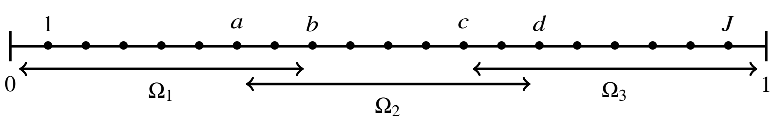


Fig. 2 Three subdomain decomposition in the 1-D case.

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 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.

The T substructured system matrix defined in (5) is implemented matrix-free in our `PETSc` implementation, using the `MatCreateShell` and `MatShellSet-Operation` tools. Each multiplication by T implies data transfer (with or without

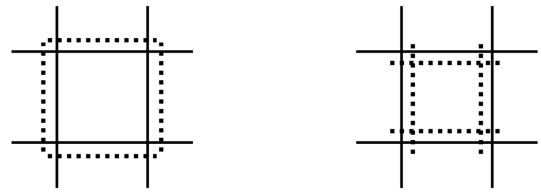


Fig. 3 Dotted are the substructure values to be received (a) or sent (b) by the central subdomain.

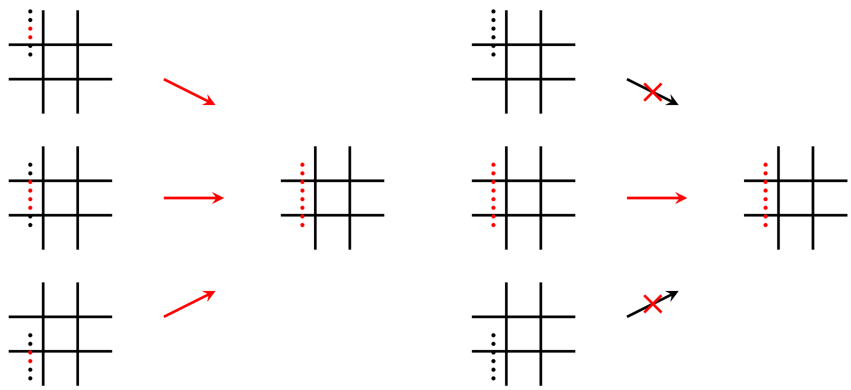
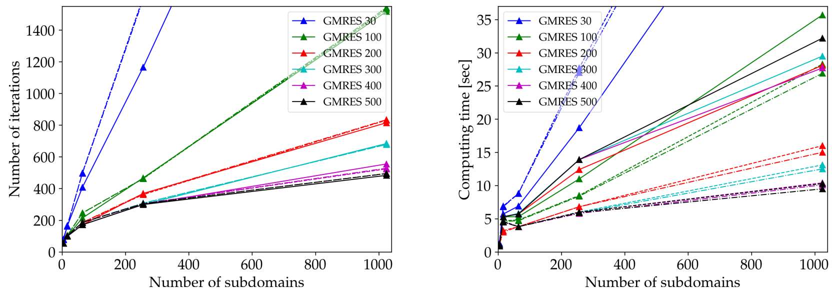


Fig. 4 Schematic representation of left-to-right data exchange with (a) or without (b) transfers from diagonal neighbours. The transferred data are in red.

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 [4] (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 (.004 fine-to-coarse mesh ratio). Several observations can be 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



(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.

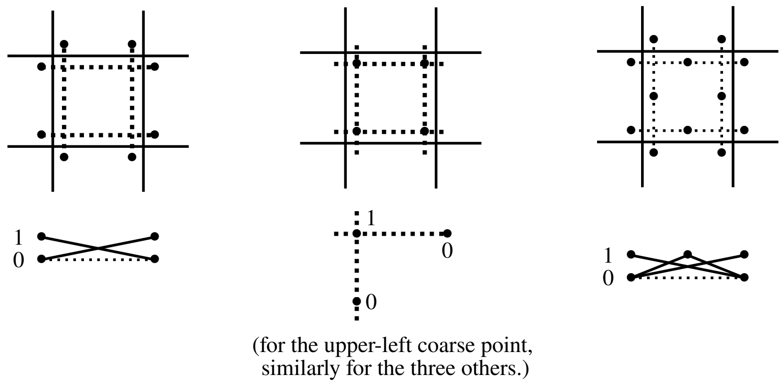
the substructured problem thus making a larger Krylov space profitable. Third, and most importantly, at high restart parameters and in particular at the optimal one, substructured methods yield better timing performances than volume methods. This appears to be due to the smaller size of the substructured 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 [9], 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 [7, 8, 9].



