MOX-Report No. 16/2022

Substructured Two-grid and Multi-grid Domain Decomposition Methods

G. Ciaramella, T. Vanzan

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

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

Noname manuscript No.
(will be inserted by the editor)

Substructured Two-grid and Multi-grid Domain Decomposition Methods

G. Ciaramella · T. Vanzan

Received: date / Accepted: date

Abstract Two-level Schwarz domain decomposition methods are very powerful techniques for the efficient numerical solution of partial differential equations (PDEs). A two-level domain decomposition method requires two main components: a one-level preconditioner (or its corresponding smoothing iterative method), which is based on domain decomposition techniques, and a coarse correction step, which relies on a coarse space. The coarse space must properly represent the error components that the chosen one-level method is not capable to deal with. In the literature, most of the works introduced efficient coarse spaces obtained as the span of functions defined on the entire space domain of the considered PDE. Therefore, the corresponding two-level preconditioners and iterative methods are defined in volume.

In this paper, we use the excellent smoothing properties of Schwarz domain decomposition methods to define, for general elliptic problems, a new class of substructured two-level methods, for which both Schwarz smoothers and coarse correction steps are defined on the interfaces (except for the application of the smoother that requires volumetric subdomain solves). This approach has several advantages. On the one hand, the required computational effort is cheaper than the one required by classical volumetric two-level methods. On the other hand, our approach does not require, like classical multi-grid methods, the explicit construction of coarse spaces, and it permits a multilevel extension, which is desirable when the high dimension of the problem or the scarce quality of the coarse space prevents the efficient numerical solution. Numerical experiments demonstrate the effectiveness of the proposed new numerical framework.

G. Ciaramella
Politecnico di Milano, Italy
E-mail: gabriele.ciaramella@polimi.it

T. Vanzan
CSQI Chair, École Polytecnique Fédérale de Lausanne, Switzerland
E-mail: tommaso.vanzan@epfl.ch

1 Introduction

Domain decomposition (DD) methods are powerful divide-and-conquer strategies that permit the solution of linear systems of equations, generally discrete partial differential equation (PDE) problems, by efficient parallelization processes [20,48, 52]. Over the course of the time, several different parallel DD strategies have been developed; see [28] for an elegant review with a historical flavor. The first idea of parallelizing a Schwarz method goes back to P.L. Lions, who introduced the classical parallel Schwarz method [47]. This method is based on Dirichlet transmission conditions and its discrete version is proved to be equivalent to the famous Restricted Additive Schwarz (RAS) method [28], which was discovered much later in [6]. Similar to RAS is the famous Additive Schwarz (AS) method introduced in [54]. Even though both RAS and AS are based on Dirichlet transmission conditions, they are not equivalent methods; see, e.g., [24] for a detailed comparison analysis. If different transmission conditions are used, one obtains different DD methods, like the optimized Schwarz method [27,36], the Neumann-Neumann (or FETI) method [52,9], the Dirichlet-Neumann method [48,7], etc. The main drawback of these classical one-level DD methods is that they are not (weakly) scalable, since their convergence generally deteriorates when the number of subdomains increases; see, e.g., [20,52,7]. Only in few cases a particular scalability behavior has been proved and investigated [7,10–12,16,17]. To overcome this scalability issue, a coarse correction step is usually used. This leads to "two-level DD methods". In the literature, with "two-level DD method" one refers to either a two-level preconditioner [1,2,4,3,19,21,23,25,26,33,39,45,44,49,50,55], or to a two-level stationary method [8,9,14,22,30–32,34,35,37]. While in the first class one seeks for a coarse-space matrix to add to the one-level DD preconditioner, in the second class the goal is to design a correction step, where the residual equation is solved on a coarse space. This second class follows an idea similar to the one of multi-grid methods [42].

For any given one-level DD method (stationary or preconditioning), the choice of the coarse space influences very strongly the convergence behavior of the corresponding two-level method. For this reason, the main focus of all the references mentioned above is the definition of different coarse spaces and new strategies to build coarse space functions, leading to efficient two-level DD stationary and preconditioning methods. Despite the mentioned references consider several one-level DD methods and different partial differential equation (PDE) problems, it is still possible to classify them in two main groups. These depend on the idea governing the definition of the coarse space. To explain it, let us consider a DD iterative method (e.g., RAS) applied to a well-posed PDE problem. Errors and residuals of the DD iterative procedure have generally very special forms. The errors are predominant in the overlaps and are harmonic, in the sense of the underlying PDE operator, in the interior of the subdomains (excluding the interfaces). The residuals are predominant on the interfaces and zero outside the overlap. For examples and more details, see, e.g., [32,15,14]. This difference motivated, sometimes implicitly, the construction of different coarse spaces. On the one hand, many references use different techniques to define coarse functions in the overlap (where the error is predominant), and then extend them on the remaining part of the neighboring subdomains; see, e.g., [19,21,23,25,26,44,45,49,50]. On the other hand, in other works the coarse space is created by first defining basis functions on the interfaces

(where the residual is non-zero), and then extending them (in different ways) on the portions of the neighboring subdomains; see, e.g., [1,2,8,9,14,30,33,32,34,35, 44,37]. For a good, compact and complete overview of several of the different coarse spaces, we refer to [44, Section 5]. For Helmholtz and time-harmonic equations, we refer to [3,4,39], where the coarse space is based on a (volumetric) coarse mesh. For other different techniques and other related discussions see, e.g., [20,22,30,31, 40,55].

The starting point of this work is related to an important property of Schwarz methods: one-level Schwarz iterative methods are generally very efficient smoothers; see, e.g., [27,10,13,37] and references therein. This property was already discussed in [42, Chapter 15], where Schwarz methods are used as classical smoothers in the context of multi-grid methods or to define local defect corrections; see, e.g., [41, 37]. In these frameworks, Schwarz methods are used to smooth and correct the approximation in some subdomains of the entire computational domain. However, as we already mentioned, after one Schwarz smoothing iteration, the residuals are generally predominant on the interfaces and in some cases even zero outside the interfaces. This remark is the key starting point of our work. We introduce for the first time so-called two-level and multi-level DD substructured methods. We call these methods Geometric 2-level Substructured (G2S) method and Geometric Multi-level Substructured (GMS) method. The term "substructured" indicates that iterations and coarse correction steps are defined on the interfaces (or more precisely on the substructures[1]) of the domain decomposition (note that volumetric subdomains solves are still required to apply the smoother). With this respect, our methods are defined in the same spirit as two-level methods whose coarse spaces are extensions in volume of interfaces basis functions: they attempt to correct the residual only where it is truly necessary. The G2S method is essentially a two-grid parallel Schwarz method defined on the substructures, for which the coarse correction is performed on coarser interface grids. The GMS is the extension of the G2S to a multi-level framework. In other words, by the G2S and GMS methods, we propose a new methodology that attempts the best use of Schwarz smoothers in the context of two-grid and multi-grid methods. Direct numerical experiments show that these methods converge in less iterations than classical two-grid methods defined in volume and using a Schwarz smoother. In many cases, this improvement in terms of iteration number is significantly high. Moreover, our new methods have, in addition, other advantages. On the one hand, like classical multi-grid methods, the G2S method does not require the explicit construction of coarse spaces, and it permits a multilevel extension, which is desirable when the dimension of the coarse space becomes too large. On the other hand, since the entire solution process is defined on the substructures, less memory storage is required and it is not necessary to store the entire approximation array on each point of the (discrete) domain. For a three-dimensional problem with mesh size h, a discrete substructure array is of size $O(1/h^2)$. This is much smaller than $O(1/h^3)$, which is the size of an array corresponding to an approximation in volume. For this reason, the resulting

[1] Notice that the term "substructured" refers very often to DD methods that are defined on non-overlapping subdomains; see, e.g., [48,52]. However, in this work it indicates methods that are purely defined on the interfaces, independently of the type of (overlapping or non-overlapping) decomposition of the domain; see, e.g., [27, Section 5]. Here, we consider an overlapping domain decomposition.

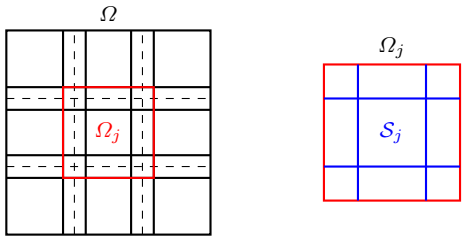

Fig. 1: Decomposition of a rectangular Ω into nine overlapping subdomains (left), and representation of the substructure $\mathcal{S}_j$ for the central subdomain (right).

interface restriction and prolongation operations are generally much cheaper and the dimension of the coarse space is much smaller.

This paper is organized as follows. In Section 2, we formulate the classical parallel Schwarz method in a substructured form. This is done at the continuous level and represents the starting point for the G2S method introduced in Section 3, where also a convergence analysis is presented for two subdomains in 2d. In Section 4, we discuss implementation details and multilevel extensions of the G2S method. Extensive numerical experiments are presented in Section 5, where the robustness of the proposed methods with respect to mesh refinement and physical parameters is studied. Finally, we present our conclusions in Section 6.

2 Substructured Schwarz domain decomposition methods

Consider a bounded Lipschitz domain $\Omega \subset \mathbb{R}^d$ for $d \in \{2, 3\}$, a general second-order linear elliptic operator $\mathcal{L}$ and a function $f \in L^2(\Omega)$. Our goal is to introduce new domain-decomposition based methods for the efficient numerical solution of the general linear elliptic problem

$$\mathcal{L}u = f \text{ in } \Omega, \; u = 0 \text{ on } \partial\Omega, \tag{1}$$

which we assume to be uniquely solved by a $u \in H_0^1(\Omega)$.

To formulate our methods, we need to fix some notation. Given a bounded set Γ with boundary $\partial\Gamma$, we denote by $\rho_\Gamma(x)$ the distance of $x \in \Gamma$ from $\partial\Gamma$. The space $H_{00}^{1/2}(\Gamma)$ is then defined as

$$H_{00}^{1/2}(\Gamma) := \{v \in H^{1/2}(\Gamma) \, : \, v/\rho_\Gamma^{1/2} \in L^2(\Gamma)\}, \tag{2}$$

and it is also known as the Lions-Magenes space; see, e.g., [46,48,51]. Equivalently, $H_{00}^{1/2}(\Gamma)$ can be defined as the space of functions in $H^{1/2}(\Gamma)$ such that their extensions by zero to a superset $\widetilde{\Gamma}$ of Γ are in $H^{1/2}(\widetilde{\Gamma})$; see, e.g., [51].

Next, consider a decomposition of Ω into N overlapping Lipschitz subdomains Ω_j, that is $\Omega = \cup_{j\in\mathcal{I}}\Omega_j$ with $\mathcal{I} := \{1, 2, \ldots, N\}$. For any $j \in \mathcal{I}$, we define the set of neighboring indexes $\mathcal{N}_j := \{\ell \in \mathcal{I} \, : \, \Omega_j \cap \partial\Omega_\ell \neq \emptyset\}$. Notice that $j \notin \mathcal{N}_j$, and $\cup_{j\in\mathcal{I}}\mathcal{N}_j = \mathcal{I}$. Given a $j \in \mathcal{I}$, we introduce the substructure of Ω_j defined as $\mathcal{S}_j := \cup_{\ell\in\mathcal{N}_j}\big(\partial\Omega_\ell \cap \Omega_j\big)$, that is the union of all portions of $\partial\Omega_\ell$ intersecting

with Ω_j with $\ell \in \mathcal{N}_j$.[2] The sets $\mathcal{S}_j$ are open and their closures are $\overline{\mathcal{S}_j} = \mathcal{S}_j \cup \partial\mathcal{S}_j$, with $\partial\mathcal{S}_j := \cup_{\ell\in\mathcal{N}_j}\big(\partial\Omega_j \cap \partial\Omega_\ell\big)$. The substructure of Ω is defined as $\mathcal{S} := \cup_{j\in\mathcal{I}}\overline{\mathcal{S}_j}$. Fig. 1 provides an illustration of substructures corresponding to a commonly used decomposition of a rectangular domain. We denote by $\mathcal{E}_j^0 : L^2(\mathcal{S}_j) \to L^2(\mathcal{S})$ the extension by zero operator. Now, we consider a set of continuous functions $\chi_j : \overline{\mathcal{S}_j} \to [0,1]$, $j = 1, \ldots, N$, such that

$$\chi_j(x) \in \begin{cases} (0,1] & \text{for } x \in \mathcal{S}_j, \\ \{1\} & \text{for } x \in \overline{\mathcal{S}_j \setminus \cup_{\ell\in\mathcal{N}_j}\mathcal{S}_\ell}, \\ \{0\} & \text{for } x \in \partial\mathcal{S}_j \setminus \partial\Omega, \end{cases}$$

and $\sum_{j\in\mathcal{I}} \mathcal{E}_j^0\chi_j \equiv 1$, which means that the functions χ_j form a partition of unity. Further, we assume that the functions χ_j, $j \in \mathcal{I}$, satisfy the condition $\chi_j/\rho_{\mathcal{S}_j}^{1/2} \in L^\infty(\mathcal{S}_j)$. This is satisfied, for example, in the case of Fig. 1 with piecewise linear partition of unity function χ_j.

For any $j \in \mathcal{I}$, we define $\Gamma_j^{\text{int}} := \partial\Omega_j \cap \big(\cup_{\ell\in\mathcal{N}_j}\Omega_\ell\big)$ and introduce the following trace and restriction operators

$$\tau_j : H^1(\Omega_j) \to H^{1/2}(\mathcal{S}_j) \text{ and } \tau_j^{\text{int}} : H^{1/2}(\mathcal{S}) \to H^{1/2}(\Gamma_j^{\text{int}}).$$

It is well known that (1) is equivalent to the domain decomposition system (see, e.g., [48])

$$\mathcal{L}u_j = f_j \text{ in } \Omega_j,\ u_j = \sum_{\ell\in\mathcal{N}_j} \mathcal{E}_\ell^0(\chi_\ell\tau_\ell u_\ell) \text{ on } \Gamma_j^{\text{int}},\ u_j = 0 \text{ on } \partial\Omega_j \setminus \Gamma_j^{\text{int}}, \tag{3}$$

where $f_j \in L^2(\Omega_j)$ is the restriction of f on Ω_j. Notice that $\chi_\ell\tau_\ell u_\ell$ lies in $H_{00}^{1/2}(\mathcal{S}_\ell)$, $\mathcal{E}_\ell^0(\chi_\ell\tau_\ell u_\ell) \in H^{1/2}(\mathcal{S})$. Moreover, for $\ell \in \mathcal{N}_j$, it holds that $\tau_j^{\text{int}}\mathcal{E}_\ell^0(\chi_\ell\tau_\ell u_\ell) \in H_{00}^{1/2}(\Gamma_j^{\text{int}})$ if $\Gamma_j^{\text{int}} \subsetneq \partial\Omega_j$, and $\tau_j^{\text{int}}\mathcal{E}_\ell^0(\chi_\ell\tau_\ell u_\ell) \in H^{1/2}(\Gamma_j^{\text{int}})$ if $\Gamma_j^{\text{int}} = \partial\Omega_j$.

Given a $j \in \mathcal{I}$ such that $\partial\Omega_j \setminus \Gamma_j^{\text{int}} \neq \emptyset$, we define the extension operator $\mathcal{E}_j : H_{00}^{1/2}(\Gamma_j^{\text{int}}) \times L^2(\Omega_j) \to H^1(\Omega_j)$ as $w = \mathcal{E}_j(v, f_j)$, where w solves the problem

$$\mathcal{L}w = f_j \text{ in } \Omega_j,\ w = v \text{ on } \Gamma_j^{\text{int}},\ w = 0 \text{ on } \partial\Omega_j \setminus \Gamma_j^{\text{int}} \tag{4}$$

for a $v \in H_{00}^{1/2}(\Gamma_j^{\text{int}})$. Otherwise, if $\Gamma_j^{\text{int}} \equiv \partial\Omega_j$, we define $\mathcal{E}_j : H^{1/2}(\Gamma_j^{\text{int}}) \times L^2(\Omega_j) \to H^1(\Omega_j)$ as $w = \mathcal{E}_j(v, f_j)$, where w solves the problem

$$\mathcal{L}w = f_j \text{ in } \Omega_j,\ w = v \text{ on } \Gamma_j^{\text{int}}, \tag{5}$$

for a $v \in H^{1/2}(\Gamma_j^{\text{int}})$.

The domain decomposition system (3) can be then written as

$$u_j = \mathcal{E}_j(0, f_j) + \mathcal{E}_j\Big(\tau_j^{\text{int}} \sum_{\ell\in\mathcal{N}_j} \mathcal{E}_\ell^0(\chi_\ell\tau_\ell u_\ell), 0\Big),\ j \in \mathcal{I}. \tag{6}$$

If we define $v_j := \chi_j\tau_j u_j$, $j \in \mathcal{I}$, then system (6) becomes

$$v_j = g_j + \sum_{\ell\in\mathcal{N}_j} G_{j,\ell}(v_\ell),\ j \in \mathcal{I}, \tag{7}$$

[2] Notice that the substructure of a subdomain is sometimes called "skeleton"; see, e.g., [17].

where $g_j := \chi_j \tau_j \mathcal{E}(0, f_j)$ and the operators $G_{j,\ell} : H_{00}^{1/2}(\mathcal{S}_\ell) \to H_{00}^{1/2}(\mathcal{S}_j)$ are defined as

$$G_{j,\ell}(\cdot) := \chi_j \tau_j \mathcal{E}_j\big(\tau_j^{\text{int}} \mathcal{E}_\ell^0(\cdot), 0\big). \tag{8}$$

System (7) is the substructured form of (3). The equivalence between (3) and (7) is explained by the following theorem.

