MOX-Report No. 60/2026

Algebraic and Two-Level Parallel Substructured Schwarz Methods

Ciaramella, G.; Gander, M.J.; 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

Algebraic and Two-Level Parallel Substructured Schwarz Methods

Gabriele Ciaramella[0000−0002−5877−4426],
Martin J. Gander[0000−0001−8450−9223],
Serge Van Criekingen[0009−0005−8401−3160],
Tommaso Vanzan[0000−0001−7554−4692]

1 Introduction

Substructured Schwarz methods are interpretations of classical volume Schwarz methods as algorithms on interface variables. While most of the time understood as non-overlapping methods, substructured methods can in fact also benefit from overlap: see [10] for a historical account of substructuring in domain decomposition from the origins to the first substructured overlapping optimized Schwarz method introduced in [13]. Recent work on overlapping substructured formulations includes the references [4, 5, 6, 7]. This work is a follow-up on [5], in which we compared the performance of classical volume overlapping Schwarz methods to equivalent overlapping substructured methods, using the PETSc library [1, 2, 3]. This comparison was performed at scale on HPC infrastructures and we are not aware of other similar comparisons at scale, except for [12] which compares the performance of a classical volume overlapping Schwarz method to a non-overlapping substructured Schwarz method (but does not consider overlapping substructured Schwarz methods) for the Helmholtz equation with multiple right-hand sides.

In our previous work [5], the trace (i.e. interface) elements constitutive of the substructure (also called skeleton) were identified geometrically using the regular-

Gabriele Ciaramella
MOX Lab, Dipartimento di Matematica, Politecnico di Milano, Italy, e-mail: gabriele.ciaramella@polimi.it

Martin J. Gander
University of Geneva, Switzerland, 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 & Université Paris-Saclay, UVSQ, CNRS, CEA, Maison de la Simulation, 91191, Gif-sur-Yvette, France, e-mail: serge.van.criekingen@idris.fr

Tommaso Vanzan
Politecnico di Torino, Italy, e-mail: tommaso.vanzan@polito.it

ity of the mesh. Here, after recalling the substructured formulation (Section 2), we propose (Section 3) a new implementation to identify these elements algebraically in a parallel context. Next, we consider (Section 4) a two-level substructured method (geometric only for now) with coarse space functions defined exclusively on the skeleton, and we propose an additive version of the two-level preconditioner which conserves the number of iterations (compared to the multiplicative approach) and significantly decreases the computing time. Weak scaling numerical results up to several thousands of `CPU` cores (one per subdomain) are presented in both Sections 3 and 4, for one- and two-level methods respectively, comparing substructured and classical volume methods. Our reference volume results are based on `PETSc`'s `PCASM` additive Schwarz method implementation (with direct exact local solves) and use the `mat_increase_overlap_scalable` option.

2 Substructured formulation

We consider the system $Au = f$ for the Laplace problem with Dirichlet boundary conditions discretized with the classical finite difference scheme. Considering for ease of presentation the 1-D case (namely the (0,1) interval subdivided into $J + 1$ mesh cells of size h with discretization points numbered from 0 to $J - 1$ as shown in Fig. 1) with a two-subdomain overlapping decomposition, the substructured parallel

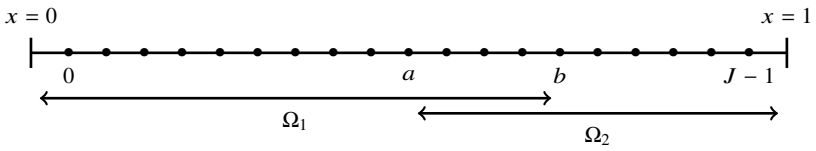


Fig. 1 Two subdomain decomposition in the 1-D case.