(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).

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

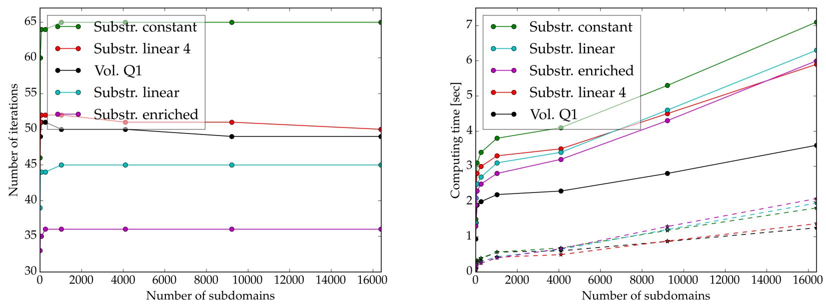
$$g^{n+1/2} = g^n + (f^g - Tg^n), \tag{6}$$

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

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`.

Figure 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



(a) Iteration count

(b) Wall-clock total (solid) and coarse solution (dashed) times

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

solution 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. It is possible to improve the substructured computational times further by noting that the two-level iteration (6)–(7) requires the computation of two actions of the operator T, and one of them can be eliminated using the strategy proposed in [5, 6]. This is possible to do in PETSc as well, but requires a substantial modification in the implementation technique that goes beyond this short manuscript, and will appear elsewhere. 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. Its coarse solution time appears very close the `Q1` one in volume as shown in Fig. 7b (dashed lines).

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 solution and wall-clock times.

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

References

1. Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E. M., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W. D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M. G., Kong, F., Kruger, S., May, D. A., McInnes, L. C., Mills, R. T., Mitchell, L., Munson, T., Roman, J. E., Rupp, K., Sanan, P., Sarich, J., Smith, B. F., Zampini, S., Zhang, H., Zhang, H., and Zhang, J. PETSc Web page. https://petsc.org/ (2022).
2. Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E. M., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W. D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M. G., Kong, F., Kruger, S., May, D. A., McInnes, L. C., Mills, R. T., Mitchell, L., Munson, T., Roman, J. E., Rupp, K., Sanan, P., Sarich, J., Smith, B. F., Zampini, S., Zhang, H., Zhang, H., and Zhang, J. PETSc/TAO users manual. Tech. Rep. ANL-21/39 - Revision 3.18, Argonne National Laboratory (2022).
3. Balay, S., Gropp, W., McInnes, L. C., and Smith, B. Efficient management of parallelism in object oriented numerical software libraries. In: Arge, E., Bruaset, A. M., and Langtangen, H. P. (eds.), *Modern Software Tools in Scientific Computing*, 163–202. Birkhäuser Press (1997).
4. Cai, X.-C. and Sarkis, M. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comp.* **21**(2), 239–247 (1999).
5. Ciaramella, G. and Vanzan, T. Spectral coarse spaces for the substructured parallel Schwarz method. *J. Sci. Comput.* **91**(69) (2022).
6. Ciaramella, G. and Vanzan, T. Substructured two-grid and multi-grid domain decomposition methods. *Numerical Algorithms* **91**, 413–448 (2022).
7. Dubois, O., Gander, M., Loisel, S., St-Cyr, A., and Szyld, D. The optimized Schwarz methods with a coarse grid correction. *SIAM J. Sci. Comp.* **34**(1), A421–A458 (2012).
8. Gander, M., Halpern, L., and Santugini, K. A new coarse grid correction for RAS/AS. In: *Domain Decomposition Methods in Science and Engineering XXI*, Lecture Notes in Computational Science and Engineering, 275–284. Springer-Verlag (2014).
9. Gander, M. and Van Criekingen, S. 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, 483–490. Springer-Verlag (2019).
10. Gander, M. and Van Criekingen, S. 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, 589–596. Springer-Verlag (2021).
11. Gander, M. J. and Halpern, L. Méthodes de décomposition de domaine, encyclopédie électronique pour les ingénieurs. Tech. rep. (2012).
12. Saad, Y. *Iterative Methods for Sparse Linear Systems*. SIAM (2003).