Theorem 1 (Relation between (3) and (7)) *Let $u_j \in H^1(\Omega_j)$, $j \in \mathcal{I}$, solve (3), then $v_j := \chi_j \tau_j u_j$, $j \in \mathcal{I}$, solve (7). Let $v_j \in H_{00}^{1/2}(\mathcal{S}_j)$, $j \in \mathcal{I}$, solve (7), then $u_j := \mathcal{E}_j(\tau_j^{\text{int}} \sum_{\ell \in \mathcal{N}_j} \mathcal{E}_\ell^0(v_\ell), f_j)$, $j \in \mathcal{I}$, solve (3).*

Proof The first statement is proved before Theorem 1, where the substructured system (7) is derived. To obtain the second statement, we use (7) and the definition of u_j to write $v_j = \chi_j \tau_j \mathcal{E}_j(\tau_j^{\text{int}} \sum_{\ell \in \mathcal{N}_j} \mathcal{E}_\ell^0(v_\ell), f_j) = \chi_j \tau_j u_j$. The claim follows by using this equality together with the definitions of u_j and $\mathcal{E}_j$.

Take any function $w \in H_0^1(\Omega)$ and consider the initialization $u_j^0 := w|_{\Omega_j}$, $j \in \mathcal{I}$. The parallel Schwarz method (PSM) is then given by

$$\mathcal{L} u_j^n = f_j \text{ in } \Omega_j,\ u_j^n = \sum_{\ell \in \mathcal{N}_j} \mathcal{E}_\ell^0(\chi_\ell \tau_\ell u_\ell^{n-1}) \text{ on } \Gamma_j^{\text{int}},\ u_j^n = 0 \text{ on } \partial\Omega_j \setminus \Gamma_j^{\text{int}}, \tag{9}$$

for $n \in \mathbb{N}^+$, and has the substructured form

$$v_j^n = g_j + \sum_{\ell \in \mathcal{N}_j} G_{j,\ell}(v_\ell^{n-1}),\ j \in \mathcal{I}, \tag{10}$$

initialized by $v_j^0 := \chi_j \tau_j u_j^0 \in H_{00}^{1/2}(\mathcal{S}_j)$. Notice that the iteration (10) is well posed in the sense that $v_j^n \in H_{00}^{1/2}(\mathcal{S}_j)$ for $j \in \mathcal{I}$ and $n \in \mathbb{N}$. Equations (10) and (7) allow us to obtain the substructured PSM in error form, that is

$$e_j^n = \sum_{\ell \in \mathcal{N}_j} G_{j,\ell}(e_\ell^{n-1}),\ j \in \mathcal{I}, \tag{11}$$

for $n \in \mathbb{N}^+$, where $e_j^n := v_j - v_j^n$, for $j \in \mathcal{I}$ and $n \in \mathbb{N}$. Equation (7) can be written in the matrix form $A\mathbf{v} = \mathbf{b}$, where $\mathbf{v} = [v_1, \dots, v_N]^\top$, $\mathbf{b} = [g_1, \dots, g_N]^\top$ and the entries of A are

$$[A]_{j,j} = I_{d,j} \text{ and } [A]_{j,\ell} = -G_{j,\ell},\ j, \ell \in \mathcal{I},\ j \neq \ell, \tag{12}$$

where $I_{d,j}$ are the identities on $L^2(\mathcal{S}_j)$, $j \in \mathcal{I}$. Similarly, we define the operator G as

$$[G]_{j,j} = 0 \text{ and } [G]_{j,\ell} = G_{j,\ell},\ j, \ell \in \mathcal{I},\ j \neq \ell,$$

which allows us to write equations (10) and (11) as $\mathbf{v}^n = G\mathbf{v}^{n-1} + \mathbf{b}$ and $\mathbf{e}^n = G\mathbf{e}^{n-1}$, respectively, where $\mathbf{v}^n := [v_1^n, \dots, v_N^n]^\top$ and $\mathbf{e}^n := [e_1^n, \dots, e_N^n]^\top$. Notice that $G = \mathbb{I} - A$, where $\mathbb{I} := \text{diag}_{j=1,\dots,N}(I_{d,j})$. Moreover, we wish to remark that neither the operator A nor G is necessarily symmetric.

If the iteration $\mathbf{v}^n = G\mathbf{v}^{n-1} + \mathbf{b}$ converges, then the limit is the solution to the problem $A\mathbf{v} = \mathbf{b}$. From a numerical point of view, this is not necessarily true if the (discretized) subproblems (9) are not solved exactly. For this reason, we assume in what follows that the subproblems (9) are always solved exactly.

3 G2S: Geometric two-level substructured DD method

In this section, we introduce our G2S method. The main drawback of many two-level DD methods (including our two-level G2S method) is that the dimension of the coarse space can grow for increasing number of subdomains. This situation becomes even worse if the basis functions are not "good enough", a fact that would require an even larger dimension of the coarse space. In this case, the extension from two-level to multi-level framework would be suitable. These comments lead to the following questions. Is it possible to avoid the explicit construction of a coarse space? Is there any practical way to implicitly define a coarse space? Can one define a framework in which an extension of the two-level method to a multi-level framework is possible and easy?

In this section, we answer the above questions by introducing the so-called Geometric 2-level Substructured (G2S) method, which is a two-grid-type method (allowing a multi-grid generalization). This is detailed in Section 3.1. The corresponding convergence analysis for a two-subdomain case is presented in Section 3.2.2.

3.1 Description of the G2S method

Let us consider a discretization of the substructures such that $\mathcal{S}_j$ is approximated by a mesh of N_j points, $j \in \mathcal{I}$. The discrete substructures are denoted by $\mathcal{S}_j^{N_j}$, $j \in \mathcal{I}$. An example is given in Figure 2 (left). Moreover, we set $N^s := \sum_{j\in\mathcal{I}} N_j$.

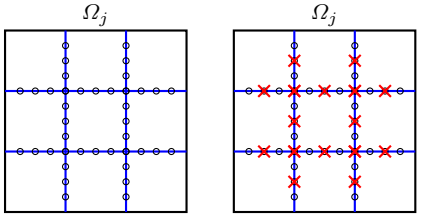

Fig. 2: Left: The subdomain Ω_j as in Figure 1, its substructure $\mathcal{S}_j$ (blue lines) and the corresponding discrete substructure $\mathcal{S}_j^{N_j}$ (black circles). Right: The coarse discrete substructure $\mathcal{S}_j^{M_j}$ is marked by red crosses.

The corresponding finite-dimensional discretization of the operators $G_{j,\ell}$ in (8) are denoted by $G_{h,j,\ell} \in \mathbb{R}^{N_j \times N_\ell}$. Similarly as in (12), we define the block operators $A_h \in \mathbb{R}^{N^s \times N^s}$ and $G_h \in \mathbb{R}^{N^s \times N^s}$ as

$$\begin{aligned} &[A_h]_{j,j} = I_{h,j},\ [A_h]_{j,\ell} = -G_{h,j,\ell},\ j,\ell \in \mathcal{I},\ j \neq \ell, \\ &[G_h]_{j,j} = 0,\ [G_h]_{j,\ell} = G_{h,j,\ell},\ j,\ell \in \mathcal{I},\ j \neq \ell, \end{aligned} \tag{13}$$

where $I_{h,j} \in \mathbb{R}^{N_j \times N_j}$ are identity matrices. Notice that $A_h = \mathbb{I}_h - G_h$, where $\mathbb{I}_h = \mathrm{diag}(I_{h,1}, \ldots, I_{h,N})$. Therefore, the substructured problem $A\mathbf{v} = \mathbf{b}$ becomes

$$A_h \mathbf{v} = \mathbf{b}_h,$$

Algorithm 1 Two-level substructured domain decomposition method

Require: $\mathbf{v}^0$ (initial guess)
1: $\mathbf{v}^n = G_h\mathbf{v}^{n-1} + \mathbf{b}_h$, $n = 1, \dots, n_1$ (DD pre-smoothing steps)
2: $\mathbf{r} = \mathbf{b}_h - A_h\mathbf{v}^{n_1}$ (compute the residual)
3: Solve $A_{2h}\mathbf{v}_c = R\mathbf{r}$ (solve the coarse problem)
4: $\mathbf{v}^0 = \mathbf{v}^{n_1} + P\mathbf{v}_c$ (coarse correction)
5: $\mathbf{v}^n = G_h\mathbf{v}^{n-1} + \mathbf{b}_h$, $n = 1, \dots, n_2$ (DD post-smoothing steps)
6: Set $\mathbf{v}^0 = \mathbf{v}^{n_2}$ (update)
7: Repeat from 1 to 6 until convergence

where $\mathbf{b}_h = [\mathbf{b}_{h,1}, \dots, \mathbf{b}_{h,N}]$, and the PSM is then

$$\mathbf{v}^n = G_h\mathbf{v}^{n-1} + \mathbf{b}_h. \tag{14}$$

The matrices G_h and $A_h = \mathbb{I}_h - G_h$ are not necessarily symmetric and never assembled explicitly. Instead their action of given vectors is computed directly. Notice that the computation of the action of $G_{h,j,\ell}$ on a given vector requires a subdomain solve. We insist on the fact that this subdomain solve is performed exactly. Furthermore, if the discrete PSM (14) converges, then $\rho(G_h) < 1$ and the matrix A_h is invertible.

Next, we introduce coarser discretizations $\mathcal{S}_j^{M_j}$, $j \in \mathcal{I}$, where the jth substructure is discretized with $M_j < N_j$ points. An example is given in Figure 2 (right). The total number of discrete coarse points is $M^s := \sum_{j\in\mathcal{I}} M_j$. For each $j \in \mathcal{I}$ we introduce restriction and prolongation matrices $R_j \in \mathbb{R}^{M_j \times N_j}$ and $P_j \in \mathbb{R}^{N_j \times M_j}$. These could be classical interpolation operators used, e.g., in multi-grid methods. If for example $\mathcal{S}_j$ is a one-dimensional interval, then the prolongation matrix can be chosen as

$$P_j := \begin{bmatrix} \frac{1}{2} & 1 & \frac{1}{2} & & \cdots & \\ & \frac{1}{2} & 1 & \frac{1}{2} & \cdots & \\ & & \frac{1}{2} & \cdots & & \\ & & & \cdots & \frac{1}{2} & \\ & & & \cdots & \frac{1}{2} & 1 & \frac{1}{2} \end{bmatrix}^\top, \tag{15}$$

and the corresponding restriction matrix would be the full weighting restriction operator $R_j := \frac{1}{2}P_j^\top$. The global restriction and prolongation matrices are defined as $R := \mathrm{diag}(R_1, \dots, R_N) \in \mathbb{R}^{M^s \times N^s}$ and $P := \mathrm{diag}(P_1, \dots, P_N) \in \mathbb{R}^{N^s \times M^s}$. The restriction of A_h on the coarse level is then defined as $A_{2h} := RA_hP$. Notice that this matrix can be either precomputed exactly or assembled in an approximate way. For more details see Section 4.1.

Remark 1 Notice that, for the definition of the G2S method, fine and coarse meshes need not to be nested and the sets $\mathcal{S}_j^{M_j}$ need not to coincide on overlapping areas. In this manuscript, we work with nested meshes. The case of non-nested meshes is beyond the scope of this paper and will be the subject of future work. Moreover, it is natural to include cross points in both fine and coarse discrete substructures, since these are generally the corners of the (overlapping) subdomains (see, e.g., Fig. 2 and 7).

The G2S procedure is defined by the following Algorithm 1, where n_1 and n_2 are the numbers of the pre- and post-smoothing steps. This is a classical two-grid type iteration, but instead of having the classical grids in volume, we consider

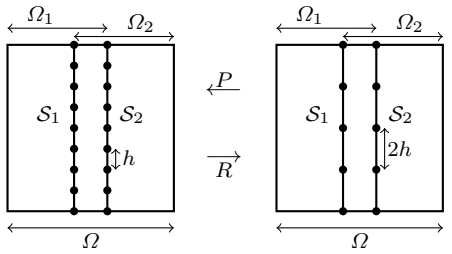

Fig. 3: Two-subdomain decomposition, substructures and their discretizations.

two discrete levels on the substructures. This has the advantage of performing all restriction and interpolation operations of smaller coarse problems. More details are given in Section 4.1. We insist on the fact that the G2S method does not require the explicit construction of a coarse space V_c, but it exploits directly a discretization of the interfaces. Moreover, it is clear that a simple recursion allows us to embed the G2S method into a multi-grid framework. Further implementation details are discussed in Section 4.

Formally, one iteration of our G2S method can be represented as

$$\mathbf{v}^{\text{new}} = G_h^{n_2}(\mathbb{I}_h - PA_{2h}^{-1}RA_h)G_h^{n_1}\mathbf{v}^{\text{old}} + \widetilde{M}\mathbf{b}_h, \tag{16}$$

where $\widetilde{M}$ is a matrix which acts on the right-hand side vector $\mathbf{b}_h$ and which can be regarded as the preconditioner corresponding to our two-level method. In error form, the iteration (16) becomes

$$\mathbf{e}^{\text{new}} = T_h\mathbf{e}^{\text{old}} \text{ with } T_h := G_h^{n_2}(\mathbb{I}_h - PA_{2h}^{-1}RA_h)G_h^{n_1},$$

where $\mathbf{e}^{\text{new}} := \mathbf{v} - \mathbf{v}^{\text{new}}$ and $\mathbf{e}^{\text{old}} := \mathbf{v} - \mathbf{v}^{\text{old}}$.

3.2 Analysis of the G2S method

In this section, we analyze the convergence of the G2S method. To do so, we recall our model problem (1) and assume a two-subdomain decomposition $\Omega = \Omega_1 \cup \Omega_2$ such that the two substructures $\mathcal{S}_1$ and $\mathcal{S}_2$ are two segments of the same length $\widetilde{L}$. Notice that in this case the substructures coincide with the interfaces. An example for Ω equal to a rectangle is given in Figure 3. For a given $\ell \in \mathbb{N}^+$, $\ell \geq 2$, we discretize (1) using a uniform grid of $N_h = 2^\ell - 1$ points on each substructure (without counting the two points on $\partial\Omega$) so that the grid size is $h = \frac{\widetilde{L}}{N_h+1}$. Notice that $N_h = N_1 = N_2$, where N_j are used in Section 3.1 to denote the number of discretization points of the substructures. We also introduce a coarser mesh of $N_c = 2^{\ell-1} - 1$ points on each substructure and mesh size $h_c = \frac{\widetilde{L}}{N_c+1}$. We define the geometric prolongation operator $P \in \mathbb{R}^{2N_h \times 2N_c}$ as $P := \text{diag}(\widetilde{P}, \widetilde{P})$, where $\widetilde{P} = P_1 = P_2$ is the matrix given in (15). The operator $R \in \mathbb{R}^{2N_c \times 2N_h}$ is defined as $R := \text{diag}(\widetilde{R}, \widetilde{R})$, where $\widetilde{R}$ is the full weighting restriction matrix $\widetilde{R} := \frac{1}{2}\widetilde{P}^\top$. Due to the special decomposition into two subdomains, let us simplify the notation defining $G_{h,1} := G_{h,1,2}$ and $G_{h,2} := G_{h,2,1}$, that is the action of $G_{h,j}$

represents a subdomain solution in the j-th subdomain. We suppose that the operators $G_{h,1}$ and $G_{h,2}$ have eigenvectors $\boldsymbol{\psi}_k$ with eigenvalues $\rho_j(k)$, $k = 1, \dots, N_h$, $j = 1, 2$. Here, $\boldsymbol{\psi}_k$ are discrete Fourier modes given by $(\boldsymbol{\psi}_k)_j = \sin(k\pi hj)$, for $j, k = 1, \dots, N_h$. Notice that $\boldsymbol{\psi}_\ell^\top \boldsymbol{\psi}_k = \delta_{\ell,k} \frac{N_c+1}{2}$, with $\delta_{\ell,k}$ the Kronecker delta.

It is well-known that the actions of $\widetilde{R}$ and $\widetilde{P}$ on the combination of a low-frequency mode $\boldsymbol{\psi}_k$ with its high-frequency companion $\boldsymbol{\psi}_{\widetilde{k}}$, with $\widetilde{k} = N_h - k + 1$, are

$$\widetilde{R}\left[\boldsymbol{\psi}_k \ \boldsymbol{\psi}_{\widetilde{k}}\right] = \boldsymbol{\phi}_k \left[c_k^2 \ -s_k^2\right], \quad \widetilde{P}\boldsymbol{\phi}_k = (c_k^2 \boldsymbol{\psi}_k - s_k^2 \boldsymbol{\psi}_{\widetilde{k}}) = \left[\boldsymbol{\psi}_k \ \boldsymbol{\psi}_{\widetilde{k}}\right] \begin{bmatrix} c_k^2 \\ -s_k^2 \end{bmatrix}, \tag{17}$$

where $c_k = \cos(k\pi \frac{h}{2})$, $s_k = \sin(k\pi \frac{h}{2})$ for $k = 1, \dots, N_c$ and $(\boldsymbol{\phi}_k)_j = \sin(k\pi 2hj)$, for $k = 1, \dots, \frac{N_h+1}{2} - 1$ and $j = 1, \dots, \frac{N_h+1}{2} - 1 = N_c$ (notice that the two points on $\partial\Omega$ are excluded); see, e.g., [42,13]. The vectors $\boldsymbol{\phi}_k$ are Fourier modes on the coarse grid. As before, the coarse matrix is $A_{2h} = RA_hP$, and the G2S iteration operator is $T_h = G_h^{n_2}(I - PA_{2h}^{-1}RA_h)G_h^{n_1}$.