Schwarz method, which is equivalent to RAS [9], can be written [5] as

$$Tg = f^g, \tag{1}$$

with

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

where $A_1 \subset \mathbb{R}^{b \times b}$ and $A_2 \subset \mathbb{R}^{(J-a-1)\times(J-a-1)}$ are the (overlapping) diagonal subblocks of A corresponding to the two subdomains, f_1 and f_2 are the corresponding subvectors of f, the trace operators G_i $(i = 1, 2)$ are defined as

$$G_1 : (v_0, \ldots, v_a, \ldots, v_{b-1}) \to v_a, \quad G_2 : (v_{a+1}, \ldots, v_b, \ldots, v_{J-1}) \to v_b,$$

and the extension by zero operators E_i as

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

Note that the local solves A_i^{-1} in (2) must be performed exactly since they are part of the system operator, and their approximation would lead to a different solution.

3 Algebraic trace characterization

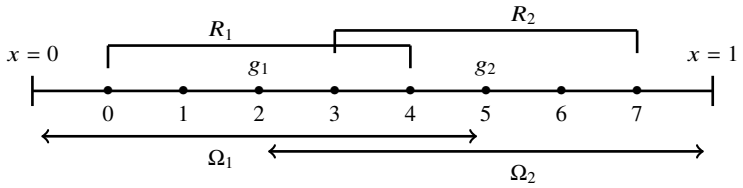


Fig. 2 Global numbering for the 8 point two subdomain configuration in 1-D.

For ease of presentation, let us consider (Fig. 2) an 8 point global domain split into two equal-sized subdomains with overlap 1 in the PETSc sense, namely one extra point for each subdomain (that is an algebraic overlap of 2; unknowns 0, 1, 2, 3 and 4 are in Ω_1, unknowns 3, 4, 5, 6 and 7 are in Ω_2). The matrix T has size 2×2 and the trace value vector $g = (g_1, g_2)^T$ is made out of the two elements of global number 2 and 5. Using (1)-(2), we have that the A_1^{-1} matrix in the upper right part of T will have to be applied to the vector $E_1 g_2 = (0, 0, 0, 0, g_2)^T$, i.e., a vector in subdomain 1 with value g_2 in local index 4 (PETSc indexing starting from 0). Similarly, the A_2^{-1} matrix in the lower left part of T will have to be applied to the vector $E_2 g_1 = (g_1, 0, 0, 0, 0)^T$, i.e., a vector in subdomain 2 with value g_1 in local index 0. Then, the trace operator G_1 will extract the element of global index 2 out of $A_1^{-1} E_1 g_2$, and G_2 will extract the element of global index 5 out of $A_2^{-1} E_2 g_1$. Adapting a technique introduced in [4, Section 3.1], we define

$$R_{s,j} = R_j\, A - R_j\, A\, R_j^T\, R_j, \tag{3}$$

where the R_j ($j = 1, 2$) are the restriction operators to the two overlapping subdomains. For subdomain 1 we have

$$R_{s,1} = \frac{-1}{h^2} \begin{pmatrix} -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & \mathbf{1} & 0 & 0 \end{pmatrix} - \frac{-1}{h^2} \begin{pmatrix} -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & \mathbf{0} & 0 & 0 \end{pmatrix}, \tag{4}$$

so that $R_{s,1}$ has only one nonzero element, namely $\frac{-1}{h^2}$ on row 4 and column 5 (again with indices starting from 0). Thus the row and column indices of the nonzero

element of $R_{s,1}$ respectively give us the local index 4 (in subdomain 1) where the g_2 value has to be assigned in the extension by zero phase before applying A_1^{-1} in (2), and the global index 5 corresponding to this g_2 value. Similarly for subdomain 2, we have that $R_{s,2}$ has only a nonzero element in row 0 and column 2, which gives us the local index 0 (in subdomain 2) where g_1 has to be assigned in the extension by zero phase, and the global index 2 corresponding to this g_1 value.

We therefore state (and this can be verified on a less basic example) that the nonzero elements of the $R_{s,j}$ matrix give us the necessary information for the algebraic characterization of the skeleton: for each nonzero element in $R_{s,j}$,

- its row gives the local index where the corresponding trace element has to be assigned in the extension by zero phase before the local solve in (2),
- its column gives the global index of the corresponding skeleton element,
- its value gives the scaling coefficient $(-1/h^2)$ to be applied to the trace value.