In the following subsections, we prove that G2S method is well posed and convergent. Well-posedness follows by the invertibility of the coarse matrix A_{2h}, which is proved in Section 3.2.1 along with an interpretation of the G2S method. A detailed convergence analysis is presented in Section 3.2.2, where sharp estimates of the spectral radius of T_h are derived under certain assumptions. Finally, in Section 3.2.3 we discuss the relations between our G2S method and a classical two-grid method in volume using the PSM as a smoother.

3.2.1 Interpretation of G2S as a general two-level method

Let us begin by considering any invertible matrix $U \in \mathbb{R}^{2N_c \times 2N_c}$ and compute

$$\begin{aligned} T_h &= G_h^{n_2}(I - PA_{2h}^{-1}RA_h)G_h^{n_1} \\ &= G_h^{n_2}(I - PUU^{-1}(RA_hP)^{-1}UU^{-1}RA_h)G_h^{n_1} \\ &= G_h^{n_2}(I - PUU^{-1}[U(U^{-1}RA_hPU)U^{-1}]^{-1}UU^{-1}RA_h)G_h^{n_1} \\ &= G_h^{n_2}(I - PU(U^{-1}RA_hPU)^{-1}U^{-1}RA_h)G_h^{n_1} \\ &= G_h^{n_2}(I - \widehat{P}\widehat{A}_{2h}^{-1}\widehat{R}A_h)G_h^{n_1} =: \widehat{T}_h, \end{aligned} \tag{18}$$

where $\widehat{P} := PU$, $\widehat{R} = U^{-1}R$ and $\widehat{A}_{2h} := \widehat{R}A_h\widehat{P}$.

Let us define the orthogonal matrices $\Phi = \frac{2}{N_c+1}[\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_{N_c}]$ and $U := \mathrm{diag}(\Phi, \Phi)$, and the operators $\widehat{P} := PU$, $\widehat{R} = U^\top R$[3] and $\widehat{A}_{2h} := \widehat{R}A_h\widehat{P}$. Notice that the columns of $\widehat{P} := PU$ are the vectors spanning the space

$$V_c = (\mathrm{span}_{k=1,\dots,N_c}\{\widetilde{P}\boldsymbol{\phi}_k\})^2 = (\mathrm{span}_{k=1,\dots,N_c}\{c_k^2\boldsymbol{\psi}_k - s_k^2\boldsymbol{\psi}_{\widetilde{k}}\})^2 \subset \mathbb{R}^{2N_h}, \tag{19}$$

where the relation (17) is used. This means that the G2S method can be written as a two-level method characterized by an iteration operator $\widehat{T}_h$ defined via the

[3] Notice that $(\widehat{P})^\top = U^\top P^\top = 2U^\top R = 2\widehat{R}$, since $\widetilde{R} = \frac{1}{2}\widetilde{P}^\top$.

prolongation and restriction operators $\widehat{P}$ and $\widehat{R}$. Moreover, in this case the actions of $\widehat{P}$ and $\widehat{R}$ on two vectors can be expressed by

$$\begin{aligned}\widehat{P}\begin{bmatrix}\mathbf{v}\\ \mathbf{w}\end{bmatrix} &= \left[\sum_{k=1}^{N_c}(\mathbf{v})_k\widetilde{P}\boldsymbol{\phi}_k,\ \sum_{k=1}^{N_c}(\mathbf{w})_k\widetilde{P}\boldsymbol{\phi}_k\right]^\top,\\ \widehat{R}\begin{bmatrix}\mathbf{f}\\ \mathbf{g}\end{bmatrix} &= \left[\langle\tfrac{1}{2}\widetilde{P}\boldsymbol{\phi}_1,\mathbf{f}\rangle,\cdots,\langle\tfrac{1}{2}\widetilde{P}\boldsymbol{\phi}_{N_c},\mathbf{f}\rangle,\langle\tfrac{1}{2}\widetilde{P}\boldsymbol{\phi}_1,\mathbf{g}\rangle,\cdots,\langle\tfrac{1}{2}\widetilde{P}\boldsymbol{\phi}_{N_c},\mathbf{g}\rangle\right]^\top,\end{aligned} \tag{20}$$

for any $\mathbf{v},\mathbf{w}\in\mathbb{R}^{N_c}$ and any $\mathbf{f},\mathbf{g}\in\mathbb{R}^{N_h}$, where $\langle\cdot,\cdot\rangle$ denotes the usual Euclidean scalar product.

Now, we turn our attention to the matrix A_{2h}, whose invertibility is proved in the following lemmas.

Lemma 1 (Invertibility of a coarse operator A_c) *Let $(\mathcal{X}_j,\langle\cdot,\cdot\rangle_j)$, $j=1,2$ be two inner-product spaces. Define the space $\mathcal{X}:=\mathcal{X}_2\times\mathcal{X}_1$ endowed with the inner product $\langle(a,b),(c,d)\rangle:=\langle a,c\rangle_2+\langle b,d\rangle_1$ for all $(a,b),(c,d)\in\mathcal{X}$. Consider some bases $\{\psi_\ell^j\}_{\ell\in\mathbb{N}}\subset\mathcal{X}_j$, $j=1,2$. Let V_c be a finite-dimensional subspace of $\mathcal{X}$ given by the span of the basis vectors $(\psi_1^2,0),\dots,(\psi_m^2,0)$ and $(0,\psi_1^1),\dots,(0,\psi_m^1)$, for a finite integer $m>0$. Let $\mathbb{P}_{V_c}$ be the orthogonal projection operator onto V_c. Consider an invertible operator $A:\mathcal{X}\to\mathcal{X}$ and the matrix $A_c=\mathcal{R}A\mathcal{P}\in\mathbb{R}^{2m\times 2m}$, where $\mathcal{P}$ and $\mathcal{R}$ are defined as*

$$\begin{aligned}\mathcal{P}\begin{bmatrix}\mathbf{v}\\ \mathbf{w}\end{bmatrix} &:= \left[\sum_{k=1}^{m}(\mathbf{v})_k\psi_k^2,\ \sum_{k=1}^{m}(\mathbf{w})_k\psi_k^1\right]^\top,\\ \mathcal{R}\begin{bmatrix}f\\ g\end{bmatrix} &:= \left[\langle\psi_1^2,f\rangle_2,\cdots,\langle\psi_m^2,f\rangle_2,\langle\psi_1^1,g\rangle_1,\cdots,\langle\psi_m^1,g\rangle_1\right]^\top.\end{aligned} \tag{21}$$

Then A_c has full rank if and only if $\mathbb{P}_{V_c}(A\mathbf{v})\neq 0\ \forall\mathbf{v}\in V_c\setminus\{0\}$.

Proof We first show that if $\mathbb{P}_{V_c}(A\mathbf{v})\neq 0$ for any $\mathbf{v}\in V_c\setminus\{0\}$, then $A_c=\mathcal{R}A\mathcal{P}$ has full rank. This result follows from the rank-nullity theorem, if we show that the only element in the kernel of A_c is the zero vector. To do so, we recall the definitions of $\mathcal{P}$ and $\mathcal{R}$ given in (21). Clearly, $\mathcal{P}\mathbf{z}=0$ if and only if $\mathbf{z}=0$. For any $\mathbf{z}\in\mathbb{R}^{2m}$ the vector $\mathcal{P}\mathbf{z}$ is in V_c. Since A is invertible, then $A\mathcal{P}\mathbf{z}=0$ if and only if $\mathbf{z}=0$. Moreover, by our assumption it holds that $\mathbb{P}_{V_c}(A\mathcal{P}\mathbf{z})\neq 0$. Now, we notice that $\mathcal{R}\mathbf{w}\neq 0$ for all $\mathbf{w}\in V_c\setminus\{0\}$, and $\mathcal{R}\mathbf{w}=0$ for all $\mathbf{w}\in V_c^\perp$, where $V_c^\perp$ denotes the orthogonal complement of V_c in $\mathcal{X}$ with respect to $\langle\cdot,\cdot\rangle$. Since $(\mathcal{X},\langle\cdot,\cdot\rangle)$ is an inner-product space, we have $A\mathcal{P}\mathbf{z}=\mathbb{P}_{V_c}(A\mathcal{P}\mathbf{z})+(I-\mathbb{P}_{V_c})(A\mathcal{P}\mathbf{z})$ with $(I-\mathbb{P}_{V_c})(A\mathcal{P}\mathbf{z})\in V_c^\perp$. Hence, $\mathcal{R}A\mathcal{P}\mathbf{z}=\mathcal{R}\mathbb{P}_{V_c}(A\mathcal{P}\mathbf{z})\neq 0$ for any non-zero $\mathbf{z}\in\mathbb{R}^{2m}$.

Now we show that, if $A_c=\mathcal{R}A\mathcal{P}$ has full rank, then $\mathbb{P}_{V_c}(A\mathbf{v})\neq 0$ for any $\mathbf{v}\in V_c\setminus\{0\}$. We proceed by contraposition and prove that if there exists a $\mathbf{v}\in V_c\setminus\{0\}$ such that $A\mathbf{v}\in V_c^\perp$, then $A_c=\mathcal{R}A\mathcal{P}$ has not full rank. Assume that there is a $\mathbf{v}\in V_c\setminus\{0\}$ such that $A\mathbf{v}\in V_c^\perp$. Since $\mathbf{v}$ is in V_c, there exists a nonzero vector $\mathbf{z}\in\mathbb{R}^{2m}$ such that $\mathbf{v}=\mathcal{P}\mathbf{z}$. Hence $A\mathcal{P}\mathbf{z}\in V_c^\perp$. We can now write that $A_c\mathbf{z}=\mathcal{R}(A\mathcal{P}\mathbf{z})=0$, which implies that A_c has not full rank.

Lemma 2 (Invertibility of A_{2h}) *Assume that $\rho_1(k),\rho_2(k)\in[0,1)$ for all k and that $\rho_1(k)\geq\rho_1(\widetilde{k})$ and $\rho_2(k)\geq\rho_2(\widetilde{k})$ for any $k=1,\dots,N_c$ and $\widetilde{k}=N_h-k+1$. The matrix $A_{2h}:=RA_hP\in\mathbb{R}^{2N_c\times 2N_c}$ has full rank.*

Proof Since $A_{2h} = U\widehat{A}_{2h}U^\top$, it is enough to show that $\widehat{A}_{2h}$ is invertible. To do so, we recall that $\widehat{A}_{2h} = \widehat{R}A_h\widehat{P}$ and we wish to prove that for any $\mathbf{z} \in V_c \setminus \{0\}$ (with V_c defined in (19)) it holds $\mathbb{P}_{V_c}(A_h\mathbf{z}) \neq 0$ and then invoke Lemma 1. Here the orthogonality is understood with respect to the classical scalar product of $\mathbb{R}^{2N_h}$. First, it is possible to show that the orthogonal complement of V_c is

$$V_c^\perp = (\text{span}_{k=1,\dots,N_c}\{c_k^{-2}\boldsymbol{\psi}_k + s_k^{-2}\boldsymbol{\psi}_{\widetilde{k}}, \boldsymbol{\psi}_{(N_h+1)/2}\})^2.$$

Notice that $\dim(V_c) = 2N_c$, $\dim(V_c^\perp) = 2(N_c+1)$, and $\dim(V_c)+\dim(V_c^\perp) = 2N_h$, since $N_h = 2N_c + 1$.

Since the vectors spanning V_c in (19) are orthogonal, we have $\mathbb{P}_{V_c}(\mathbf{w}) = \widehat{P}\widehat{P}^\top\mathbf{w}$ for any $\mathbf{w} \in \mathbb{R}^{2N_h}$, Since $\widehat{P}$ has full rank, to prove that $\mathbb{P}_{V_c}(A_h\mathbf{z}) \neq 0$ for any $\mathbf{z} \in V_c \setminus \{0\}$ it is sufficient to show that $\widehat{P}^\top A_h\mathbf{v} \neq 0$ holds for any column $\mathbf{v}$ of $\widehat{P}$, that is any element of the form $[(\widetilde{P}\boldsymbol{\phi}_k)^\top, (\widetilde{P}\boldsymbol{\phi}_\ell)^\top]^\top$. Therefore, we use (17) and compute

$$\begin{aligned} A_h\begin{bmatrix}\widetilde{P}\boldsymbol{\phi}_k\\ \widetilde{P}\boldsymbol{\phi}_\ell\end{bmatrix} &= A_h\begin{bmatrix}c_k^2\boldsymbol{\psi}_k - s_k^2\boldsymbol{\psi}_{\widetilde{k}}\\ c_\ell^2\boldsymbol{\psi}_\ell - s_\ell^2\boldsymbol{\psi}_{\widetilde{\ell}}\end{bmatrix} = \begin{bmatrix}c_k^2\boldsymbol{\psi}_k - s_k^2\boldsymbol{\psi}_{\widetilde{k}} - (\rho_1(\ell)c_\ell^2\boldsymbol{\psi}_\ell - \rho_1(\widetilde{\ell})s_\ell^2\boldsymbol{\psi}_{\widetilde{\ell}})\\ c_\ell^2\boldsymbol{\psi}_\ell - s_\ell^2\boldsymbol{\psi}_{\widetilde{\ell}} - (\rho_2(k)c_k^2\boldsymbol{\psi}_k - \rho_2(\widetilde{k})s_k^2\boldsymbol{\psi}_{\widetilde{k}})\end{bmatrix}\\ &= \begin{bmatrix}\widetilde{P}\boldsymbol{\phi}_k - \rho_1(\ell)\widetilde{P}\boldsymbol{\phi}_\ell - (\rho_1(\ell) - \rho_1(\widetilde{\ell}))s_\ell^2\boldsymbol{\psi}_{\widetilde{\ell}}\\ \widetilde{P}\boldsymbol{\phi}_\ell - \rho_2(k)\widetilde{P}\boldsymbol{\phi}_k - (\rho_2(k) - \rho_2(\widetilde{k}))s_k^2\boldsymbol{\psi}_{\widetilde{k}}\end{bmatrix}, \end{aligned} \tag{22}$$

for any $k, \ell = 1, \dots, N_c$, where $\widetilde{k} = N_h - k + 1$ and $\widetilde{\ell} = N_h - \ell + 1$, for $k, \ell = 1, \dots, N_c$. Now, a direct calculation shows that

$$s_k^2\boldsymbol{\psi}_{\widetilde{k}} = \underbrace{-\frac{s_k^4}{s_k^4 + c_k^4}\widetilde{P}\boldsymbol{\phi}_k}_{\mathbb{P}_{V_c}(s_k^2\boldsymbol{\psi}_{\widetilde{k}})} + \underbrace{\frac{1}{s_k^{-4} + c_k^{-4}}(c_k^{-2}\boldsymbol{\psi}_k + s_k^{-2}\boldsymbol{\psi}_{\widetilde{k}})}_{\mathbb{P}_{V_c^\perp}(s_k^2\boldsymbol{\psi}_{\widetilde{k}})},$$

for any $k = 1, \dots, N_c$. Inserting this equality into (22), multiplying to the left with $[(\widetilde{P}\boldsymbol{\phi}_k)^\top, (\widetilde{P}\boldsymbol{\phi}_\ell)^\top]$ and using (17) together with the orthogonality relation $\boldsymbol{\psi}_\ell^\top\boldsymbol{\psi}_k = \delta_{\ell,k}\frac{N_c+1}{2}$, we obtain for $k \neq \ell$ that

$$\begin{bmatrix}\widetilde{P}\boldsymbol{\phi}_k\\ \widetilde{P}\boldsymbol{\phi}_\ell\end{bmatrix}^\top A_h\begin{bmatrix}\widetilde{P}\boldsymbol{\phi}_k\\ \widetilde{P}\boldsymbol{\phi}_\ell\end{bmatrix} = \|\widetilde{P}\boldsymbol{\phi}_k\|_2^2 + \|\widetilde{P}\boldsymbol{\phi}_\ell\|_2^2 \neq 0.$$

Similarly, for $k = \ell$ we obtain that

$$\begin{bmatrix}\widetilde{P}\boldsymbol{\phi}_k\\ \widetilde{P}\boldsymbol{\phi}_\ell\end{bmatrix}^\top A_h\begin{bmatrix}\widetilde{P}\boldsymbol{\phi}_k\\ \widetilde{P}\boldsymbol{\phi}_\ell\end{bmatrix} = \Big(2 - (\rho_1(k) - \rho_2(k)) + \frac{s_k^4(\rho_2(k) - \rho_2(\widetilde{k}) + \rho_1(k) - \rho_1(\widetilde{k}))}{s_k^4 + c_k^4}\Big)\|\widetilde{P}\boldsymbol{\phi}_k\|_2^2.$$

A direct calculation using the assumptions on $\rho_j(k)$ shows that this is nonzero.

3.2.2 Convergence of the G2S method

The previous section focused on the well-posedness of the method. In particular, we proved Lemma 2 that guarantees that A_{2h} is invertible and that the G2S method is well posed. In this section, our attention is turned to the analysis of the G2S convergence behavior. This is performed by studying the spectral properties of the G2S iteration operator. Our first key result is the following technical lemma.

Lemma 3 *Consider the G2S matrix* $T_h := G_h^{n_2}(I - PA_{2h}^{-1}RA_h)G_h^{n_1}$. *The action of* T_h *on* $\begin{bmatrix} \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0 \\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} \end{bmatrix}$ *is given by*

$$T_h \begin{bmatrix} \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0 \\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0 \\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} \end{bmatrix} \widetilde{G}_k, \tag{23}$$

where $\widetilde{G}_k := D_{n_2}(k)(D_{n_1}(k) - V(k)\Lambda_2^{-1}(k)\Lambda_1(k))$ *with*

$$\Lambda_1(k) := V(k)^\top H(k) D_{n_1}(k), \quad \Lambda_2(k) := V(k)^\top H(k) V(k),$$

$$V(k) := \begin{bmatrix} c_k^2 & 0 \\ -s_k^2 & 0 \\ 0 & c_k^2 \\ 0 & -s_k^2 \end{bmatrix}, \qquad H(k) := \begin{bmatrix} 1 & 0 & -\rho_1(k) & 0 \\ 0 & 1 & 0 & -\rho_1(\widetilde{k}) \\ -\rho_2(k) & 0 & 1 & 0 \\ 0 & -\rho_2(\widetilde{k}) & 0 & 1 \end{bmatrix},$$

and $D_n(k)$ *is given by*

$$D_n(k) := \begin{bmatrix} \pi(k)^n & 0 & 0 & 0 \\ 0 & \pi(\widetilde{k})^n & 0 & 0 \\ 0 & 0 & \pi(k)^n & 0 \\ 0 & 0 & 0 & \pi(\widetilde{k})^n \end{bmatrix}, \quad D_n(k) := \begin{bmatrix} 0 & 0 & \pi_{21}(k,n) & 0 \\ 0 & 0 & 0 & \pi_{21}(\widetilde{k},n) \\ \pi_{12}(k,n) & 0 & 0 & 0 \\ 0 & \pi_{12}(\widetilde{k},n) & 0 & 0 \end{bmatrix}$$

for n *even and for* n *odd, respectively, whose entries are* $\pi(k) := (\rho_1(k)\rho_2(k))^{1/2}$, $\pi_{12}(k,n) := \rho_1(k)^{\frac{n-1}{2}}\rho_2(k)^{\frac{n+1}{2}}$, *and* $\pi_{21}(k,n) := \rho_1(k)^{\frac{n+1}{2}}\rho_2(k)^{\frac{n-1}{2}}$.

Proof We consider the case in which both n_1 and n_2 are even. The other cases can be obtained by similar arguments. Since n_1 is even, we have that

$$G_h^{n_1} = \begin{bmatrix} (G_{h,1}G_{h,2})^{n_1/2} & 0 \\ 0 & (G_{h,2}G_{h,1})^{n_1/2} \end{bmatrix}.$$

Because of the relation $(G_{h,1}G_{h,2})^{n_1/2}\boldsymbol{\psi}_k = (G_{h,2}G_{h,1})^{n_1/2}\boldsymbol{\psi}_k = \pi(k)^{n_1}\boldsymbol{\psi}_k$, where $\pi(k) := (\rho_1(k)\rho_2(k))^{1/2}$, we get

$$G_h^{n_1}\begin{bmatrix} \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0 \\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0 \\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} \end{bmatrix} \begin{bmatrix} \pi(k) & 0 & 0 & 0 \\ 0 & \pi(\widetilde{k}) & 0 & 0 \\ 0 & 0 & \pi(k) & 0 \\ 0 & 0 & 0 & \pi(\widetilde{k}) \end{bmatrix}^{n_1} = \begin{bmatrix} \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0 \\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} \end{bmatrix} D_{n_1}(k).$$

Similarly, we obtain that $G_h^{n_2}\begin{bmatrix}\boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0\\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0\\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}}\end{bmatrix} D_{n_2}(k)$. Moreover, direct calculations reveal that

$$A_h\begin{bmatrix}\boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0\\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}}\end{bmatrix}=\begin{bmatrix}\boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0\\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}}\end{bmatrix}\begin{bmatrix}1 & 0 & -\rho_1(k) & 0\\ 0 & 1 & 0 & -\rho_1(\widetilde{k})\\ -\rho_2(k) & 0 & 1 & 0\\ 0 & -\rho_2(\widetilde{k}) & 0 & 1\end{bmatrix}=\begin{bmatrix}\boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0\\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}}\end{bmatrix}H(k) \tag{24}$$

and

$$R\begin{bmatrix}\boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0\\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\phi}_k & 0\\ 0 & \boldsymbol{\phi}_k\end{bmatrix}\begin{bmatrix}c_k^2 & -s_k^2 & 0 & 0\\ 0 & 0 & c_k^2 & -s_k^2\end{bmatrix} = \begin{bmatrix}\boldsymbol{\phi}_k & 0\\ 0 & \boldsymbol{\phi}_k\end{bmatrix}V(k)^\top, \tag{25}$$

where we used (17). It follows that $RA_hG_h^{n_1}\begin{bmatrix}\boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0\\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\phi}_k & 0\\ 0 & \boldsymbol{\phi}_k\end{bmatrix}\Lambda_1(k)$. Let us now study the action of the coarse matrix A_{2h} on $\begin{bmatrix}\boldsymbol{\phi}_k & 0\\ 0 & \boldsymbol{\phi}_k\end{bmatrix}$. We use (17), (24) and (25) to write

$$\begin{aligned}A_{2h}\begin{bmatrix}\boldsymbol{\phi}_k & 0\\ 0 & \boldsymbol{\phi}_k\end{bmatrix} &= RA_hP\begin{bmatrix}\boldsymbol{\phi}_k & 0\\ 0 & \boldsymbol{\phi}_k\end{bmatrix} = RA_h\begin{bmatrix}\boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0\\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}}\end{bmatrix}V(k)\\ &= R\begin{bmatrix}\boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0\\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}}\end{bmatrix}H(k)V(k) = \begin{bmatrix}\boldsymbol{\phi}_k & 0\\ 0 & \boldsymbol{\phi}_k\end{bmatrix}V(k)^\top H(k)V(k).\end{aligned}$$

Thus, we have $A_{2h}\begin{bmatrix}\boldsymbol{\phi}_k & 0\\ 0 & \boldsymbol{\phi}_k\end{bmatrix} = \begin{bmatrix}\boldsymbol{\phi}_k & 0\\ 0 & \boldsymbol{\phi}_k\end{bmatrix}\Lambda_2(k)$, and since A_{2h} is invertible by Lemma 2, we get

$$\begin{bmatrix}\boldsymbol{\phi}_k & 0\\ 0 & \boldsymbol{\phi}_k\end{bmatrix} = A_{2h}^{-1}\begin{bmatrix}\boldsymbol{\phi}_k & 0\\ 0 & \boldsymbol{\phi}_k\end{bmatrix}\Lambda_2(k). \tag{26}$$

A direct calculation reveals that the eigenvalues of $\Lambda_2(k)$ are $\lambda_{1,2} = c_k^4 + s_k^4 \pm \sqrt{(c_k^4\rho_1(k) + s_k^4\rho_1(\widetilde{k}))(c_k^4\rho_2(k) + s_k^4\rho_2(\widetilde{k}))}$ and they are nonzero for $k = 1, \dots, N_c$. Hence, $\Lambda_2(k)$ is invertible and, using (26), we get

$$A_{2h}^{-1}\begin{bmatrix}\boldsymbol{\phi}_k & 0\\ 0 & \boldsymbol{\phi}_k\end{bmatrix}\Lambda_1(k) = A_{2h}^{-1}\begin{bmatrix}\boldsymbol{\phi}_k & 0\\ 0 & \boldsymbol{\phi}_k\end{bmatrix}\Lambda_2(k)\Lambda_2^{-1}(k)\Lambda_1(k) = \begin{bmatrix}\boldsymbol{\phi}_k & 0\\ 0 & \boldsymbol{\phi}_k\end{bmatrix}\Lambda_2^{-1}(k)\Lambda_1(k),$$

Summarizing our results and using the definition of T_h, we conclude that

$$T_h\begin{bmatrix}\boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0\\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0\\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}}\end{bmatrix}D_{n_2}(k)\left[D_{n_1}(k) - \begin{bmatrix}c_k^2 & 0\\ -s_k^2 & 0\\ 0 & c_k^2\\ 0 & -s_k^2\end{bmatrix}\Lambda_2^{-1}(k)\Lambda_1(k)\right]$$

and our claim follows.

Using Lemma 3, it is possible to factorize the iteration matrix T_h. This factorization is obtained in the following theorem.

Theorem 2 (Factorization of the iteration matrix T_h) *There exists an invertible matrix Q such that $T_h = Q\widetilde{G}Q^{-1}$, where the G2S iteration matrix T_h is defined in Lemma 3 and*

$$\widetilde{G} = \begin{bmatrix} \widetilde{G}_1 & & & & \\ & \ddots & & & \\ & & \widetilde{G}_{N_c} & & \\ & & & \gamma_1(\frac{N_h+1}{2}) & \\ & & & & \gamma_2(\frac{N_h+1}{2}) \end{bmatrix},$$

where the matrices $\widetilde{G}_k \in \mathbb{R}^{4\times 4}$ are defined in Lemma 3 and $\gamma_j(\frac{N_h+1}{2})$ depend on n_1, n_2 and the eigenvalues $\rho_j(\frac{N_h+1}{2})$ of $G_{h,j}$, for $j = 1, 2$.

Proof We define the invertible matrix

$$Q := \begin{bmatrix} \boldsymbol{\psi}_1 & \boldsymbol{\psi}_{N_h} & 0 & 0 & \cdots & \boldsymbol{\psi}_{N_c} & \boldsymbol{\psi}_{N_c+2} & 0 & 0 & \boldsymbol{\psi}_{\frac{N_h+1}{2}} & 0 \\ 0 & 0 & \boldsymbol{\psi}_1 & \boldsymbol{\psi}_{N_h} & \cdots & 0 & 0 & \boldsymbol{\psi}_{N_c} & \boldsymbol{\psi}_{N_c+2} & 0 & \boldsymbol{\psi}_{\frac{N_h+1}{2}} \end{bmatrix}.$$

Equation (23) says that $T_h \begin{bmatrix} \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0 \\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} & 0 & 0 \\ 0 & 0 & \boldsymbol{\psi}_k & \boldsymbol{\psi}_{\widetilde{k}} \end{bmatrix} \widetilde{G}_k$, for every $k = 1, \ldots, N_c = \frac{N_h+1}{2} - 1$ and $\widetilde{k} = N_h - k + 1$. Moreover, the frequency $\boldsymbol{\psi}_{\frac{N_h+1}{2}}$ is mapped to zero by the restriction operator, $R \begin{bmatrix} \boldsymbol{\psi}_{\frac{N_h+1}{2}} & 0 \\ 0 & \boldsymbol{\psi}_{\frac{N_h+1}{2}} \end{bmatrix} = 0$, and we get

$$T_h \begin{bmatrix} \boldsymbol{\psi}_{\frac{N_h+1}{2}} & 0 \\ 0 & \boldsymbol{\psi}_{\frac{N_h+1}{2}} \end{bmatrix} = G_h^{n_2} G_h^{n_1} \begin{bmatrix} \boldsymbol{\psi}_{\frac{N_h+1}{2}} & 0 \\ 0 & \boldsymbol{\psi}_{\frac{N_h+1}{2}} \end{bmatrix} = \begin{bmatrix} \gamma_1(\frac{N_h+1}{2})\boldsymbol{\psi}_{\frac{N_h+1}{2}} & 0 \\ 0 & \gamma_2(\frac{N_h+1}{2})\boldsymbol{\psi}_{\frac{N_h+1}{2}} \end{bmatrix},$$

where the expressions of $\gamma_1(\frac{N_h+1}{2})$ and $\gamma_2(\frac{N_h+1}{2})$ depend on n_1 and n_2. For instance, if n_1+n_2 is an even number, then $\gamma_1(\frac{N_h+1}{2}) = \gamma_2(\frac{N_h+1}{2}) := (\rho_1(\frac{N_h+1}{2})\rho_2(\frac{N_h+1}{2}))^{\frac{n_1+n_2}{2}}$. Hence, we conclude that $T_hQ = Q\widetilde{G}$ and our claim follows.

The factorization of T_h proved in Theorem 2 allows one to obtain accurate convergence results of a G2S method. Clearly, an optimal result would be a direct calculation of the spectral radii of the matrices $\widetilde{G}_k$. However, this is in general a difficult task that requires cumbersome calculations. Nevertheless, in Theorem 3 we obtain an explicit expression for the spectral radii of $\widetilde{G}_k$ under some reasonable assumptions that are in general satisfied in case of Schwarz methods and symmetric decompositions; see, e.g., [29, Section 3]. Notice also that Theorem 3 guarantees that only one (pre- or post-) smoothing step is necessary for the G2S method to converge.

Theorem 3 *Assume that $1 > \rho_1(k) = \rho_2(k) = \rho(k) \geq 0$ for any k and that $\rho(k)$ is a decreasing function of k. The convergence factor of the G2S method is*

$$\rho_{G2S}(T_h) = \max_{k\in\{1,\ldots,N_c,\frac{N_h+1}{2}\}} \left(\frac{c_k^4(1-\rho(k))\rho(\widetilde{k})^{n_1+n_2} + s_k^4(1-\rho(\widetilde{k}))\rho(k)^{n_1+n_2}}{c_k^4(1-\rho(k)) + s_k^4(1-\rho(\widetilde{k}))} \right) < 1.$$

Proof The convergence factor of the G2S is given by the spectral radius of the iteration matrix T_h. Theorem 2 implies that

$$\rho_{G2S}(T_h) = \max\Big\{\max_{k\in\{1,\dots,N_c\}} \rho(\widetilde{G}_k), \gamma_1\left(\frac{N_h+1}{2}\right), \gamma_2\left(\frac{N_h+1}{2}\right)\Big\}.$$

Regardless of the values of n_1 and n_2, direct calculations show that the matrices $\widetilde{G}_k$ have four eigenvalues:

$$\begin{aligned}
\lambda_1(k) &= \lambda_2(k) = 0,\\
|\lambda_3(k)| &= \frac{c_k^4(1-\rho(k))\rho(\widetilde{k})^{n_1+n_2} + s_k^4(1-\rho(\widetilde{k}))\rho(k)^{n_1+n_2}}{c_k^4(1-\rho(k)) + s_k^4(1-\rho(\widetilde{k}))},\\
|\lambda_4(k)| &= \frac{c_k^4(1+\rho(k))\rho(\widetilde{k})^{n_1+n_2} + s_k^4(1+\rho(\widetilde{k}))\rho(k)^{n_1+n_2}}{c_k^4(1+\rho(k)) + s_k^4(1+\rho(\widetilde{k}))}.
\end{aligned}$$

Moreover, we observe that

$$|\lambda_3(k)| - |\lambda_4(k)| = \frac{2c_k^4 s_k^4(\rho(k)-\rho(\widetilde{k}))(\rho(k)^{n_1+n_2} - \rho(\widetilde{k})^{n_1+n_2})}{((\rho(k)+1)c_k^4 + s_k^4(\rho(\widetilde{k})+1))((1-\rho(k))c_k^4 + s_k^4(1-\rho(\widetilde{k})))} \geq 0,$$

where we used the monotonicity of $\rho(k)$. On the other hand, since $\rho_1(k) = \rho_2(k) = \rho(k)$, we have $\gamma_1(\frac{N_h+1}{2}) = \gamma_2(\frac{N_h+1}{2}) = \rho(\frac{N_h+1}{2})^{n_1+n_2}$. Therefore, we have that

$$\max\left\{\max_{k\in\{1,\dots,N_c\}} \rho(\widetilde{G}_k), \rho\left(\frac{N_h+1}{2}\right)^{n_1+n_2}\right\} = \max\left\{\max_{k\in\{1,\dots,N_c\}} |\lambda_3(k)|, \rho\left(\frac{N_h+1}{2}\right)^{n_1+n_2}\right\},$$

and the result follows by observing that $\lambda_3\left(\frac{N_h+1}{2}\right) = \rho\left(\frac{N_h+1}{2}\right)^{n_1+n_2}$, since $\rho(\widetilde{k}) = \rho(k)$ for $k = \frac{N_h+1}{2}$.

3.2.3 Two-level substructured and volumetric methods

At this stage, it is fair to pose the following questions: What is the difference between our G2S and other two-level DD methods? Is our G2S different from a classical two-grid method that uses a PSM as smoother? Is there any relation between these two apparently similar approaches? The answers are given in this section.