Definition (3) uses the full A matrix but the information necessary for the trace characterization lies, for each subdomain, only "one unknown away" from it (in case of stencil width 1 as we consider here) and can be extracted by computing

$$\tilde{R}_{s,j} = R_{s,j}\,(R_j^{ext})^T = R_j\,A\,(R_j^{ext})^T - R_j\,A\,R_j^T\,R_j\,(R_j^{ext})^T \tag{5}$$

where R_j^{ext} is the restriction matrix to a subdomain with an overlap increased by 1. Definition (5) (where the first term is a submatrix of A and the second term the extension of the submatrix $R_j\,A\,R_j^T$ with some columns of zeros) avoids scalability issues and amounts to about 2% of the computing time in our tests below.

Another point to notice in view of large-scale parallel experiments is that the column information of $R_{s,j}$ (or $\tilde{R}_{s,j}$), i.e., the global index of the data to be received from a neighboring element, is non-local. In our above example, $R_{s,1}$ is computed on subdomain 1 but unknown to subdomain 2, so the value of its non-zero column (namely 5) has to be sent (in a setup phase) to subdomain 2 so that this subdomain knows which data it has to send (in the subsequent `GMRES` iterations) to subdomain 1. A simple way to do this is through `PETSc`'s `ISgetTotalIndices` (which amounts to `MPI_Allgather`) operation on the whole communicator followed by `ISGlobal-ToLocalMappingApply`. In more than 1D, one can then, in a way to apply a partition of unity, ignore the indices in the overlap, and this is equivalent to the "with diagonal transfers" option of the geometric implementation [5]. However, applying `ISget-TotalIndices` on the whole communicator is not scalable with respect to memory consumption in more than 1D. Therefore, we designed and implemented another way to proceed, where `ISgetTotalIndices` is applied sequentially on unidirectional sub-communicators, each of them being one-dimensional in one of the problem dimensions. This is equivalent to the "without diagonal transfers" option of the geometric implementation [5]. Note that taking unidirectional sub-communicators is non-algebraic by nature, but the trace characterization remains algebraic.

One-level 2D weak scaling results for the Laplace problem (on the unit square and with a cartesian mesh) obtained with the new algebraic trace characterization are presented in Fig. 3. System (1) is solved using `GMRES` without preconditioner (the Schwarz method being already embedded in the formulation). The no-restart volume

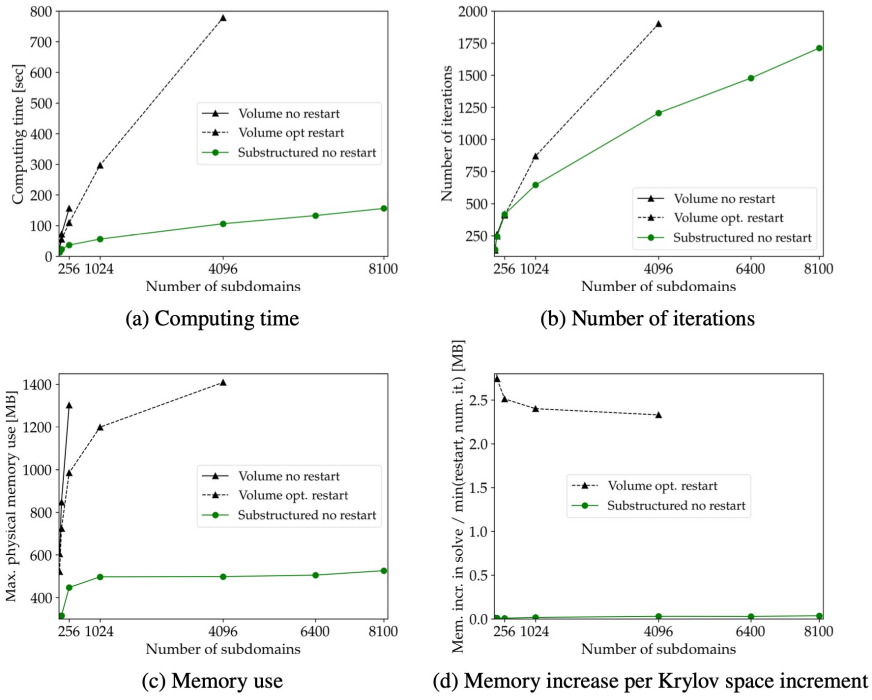


Fig. 3 One-level weak scaling experimental results with 512×512 local mesh and square domain decompositions using up to 8100 subdomains (i.e., `CPUs`), using `GMRES` acceleration with optimal or no restart.