Let $A_v\mathbf{u} = \mathbf{f}$ be a discretization of our problem (1). In particular, $A_v \in \mathbb{R}^{N^v\times N^v}$ is the discretization of the elliptic operator $\mathcal{L}$, while $\mathbf{u} \in \mathbb{R}^{N^v}$ and $\mathbf{f} \in \mathbb{R}^{N^v}$ are the discrete counterparts of the solution u and the right-hand side function f. Consider the following splittings of the matrix A_v:

$$A_v = \begin{bmatrix} A_1 & E_1\widehat{R}_1 \\ \times & \times \end{bmatrix} = \begin{bmatrix} \times & \times \\ E_2\widehat{R}_2 & A_2 \end{bmatrix},$$

where $A_j \in \mathbb{R}^{N_j^a\times N_j^a}$ for $j = 1, 2$. Notice that these correspond to a two-subdomain decomposition. We assume that A_v, A_1 and A_2 are invertible. The matrices $\widehat{R}_1 \in \mathbb{R}^{N_1\times(N^v-N_1^a)}$ and $\widehat{R}_2 \in \mathbb{R}^{N_2\times(N^v-N_2^a)}$ are restriction operators that take as input vectors of sizes $N^v - N_1^a$ and $N^v - N_2^a$ and returns as output substructure vectors

of sizes N_1 (substructure $\mathcal{S}_1$) and N_2 (substructure $\mathcal{S}_2$). The two matrices $E_1 \in \mathbb{R}^{N_1^a \times N_1}$ and $E_2 \in \mathbb{R}^{N_2^a \times N_2}$ are extension by zero operators. In order to obtain a discrete substructured problem, we introduce the augmented system

$$A_a \mathbf{u}_a = \mathbf{f}_a, \tag{27}$$

where $A_a = \begin{bmatrix} A_1 & E_1R_1 \\ E_2R_2 & A_2 \end{bmatrix}$, $\mathbf{u}_a = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}$, and $\mathbf{f}_a = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}$, with $A_j \in \mathbb{R}^{N_j^a \times N_j^a}$ and $\mathbf{u}_j, \mathbf{f}_j \in \mathbb{R}^{N_j^a}$, for $j = 1, 2$. The matrices $R_1 \in \mathbb{R}^{N_1 \times N_2^a}$ and $R_2 \in \mathbb{R}^{N_2 \times N_1^a}$ are restriction operators that map volume vectors, of sizes N_2^a (second subdomain) and N_1^a (first subdomain), respectively, to substructure vectors, of sizes N_1 (substructure $\mathcal{S}_1$) and N_2 (substructure $\mathcal{S}_2$), respectively. Notice that $R_j R_j^\top = I_{N_j}$, with I_{N_j} the identity of size N_j, for $j = 1, 2$. Moreover, we define $N^s := N_1 + N_2$ and $N^a := N_1^a + N_2^a$.

The substructure vectors $\mathbf{v}_{21} := R_1\mathbf{u}_2$ and $\mathbf{v}_{12} := R_2\mathbf{u}_1$ solve the discrete substructured system

$$A_h \begin{bmatrix} \mathbf{v}_{12} \\ \mathbf{v}_{21} \end{bmatrix} = \begin{bmatrix} R_2A_1^{-1}\mathbf{f}_1 \\ R_1A_2^{-1}\mathbf{f}_2 \end{bmatrix}, \tag{28}$$

where $A_h = \begin{bmatrix} I_{N_2} & R_2A_1^{-1}E_1 \\ R_1A_2^{-1}E_2 & I_{N_1} \end{bmatrix}$. The vectors $\mathbf{v}_{12}$ and $\mathbf{v}_{21}$ are restrictions on the substructures $\mathcal{S}_2$ and $\mathcal{S}_1$ of the solution vectors $\mathbf{u}_1$ and $\mathbf{u}_2$, and (28) is the substructured form of (27). Notice that (28) is the discrete counterpart of the substructured problem (10).

The block-Jacobi method applied to (27) and (28) leads to the iteration matrices

$$G_a = \begin{bmatrix} 0 & -A_1^{-1}E_1R_1 \\ -A_2^{-1}E_2R_2 & 0 \end{bmatrix} \quad \text{and} \quad G_h = \begin{bmatrix} 0 & -R_2A_1^{-1}E_1 \\ -R_1A_2^{-1}E_2 & 0 \end{bmatrix},$$

where G_h is the discretization of G, as denoted in Sections 3.1 and 3.2.

Let us now introduce the matrices

$$D := \begin{bmatrix} A_1^{-1} & 0 \\ 0 & A_2^{-1} \end{bmatrix}, \quad \widetilde{T} := \begin{bmatrix} R_2 & 0 \\ 0 & R_1 \end{bmatrix} \quad \text{and} \quad \widetilde{E} := \begin{bmatrix} 0 & E_1 \\ E_2 & 0 \end{bmatrix}.$$

It is easy to verify the relations

$$\widetilde{T}\widetilde{T}^\top = I_{N^s}, \quad A_h\widetilde{T} = \widetilde{T}DA_a, \quad G_a = -D\widetilde{E}\widetilde{T} \quad \text{and} \quad G_h\widetilde{T} = \widetilde{T}G_a. \tag{29}$$

In particular, the relation $\widetilde{T}\widetilde{T}^\top = I_{N^s}$ is trivial, and $A_h\widetilde{T} = \widetilde{T}DA_a$ can be obtained by calculating

$$\begin{aligned} \widetilde{T}DA_a &= \begin{bmatrix} R_2 & 0 \\ 0 & R_1 \end{bmatrix} \begin{bmatrix} A_1^{-1} & 0 \\ 0 & A_2^{-1} \end{bmatrix} \begin{bmatrix} A_1 & E_1R_1 \\ E_2R_2 & A_2 \end{bmatrix} = \begin{bmatrix} R_2 & 0 \\ 0 & R_1 \end{bmatrix} \begin{bmatrix} I_{N_1^a} & A_1^{-1}E_1R_1 \\ A_2^{-1}E_2R_2 & I_{N_2^a} \end{bmatrix} \\ &= \begin{bmatrix} R_2 & R_2A_1^{-1}E_1R_1 \\ R_1A_2^{-1}E_2R_2 & R_1 \end{bmatrix} = \begin{bmatrix} I_{N_2} & R_2A_1^{-1}E_1 \\ R_1A_2^{-1}E_2 & I_{N_1} \end{bmatrix} \begin{bmatrix} R_2 & 0 \\ 0 & R_1 \end{bmatrix} = A_h\widetilde{T}. \end{aligned}$$

A similar calculation allows us to obtain that $G_h\widetilde{T} = \widetilde{T}G_a$.

Since the matrices G_h and G_a are two different representations of the PSM, one expects that their spectra coincide. This is shown in the next lemma.

Lemma 4 *The matrices $G_h \in \mathbb{R}^{N^s \times N^s}$ and $G_a \in \mathbb{R}^{N^a \times N^a}$ have the same nonzero eigenvalues, that is $\sigma(G_h) = \sigma(G_a) \setminus \{0\}$.*

Proof Recalling the structure of G_a, one can clearly see that $\text{rank}(G_a) = N^s$, because the matrices $E_j R_j$ have rank N_j for $j = 1, 2$. Hence, G_a has N^s nonzero eigenvalues. Take any eigenvector $\mathbf{v} \in \mathbb{R}^{N^a}$ of G_a with eigenvalue $\lambda \neq 0$. We note that $\widetilde{T}\mathbf{v} \neq 0$, otherwise we would have $G_a\mathbf{v} = -D\widetilde{E}\widetilde{T}\mathbf{v} = 0$, which contradicts the hypothesis $\lambda \neq 0$. Using the last relation in (29), we write $G_h\widetilde{T}\mathbf{v} = \widetilde{T}G_a\mathbf{v} = \lambda\widetilde{T}\mathbf{v}$. Hence $(\widetilde{T}\mathbf{v}, \lambda)$ is an eigenpair of G_h. Since this holds for any eigenpair $(\mathbf{v}, \lambda)$ of G_a, the result follows.

Let us now consider arbitrary restriction and prolongation operators R_s and P_s (with $R_s = P_s^\top$). The corresponding discrete substructured two-level iteration matrix is then given by

$$G_h^{2L} := \big[I_{N^s} - P_s(R_s A_h P_s)^{-1} R_s A_h\big] G_h. \tag{30}$$

Our goal is to find a volumetric two-level iteration operator G_a^{2L} that has the same spectrum of G_h^{2L}. Such a volumetric operator must be formulated for the augmented system (27) and based on the iteration matrix G_a. Let us recall (29) and compute

$$\begin{aligned}
G_h^{2L}\widetilde{T} &= \big[I_{N^s} - P_s(R_s A_h P_s)^{-1} R_s A_h\big] G_h \widetilde{T} \\
&= \big[I_{N^s} - P_s(R_s A_h P_s)^{-1} R_s A_h\big] \widetilde{T} G_a \\
&= \big[\widetilde{T} - P_s(R_s A_h P_s)^{-1} R_s A_h \widetilde{T}\big] G_a \\
&= \widetilde{T}\big[I_{N^a} - \widetilde{T}^\top P_s(R_s A_h P_s)^{-1} R_s A_h \widetilde{T}\big] G_a \\
&= \widetilde{T}\big[I_{N^a} - \widetilde{T}^\top P_s(R_s A_h \widetilde{T}\widetilde{T}^\top P_s)^{-1} R_s \widetilde{T} D A_a\big] G_a \\
&= \widetilde{T}\big[I_{N^a} - \widetilde{T}^\top P_s(R_s \widetilde{T} D A_a \widetilde{T}^\top P_s)^{-1} R_s \widetilde{T} D A_a\big] G_a \\
&= \widetilde{T}\big[I_{N^a} - P_a(R_a D A_a P_a)^{-1} R_a D A_a\big] G_a = \widetilde{T} G_a^{2L},
\end{aligned}$$

where $P_a := \widetilde{T}^\top P_s$, $R_a := R_s\widetilde{T} = P_a^\top$ and

$$G_a^{2L} := \big[I_{N^a} - P_a(R_a D A_a P_a)^{-1} R_a D A_a\big] G_a. \tag{31}$$

We obtained that $G_h^{2L}\widetilde{T} = \widetilde{T}G_a^{2L}$. Similarly as in the proof of Lemma 4, one can show that $\sigma(G_h^{2L}) = \sigma(G_a^{2L}) \setminus \{0\}$. This means that we have found a two-level volumetric iteration operator that is spectrally equivalent to our substructured two-level operator. Moreover, for any invertible matrix $U \in \mathbb{R}^{N^a \times N^a}$ we can repeat the calculations done in (18), to obtain

$$G_a^{2L} = \big[I_{N^a} - \widetilde{P}_a(\widetilde{R}_a D A_a \widetilde{P}_a)^{-1} \widetilde{R}_a D A_a\big] G_a, \tag{32}$$

where $\widetilde{P}_a = P_a U$ and $\widetilde{R}_a = U^{-1} R_a$ (with $\widetilde{R}_a = \widetilde{P}_a^\top$ if U is orthogonal). This means that there exist many two-level DD methods in volume that are equivalent to our substructured two-level methods. We can summarize the obtained result in the following theorem.

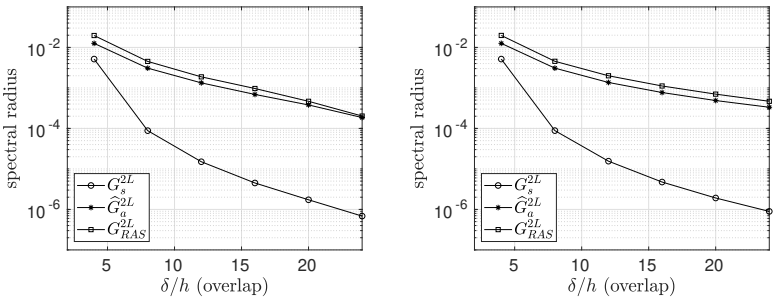

Fig. 4: Spectral radii of the matrices G_h^{2L}, $\widehat{G}_a^{2L}$ and G_{RAS}^{2L} and corresponding to $\ell = 5$ (left) and $\ell = 6$ (right).

Theorem 4 (Volumetric formulation of substructured methods) *Consider the substructured two-level iteration operator G_h^{2L} given in* (30) *and denote its spectrum by $\sigma(G_h^{2L})$. For any invertible matrix $U \in \mathbb{R}^{N^a \times N^a}$, the spectrum of the matrix G_a^{2L} given in* (32) *satisfies the relation $\sigma(G_h^{2L}) = \sigma(G_a^{2L}) \setminus \{0\}$.*

The matrix G_a^{2L} has a special structure. Since D is the block-Jacobi preconditioner for the augmented system (27), one can say that G_a^{2L} corresponds to a two-level method applied to the preconditioned system $DA_a\mathbf{u}_a = D\mathbf{f}_a$, in a similar spirit of the smoothed aggregation method defined in [5, Section 2].

Let us now focus on the question: what is the relation between our G2S method and a two-grid (volumetric) method that uses the same smoother (PSM)? A two-grid method in volume applied to the augmented system (27), would correspond to an iteration operator $\widehat{G}_a^{2L}$ of the form

$$\widehat{G}_a^{2L} = \big[I_{N^a} - \widehat{P}_a(\widehat{R}_a A_a \widehat{P}_a)^{-1}\widehat{R}_a A_a\big]G_a.$$

Natural choices for $\widehat{P}_a$ and $\widehat{R}_a$ are the usual (volumetric) restriction and prolongation operators. For example, for a one-dimensional problem a natural choice is the prolongation matrix $\widehat{P}_a$ given in (15) and $\widehat{R}_a = \frac{1}{2}P_a^\top$. On the other hand, our prolongation operator $P_a := \widetilde{T}^\top P_s$ is an extension by zero of a coarse substructure vector to a fine volumetric vector. Moreover, $R_a := R_s\widetilde{T}$ restricts a fine volumetric vector $\mathbf{v}$ to a coarse substructure vector by only interpolating the components of $\mathbf{v}$ belonging to the (fine) substructures. Another crucial difference is that G_a^{2L} is constructed on DA_a, while $\widehat{G}_a^{2L}$ is obtained using the matrix A_a. Therefore, $\widehat{G}_a^{2L}$ is constructed on the original augmented system $A_a\mathbf{u}_a = \mathbf{f}_a$, while G_a^{2L} is defined over the preconditioned system $DA_a\mathbf{u}_a = D\mathbf{f}_a$.

These facts indicate clearly that our method is by far distant from a classical volumetric two-grid method that uses the PSM as smoother. This is also confirmed by the numerical results shown in Figure 4, where the spectral radii of three different two-level iteration matrices are depicted. In particular, we consider the Laplace problem defined on a unit square Ω (of side $\widetilde{L} = 1$). This domain is decomposed into two overlapping rectangles of width $L = \frac{1}{2} + \delta$. Hence, the length of the overlap is 2δ. This problem is discretized using a classical second-order finite-difference scheme with a uniform grid of size $h = \frac{1}{N_h+1}$, where $N_h = 2^\ell - 1$.

The length of the overlap is $\delta = (N_{ov} + 1)h$, for some positive odd integer N_{ov}. We consider three different two-level iterations matrices G_h^{2L}, $\widehat{G}_a^{2L}$ and G_{RAS}^{2L}. The first one G_h^{2L} is the iteration matrix corresponding to our G2S method. The second one $\widehat{G}_a^{2L}$ is the iteration matrix of a two-level method applied on the augmented volumetric system (27). In both cases, the same classical Schwarz method is used as smoother. The third matrix G_{RAS}^{2L} is the iteration operator of a classical two-grid method applied to the volumetric system $A_v\mathbf{u} = \mathbf{f}$ and using as smoother the RAS method. In all cases, restriction and prolongation operators correspond to linear interpolation matrices (as in (15)) and to the full weighting restriction matrices, respectively. Indeed, for our G2S method these are one-dimensional operators, while for the other two methods they are two-dimensional operators. In particular, for the augmented system these interpolation and restriction operators take into account the non-zero values of the discrete functions on the substructures. For the two-level RAS method, they are obtained by a two-dimensional extension of (15).

In Figure 4, we show the spectral radii of G_h^{2L}, $\widehat{G}_a^{2L}$ and G_{RAS}^{2L}, obtained by a direct numerical computation, as a function of N_{ov}, hence the size of the overlap. The two figures correspond to two different discretizations. It is clear that our G2S method outperforms the other two methods, which have also very small contraction factors. Moreover, by comparing the two plots, we observe that the coarse correction makes all the methods very robust with respect to the number of discretization points.

4 Implementation details and multilevel algorithm

In Section 4.1, after having explained pro and cons of substructured and volume two-level methods, we reformulate Algorithm 1 in equivalent forms, which are essential to make our method computationally efficient. In Section 4.2, we explain how to extend our G2S method to a multi-grid strategy.

4.1 A practical form of two-level substructured methods

One of the advantages of our new substructured framework is that a large part of the computations are performed with objects (vectors, matrices, arrays, etc.) that are defined on the substructures and hence have very small sizes if compared to their volumetric counterparts. This is clear if one carefully studies Algorithm 1, where for example the products $R\mathbf{r}$ and $P\mathbf{v}_c$ are performed on substructure vectors. In volumetric two-level methods, the same prolongation and restriction operators involve volume entities, thus their application is more costly and they might be generally more difficult to implement due to the higher dimensions.

We now compare the computational costs of one iteration of the G2S and of a 2-grid method in volume that uses the same smoother. Let N^v be the size of the volume problem and N^s the size of the substructured problem ($N^s \ll N^v$). The size of each subdomain volume problem is N_{sub}. The coarse spaces are of dimension M^s for the G2S method and M^v for the volume method. The restriction and prolongation operators in volume are denoted by R_v and P_v. For simplicity we assume $n_1 = 1$, $n_2 = 0$.

The computational costs of one iteration are reported in Table 1. The first

G2S	G2S C.C.	Volume two-level	Volume C.C.
$\mathbf{v}^{n+\frac{1}{2}} = G_h\mathbf{v}^n + \mathbf{b}_h$	$O(\gamma_c(N_{\text{sub}}))$	$\mathbf{u}_v^{n+\frac{1}{2}} = N\mathbf{u}_v^n + M^{-1}\mathbf{b}_v$	$O(\gamma_c(N_{\text{sub}}))$
$\mathbf{r}^{n+\frac{1}{2}} = \mathbf{b}_h - A_h\mathbf{v}^{n+\frac{1}{2}}$	$O(\gamma_c(N_{\text{sub}}))$	$\mathbf{r}_v^{n+\frac{1}{2}} = \mathbf{b}_v - A_v\mathbf{u}_v^{n+\frac{1}{2}}$	$O((N^v)^{\gamma_m})$
$\mathbf{v}_c^{n+1} = A_{2h}^{-1}(R\mathbf{r}^{n+\frac{1}{2}})$	$O(\gamma_s(M^s))$	$\mathbf{u}_{vc}^{n+1} = A_{vc}^{-1}(R_v\mathbf{r}_v^{n+\frac{1}{2}})$	$O(\gamma_v(M^v))$
$\mathbf{v}^{n+1} = \mathbf{v}^{n+\frac{1}{2}} + P\mathbf{v}_c^{n+1}$	$O(N^s)$	$\mathbf{u}_v^{n+1} = \mathbf{u}_v^{n+\frac{1}{2}} + P_v\mathbf{u}_{vc}^{n+1}$	$O(N^v)$

Table 1: Computational cost (C.C.) per iteration. Notice that the smoother in volume is written as a standard stationary method based on the splitting $A_v = M - N$.

row of this table corresponds to the smoothing step performed by G2S and a DD method in volume. Since we assumed that both strategies use the same DD smoother, their computational costs coincide and are equal to $O(\gamma_c(N_{\text{sub}}))$, where γ_c depends on the choice of the linear solver. For example, for a Poisson problem, one has $\gamma_c(N_{\text{sub}}) = N_{\text{sub}}\log(N_{\text{sub}})$ if a fast Poisson solver is used, or $\gamma_c(N_{\text{sub}}) = bN_{\text{sub}}$ for sparse banded matrices with bandwidth b; see, e.g., [38]. For a general problem, the complexity of sparse direct solvers is a power of N_{sub}, which depends on the dimension. Moreover, one could consider the precomputation of the factorization of the subdomain matrices and just using forward and backward substitutions along the iterations.

For simplicity, we assume that restriction and prolongation operations are classical nodal operations whose computational costs are supposed to grow linearly with the dimension of the problem. Notice that since $N^s \ll N^v$, assuming that the same interpolation method is used, the cost in the substructured case is much lower than the corresponding cost in volume; see last row in Table 1.

Let us now discuss the third row of Table 1, which corresponds to the solution of the coarse problems. Since the dimension of the substructured coarse space is smaller, the G2S could require much less computational effort in the solution of the coarse problem. We remark that on the one hand, the coarse matrix A_{2h} is typically block sparse, where the block structure is related to the connectivity among the subdomains (namely the jth block-row of G_h has a sparsity pattern governed by the set $\mathcal{N}_j$). Furthermore, for a large class of PDE problems, these blocks admit very accurate low-rank approximations that can make the solution process more efficient; see, e.g., [43] and references therein. On the other hand, A_{vc} is typically a sparse matrix, whose sparsity pattern depends on the discretization method used (e.g., finite differences, finite elements, etc). In both cases there exist sophisticated algorithms for the solution of the corresponding linear systems; see, e.g., [38, 18, 43] and references therein. For this reason, we use the two functions γ_s and γ_v to indicate the computational cost of the coarse solvers. A general direct comparison in this sense is problem dependent, it could be very complicated, and it is beyond the scope of this paper. Nevertheless, we provide in Section 5.1 a detailed analysis for a specific test case.

Let us now turn our attention to the second row of Table 1, which corresponds to the computation of the residual. Here, a volumetric method requires $O((N^v)^{\gamma_m})$ operations, where γ_m depends on the sparsity structure of A_v. For example, if A_v is a second-order finite-difference matrix, then $\gamma_m = 1$. In contrast to this favorable situation, the computation of the residual for a G2S method requires the action of

Algorithm 2 G2S-B1

Require: $\mathbf{v}^0$ and $\widetilde{P} = G_h P$.
1: $\mathbf{v}^1 = G_h\mathbf{v}^0 + \mathbf{b}_h$,
2: $\mathbf{w} = G_h\mathbf{v}^1$,
3: $\mathbf{r} = \mathbf{b}_h - \mathbf{v}^1 + \mathbf{w}$,
4: $\mathbf{v}_c = A_{2h}^{-1} R\mathbf{r}$,
5: $\mathbf{v}^0 = \mathbf{v}^1 + P\mathbf{v}_c$,
Iterations:
6: $\mathbf{v}^1 = \mathbf{w} + \widetilde{P}\mathbf{v}_c + \mathbf{b}_h$,
7: $\mathbf{w} = G_h\mathbf{v}^1$,
8: $\mathbf{r} = \mathbf{b}_h - \mathbf{v}^1 + \mathbf{w}$,
9: $\mathbf{v}_c = A_{2h}^{-1} R\mathbf{r}$,
10: $\mathbf{v}^0 = \mathbf{v}^1 + P\mathbf{v}_c$,
11: Repeat 6 to 10 until convergence.

Algorithm 3 G2S-B2

Require: $\mathbf{v}^0$ and $\widetilde{P} = G_h P$.
1: $\mathbf{v} = G_h\mathbf{v}^0$,
2: $\mathbf{r} = \mathbf{b}_h - \mathbf{v}^0 + \mathbf{v}$,
3: $\mathbf{v}_c = A_{2h}^{-1} R\mathbf{r}$,
4: $\mathbf{v}^0 = \mathbf{w} + \widetilde{P}\mathbf{v}_c + \mathbf{b}_h$,
5: Repeat 1 to 5 until convergence.

A_h on a vector $\mathbf{v}^{n+\frac{1}{2}}$, which in turn requires a subdomain solve that is assumed to cost $O(\gamma_c(N_{\text{sub}}))$ (as discussed above). Hence, two smoothing steps are needed by the G2S method. If we could avoid this extra cost, then all the other steps of the G2S methods are cheaper since they are performed on arrays of much smaller sizes. Moreover, we wish to remark that, as we are going to see in Section 5, the G2S method requires in general less iterations than the corresponding method in volume. Hence, if we could avoid one of the two smoothing applications in each iteration, we would get a method which is faster in terms of iterations and computational cost per iteration. To avoid one of the two applications of the smoother in the G2S method, we exploit the special form of the matrix $A_h = \mathbb{I}_h - G_h$ and propose two new versions of Algorithm 1. These are called G2S-B1 and G2S-B2 and given by Algorithms 2 and 3. These substructured algorithms require only one smoothing step per iteration. Hence, they are potentially cheaper than a two-grid method using the same smoother. Moreover, it turns out that G2S and G2S-B1 are equivalent and they have the same spectral properties of G2S-B2. These relations are proved in the following theorem.

Theorem 5 (Equivalence between G2S, G2S-B2 and G2S-B1)

(a) *Algorithm 2 generates the same iterates of Algorithm 1.*
(b) *Algorithm 3 corresponds to the stationary iterative method*

$$\mathbf{v}^n = G_h(\mathbb{I}_h - PA_{2h}^{-1}RA_h)\mathbf{v}^{n-1} + \widetilde{M}\mathbf{b}_h,$$

where $G_h(\mathbb{I}_h - PA_{2h}^{-1}RA_h)$ is the iteration matrix and $\widetilde{M}$ the relative preconditioner. Moreover, Algorithm 3 and Algorithm 2 have the same convergence behavior.

Proof For simplicity, we suppose to work with the error equation and thus $\mathbf{b}_h = 0$. We call $\widetilde{\mathbf{v}}^0$ the output of the first five steps of Algorithm 2 and $\widehat{\mathbf{v}}^0$ the output of Algorithm 1. Then given an initial guess $\mathbf{v}^0$, we have

$$\begin{aligned}\widetilde{\mathbf{v}}^0 &= \mathbf{v}^1 + P\mathbf{v}_c = \mathbf{v}^1 + PA_{2h}^{-1}R(-\mathbf{v}^1 + \mathbf{w}) \\ &= G_h\mathbf{v}^0 + PA_{2h}^{-1}R(-A_hG_h\mathbf{v}^0) = (\mathbb{I}_h - PA_{2h}^{-1}RA_h)G_h\mathbf{v}^0 = \widehat{\mathbf{v}}^0.\end{aligned}$$

To verify that steps 6-10 of G2S-B1 are equivalent to an iteration of Algorithm 1, let $\mathbf{v}^{0,k-1} := \mathbf{v}^{1,k-1} + P\mathbf{v}_c$ be the output on line 10 of the $(k-1)$-th iteration of the G2S-B1 algorithm. Then the smoothing step on line 6 of the k-iteration reads

$$\mathbf{v}^{1,k} = G_h\mathbf{v}^{0,k-1} + \mathbf{b}_h = G_h\mathbf{v}^{1,k-1} + G_hP\mathbf{v}_c + \mathbf{b}_h = \mathbf{w}^{k-1} + \widetilde{P}\mathbf{v}_c + \mathbf{b}_h,$$

where we use the definition of $\widetilde{P}$ and the quantity $\mathbf{w}^{k-1}$ computed at the previous iteration. Steps 7-8 are just the residual computation using the identity $A_h = I - G_h$ and the remaining steps are standard. For the second part of the Theorem, we write one iteration of Algorithm 3 as

$$\mathbf{v}^1 = \mathbf{w} + \widetilde{P}\mathbf{v}_c = G_h\mathbf{v}^0 + G_hPA_{2h}^{-1}R(-A_h\mathbf{v}^0) = G_h(\mathbb{I}_h - PA_{2h}^{-1}RA_h)\mathbf{v}^0.$$

Hence, Algorithm 3 performs a post-smoothing step instead of a pre-smoothing step as Algorithm 2 does. The method still has the same convergence behavior since the matrices $G_h(\mathbb{I}_h - PA_{2h}^{-1}RA_h)$ and $(\mathbb{I}_h - PA_{2h}^{-1}RA_h)G_h$ have the same eigenvalues[4].

Notice that Algorithm 2 requires for the first iteration two applications of the smoothing operator G_h. The next iterations, given by Steps 6-10, need only one application of the smoothing operator G_h. Theorem 5 (a) shows that Algorithm 2 is equivalent to Algorithm 1. This means that each iteration of Algorithm 2 after the first one is computationally less expensive than one iteration of a volume two-level DD method. Since two-level DD methods perform generally few iterations, it could be important to get rid of the expensive first iteration. For this reason, we introduce Algorithm 3, which overcomes the problem of the first iteration. Theorem 5 (b) guarantees that Algorithm 3 is exactly a G2S method with no pre-smoothing and one post-smoothing step. Moreover, it has the same convergence behavior of Algorithm 2.

We wish to remark that, the reformulations G2S-B1 and G2S-B2 require to store the (substructured) matrix $\widetilde{P} := G_hP$. This matrix is anyway computed in a pre-computation phase to assemble the coarse matrix $A_{2h} = RA_hP = RP - RG_hP = RP - R\widetilde{P}$. Hence, no extra cost is required. These implementation tricks can be readily generalized to a general number of pre- and post-smoothing steps.

Concerning the specific implementation details for the G2S, we remark that one can lighten the off-line assembly of the matrix $A_{2h} = RA_hP$, using instead the matrix

$$\widetilde{A}_{2h} := \begin{bmatrix} I_{2h,2} & -G_{2h,1} \\ -G_{2h,2} & I_{2h,1} \end{bmatrix}, \tag{33}$$

which corresponds to a direct discretization of A on the coarse level. Moreover, since our two-level method works directly on the interfaces, we have more freedom in the discretization of the smoothing operators on each level. For instance, one could keep the corresponding volume mesh close to the substructures, while having a coarser grid away from them. This strategy would follow a similar idea of the methods discussed in, e.g., [15] and references therein.

[4] Given two matrices A and B, AB and BA share the same non-zero eigenvalues.

Algorithm 4 Geometric multilevel substructured DD method - GMS($\mathbf{u}^0$,$\mathbf{b}$,ℓ)

if $\ell = \ell_{\min}$, **then**
  set $\mathbf{v}^0 = A_{\ell_{\min}}^{-1}\mathbf{b}$. (direct solver)
else
  $\mathbf{v}^n = G_\ell(\mathbf{v}^{n-1}, \mathbf{b})$, $n = 1, \dots, n_1$ (DD pre-smoothing steps)
  $\mathbf{r} = \mathbf{b} - A_\ell \mathbf{v}^{n_1}$ (compute the residual)
  $\mathbf{v}_c$ =GMS($\mathbf{0}, R_{\ell-1}^{\ell}\mathbf{r}, \ell-1$). (recursive call)
  $\mathbf{v}^0 = \mathbf{v}^{n_1} + P_{\ell-1}^{\ell}\mathbf{v}_c$ (coarse correction)
  $\mathbf{v}^n = G_\ell(\mathbf{v}^{n-1}, \mathbf{b})$, $n = 1, \dots, n_2$ (DD post-smoothing steps)
  Set $\mathbf{v}^0 = \mathbf{v}^{n_2}$ (update)
end if
return $\mathbf{u}^0$.

4.2 GMS: Extension to multilevel framework

The solution of large problems can be challenging using classical two-level methods in volume. This is mainly due to the dimension of the coarse space, which can still be too large in volume to be solved exactly. In our substructured framework, the size of the substructured coarse matrix corresponds to the number of degrees of freedom on the coarse substructures, and thus it is already much smaller if compared to the volume case (see Section 5.1 for a comparison of their sizes in a concrete model problem). However, there might be problems for which the direct solution of the coarse problem is inconvenient also in our substructured framework. For instance, if we considered multiple subdomains, then we would have several substructures and therefore the size of the substructured coarse matrix increases.

The G2S is suitable to a multilevel generalization following a classical multigrid strategy [42]. Given a sequence of grids on the substructures labeled from the coarsest to the finest by $\{\ell_{\min}, \ell_{\min}+1, \dots, \ell_{\max}\}$, we denote by $P_{\ell-1}^{\ell}$ and $R_{\ell-1}^{\ell}$ the interpolation and restriction operators between grids ℓ and $\ell-1$. To build the substructured matrices on the different grids we have two possible choices. The first one corresponds to the standard Galerkin projection. Being $A_{\ell_{\max}}$ the substructured matrix on the finest grid, we can define the coarse matrices $A_\ell := R_\ell^{\ell+1} A_{\ell+1} P_\ell^{\ell+1}$, for $\ell \in \{\ell_{\min}, \ell_{\min}+1, \dots, \ell_{\max}-1\}$. The second choice consists in defining A_ℓ directly as the discretization of (12) on the grid labeled by ℓ, and corresponds exactly to (33) for the two-grid case. The two choices are not equivalent. On the one hand, the Galerkin approach leads to a faster method in terms of iterations. However, the Galerkin matrices A_ℓ do not have the block structure as in (12). For instance, $A_{\ell_{\max}-1} = R_{\ell_{\max}-1}^{\ell_{\max}} A_{\ell_{\max}} P_{\ell_{\max}-1}^{\ell_{\max}} = R_{\ell_{\max}-1}^{\ell_{\max}} P_{\ell_{\max}-1}^{\ell_{\max}} - R_{\ell_{\max}-1}^{\ell_{\max}} G_{\ell_{\max}} P_{\ell_{\max}-1}^{\ell_{\max}}$. Thus, the identity matrix is replaced by the sparse matrix $R_{\ell_{\max}-1}^{\ell_{\max}} P_{\ell_{\max}-1}^{\ell_{\max}}$. On the other hand, defining A_ℓ directly on the current grid ℓ as in (33) leads to a minimum increase of the iteration number, but it permits to preserve the original block-diagonal structure (which is important if one wants to use G2S-B1 and G2S-B2). The difference between the two approaches is also studied by numerical experiments in Section 5.

In spite of the choice for A_ℓ, we define the geometric multilevel substructured DD method (GMS) in Algorithm 4, which is a substructured multi-grid V-cycle.

5 Numerical experiments

In this section, we demonstrate the effectiveness of our new computational framework by extensive numerical experiments. These experiments have two main purposes. On the one hand, we wish to validate the theoretical results of Section 3.1, while discussing the implementation details of Section 4.1 and comparing our new method with other classical existing methods, like a two-grid method in volume using RAS as smoother. This is the focus of Section 5.1, where a Poisson problem on two-dimensional and three-dimensional boxes is considered. Both convergence rates and computational times are studied.

On the other hand, we wish to show the effectiveness of our new methods in case of multiple-subdomain decompositions and for classical test problems. In particular, in Section 5.2 we consider a multiple-subdomain decomposition for a classical Poisson problem, while in Section 5.3 a diffusion problem with highly jumping diffusion coefficients is solved. For the efficient solution of these two problems different discretization methods are required. These are the finite-difference method, for the classical Poisson problem, and the finite-volume method, in case of jumping diffusion coefficients. These two methods require different definitions of restriction and prolongation operators. We thus sketch some implementation details of our algorithms for a regular decomposition. In both cases, the robustness of our methods for increasing number of subdomains is studied and compared to classical two-grid and multi-grid methods defined in volume and using RAS as smoother. The obtained numerical results show clearly, and particularly for the jumping diffusion coefficient case, that our methods converge in less iterations than the classical two-level RAS method. In Sections 5.1 and 5.2, all the methods are used as iterative solvers, without any Krylov acceleration, while in Section 5.3 we test their efficiency as preconditioners for GMRES.