results are displayed only up to 256 subdomains because at 1024 subdomains the memory limit was reached. While using an optimized (w.r.t. computing time) restart parameter improves the volume results, the no-restart option always appeared the best for the substructured case. We observe a very clear computing time and memory gain in using the substructured method. The memory gain is due to the fact that the Krylov space in the `GMRES` process is built on the substructure, and is thus much smaller than in the volume case where it is built on the whole domain. The number of iterations of the volume (with or without restart) and substructured methods nearly coincide up to 256 subdomains, where the time and memory gain are already visible, and these gains grow even more beyond 256 subdomains, where the volume number of iterations grows larger than the substructured one. One can estimate the memory increase per Krylov space increment as the total memory use increase during the solve phase divided by the minimum of the restart parameter and the number of iterations: Fig. 3(d) shows that the difference between the substructured and volume values tends to a value higher than 2MB.

4 Two-level parallel implementation and results

The two-level parallel Schwarz method for (1) reads

$$g^{n+1/2} = g^n + I(f^g - Tg^n), \tag{6}$$
$$g^{n+1} = g^{n+1/2} + R_c^T T_c^{-1} R_c \ (f^g - Tg^{n+1/2}), \tag{7}$$

where I is the identity and $T_c = R_c T R_c^T$ is the substructured coarse matrix, R_c being the restriction matrix to the substructured coarse space. The (geometric) substructured coarse functions used here are the same as in [5, Fig. 6], namely `Nicolaides` (one constant function per substructure portion), `Linear4` (same number of functions as `Nicolaides` but with linear functions), `Linear` (two linear functions per portion) and `Enriched` (three linear functions per portion), see [5] for details.

As for the implementation of (6)-(7), first note that, while T is implemented matrix-free, T_c is built by applying it to the identity matrix. Then `PETSc` has its LU decomposition computed once and an LU solve performed at each application. Moreover, note that since T has identity diagonal blocks (see (2)), it can be written as $T = I - G$. Using this and inserting $g^{n+1/2}$ from (6) into (7), we obtain that the two-level multiplicative preconditioning of (6)-(7), namely

$$R_c^T T_c^{-1} R_c \circ I \tag{8}$$

can be replaced by the two-level additive one

$$I + R_c^T T_c^{-1} R_c G. \tag{9}$$

Next, since for any two matrices M_1 and M_2, their products $M_1 M_2$ and $M_2 M_1$ share the same nonzero eigenvalues ($M_1 M_2 v = \lambda v \Rightarrow M_2 M_1 w = \lambda w$ with $w = M_2 v$; $w \notin \ker M_1$ and $v \notin \ker M_2$ since $M_1 M_2 v = \lambda v \neq 0$), this has the same convergence behavior as