5.1 Laplace equation on 2D and 3D boxes

We first consider the Poisson equation $-\Delta u = f$ with homogeneous Dirichlet boundary condition. The geometry of the domain and its decomposition are shown in Figure 3, where Ω is decomposed into two overlapping rectangles $\Omega_1 = (-1, \delta) \times (0, 1)$ and $\Omega_2 = (-\delta, 1) \times (0, 1)$. The length of the overlap is 2δ. On each subdomain, we use a standard second-order finite difference scheme based on a uniform grid of $N_y = 2^\ell - 1$ interior points in direction y and $N_x = N_y$ interior points in direction x. Here, ℓ is a positive integer. The grid size is denoted by h. The overlap is assumed to be $\delta = hN_{ov}$, where N_{ov} represents the number of interior points in the overlap in direction x.

The results of our numerical experiments are shown in Figures 5. All figures show the decay of the relative errors with respect to the number of iterations. To study the asymptotic convergence behavior of the G2S and compare it with the theoretical results of Section 3.2, all methods are stopped if the relative error is smaller than the very low tolerance 10^{-12}.

The problem is solved by the classical parallel Schwarz method, a classical two-grid and a three-grid method using RAS as smoother ("2L-RAS" and "3L-RAS" in the figures), the G2S method, and its extension to three-grid denoted by G3S. For the G2S method, we further distinguish two cases: "G2S" indicates the G2S

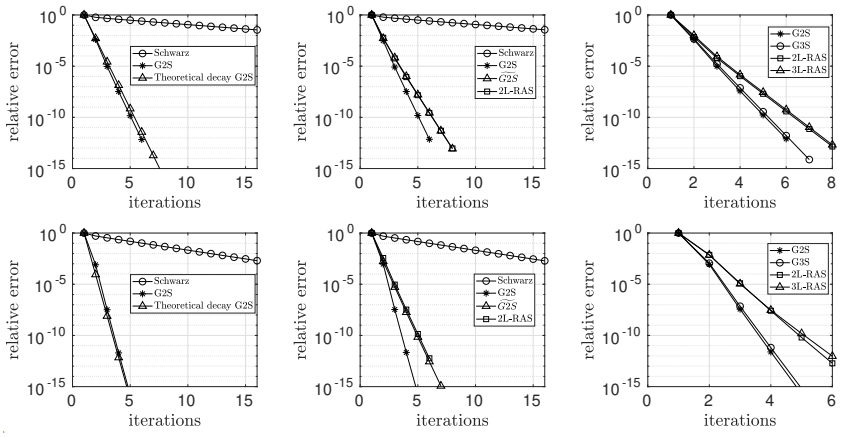

Fig. 5: Convergence curves for $\ell = 6$, $N_{ov} = 4$ (top row), $N_{ov} = 8$ (bottom row).

method using the coarse matrix $A_{2h} := RA_hP$, while "$\widetilde{\text{G2S}}$" refers to the G2S method using the coarse matrix obtained by a direct discretization of A on the coarse grid (instead of $A_{2h} := RA_hP$), see (33) and the discussion in Section (4.2). For the G2S and $\widetilde{\text{G2S}}$ methods, we use the one-dimensional linear interpolation operator $P = \text{diag}(P_1, P_2)$, where the expression of P_j, $j = 1, 2$ is given in (15), and $R = \frac{1}{2}(P)^\top$ (as explained in Section 3.1 and Figure 3). For 2L-RAS, we use the classical full weighting restriction and interpolation operators, $P_V = \text{kron}(P_x, P_y)$, $R_V = \frac{1}{4}P_V^\top$, where P_x, P_y are one-dimensional interpolation operators of the same form of (15).

The left panels of Figures 5 validate the theoretical convergence factor obtained in Theorem 3. The center panels compare the classical one-level PSM, the G2S and $\widetilde{\text{G2S}}$ method and the 2L-RAS method. The slower performance of 2L-RAS with respect to G2S can be traced back to the interpolation step. This operation breaks the harmonicity of the obtained correction, which therefore does not lie any more in the space of the error; see, e.g., [34]. One could use interpolators which extend harmonically the correction inside the overlapping subdomains although this would increase significantly the computational cost of each iteration. We refer also to [37] for a similar observation. Further notice that, while the G2S coarse space has dimension about N_y, the one corresponding to the 2L-RAS method has dimension about $N_xN_y/4 \approx N_y^2/2 \gg N_y$. In the setting of Figure 5, the dimensions of the coarse spaces of G2S and 2L-RAS are about 60 and 1900, respectively. Notice that, the convergence of $\widetilde{\text{G2S}}$ is comparable to 2L-RAS, hence a little slower than G2S, but the assembly of the coarse matrix is cheaper.

The right panels compare the convergence behavior of the two-grid methods, G2S and 2L-RAS, with their three-level variants. We remark that the addition of a third level does not result in a noticeable convergence deterioration.

Next, we are interested in computational times and in numerically validating the computational cost presented in Table 1. To do so, we consider a three-dimensional box $\Omega = (-1, 1) \times (0, 1) \times (0, 1)$ decomposed into two overlapping subdomains $\Omega_1 = (-1, \delta) \times (0, 1) \times (0, 1)$ and $\Omega_2 = (-\delta, 1) \times (0, 1) \times (0, 1)$. We solve the problem (up to a tolerance of 10^{-6} on the relative error) using the G2S method,

# (volume)	G2S	G2S-B1	G2S-B2	2L-RAS
539	4	4	4	6
6075	5	5	5	6
56699	5	5	4	6
488187	4	4	4	6

Table 2: Number of iterations performed by the different methods and for different number of degrees of freedom.

# (volume)	G2S	G2S-B1	G2S-B2	2L-RAS
539	0.014 (0.004)	0.008 (0.002)	0.006 (0.002)	0.009 (0.005)
6075	0.1346 (0.027)	0.082 (0.016)	0.083 (0.016)	0.106 (0.03)
56699	2.040 (0.408)	1.2282 (0.246)	0.818 (0.204)	1.367 (0.228)
488187	53.2873 (13.321)	33.309 (6.662)	26.687 (6.671)	43.635 (7.272)

Table 3: Computational times performed by the different methods. In parentheses we indicate the computational time per iteration.

its equivalent forms G2S-B1 and G2S-B2, introduced in Section 4.1, and 2L-RAS. The length of the overlap is $\delta = hN_{ov}$, where h is the grid size and N_{ov} is fixed to 4. Hence the overlap is proportional to the grid size. The experiments have been performed on a workstation with a processor Intel Core i9-10900X CPU 3.7GHz and with 32GB of RAM. The subdomain problems are solved sequentially using the MATLAB backslash command, which calls a direct solver for sparse banded matrices (with small band density threshold) with almost linear complexity. The smoothing steps have the same cost for both 2L-RAS and G2S implementations, and it permits to better remark the advantages of the substructured methods in the coarse step and the prolongation/restriction steps. The results are shown in Tables 2 and 3. The G2S method outperforms 2L-RAS in terms of iteration numbers and computational times. To better understand why the G2S method is faster and to validate the computational cost analysis presented in Table 1, Figures 6 show the computational time spent by the G2S and 2L-RAS methods in the different steps of a two-level method. As expected, the smoothing step requires the same effort in both methods. This is shown in Figure 6 (left). In Figure 6 (right) we compare the computational times required by one coarse correction step performed by the two methods. The two curves corresponds to the same volumetric dimensions of the problem (as in Table 3), but the coarse space dimensions corresponding to G2S and 2L-RAS are different. This means that, the kth point (circle) from the left of the G2S curve has to be compared with the kth point (cross) from the left of the 2L-RAS curve. It must also be said that for both cases we use the Matlab backslash command. This is clearly a choice more favorable for the 2L-RAS coarse problem (which is sparse and banded). A different and more appropriate solver for the G2S coarse matrix exploiting the block-sparse structure (see, e.g., [38]) could lead to further improvement of these computational times.

5.2 Decompositions into many subdomains

In this section, we consider a square domain Ω decomposed into $M \times M$ nonoverlapping square subdomains $\widetilde{\Omega}_j$, $j = 1, \ldots, M^2 = N$. Each subdomain $\widetilde{\Omega}_j$ contains

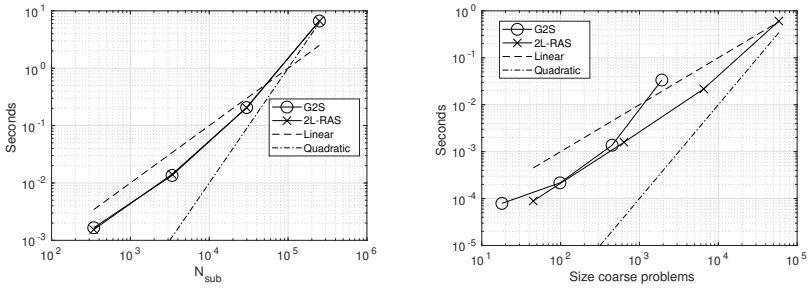

Fig. 6: Time in seconds spent by the G2S and 2L-RAS methods in the smoothing step (left) and in the coarse solver step (right).

$N_{\text{sub}} = (2^\ell - 1)^2$ interior degrees of freedom. Extending the subdomains $\widetilde{\Omega}_j$ by N_{ov} points, we obtain the overlapping subdomains Ω_j with overlap $\delta = 2N_{ov}h$. On each subdomain Ω_j, we locate the discrete substructure $\mathcal{S}_j^{N_j}$, marked with blue lines in Figure 7, which is made by four (one-dimensional) segments. For each dis-

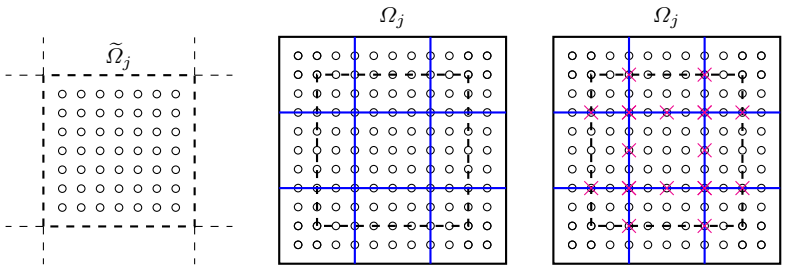

Fig. 7: An interior nonoverlapping subdomain $\widetilde{\Omega}_j$ is enlarged by $N_{ov} = 2$ points in each direction. The discrete substructure $\mathcal{S}_j^{N_j}$ is denoted by a blue line. On the right panel, the coarse discrete substructure $\mathcal{S}_j^{M_j}$ is marked by red crosses.

crete substructure $\mathcal{S}_j^{N_j}$, the interpolation operator P_j acts block-wise on each one-dimensional interval, i.e. $P_j = \text{diag}\{\widetilde{P}_1, \widetilde{P}_2, \widetilde{P}_3, \widetilde{P}_4\}$, where each $\widetilde{P}_k$, $k = 1, \dots, 4$, corresponds to the prolongation matrix (15). We remark that using $\widetilde{P}_k$ implies assuming that on the boundary of $\mathcal{S}_j$ the function attains zero. This holds since on each substructure $v_j \in H_{00}^{\frac{1}{2}}(\mathcal{S}_j)$, due to the partition of unity.

The results of our numerical experiments are reported in Figure 8. The left panel shows the dependence of the spectral radius on the size of the overlap for the different methods and $N = 16$, $\ell = 5$. We then study the robustness of the method with respect to an increasing number of subdomains. We first keep the size of each subdomain fixed, $N_{\text{sub}} = (2^5 - 1)^2$, and thus we consider larger global problems as N grows. Then, we fix a global domain Ω with approximately $17 \cdot 10^3$ interior degrees of freedom, and we get smaller subdomains as N grows. In both

cases, we observe that the spectral radius of both 2L-RAS and G2S does not deteriorate as the number of subdomains increases.

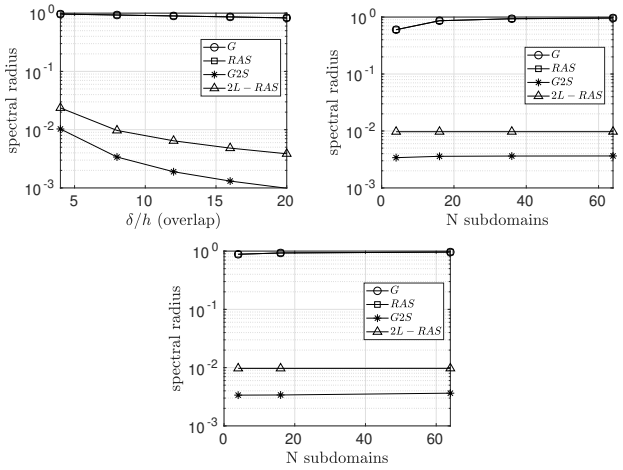

Fig. 8: Dependence of spectral radius on the overlap (left) and robustness of the two-level methods when increasing the number of subdomains for subdomains with same size (center) and global problem fixed (right).

We further compare G2S, 2LRAS and Geometric MultiGrid (GMG) for the solution of the Poisson equation $-\Delta u = 1$. We decompose Ω into $N = 9$ and $N = 16$ subdomains and set $N_{ov} = 2$. Table 4 reports the number of iterations and computational times to reach a relative tolerance of 10^{-6}. For the G2S method, we preassembled the coarse matrix. Concerning GMG, we implemented a V-cycle with

$\ell - N^v$	G2S	2L-RAS	GMG
7-146689	3 (0.31)	3 (0.42)	6 (0.22)
8-588289	3 (0.68)	3 (1.34)	6 (0.97)
9-2356225	3 (2.29)	3 (5.94)	6 (4.06)

$\ell - N^v$	G2S	2L-RAS	GMG
7-261121	3 (0.50)	3 (0.72)	6 (0.44)
8-1046529	3 (1.64)	3 (2.57)	6 (1.73)
9-4190209	3 (5.98)	3 (11.38)	6 (7.30)

Table 4: Number of iterations and seconds (in brackets) required by the G2S, 2L-RAS and GMG methods to reach a tolerance of 10^{-6}. The left table refers to $N = 9$, while the right table to $N = 16$.

two pre- and post-smoothing steps using a damped Jacobi smoother with optimal damping parameter $\omega = 4/5$ [53]. The coarsest level of GMG corresponds to $\ell = 3$ and the size of the coarse matrix is 961. Concerning the implementation of the DD methods, the subdomain problems are solved in parallel, using the Matlab parallel Toolbox. The G2S method is implemented according to Algorithm 3. The sizes of the G2S coarse matrices are 3096, 6168, 12312 for $\ell = 7, 8, 9$, respectively. For both G2S and GMG, we compute once for all the LU decompositions of the corresponding coarse matrices as their size is small. The cost of the LU decompo-

sitions is included in the computational times reported. Table 4 shows the G2S is competitive with GMG.

5.3 Diffusion problem with jumping diffusion coefficients

In this section, we test our method for the solution of a diffusion equation $-\text{div}(\alpha\nabla u) = f$ in a square domain $\Omega := (0,1)^2$ with $f := \sin(4\pi x)\sin(2\pi y)\sin(2\pi xy)$. The domain Ω is decomposed into 16 non-overlapping subdomains.

We suppose $\alpha = 1$ everywhere except in some channels where α takes the values 10^2, 10^4 and 10^6. We consider two configurations represented in Figure 9.

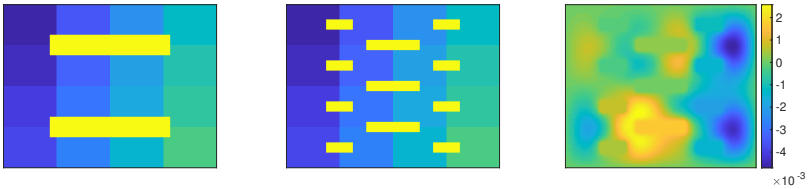

Fig. 9: Decomposition of Ω into 16 subdomains with two different patterns of channels (left and center). The yellow regions correspond to large values of the diffusion coefficient. The blue-green area shows the nonoverlapping decomposition. The right panel shows the solution of the equation with the central pattern.

We use a finite-volume discretization, where each non-overlapping subdomain is discretized with $N_{\text{sub}} = 2^{2\ell}$ cells and it is enlarged by N_{ov} cells to create an overlapping decomposition with overlap $\delta = 2N_{ov}h$. We further assume that the discontinuities of the diffusion coefficient are aligned with the edges of the cells and they do not cross any cell. The mapping between the fine and coarse mesh is illustrated in Figure 10. At the volume level, the restriction operator maps four fine cells to a single coarse cell by averaging the four cell values and the interpolation operator is its transpose. At the substructured level, the restriction operator maps two fine cells to a single coarser cell by averaging. The interpolation operator splits one coarse cell to two fine cells assigning the same coarse value to each new cell.

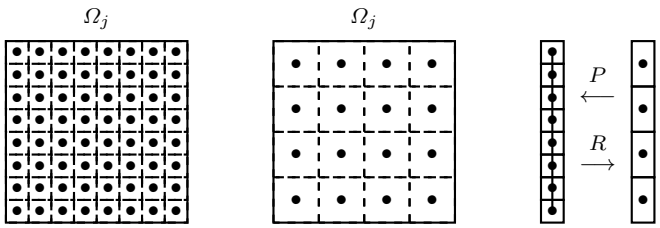

Fig. 10: Illustration of the action of the restriction operator in volume (left) and of the restriction and interpolation operators on a one-dimensional substructure (right).

dimV_c	456	840	1608
α \ N^v	4096	16384	65536
10^2	4 (17)	4 (16)	4 (16)
10^4	4 (18)	4 (16)	4 (16)
10^6	6 (17)	5 (17)	5 (16)

dimV_c	456	840	1608
α \ N^v	4096	16384	65536
10^2	4 (19)	4 (19)	4 (18)
10^4	5 (20)	5 (19)	4 (18)
10^6	6 (20)	6 (19)	7 (18)

Table 5: Number of iterations performed by the G2S and 2L-RAS (in brackets) methods with $N_{ov} = 2$ and for different values of jumps of α and different numbers of degrees of freedom N^v. The dimension of the substructured coarse space is dimV_c. The left table refers to the two channels configuration and the right table to the multiple channels one.

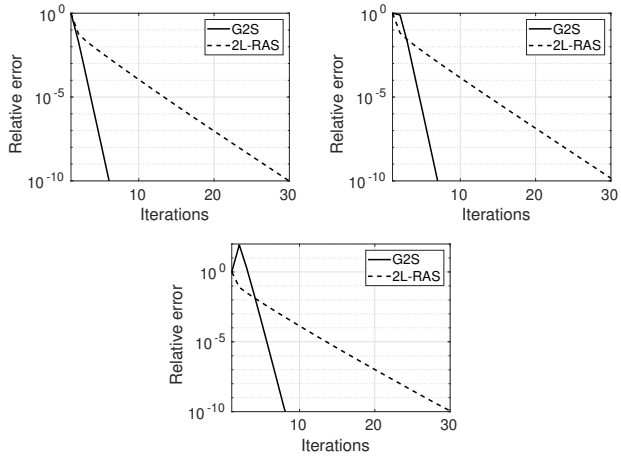

Fig. 11: Convergence curves for $\ell = 5$, $N_{ov} = 2$ for the two channels configuration. The parameter α is equal to 10^2 (left), 10^4 (center), 10^6 (right).

It still holds that the interpolation operator is the transpose of the restriction operator.

In this setting, we study the robustness of the G2S method with respect to the mesh size and the amplitudes of the jumps of α and we compare it to the 2L-RAS method. In Table 5 we report the number of iterations to reach a relative error of Tol $= 10^{-6}$. Both methods are used as iterative solvers. The iterations performed by the G2S method are the numbers on the left in each cell of the table, while the iterations of the 2L-RAS are the numbers in brackets on the right. We can observe that the G2S outperforms 2L-RAS. Figures 11 show the convergence curves for a fixed mesh size and three different values of α. These results show that the G2S method is robust both with respect to the jumps of the diffusion coefficient and the mesh size, and that it outperforms the 2L-RAS method.

Now, we study the performance of G2S and 2L-RAS as preconditioners for GMRES. Table 6 reports the number of iterations when both methods are used to accelerate GMRES. We further specify the final size of the Krylov subspaces. GMRES preconditioned by the G2S method builds a much smaller Krylov subspace as the system and preconditioner have dimensions equal to the size of the substructured space.

6 Conclusions

In this work we introduced a new framework of two-level and multi-level substructured DD methods, namely the G2S method and its extension called GMS method. These are formulated on the substructures of the considered overlapping domain decomposition. Under certain reasonable hypotheses, for 2 subdomains in 2d, we proved that the G2S method is well posed and convergent, and we also estimated the corresponding convergence factor. The effectiveness of our new methods is confirmed by extensive numerical experiments, where elliptic PDE problems with possibly highly jumping diffusion coefficients are efficiently solved.

References

1. Aarnes, J., Hou, T.Y.: Multiscale domain decomposition methods for elliptic problems with high aspect ratios. Acta Math. Appl. Sin. **18**(1), 63–76 (2002)
2. Bjorstad, P., Gander, M.J., Loneland, A., Rahman, T.: Does SHEM for Additive Schwarz work better than predicted by its condition number estimate? Domain Decomposition Methods in Science and Engineering XXIV, LNCSE, Springer −(–), 129–138 (2018)
3. Bonazzoli, M., Dolean, V., Graham, I., Spence, E., Tournier, P.H.: Domain decomposition preconditioning for the high-frequency time-harmonic maxwell equations with absorption. Mathematics of Computation **88**(320), 2559–2604 (2019)
4. Bonazzoli, M., Dolean, V., Graham, I.G., Spence, E.A., Tournier, P.H.: Two-level preconditioners for the helmholtz equation. In: Domain Decomposition Methods in Science and Engineering XXIV. Springer International Publishing (2018)
5. Brezina, M., Manteuffel, T., McCormick, S., Ruge, J., Sanders, G.: Towards adaptive smoothed aggregation (αSA) for nonsymmetric problems. SIAM J. Sci. Comput. **32**(1), 14–39 (2010)
6. Cai, X.C., Sarkis, M.: A restricted additive Schwarz preconditioner for general sparse linear systems. SIAM J. Sci. Comp. **21**(2), 792–797 (1999)
7. Chaouqui, F., Ciaramella, G., Gander, M.J., Vanzan, T.: On the scalability of classical one-level domain-decomposition methods. Vietnam J. Math. **46**(4), 1053–1088 (2018)
8. Chaouqui, F., Gander, M.J., Santugini-Repiquet, K.: A coarse space to remove the logarithmic dependency in Neumann–Neumann methods. In: Domain Decomposition Methods in Science and Engineering XXIV, pp. 159–167. Springer International Publishing, Cham (2018)
9. Chaouqui, F., Gander, M.J., Santugini-Repiquet, K.: A local coarse space correction leading to a well-posed continuous Neumann-Neumann method in the presence of cross points. In: Domain Decomposition Methods in Science and Engineering XXV, pp. 83–91. Springer International Publishing, Cham (2020)
10. Ciaramella, G., Gander, M.J.: Analysis of the parallel Schwarz method for growing chains of fixed-sized subdomains: Part I. SIAM J. Numer. Anal. **55**(3), 1330–1356 (2017)

$\dim \mathcal{K}$	0.03 (0.38)	0.05 (1.17)	0.1 (4.71)
α \ N^v	4096	16384	65536
10^2	3 (8)	3 (8)	3 (8)
10^4	3 (8)	3 (8)	3 (8)
10^6	3 (8)	3 (8)	3 (8)

$\dim \mathcal{K}$	0.03 (0.38)	0.05 (1.17)	0.1 (4.71)
α \ N^v	4096	16384	65536
10^2	3 (9)	3 (8)	3 (8)
10^4	3 (9)	3 (8)	3 (8)
10^6	3 (9)	3 (8)	3 (8)

Table 6: Number of iterations performed by GMRES preconditioned by G2S and 2L-RAS (in brackets) with $N_{ov} = 2$ and for different values of jumps of α and different numbers of degrees of freedom N^v. The dimension of the final Krylov subspace is $\dim \mathcal{K}$ and expressed in megabytes. The left table refers to the two channels configuration and the right table to the multiple channels one.

11. Ciaramella, G., Gander, M.J.: Analysis of the parallel Schwarz method for growing chains of fixed-sized subdomains: Part II. SIAM J. Numer. Anal. **56 (3)**, 1498–1524 (2018)
12. Ciaramella, G., Gander, M.J.: Analysis of the parallel Schwarz method for growing chains of fixed-sized subdomains: Part III. Electron. Trans. Numer. Anal. **49**, 201–243 (2018)
13. Ciaramella, G., Gander, M.J.: Iterative Methods and Preconditioners for Systems of Linear Equations. Fundamentals of Algoriths. SIAM (2022)
14. Ciaramella, G., Gander, M.J., Halpern, L., Salomon, J.: Methods of reflections: relations with Schwarz methods and classical stationary iterations, scalability and preconditioning. The SMAI journal of computational mathematics **5**, 161–193 (2019)
15. Ciaramella, G., Gander, M.J., Mamooler, P.: The domain decomposition method of Bank and Jimack as an optimized Schwarz method. In: Domain Decomposition Methods in Science and Engineering XXV, pp. 285–293. Springer International Publishing, Cham (2020)
16. Ciaramella, G., Hassan, M., Stamm, B.: On the scalability of the parallel Schwarz method in one-dimension. In: Domain Decomposition Methods in Science and Engineering XXV, pp. 151–158. Springer International Publishing, Cham (2020)
17. Ciaramella, G., Hassan, M., Stamm, B.: On the scalability of the Schwarz method. The SMAI journal of computational mathematics **6**, 33–68 (2020)
18. Davis, T.A.: Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms 2). Society for Industrial and Applied Mathematics, USA (2006)
19. Dohrmann, C.R., Klawonn, A., Widlund, O.B.: A family of energy minimizing coarse spaces for overlapping Schwarz preconditioners. In: Domain Decomposition Methods in Science and Engineering XVII, pp. 247–254 (2008)
20. Dolean, V., Jolivet, P., Nataf, F.: An Introduction to Domain Decomposition Methods. SIAM, Philadelphia, PA (2015)
21. Dolean, V., Nataf, F., Scheichl, R., Spillane, N.: Analysis of a two-level Schwarz method with coarse spaces based on local Dirichlet-to-Neumann maps. Comput. Meth. in Appl. Math. **12**(4), 391–414 (2012)
22. Dubois, O., Gander, M.J., Loisel, S., St-Cyr, A., Szyld, D.B.: The optimized Schwarz method with a coarse grid correction. SIAM J. Sci. Comput. **34**(1), 421–458 (2012)
23. Efendiev, Y., Galvis, J., Lazarov, R., Willems, J.: Robust domain decomposition preconditioners for abstract symmetric positive definite bilinear forms. ESAIM Math. Model. Numer. Anal. **46**(5), 1175–1199 (2012)
24. Efstathiou, E., Gander, M.J.: Why Restricted Additive Schwarz converges faster than Additive Schwarz. BIT Numerical Mathematics **43**(5), 945–959 (2003)
25. Galvis, J., Efendiev, Y.: Domain decomposition preconditioners for multiscale flows in high-contrast media. Multiscale Model. Sim. **8**(4), 1461–1483 (2010)
26. Galvis, J., Efendiev, Y.: Domain decomposition preconditioners for multiscale flows in high contrast media: Reduced dimension coarse spaces. Multiscale Model. Sim. **8**(5), 1621–1644 (2010)
27. Gander, M.J.: Optimized Schwarz methods. SIAM J. Numer. Anal. **44**(2), 699–731 (2006)
28. Gander, M.J.: Schwarz methods over the course of time. Electron. Trans. Numer. Anal. **31**, 228–255 (2008)
29. Gander, M.J.: On the influence of geometry on optimized schwarz methods. SeMA Journal **53**(1), 71–78 (2011)
30. Gander, M.J., Halpern, L., Repiquet, K.: A new coarse grid correction for RAS/AS. In: Domain Decomposition Methods in Science and Engineering XXI, pp. 275–283. Springer (2014)
31. Gander, M.J., Halpern, L., Santugini-Repiquet, K.: On optimal coarse spaces for domain decomposition and their approximation. In: Domain Decomposition Methods in Science and Engineering XXIV, pp. 271–280. Springer International Publishing, Cham (2018)
32. Gander, M.J., Loneland, A.: SHEM: An optimal coarse space for RAS and its multiscale approximation. In: Domain Decomposition Methods in Science and Engineering XXIII, pp. 313–321. Springer (2017)
33. Gander, M.J., Loneland, A., Rahman, T.: Analysis of a new harmonically enriched multiscale coarse space for domain decomposition methods. preprint arXiv:1512.05285 (2015)
34. Gander, M.J., Song, B.: Complete, optimal and optimized coarse spaces for additive Schwarz. In: Domain Decomposition Methods in Science and Engineering XXIV. Springer (2018)
35. Gander, M.J., Van Criekingen, S.: New coarse corrections for optimized restricted additive Schwarz using PETSc. In: Domain Decomposition Methods in Science and Engineering XXV, pp. 483–490. Springer International Publishing, Cham (2020)

36. Gander, M.J., Vanzan, T.: Heterogeneous optimized Schwarz methods for second order elliptic PDEs. SIAM Journal on Scientific Computing **41**(4), A2329–A2354 (2019)
37. Gander, M.J., Vanzan, T.: Multilevel optimized Schwarz methods. SIAM Journal on Scientific Computing **42**(5), A3180–A3209 (2020)
38. Golub, G.H., Van Loan, C.F.: Matrix Computations (Fourth Edition). Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD (2013)
39. Graham, I., Spence, E., Vainikko, E.: Domain decomposition preconditioning for high-frequency helmholtz problems with absorption. Mathematics of Computation **86**(307), 2089–2127 (2017)
40. Graham, I.G., Lechner, P.O., Scheichl, R.: Domain decomposition for multiscale PDEs. Numer. Math. **106**(4), 589–626 (2007)
41. Hackbusch, W.: Local Defect Correction Method and Domain Decomposition Techniques, pp. 89–113. Vienna (1984)
42. Hackbusch, W.: Multi-Grid Methods and Applications. Series in Computational Mathematics. Springer Berlin Heidelberg (2013)
43. Hackbusch, W.: Hierarchical Matrices: Algorithms and Analysis, 1st edn. Springer Publishing Company, Incorporated (2015)
44. Heinlein, A., Klawonn, A., Knepper, J., Rheinbach, O.: Multiscale coarse spaces for overlapping Schwarz methods based on the ACMS space in 2D. Electron. Trans. Numer. Anal. **48**, 156–182 (2018)
45. Klawonn, A., Radtke, P., Rheinbach, O.: FETI-DP methods with an adaptive coarse space. SIAM J. Numer. Anal. **53**(1), 297–320 (2015)
46. Lions, J., Magenes, E.: Non-homogeneous Boundary Value Problems and Applications (Vol I). Die Grundlehren der mathematischen Wissenschaften. Springer-Verlag Berlin Heidelberg (1972)
47. Lions, P.L.: On the Schwarz alternating method. I. First international symposium on domain decomposition methods for partial differential equations pp. 1–42 (1988)
48. Quarteroni, A., Valli, A.: Domain Decomposition Methods for Partial Differential Equations. Numerical Mathematics and Scientific Computation. Oxford Science Publications (1999)
49. Spillane, N., Dolean, V., Hauret, P., Nataf, F., Pechstein, C., Scheichl, R.: A robust two-level domain decomposition preconditioner for systems of PDEs. C. R. Math. **349**(23), 1255 – 1259 (2011)
50. Spillane, N., Dolean, V., Hauret, P., Nataf, F., Pechstein, C., Scheichl, R.: Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. Numer. Math. **126**(4), 741–770 (2014)
51. Tartar, L.: An Introduction to Sobolev Spaces and Interpolation Spaces. Lecture Notes of the Unione Matematica Italiana. Springer Berlin Heidelberg (2007)
52. Toselli, A., Widlund, O.: Domain Decomposition Methods: Algorithms and Theory, Series in Computational Mathematics, vol. 34. Springer, New York (2005)
53. Trottenberg, U., Ulrich Trottenberg, C., Oosterlee, C., Schuller, A., Brandt, A., Oswald, P., Stüben, K.: Multigrid. Elsevier Science (2001)
54. Widlund, O., Dryja, M.: An additive variant of the Schwarz alternating method for the case of many subregions. Tech. rep., Department of Computer Science, Courant Institute (1987)
55. Zampini, S., Tu, X.: Multilevel balancing domain decomposition by constraints deluxe algorithms with adaptive coarse spaces for flow in porous media. SIAM J. Sci. Comput. **39**(4), A1389–A1415 (2017)

MOX Technical Reports, last issues

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

14/2022 Zappon, E.; Manzoni, A.; Quarteroni A.
Efficient and certified solution of parametrized one-way coupled problems through DEIM-based data projection across non-conforming interfaces

15/2022 G. Ciaramella, T. Vanzan
Spectral coarse spaces for the substructured parallel Schwarz method

13/2022 Grasselli, M.; Parolini, N.; Poiatti, A.; Verani, M.
Non-isothermal non-Newtonian fluids: the stationary case

12/2022 Antonietti, P.F.; Dassi, F.; Manuzzi, E.
Machine Learning based refinement strategies for polyhedral grids with applications to Virtual Element and polyhedral Discontinuous Galerkin methods

11/2022 Sampaoli, S.; Agosti, A.; Pozzi, G.; Ciarletta,P.
A toy model of misfolded protein aggregation and neural damage propagation in neudegenerative diseases

09/2022 Corti, M.; Zingaro, A.; Dede', L.; Quarteroni, A.
Impact of Atrial Fibrillation on Left Atrium Haemodynamics: A Computational Fluid Dynamics Study

10/2022 Fresca, S.; Manzoni, A.
Real-time simulation of parameter-dependent fluid flows through deep learning-based reduced order models

08/2022 Gobat, G.; Opreni, A.; Fresca, S.; Manzoni, A.; Frangi, A.
Reduced order modeling of nonlinear microstructures through Proper Orthogonal Decomposition

07/2022 Sinigaglia, C.; Quadrelli, D.E.; Manzoni, A.; Braghin, F.
Fast active thermal cloaking through PDE-constrained optimization and reduced-order modeling

06/2022 Pozzi, G.; Grammatica, B.; Chaabane, L.; Catucci, M.; Mondino, A.; Zunino, P.; Ciarletta, P.
T cell therapy against cancer: a predictive diffuse-interface mathematical model informed by pre-clinical studies