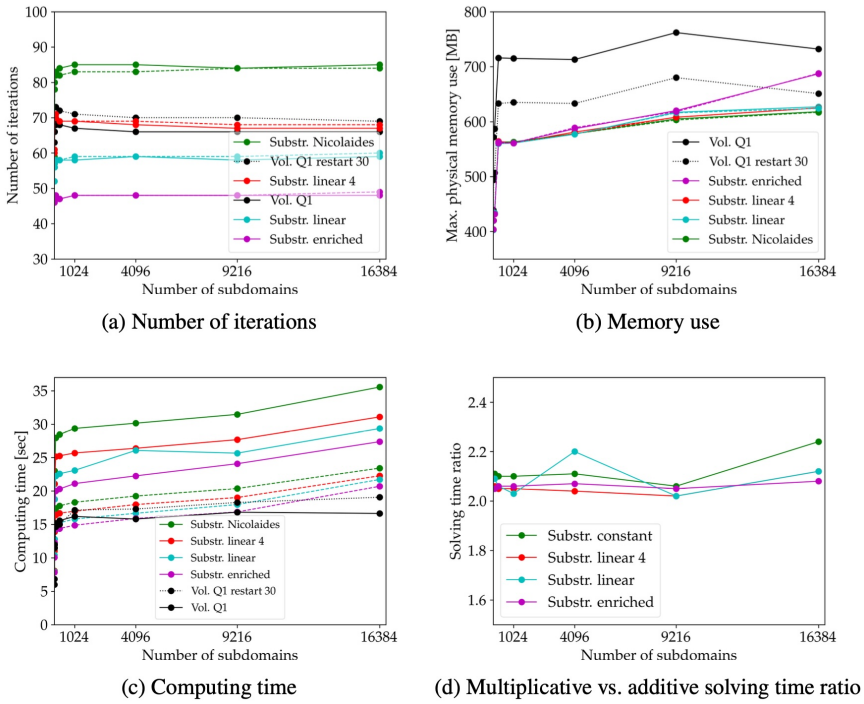
$$I + Q_c^T T_c^{-1} R_c \tag{10}$$

where $Q_c^T = G R_c^T$. Note that Q_c in (10) is known from the $T_c = R_c (I - G) R_c^T$ computation and thus comes at no extra cost, and that (10) yields the recurrence

$$g^{n+1} = g^n + (I + Q_c^T T_c^{-1} R_c) \ (f^g - Tg^n).$$

This recurrence implies only one residual computation and thus only one subdomain solve per iteration instead of two in (6)-(7), and this comes with quasi constant number of iterations given the identical convergence behavior. This derivation parallels the proof of Theorem 8 (b) in [6].

Fig. 4 presents the two-level weak scaling results with the above mentioned coarse spaces for the substructured case, and with the `Q1` coarse space (using four coarse points around each cross point [8, 11]) for the volume case. For substructured results, solid lines correspond to results with multiplicative preconditioning and dashed



(a) Number of iterations (b) Memory use

(c) Computing time (d) Multiplicative vs. additive solving time ratio

Fig. 4 Two-level weak scaling results with 512×512 local mesh and square domain decompositions using up to 16384 subdomains, using `GMRES` acceleration without restart unless otherwise stated. In (a), (b), and (c), solid/dashed lines correspond to multiplicative/additive substructured preconditioning, respectively.

lines with additive preconditioning, the latter yielding quasi the same number of iterations as expected (Fig. 4(a)), the same memory consumption (Fig. 4(b)), but better computing time results (Fig. 4(c)) due to the halving of the number of residual computations. Fig. 4(d) shows the multiplicative vs. additive ratio of the solving time (i.e., the `KSPSolve` time in `PETSc`), excluding the time for the coarse solve T_c^{-1}. This ratio remains close to the expected value of 2.

Comparing volume vs. best substructured results, we observe similar computing times (at least with the enriched coarse space) up to 9216 `CPU` cores, and at 16384 `CPU` cores some scalability issues appear for the substructured case (from the log files, within the communication routines used in the coarse solve phase). On the other hand, we observe in Fig. 4(b) a memory gain in using the substructured method. The memory use of the volume method can be reduced by introducing restarts (see the restart 30 results in Fig. 4(b)), but this comes at the cost of an increased computing time (see Fig. 4(c)). The substructured memory gain can be explained in part by the Krylov space size as in the one level case (but here the number of iterations is smaller so the influence is smaller), and, probably for the most part, because of the lower dimensionality of the coarse space defined exclusively on the substructure.

Therefore, we hope that the (still to be implemented) 3D two-level substructured method will generate a bigger advantage compared to the classical volume method.

Acknowledgements This work was performed using HPC resources from GENCI-CINES. GC and TV are members of the Indam GNCS group.

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., Zhang, J.: PETSc Web page. https://petsc.org/ (2025)
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., Zhang, J.: PETSc/TAO users manual. Tech. Rep. ANL-21/39 - Revision 3.24, Argonne National Laboratory (2025)
3. Balay, S., Gropp, W., McInnes, L.C., Smith, B.: Efficient management of parallelism in object oriented numerical software libraries. In: E. Arge, A.M. Bruaset, H.P. Langtangen (eds.) Modern Software Tools in Scientific Computing, pp. 163–202. Birkhäuser Press (1997)
4. Chaouqui, F., Gander, M., Kumbhar, P., Vanzan, T.: Linear and nonlinear substructured Restricted Additive Schwarz iterations and preconditioning. Numer. Algorithms **91**, 81–107 (2022)
5. Ciaramella, G., Gander, M., Van Criekingen, S., Vanzan, T.: A performance comparison of classical volume and new substructured one- and two-level Schwarz methods in PETSc. In: Domain Decomposition Methods in Science and Engineering XXVII, Lecture Notes in Computational Science and Engineering, pp. 133–140. Springer-Verlag (2023)
6. Ciaramella, G., Vanzan, T.: Spectral coarse spaces for the substructured parallel Schwarz method. J. Sci. Comput. **91**(69) (2022)
7. Ciaramella, G., Vanzan, T.: Substructured two-grid and multi-grid domain decomposition methods. Numer. Algorithms **91**, 413–448 (2022)
8. Dubois, O., Gander, M., Loisel, S., St-Cyr, A., Szyld, D.: The optimized Schwarz methods with a coarse grid correction. SIAM J. Sci. Comp. **34**(1), A421–A458 (2012)
9. Gander, M.: Schwarz methods over the course of time. ETNA **31**, 228–255 (2008)
10. Gander, M., Nataf, F.: Substructuring of arbitrary domain decomposition methods. In: Domain Decomposition Methods in Science and Engineering XXVII, Lecture Notes in Computational Science and Engineering, pp. 223–230. Springer-Verlag (2023)
11. Gander, M., 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, pp. 589–596. Springer-Verlag (2021)
12. Martin, B., Jolivet, P., Geuzaine, C.: Comparison of substructured non-overlapping domain decomposition methods and overlapping additive Schwarz methods for large-scale Helmoltz problems with multiple sources. J. Comput. Phys. **548** (2026)
13. Nataf, F., Rogier, F., de Sturler, E.: Optimal interface conditions for domain decomposition methods. Tech. rep., CMAP, Ecole Polytechnique, Paris (1994)

MOX Technical Reports, last issues

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

58/2026 Mapelli, A.; Massi, M.C.; Cuccuru, G.; Di Angelantonio, E.; Ieva, F.
Prior-informed conditional Gaussian graphical models: an application to protein interaction network reconstruction

57/2026 Fontana, N.; Secchi, P.; Di Angelantonio, E.; Ieva, F.
Modeling time-varying genetic effects on binary disease risk via functional Mendelian Randomization

56/2026 Botta, P.; Vitullo, P.; Ventimiglia, T.; Linninger, A.; Zunino, P.
Physics-Informed Learning of Microvascular Flow Models using Graph Neural Networks

52/2026 Bonazzoli M.; Ciaramella G.; Mazzieri I.
On the Unmapped Tent Pitching for the Heterogeneous Wave Equation

51/2026 Bellezza P.; Ciaramella G.; Macchini C.; Mazzieri I.; Verani M.
ParaFlow: Parareal Acceleration of Gradient-Flow Minimization

53/2026 Dong Z., Jiang Y., Ng M., Ciaramella G., Yin J.
Chebyshev-Filtered Randomized Low-rank Preconditioners for Symmetric Positive Definite Linear Systems

55/2026 Beirao da Veiga, L.; Canuto, C.; Nochetto, R.H.; Vacc, G ; Verani, M.
A Virtual Element Method for elliptic problems on trimmed background meshes

54/2026 Antonietti, P. F.; Corti, M.; Leimer Saglio, C. B.; Pagani, S.
The lymph 2.0 library: p-adaptive algorithms and parallel assembly strategies for polytopal DG methods

50/2026 Donnarumma, A.; Guagliardi, O.; Di Stazio F.; Mazza E.; Tanelli M.; Paganoni A.M.
Modelling Well-Being and Psychological Risk in Doctoral Education: An Integrated Latent Trait Approach